1 2 9 0

UNIVERSIDADE Đ
COIMBRA

António José Preto Martins Gomes

Deep Learning Application to *in silico* Drug Design

Doctoral Thesis submitted for the PhD degree in Experimental Biology and Biomedicine – Biotechnology and Health, supervised by Professor Irina de Sousa Moreira and Professor Alexandre Bonvin and presented to the Institute of Interdisciplinary Research of the University of Coimbra

December 2022

Institute of Interdisciplinary Research of the University of Coimbra

Deep-Learning application to *in silico* Drug Design

António José Preto Martins Gomes

Doctoral Thesis submitted for the PhD degree in Experimental Biology and Biomedicine - Biotechnology and Health, supervised by Professor Irina de Sousa Moreira and Professor Alexandre Bonvin and presented to the Institute of Interdisciplinary Research of the University of Coimbra

December 2022

1 2 9 0

UNIVERSIDADE Ð
COIMBRA

”All generalisations - perhaps except this one - are false.” - Kurt Gödel

Agradecimentos

Antes de mais quero agradecer à minha orientadora, a Professora Irina Moreira, a oportunidade e o apoio que me foram dados. Não é comum encontrar um líder de grupo disposto a investir nas pessoas como a professora investe. Quando enviei o primeiro email, em 2017, a perguntar se haveria possibilidade de reunir para discutir a possibilidade de fazer tese de Mestrado não antevia que isso se fosse desenrolar numa tese de Mestrado, uma tese de Doutoramento, cinco EJIBCES, vários projetos, diversas publicações e, mais importante, uma jornada científica. Obrigado.

I would also like to thank Professor Alexandre Bonvin, my co-advisor, for the opportunity to communicate and collaborate with international peers and the support in multiple publications and exciting projects.

Agradeço também todo o apoio financeiro fornecido por parte da Fundação para a Ciência e a Tecnologia (FCT), Ministério da Ciência, Tecnologia e Ensino Superior de Portugal (Bolsa de Doutoramento: SFRH/BD/144966/2019) e do Centro de Neurociências e Biologia Celular, bem como à Universidade de Coimbra.

Quero também agradecer a amigos que Coimbra me deu, o Chicória, o Gonçalo, a Ana Rita, a Margarida, o Pedro António (a.k.a. Mágico), o Piedade, o Bidarra, a Dani Costa, o Xavier, o Frias, a Baptista, o Diogo, o Cláudio, o Brandão, a Ana Costa, o Roque, o Cruz. Ao grande Miguel Oliveira (não o piloto, mas sim a versão melhorada). E a amigos que já Viseu me tinha trazido, o João Pedro, o Ferrolho e o Routar.

À Joana, num sistema científico que atravessa sérios problemas no que toca a publicações, financiamento, entre outros, a Joana relembra constantemente aqueles com quem trabalha daquilo que verdadeiramente importa, a ciência. Adicionalmente, quero agradecer-lhe o cuidado e rigor que teve ao reler este documento (e muitos outros); sobretudo, quero agradecer-lhe a amizade. Quero também agradecer ao Carlos, o meu camarada nas trincheiras que é a ciência para alunos de doutoramento em Portugal. Ao Pedro, à Salete, à Teresa, ao Manel, ao Piochi, à Catarina, à Raquel, ao Ramalhão e a todos os membros e amigos do grupo com quem tive a oportunidade de aprender.

Ao Zé Gui quero agradecer a amizade em tempos de cólera, o orgulho sem preconceito e a amizade sem fronteiras. Ao Navega, por ser o irmão mais velho de todos nós. Ao Coelho, por ser o Coelho. Ao Manolo por ser, com o Zé Pedro, um dos meus mais antigos e melhores amigos.

Ao meu padrinho. Como em muitas famílias, o número de pessoas com Ensino Superior na minha só recentemente começou a crescer. Até ao momento, o meu padrinho é a única pessoa na minha família com Doutoramento. O mais engraçado é que isso não é o mais que lhe tenho a agradecer. Ele não me incitou propriamente a querer ter Doutoramento. Ele foi a pessoa que mais confortável me fez sentir com a vontade de ser cientista e, por isso, deu-me o melhor incentivo possível.

Ao meu afilhado, o Afonso, espero poder fazer por ele tanto como o meu padrinho fez por mim.

Ao Tiago, cunhado é uma palavra estranha, mas amigo é também um termo insuficiente para o que o Tiago é. Entre muitos Legos, jogos de estratégia e tabuleiro, puzzles, pancadaria na garagem, qualquer pessoa teria sorte de ter o Tiago como amigo, eu tenho a sorte de o ter como família.

À Lau. Quando eu tenho dúvidas se consigo fazer alguma coisa, por mais descabidamente ambiciosa que seja, ela não. E não me refiro a aqueles “tu consegues” vazios que, por vezes, as pessoas dizem para tranquilizar os outros sem terem verdadeiramente a certeza. A Lau sabe que eu consigo, sem sombra de dúvidas, mesmo antes de eu saber se é possível. A Lau é a minha casa quando chego a casa.

À Ana. Quando andávamos no colégio, um rapaz vários anos mais velho por várias vezes me azucrinou o juízo – no vernáculo atual poder-se-ia chamar de bullying - e a Ana chegou ao pé dele, encostou-o à parede, e deu-lhe um sermão de fazer inveja ao Padre António Vieira. Desde então, eu sei que terei sempre alguém com quem contar.

À minha mãe. É difícil escrever acerca de alguém que não cabe nas páginas. Alguém que suportou dois filhos sozinha tendo um emprego a tempo inteiro sem lhes permitir abdicar de qualquer oportunidade. Sem exigir nada em troca que não felicidade. É difícil de conceber como alguém se pode desviar tanto dos interesses próprios por outros, filhos ou não. Estudei filosofia, bioquímica, biologia, física, química, matemática, programação, informática, fisiologia, neuropsicologia, inteligência artificial, entre outras, e não consigo explicar. Mas agradeço, e espero vir a ser uma fração da pessoa que a minha mãe é.

Resumo

Tem havido um aumento significativo no investimento e contribuição de ferramentas computacionais para a descoberta de fármacos. A aprendizagem automática tem esculpido um lugar confortável no campo, com particular destaque para o conjunto específico de ferramentas que é a aprendizagem profunda. A sua utilização tem-se mostrado capaz de reduzir custos, acelerar o processo entre o desenho e a produção e limitar o erro humano. De facto, técnicas centradas nos dados têm sido utilizadas para propulsionar muitos passos no processo de desenvolvimento de fármacos. Iterativamente, isto gera nova informação que pode ser reciclada para melhorar soluções já existentes ou permitir o aparecimento de novas.

Uma componente da investigação em desenvolvimento de fármacos foca-se em perceber e modular os componentes moleculares que são alvos dos fármacos. Comummente, estes são proteínas. As proteínas frequentemente contêm aminoácidos específicos que são particularmente propícios a manter a estrutura e função – Hot-Spots (HS). Devido à sua contribuição para o desempenho dos principais papéis proteicos, os HS assumem o cargo adicional de se tornarem localizações privilegiadas para a ligação dos fármacos. Uma parte deste trabalho descreve o SPOTONE, uma ferramenta de previsão de HS a partir, somente, de informação de sequência com elevado desempenho num conjunto de dados independente (accuracy = 0.82, AUROC=0.83, precision=0.91, recall=0.82 e F1-score=0.85)[1].

Embora sejam os alvos farmacológicos mais comuns, as proteínas variam em muitos aspetos, tais como a constituição, a localização e a função. Um conjunto de proteínas destaca-se como sendo de particular interesse para o desenho de fármacos, devido à sua função e especificidade. As proteínas membranares são mediadoras entre o ambiente interno e externo à célula. Como tal, são as guardiãs que permitem a comunicação entre estímulos externos e o funcionamento celular. O MENSAdb caracteriza um vasto conjunto de proteínas membranares, apresentando dímeros manualmente processados para informação útil, tornando-a disponível para consulta.

Outros componentes vastamente abordados na investigação de desenho de fármacos são, sem surpresas, os fármacos. Habitualmente moléculas, idealmente os fármacos interagem especificamente com alvos únicos, limitando a sua interação com outras moléculas biológicas. O DrugTax é uma ferramenta, implementada e distribuída como ferramenta de Python, que foi desenvolvida para facilitar a interpretação de dados de pequenas moléculas. O DrugTax possibilita a caracterização de taxonomia química para obter descritores farmacológicos explicáveis. Adicionalmente, permite análise simultânea de múltiplos compostos para visualização e aprendizagem automática. A caracterização de alvos e fármacos é necessária para a maior parte das tarefas finais no processo de desenho de fármacos, tais como a previsão de interação entre fármacos e alvos, a previsão de reposta a fármacos e a previsão de resposta a combinação de fármacos. A última tem ganho particular interesse sob a forma de previsão de sinergia de combinações de fármacos em linhas celulares de cancro. Este interesse justifica-se pela natureza da doença e dos seus alvos, visto que os perfis de cancro podem variar abundantemente em diversos fatores como tecido, indivíduo, entre outros. Por este motivo, para fazer frente ao cancro é necessário desenvolver soluções flexíveis que possam ser adaptadas

[1] A tradução destes conceitos levaria, quase inevitavelmente, a imprecisões

e otimizadas para cada caso. A sinergia de combinação de fármacos permite isto, pois, ao administrar doses menores dos mesmos fármacos e obter resultados semelhantes ou melhores, permite diminuir a probabilidade de resistência farmacológica e, dessa forma, aumentar a probabilidade de sucesso. O SYNPRED é um conjunto de previsores para previsão de sinergia de combinações de fármacos em linhas celulares. O SYNPRED foi desenvolvido considerado cinco modelos de sinergia de referência, um esquema de validação especificamente desenhado para o efeito e os métodos de aprendizagem automática e profunda mais atuais. O modelo de previsão do SYNPRED com melhor desempenho tenta prever o Combination Sensitivity Score (RMSE, 11.07; MSE, 122.61; Pearson, 0.86; MAE, 7.43; Spearman, 0.87)[2].

Em resumo, ao longo deste trabalho fizeram-se diversos avanços em secções distintas do processo de desenho de fármacos. O presente trabalho resultou em 8 publicações científicas indexadas (5 artigos de investigação original, 1 base de dados e 2 artigos de revisão sob a forma de capítulos de livro), 5 repositórios de GitHub, 3 websites e 1 biblioteca de Python de distribuição gratuita.

Palavras-chave

Desenho de fármacos; Fármaco; Alvo; Proteína; Hot-Spot; SPOTONE; Proteína Membranar; MENSAdb; Aprendizagem automática; Inteligência Artificial; Aprendizagem Profunda; DrugTax; Cancro; Sinergia; SYNPRED.

[2] A tradução destes conceitos levaria, quase inevitavelmente, a imprecisões

Abstract

There has been a significant investment and contribution increase from computational tools to drug discovery pipelines. Machine Learning (ML) has carved a comfortable spot in the field, with a particular highlight for the specific set of tools that is Deep Learning (DL). Their utilization has proven to reduce costs, speed up time from design to production and limit human error. In fact, data-centric techniques have been used to boost many steps of the drug design pipeline. Iteratively, this generates new information that can be recycled into improving already existing solutions or allowing the sprout of new ones.

One part of drug design research is heavily focused on understanding and modulating the molecular components targeted by the drugs. Most commonly, these are proteins. Proteins often feature specific amino acids that are particularly adept at maintaining protein structure and function - Hot-Spots (HS). For their key contribution to proteins' main roles, HS take on the additional burden of becoming optimal drug binding locations. A part of this work describes SPOTONE, a state-of-the-art freely available HS prediction tool from sequence-only information with accuracy, AUROC, precision, recall and F1-score of 0.82, 0.83, 0.91, 0.82 and 0.85, respectively, on an independent testing set.

Although the most common drug targets, proteins vary widely in many regards, such as constitution, location, and function. One set of proteins stands out as particularly interesting for drug design, due to their role and specificity. Membrane Proteins (MP) are mediators between the cell inner and outer environment, as such, they are gatekeepers between external stimuli and cellular functioning. MENSAdb characterises a wide array of MPs, manually curating MP dimers into useful information, making it available for easy consultation.

Other components heavily focused in drug design research are, non-surprisingly, the drugs. Most commonly small molecules, ideally drugs interact specifically with single targets, limiting their interactions with other biological molecules. DrugTax is a tool, implemented and distributed as a Python package, that was developed to facilitate interpretable small molecule data. DrugTax explores chemical taxonomical characterization to deliver explainable drug features. Furthermore, it allows bulk analysis for visualization and ML purposes.

Target and drug characterisation are required for most end-goal drug design tasks, such as Drug-Target Interaction (DTI) prediction, drug response prediction and drug combination response prediction. The latter has gained particular interest as drug combination synergy prediction in cancer cell lines. This added focus traces back to the nature of the disease and its targets, as cancer profiles vary widely among several factors such as tissue, individual, among others. For this reason, to tackle cancer it is necessary to develop flexible solutions that can be adapted and tuned for each case. Drug combination synergy is a venue that allows this, since by delivering smaller dosages of the same drugs and achieving the same or better results, it diminishes the likeliness of drug resistance and thus increases the probability of success. SYNPRED is a set of predictors for drug combination synergy in cancer cell lines. SYNPRED was developed considering five different synergy reference models, a problem-tailored validation scheme and the most state-of-the-art ML and DL methods. The best-performing prediction model in SYNPRED targets the Combination Sensitivity Score (RMSE, 11.07; MSE, 122.61; Pearson, 0.86; MAE, 7.43; Spearman, 0.87).

In sum, throughout this work, several advances were made regarding the different sections of the drug design pipeline. The present work resulted in 8 indexed scientific publications (5 original research papers, 1 database and 2 reviews in the form of book chapters), 5 GitHub repositories, 3 websites and 1 freely distributed Python package.

Keywords

Drug design; Drug; Target; Protein; Hot-Spot; SPOTONE; Membrane Protein; MENSAdb; Machine Learning; Artificial Intelligence; Deep Learning; DrugTax; Cancer; Synergy; SYNPRED.

Index

List of Abbreviations

1D - one-Dimensional
2D - two-Dimensional
3D - three-Dimensional
5-HT2AR - Serotonin 5-HT2A Receptor
$\Delta\Delta$G - change in the change of free Gibbs energy
ADMET - Absorption, Distribution, Metabolism, Excretion and Toxicity
AI - Artificial Intelligence
ANN - Artificial Neural Network
AUROC - Area Under the Receiver Operating Characteristics curve
BLAST - Basic Local Alignment Search Tool
CADD - Computer-Aided Drug Design
CCLE - Cancer Cell Line Encyclopedia
CNV - Copy Number Variation
COVID-19 - COronaVIrus Disease 2019
CRD - Cysteine-Rich-Domain
Cryo-EM - Cryo-Electron Microscopy
CSS - Combination Sensitivity Score
CV - Cross-Validation
DCDB - Drug Combination DataBase
DL - Deep Learning
DNA - DeoxyriboNucleic Acid
DNN - Deep Neural Network
DT - Decision Tree
DTI - Drug-Target Interaction
ECD - ExtraCelullar Domain
ECL - ExtraCelullar Loop
ERT - Extreme Randomized Trees
FAIR - Findability, Accessibility, Interoperability, and Reusability
FDA - Food and Drug Administration
FDR - False Discovery Rate
FN - False Negatives
FNR - False Negative Rate
FP - False Positives
GAS - Glycine, Alanine, Serine
GDSC - Genomics of Drug Sensitivity in Cancer
GPCR - G-Protein Coupled Receptor
GNN - Graph Neural Network
HMM - Hidden Markov Model
HTS - High-Throughput Screening
HS - Hot-Spots
HSA - Highest Single Agent
ICL - IntraCellular Loop

KEGG - Kyoto Encyclopedia of Genes and Genomes
kNN - k-Nearest Neighbor
LBDD - Ligand-Based Drug Design
LD50 - Lethal Dose 50%
MAE - Mean Absolute Error
MDS - MultiDimensional Scaling
ML - Machine Learning
MP - Membrane Protein
MD - Molecular Dynamics
MDS - MultiDimensional Scaling
MSA - Multiple Sequence Alignment
MSE - Mean Squared Error
MW - Molecular Weight
NB - Naïve Bayes
NCI-ALMANAC - National Cancer Institute - A Large Matrix of Anti-Neoplastic Agent Combinations
NLP - Natural Language Processing
NMR - Nuclear Magnetic Resonance
NPSA - Non-Polar Surface Area
NPV - Negative Predictive Value
NS - Null-Spots
PCA - Principal Component Analysis
PDB - Protein DataBank
POPC - PhOsPhatidylCholine
PPI - Protein-Protein Interactions
PPV - Positive Predictive Value
PSA - Polar Surface Area
PSSM - Position-Specific Scoring Matrix
QM - Quantum Mechanics
QSAR - Quantitative Structure-Activity Relationship
RF - Random Forest
RGB - Red Green Blue
RNA - RiboNucleic Acid
ROC - Receiver Operating Characteristics curve
R&D - Research and Development
SASA - Solvent Accessible Surface Area
SBDD - Structure-Based Drug Design
SMO - SMOothened
SVM - Support Vector Machine
TM - Text Mining
TN - True Negatives
TNR - True Negative Rate
TP - True Positives
TPR - True Positive Rate

UMAP - Uniform Manifold Approximation and Projection
URL - Uniform Resource Locator
USA - United States of America
vHTS - virtual High Throughput Screening
VS - Virtual Screening
XGB - eXtreme Gradient Boosting
ZIP - Zero Interaction Potency

List of Figures

List of Tables

List of Equations

Chapter 1: Introduction

1.1. Drug Design and Development

Drug design, development and discovery is the inventive process of finding new drugs based on the knowledge of a biological target. In the most basic sense, drug design involves the design of molecules capable of interacting with one or several molecular targets, usually by binding through complementary shapes and/or charge [1] (**Figure 1**). According to the United States Food and Drug Administration (FDA) glossary, drugs are substances whose activity towards a biological target has been identified to cure, mitigate, treat, prevent, or diagnose a disease [2]. Currently, over 90% of all approved drugs are small molecules with Molecular Weight (MW) below 900 Da [3]. This broad definition allows for a wide array of possible drugs, both organic and inorganic, with various chemical properties and groups.

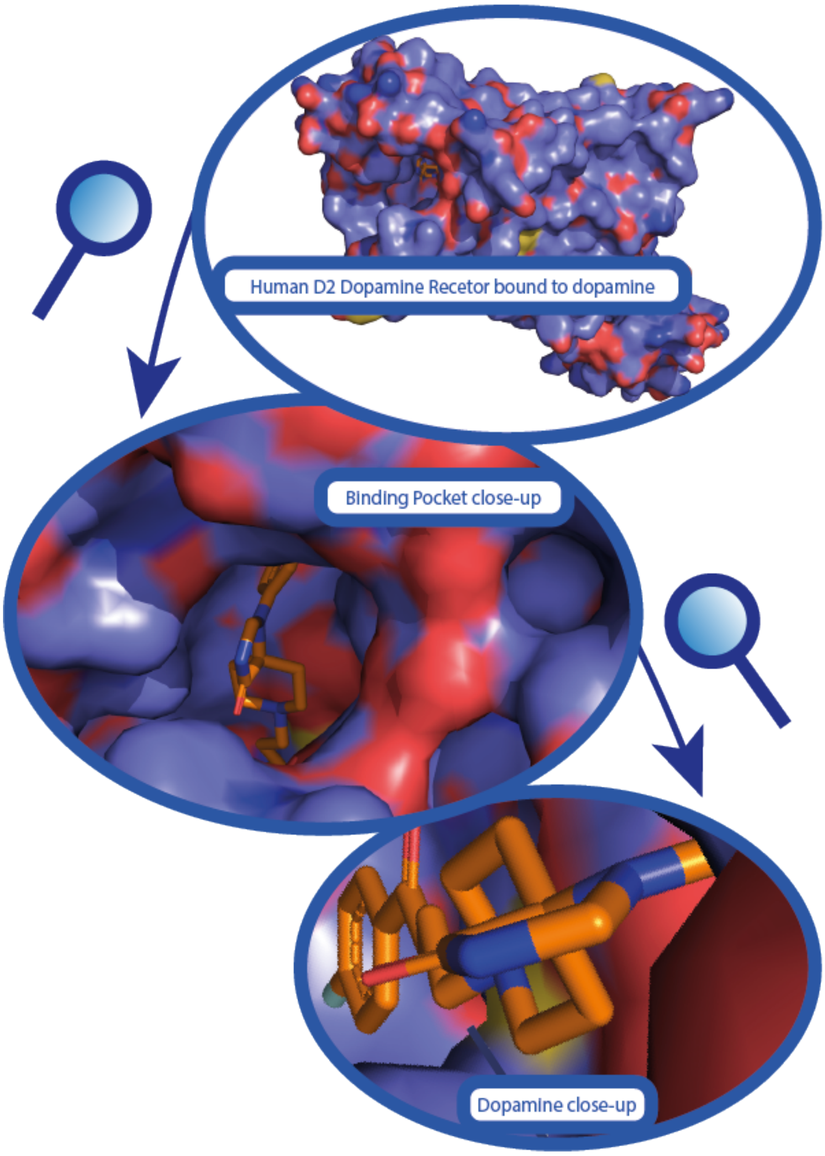


Figure 1: Representation of the binding of the drug – spiperone – to the human dopamine D2 receptor with spiperone [4], (created with PyMOL [5])).

A typical drug design pipeline uses a combination of computational, experimental, translational, and clinical models included in the following steps: basic research, preclinical development, clinical trials and, finally, drug approval [6, 1](**Figure 2**). The usage of computational approaches has been pivotal to optimize findings concerning the first steps of basic research. This covers subjects such as target identification, active compound screening and lead optimization. Despite advances in biotechnology and the understanding of biological systems, the Research & Development (R&D) pipeline is still a time-consuming, expensive and difficult process. The usage of computational approaches in the first step of basic research (e.g., target identification, active compound screening, and lead optimization) has been pivotal to minimize some of these disadvantages by selecting and prioritizing compounds for further *in vivo* and clinical testing. Optimizing these steps maximises the chance of success in the subsequent, more expensive ones.

The application of computational approaches can branch out in two different directions: new drug development [7], or more recently, the scaffold repurposing of existing ones (also called drug repositioning or drug reprofiling) [8]. When considering drug repurposing [9], the benefits of *in silico*-boosted research are even higher as it reutilizes previous research by revisiting already approved or investigational drugs with the aim of finding new therapeutical approaches different from the original ones [9]. Therefore, the latter makes the R&D pipeline faster and less expensive due to the possibility of skipping parts of the process, which lends it additional interest [10].

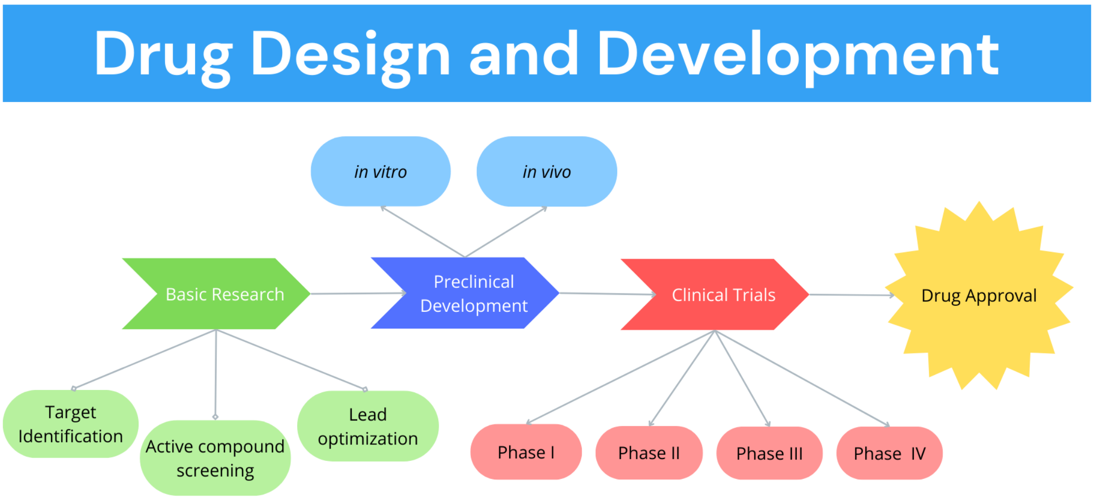


Figure 2: Drug design and development pipeline representation [11].

While many fields of science emerge mainly from publicly funded research, drug design has long been fuelled also by private companies. Between 1985 and 2005, 91% of FDA approved drugs were associated with a patent detained by the private sector. However, almost half of these patents cited a public research paper or patent [12]. The private sector is leveraging the generated knowledge from the public sector to meet market needs, denoting the importance of valuing both public and private research. Although the private sector clearly dominates the development, the research is still largely driven by academia.

According to a recent market analysis provided by Signify Research, there has been a significant increase in venture capitalists' funding (i.e., investor – person or company - that provides young companies with capital in exchange for equity) for the use of Artificial Intelligence (AI) in drug design. Particularly, the United States of America (USA) report the highest funding, going over 7.6 billion dollars between the 2011-2021 period (about 71% of the worldwide total), also displaying the highest number of deals, at 197. Interestingly, China was the country that, in the same period, invested most in start-ups, with an investment of 136.9 million dollars, also exhibiting the highest average funding per deal (47.6 million dollars). There was a slowdown in funding rounds that coincides with the COronaVIrus Disease 2019 (COVID-19) in the years 2019 and, particularly 2020, which dropped below one billion dollars, while in 2018 it was close to reaching the two billion dollars mark. However, in 2021, in the post-pandemic, drug design funding gained renewed traction, for the first time surpassing the three billion dollars mark. All this information shows an increased interest in AI applications with no signs of slowing down in the immediate future. However, investors are taking a closer interest in end-to-end solutions and slightly shifting from inflexible drug design pipelines that yield a single drug and have limited further uses [13].

1.1.1. Computer-Aided Drug Design

Computer-Aided Drug Design (CADD) leverages computational methods (e.g., computer modelling and simulation techniques) to accelerate and render more efficient the design and development of new lead compounds [14]. CADD is frequently coupled with High-Throughput Screening (HTS), allowing quick and efficient screening of datasets to identify the most likely effective compounds against a given target [15, 16, 17]. The computer-aided search for a compound with the potential to be a useful drug candidate able to modulate a specific target involves screening major chemical libraries in a process called Virtual Screening (VS) [18]. When combining HTS with VS, to perform computer-enhanced compound screening, emerges the combined concept of virtual High Throughput Screening (vHTS) [19]. vHTS is vital to CADD, as it requires minimal compound design and can yield multiple compound candidates. These methods have proven to be very useful to arrive at a small set of candidates [20, 21].

Sliwoski et al. define three major CADD applications: (i) selection of sets of active compounds from chemical libraries, that can be further submitted to experimental assays; (ii) optimization of lead compounds (in terms of affinity, metabolism, or pharmacokinetic properties); and (iii) design of novel compounds [22]. It can be added to these a fourth application: (iv) understanding of Drug-Target Interactions (DTI) to detect new targets for existing drugs. Conventionally, CADD approaches are broadly classified into Ligand-Based Drug Design (LBDD) and Structure-Based Drug Design (SBDD).

LBDD uses known ligands and follows the premise that similar ligands display similar properties, thereby binding to similar proteins. As such, through the chemical structure, the aim is to understand which are the functional groups (and why they are) responsible for binding to the target towards the development of novel analogues [23, 22]. LBDD is extremely useful when the target's three-dimensional (3D) structure is not available [24, 22]. The most used LBDD methods are Quantitative Structure-Activity Relationship (QSAR) and ligand-based pharmacophore modelling. QSAR evaluates the relationship between a structure and its biological activity to predict the activity of analogues. Pharmacophore modelling depends on common properties between ligands with the same biological activity [24]. LBDD models' performance is limited by the number of ligands used in the process, meaning that the lower the number of ligands used to build the model, the lower its accuracy [22].

Contrarily to LBDD, in SBDD, the targets' 3D structure is required, which is usually obtained through experimental approaches such as X-Ray Crystallography and Nuclear Magnetic Resonance (NMR) or, more recently, via Cryo-EM [24]. When the targets' 3D structure is not available, it can be predicted by several *in silico* approaches (such as homology modelling and *de novo* modelling). Homology modelling can be used to develop a model of the target with an unknown structure (e.g., Modeller [25], i-Tasser[26], Swiss-model [27]). For that, we use a known template structure with enough sequence similarity with the target, which should be as high as possible to minimize the model's error. This approach assumes that similar structures hold similar functions and binding site conservation [24, 28]. *De novo* approaches can include AlphaFold [29] and Rosetta [30, 31], as they generate the protein structure, taking only the amino acid sequence as the starting point.

The main methods used for SBDD are docking and Molecular Dynamics (MD). Docking simulations identify promising drug candidates for a given target by using their 3D structures and studying structure-activity relationships [32, 21]. These simulations also allow the ranking of drug-target candidates based on binding affinity estimates [32, 33]. VS complements these methods, by narrowing down both the most appropriate ligands and targets [24]. Docking and VS are not mutually exclusive. It is common to use VS on large sets of docked structural models [34]. MD simulations provide further insights into protein-ligands binding [35]. More recently, DTI prediction has become possible using HTS information and computational tools. DTI approaches interact (often even overlap) closely with CADD (**Figure 3**) [36, 37].

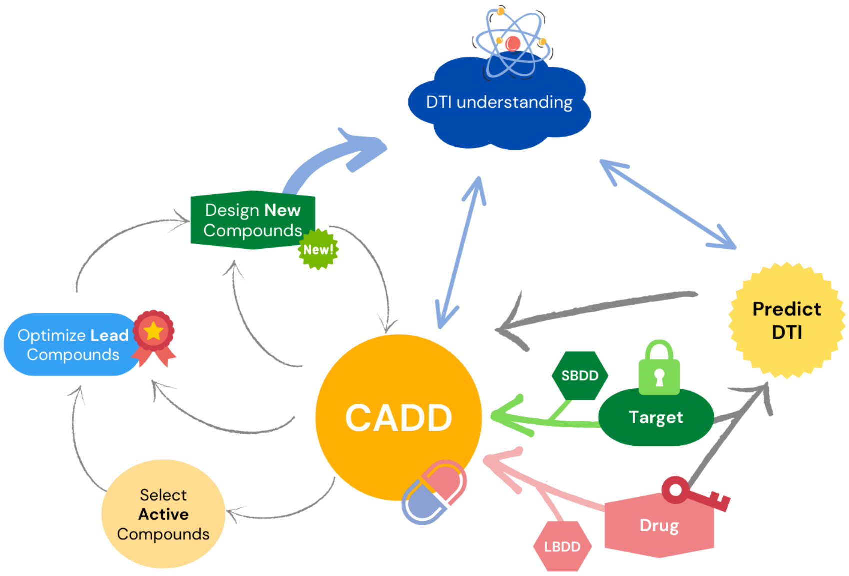


Figure 3: Importance of CADD and how it can mediate different phases of drug design and development. It is also shown how DTI understanding is expanded and helps expand both CADD and DTI prediction approaches [38].

1.1.2. Drug Characterization

Drugs are often small molecules that bind to specific targets. PubChem[39] registers over 112 million compounds and 298 million substances (November 2022). Currently, Drugbank [40, 41] lists 12.012 drugs out of which 2.729 are FDA-approved. ChEMBL [39] reports over 2.3 million compounds and 14.000 drugs. The abundance of drugs or drug-like compounds is evidently overwhelming, thus demonstrating the importance of using automatized approaches.

It is advantageous to be able to provide a deeper understanding of the drugs' characteristics while also being able to mathematically describe them [42]. Feature extraction is a focus when considering computational-based approaches, as it is a crucial and a necessary step for any algorithms to distinguish between the different patterns within the data. Under the scope of drug design, several packages have been developed to this end. Open Babel [43]is a broad example, providing a set of chemical tools to describe and manipulate drugs and other small molecules. More recently, packages such as Mordred [44] and ChemmineR [45] have also been developed. These packages usually determine straightforward features – designated molecular descriptors – which must be representative and form a unique characterization for the molecule, ensuring no two molecules can be represented with the same set of features. Such descriptors usually include MW, charge, pH, hydrogen bonds, occurrence of specific atoms, aromaticity, and others. In addition, they can expand to more abstracted features involving hydrophobicity and topological indexes, as well as 3D-derived features [44].

Alternatively, different types of approaches have been gaining attention towards faithfully representing the molecules' 3D information while simultaneously providing additional insights, such as the ones based on graph [46, 47] and voxel-based [48] drug representations. The chemical characterization of small molecules is a cornerstone for further understanding and essential for bulk data approaches, and as such this can used for data grouping and feature extraction, some of the characterizations stemming from the root biochemical definitions [49].

1.1.3. Absorption, Distribution, Metabolism, Excretion and Toxicity

From the first tests to final approval, getting a single drug to the market takes a long time and involves many resources [50]. However, only a few drug candidates that reach clinical trials are approved for human use, representing a substantial waste of time and money [51]. Most of the issues related to this enormous failure rate in drug development are associated with undesirable pharmacokinetics and toxicity. Therefore, it has been widely accepted that Absorption, Distribution, Metabolism, Excretion and Toxicity (ADMET) properties should be considered in the early stages of drug discovery to increase drug development success, especially later, during the clinical phase [52]. Moreover, post-marketing safety issues have led to several drug withdrawals and unexpected mortality and morbidity concerns boosting the need to apply ADMET prediction even after drug approval. Computational approaches emerged as crucial tools to evaluate ADMET properties in a cheaper but still efficient way [53]. Several ADMET-related databases, which incorporate pharmacokinetics and toxicity parameters can be used for shape and/or pharmacophore screening to obtain further information about bioactivity on similar models that match the input query compound [52]. Furthermore, databases like ADME Database [54], SuperToxic [55], PKKB [56], CompTox [57] and DSSTox [58] were reported as reliable and comprehensive sources for the training and development of ADMET prediction models. Besides they are also useful to predict drug metabolites and toxicity, which are ultimately responsible for drug efficacy and safety [53, 59].

Traditionally, several *in silico* ADMET approaches tend to establish a relationship between different molecular descriptors and ADME properties by applying statistical models or prediction algorithms [60]. Among them is the widely used SwissADME tool, a free web tool able to assess small molecules' pharmacokinetic profile, alongside their physicochemical properties and drug-likeness [61]. Other methodologies were applied to study the ADMET profile of drug candidates, particularly its toxicity, by testing several drug features combined with target-based predictions and QSAR studies. QSAR models were mainly applied to assess several drug safety endpoints, such as Lethal Dose 50% (LD50), tissue-specific toxicity, and skin and eye irritation [62]. PrOCTOR [63] and TargeTox [51] are two examples of freely target-based toxicity prediction tools based on QSAR models.

Specific models for metabolism prediction were also developed in recent years due to their recognized impact on the pharmacokinetics and pharmacodynamics of xenobiotics and their derivatives [59]. However, most of these models have a limited scope, coverage, and performance. To overcome this issue, a freely available software package, BioTransformer [64] was developed allowing both metabolism prediction and compound identification. Another interesting tool, MetaTrans [59], predicts human metabolites and associated features from small molecules, not to be confused with its homonymous tool from 2016 [65], which is an open-source pipeline for metatranscriptomics. An additional approach with the same aim was developed by first using chemical reaction data to pretrain a transformer model. This model was further fined-tuned using freely available databases of human metabolic reactions, which includes metabolism not only of xenobiotics but also of endogenous molecules, comprising the full spectrum of enzyme classes [59]. To conclude the previous protocol, an ensemble prediction model combining the output of several fined-tuned models and considering different metabolites was built. Authors showed that their method displayed an equivalent performance in comparison with other drugs metabolite prediction approaches, such as SyGMa [66], GLORYx [67] and BioTransformer [64], considering the major enzyme families screened. Furthermore, it seems able to identify metabolites using fewer common enzymes [59].

1.1.4. Drug activity evaluation

Drug activity can be evaluated using experimental techniques by testing them against a panel of cells. Among other metrics, this activity can be measured by using minimal concentration for 50% activity inhibition (IC50). This data is often available in large databases such as the Cancer Cell Line Encyclopedia (CCLE) [68], Genomics of Drug Sensitivity in Cancer (GDSC) [69] and the National Cancer Institute - A Large Matrix of Anti-Neoplastic Agent Combinations (NCI-ALMANAC) [70]. Besides direct drug response data these databases are an invaluable resource for non-computable data. The Cancer Cell Line Encyclopedia (CCLE) [68] is a thorough example of a repository of biological samples' information, covering data from gene mutation, RNA expression, chromatin profiling to methylation, among others.

1.1.5. Drug targets

Although the drugs are heavily highlighted when talking about the subject, the target(s), - the biological entity or entities with which the drug interacts – are equally important. These interactions are a topic of extensive research, commonly referred to as DTIs [33]. Targets are heavily involved with sets of disease-associated biological molecules [71]. Most of the time, these molecules are deeply connected through several kinds of interaction networks, forming the interactome [72, 73]. The importance of the interactome is tied to its biological signalling pathway control, which is directly linked to organism functioning and, conversely, malfunctioning – herein – disease [74, 75]. Furthermore, the same drug target can participate in multiple intracellular signalling pathways known to trigger several cellular and physiological consequences, further deepening the complexity with each component [76] (**Figure 4**).

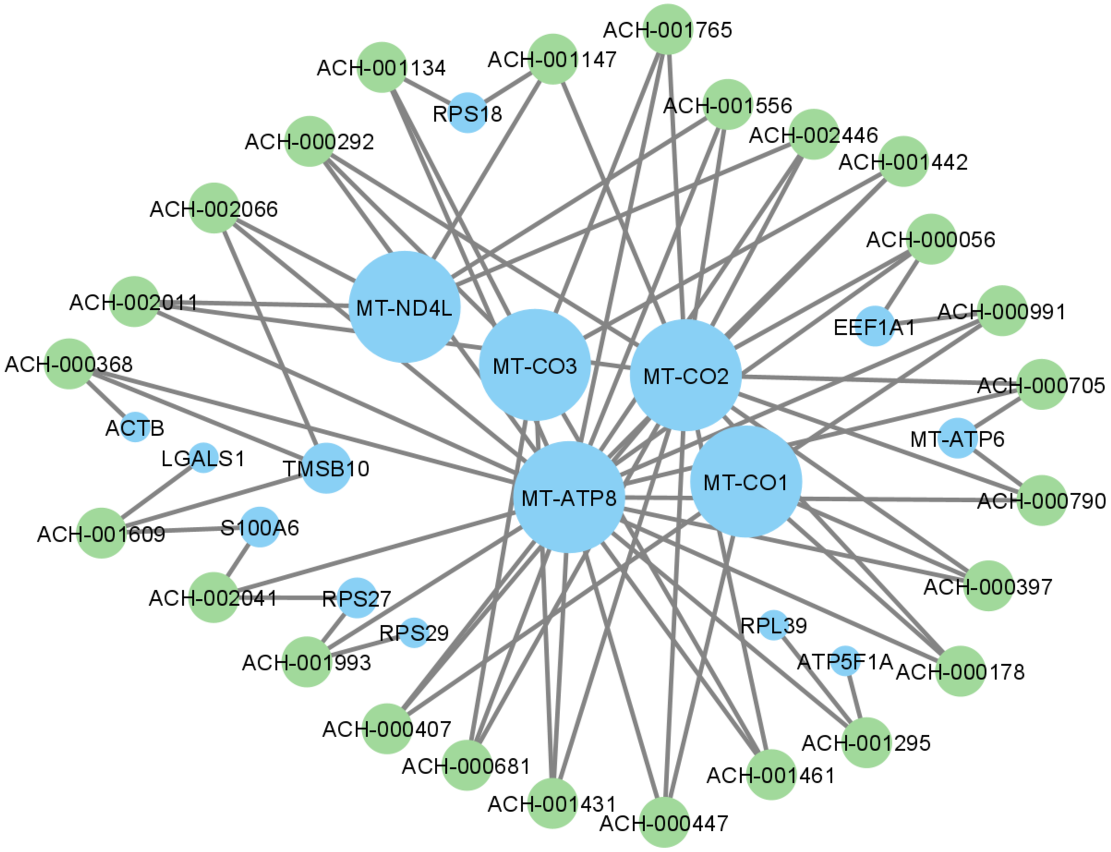


Figure 4: Drawn in Cytoscape [77], this image depicts the three most expressed genes of 25 randomly chosen cell lines from the CCLE RNA Expression dataset [68]. Genes are represented in blue and cell lines in green; circle size is proportional to the number of connections, meaning the genes that are most expressed in more cell lines are represented with larger circles.

The action of the drugs is meant to trigger effects on molecules or their interactions in disease-associated networks while attempting to affect minimally other unrelated targets [78, 79, 80]. Several studies have shown increasing evidence that drugs acting over the same target can have different physiological effects since they modulate different intracellular signalling pathways [81]. The importance of pathway mapping was explored in recent years. Diez-Alarcia and colleagues determined the probability of a molecule interacting with different targets by considering pathway information. The authors focused on predicting pharmacological compounds with affinity for Serotonin 5-HT2A Receptor (5-HT2AR). The experiment results indicate that some drugs, which previously behaved as selectively activating or inhibiting 5-HT2AR activity, can generate different effects under different circumstances. The conclusions for the 5-HT2AR showed that this computational approach could help design new antipsychotic drugs with better efficacy and tolerability profiles [82].

The most common drug targets are proteins, although there are circumstances in which DNA, RNA, lipids, and other biological molecules can also be targets. Santos et al. 2017 showed that 95.86% of the FDA-approved drug targets are proteins, with 84.73% corresponding to small molecules [83, 84]. When highlighting specific protein types, Rask-Anderson *et al.* [85] reported that G-Protein Coupled Receptors (GPCRs) make up 44% of drug targets, enzymes 29% and transporter proteins 15%. Overington et al. [86] found that over 50% of drugs target either GPCRs, nuclear receptors or ion channels. Across even more studies, the predominance of Membrane Proteins (MPs) as drug targets was well established [87].

The impact of protein activity stems, first and foremost, from their variety and amount. The collective of the human proteins is called proteome, and its size is debated to be between 20.000 to several millions, depending on whether gene splicing and post translation modifications are taken into consideration [88]. Proteins play a fundamental role in numerous (or almost all) biological processes. The complexity and dynamics of these biological processes cannot be understood without proper knowledge of the proteins that take part in them. Although they play a broad role in cell structure and activity [49], the broader messenger role they play is a major reason why they retain the number one spot as drug targets [84, 76]. It should be noted that for a protein to have any potential as a drug target it must be druggable. Most druggable proteins possess folds that favour interactions with small drug-like molecules, be they endogenous or extraneous, and therefore is one that contains a binding site. These binding sites are expected to have certain characteristics that enable high affinity site-specific binding by the drug-like molecule [87]. This, however, does not encompass drugs that target interacting proteins, which can depend on the interface formed between each other, rather than simply their individual 3D folding [89].

The number of experimentally solved 3D protein structures has increased in the last few years due to technological improvements both in the crystallography and Cryo-Electron Microscopy (Cryo-EM) fields. Since many *in silico* methods require prior 3D knowledge of protein and/or drug, this increment boosted not only the number of structural and dynamical studies but also broader approaches that use Artificial Intelligence (AI) algorithms to extract and process all the information available in these structures. Known 3D structures are generally deposited in the Protein Data Bank (PDB) [90].

Protein activity is also dependent on their interactions with other macromolecules such as nucleic acids, membranes, glycans, or other proteins. The interactions between proteins are undoubtedly the most common type of interaction, with over 650.000 Protein-Protein Interactions (PPIs) in the human organism, which dynamically contribute to the understanding of cellular function and organization [91]. Detailed characterization of PPIs is key, as their dysregulation through changes in the structure of each of the interacting proteins, the environment, or other factors can be determinant in several diseases such as cancer, neurological disorders, metabolic diseases, and others [92], this needs to be taken into account in contrast with attributing such issues to changes in the activity of single proteins. As such, PPIs involved in disease pathways have become popular targets for the development of new diagnostic and therapeutic strategies [93, 94].

1.1.6. Membrane proteins as targets of interest

The lipid membranes are essential structures to life and assume many functions within cells, such as mobility and nutrient intake, energy transduction, biosynthesis, and immunologic and nerve responses [95]. These actions are often controlled or mediated by ubiquitous MPs (**Figure 5**), with 20–30% of most organisms' genes coding for this type of protein [96]. Given the relevance of MPs, the membrane proteome warrants particular attention, as it is said to account for between 15% to 39% of the human proteome [97, 96]. They play essential roles in ion and nutrient transport, communication with the extracellular environment [98], nutrient uptake, toxin and waste product clearance, respiration and signalling [99]. MPs also regulate a lot of the communication from inside to outside the cell as well as membrane-bound subcellular structures [100, 101, 49].

Experimental characterization of MPs is difficult as the membrane imposes obstacles to its manipulation, notably its purification and crystallization. Despite these difficulties, progress in experimental techniques has generated a growing body of structural information. For instance, the mpstruc—Membrane Proteins of Known 3D Structure—database from the Stephen White Laboratory at UC Irvine (available at `http://blanco.biomol.uci.edu/mpstruc/`) [102] now lists 1509 unique MP whose 3D structures are known (as of November 3, 2022). The number of experimentally solved 3D protein structures has also increased in the last few years due to technological improvements both in the crystallography and Cryo-Electron Microscopy (Cryo-EM) fields.

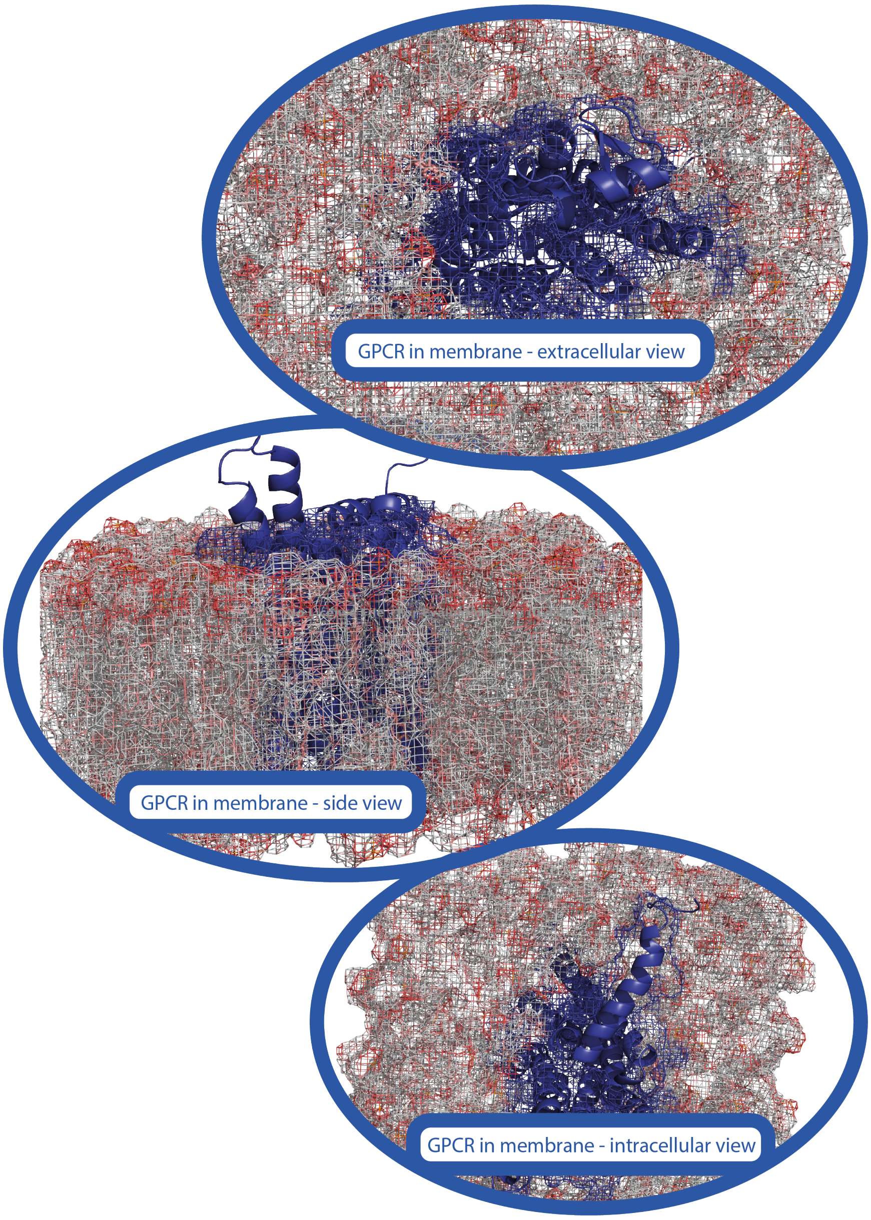

GPCR in membrane - extracellular view
GPCR in membrane - side view
GPCR in membrane - intracellular view

Figure 5: Human β2-adrenergic G-Protein Coupled Receptor (PDB identifier 2rh1 [103]) embedded on a cholesterol and PhOsPhatidylCholine (POPC) membrane extracellular (top), side (middle) and intracellular (bottom) views (created with PyMOL [5]).

Almén *et al.* 2009 categorize MPs into 225-234 families, four functional groups and 21 subgroups (**Table 1**) [97]. GPCRs represent the largest subgroup across all the functional groups and both families [104], accounting for 26.51% of the total number of analysed MPs (which is higher than any other MP subgroup) [97]. On top of this, it has been reported that 35% of all FDA-approved drugs target GPCRs [105]. Some reasons make this subgroup attractive drug targets besides their large number. For example, their intracellular coupling with G-proteins and arrestins induces the activation of various downstream signalling pathways [106, 107]. Although there are many GPCRs they share structural motifs, seven TransMembrane α-helices (TM1–7), separated by three loops on each side of the cellular membrane, three Extracellular Loops (ECL1–3) and three Intracellular Loops (ICL1–3) [108]. They also possess an extracellular N-terminus and a C-terminus located in the intracellular space [109]. Despite all the similarity between GPCRs, they are individually modulated by a very broad spectrum of ligands, such as small molecules like dopamine, sub-atomic particles such as photons or even larger molecules such as other proteins [110].

Table 1: MPs distribution and categorization according to Almén *et al.* [97]

Families	Functional groups	Subgroups	MPs	Group MPs
151	Receptors (63)	GPCRs	901	1.352
		Tyrosine Kinase	72	
		Immunoglobulin	149	
		Scavenger	63	
		Other	167	
	Transporters (89)	Channels	247	814-817
		Solute carriers	393	
		Active	81	
		Other	51	
		Auxiliary	42	
	Enzymes (7)	Oxidoreductases	123	533
		Transferases	194	
		Hydrolases	178	
		Lyases	17	
		Isomerases	8	
		Ligases	7	
		Varied	6	
74	Miscellaneous (3)	Ligands	57	697
		Other	272	
		Structural/Adhesion	187	
		Unknown function	181	
Total: 225-234	Starting at 19.523 protein-coding genes, of these, only 5.369 were valid MPs. Excluding unclassified proteins 3.145-3.399 MPs remained.			

Two GPCR-specific databases are currently available: the G Protein-Coupled Receptor database (GPCRdb) [111] and the GPCR-EXP [112], providing more organized and detailed information about the currently available structures. Both GPCRdb and GPCR-EXP provide structures from homology model protocols for receptors, or different activation states of receptors, that are still not available [112, 111].

There are a variety and abundant number of MPs in monomeric form (individual units), although they frequently assemble as dimers or even higher-order oligomers. These higher-order assemblies can have specific roles that do not necessarily coincide with those of their monomeric constituents [113, 114]. This makes the structural biology of MPs even more complex and demands the development of new experimental and theoretical methods to elucidate their function. Dimers or higher-order assemblies of MPs are often the subject of computational studies through MD, but also alternative venues, since MD can be very computationally expensive due to the large size of the biological system implied [101].

1.1.7. Protein hot-spots as focalized targets

Alanine scanning mutagenesis allows the characterization of protein Hot-Spots (HS) as amino acid residues that upon alanine mutation generate a change in binding free energy (ΔΔGbinding) higher than 2.0 kcal mol^{-1}, in opposition to Null-Spots (NS), which are unable to meet this threshold (**Figure 6**) [115, 116, 117, 118, 119, 120]. Protein HS are typically conserved residues or clusters of residues that have been identified as crucial for the interaction and stability of proteins with other molecules (e.g., proteins, DNA) [119]). Drugs can be designed to specifically interact with these regions in other to either activate or inhibit the protein's function. Besides, HS are key elements in PPIs and, as such, fundamental for a variety of biochemical functions. The disruption of these interactions can alter entire pathways and is of interest to therapeutic approaches [121, 119]. These residues are also known to be important for protein dimerization [122]. Furthermore, HS tend to be associated with the binding of small ligands, hence becoming ideal subjects of study on target proteins for drug design approaches [123, 124, 125].

The time and resource expenditure needed for protein HS/NS determination favours the usage of computational approaches that expedite this process. By deploying current tools, it is possible to leverage the relatively low and sparse amount of data available on the subject (less than 1.000 samples) [126] and build methodologies able to fully characterize a protein regarding its HS/NS profile. Furthermore, these methods have the advantage of being translatable to previously unseen proteins [127, 128, 119, 126, 129].

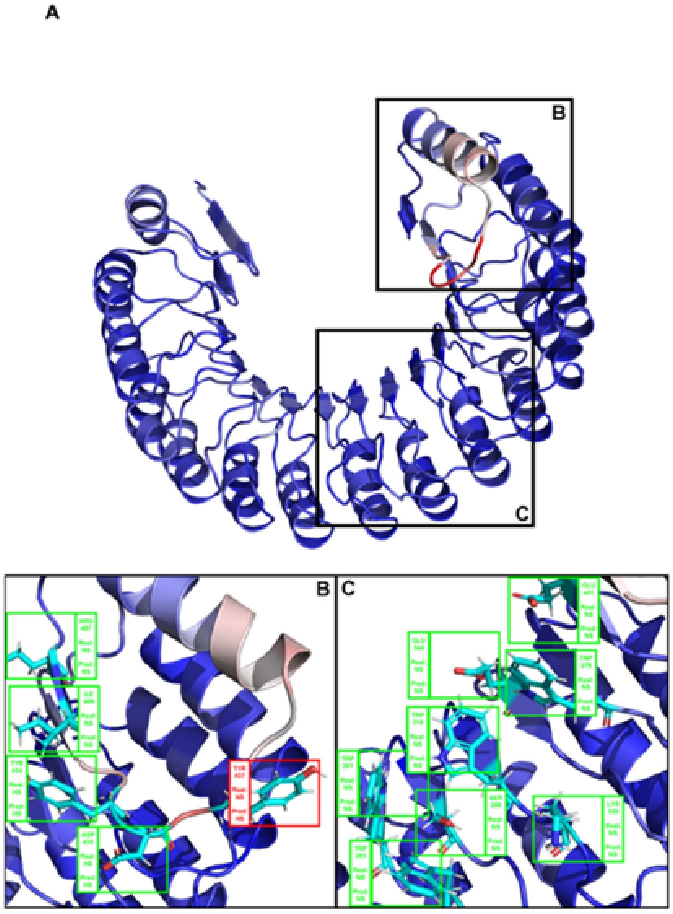


Figure 6: (A) Structural representation of the complex between angiogenin and a ribonuclease inhibitor: PDB identifier 1a4y [130]. Brighter red colours were attributed to residues with a higher probability of being classified as HS. (B, C) Close ins of all interfacial residues for which there is an experimental ΔΔGbinding value, and as such a HS/NS classification. Green boxes represent correctly predicted residues, whereas red boxes represent incorrectly classified residues.

1.2. Machine Learning Fundamentals

Table 2: Key concepts in Machine Learning

Algorithm	Step-by-step procedure required to perform a task or reach a solution/goal to a given problem or question.
Artificial Neural Networks	Prediction models that take inspiration from their biological counterparts, neuronal networks. They display complex webs of perceptrons – neurons – whose joint and layered functioning allows for the fulfilment of a wide variety of prediction tasks.
Classification task	Type of supervised learning approaches in which the algorithm is trained with labelled discrete datasets.
Confusion Matrix	Matrix with the number of true positives, true negatives, false positives, and false negatives used to evaluate the performance metrics of classification-based ML approaches.
Correlations	Scale independent performance evaluation metrics for regression-based ML approaches.
Deep Learning	Artificial Neural Networks models with more than one hidden layer that can serve a wide variety of purposes.
Dimensionality Reduction	The process of lowering the number of variables in the feature space.
Error function	Scale dependent performance evaluation metrics for regression-based ML approaches.
Feature	Information use to describe samples.
Feature space	The totality of all the features in a dataset.
Generalizability	A model's ability to accurately predict previously unknown/unseen features.
Instances	Data samples.
Label	Information associated with the instance that is, in supervised learning approaches, the target variable (i.e., what the model predicts).
Machine Learning	The process of getting computers to act without being explicitly programmed how to do so. A field of Artificial Intelligence that focuses on the development of algorithms that can learn from data and make predictions or decisions based on that learning.
Missing Values	The lack of information when characterizing data instances.
Programming Language	A language with rules - syntactic and semantic - as well as lexical elements, that allow the passage of instructions from the programmer to the computer.
Regression task	Type of supervised learning approaches in which the algorithm is trained with discrete continuous datasets.
Sample	The minimal unit that can be fed to a ML model.
Software	Set of instructions delivered to the computer hardware to achieve an objective.
Software 2.0	The designation of data-boosted software programming.
Supervised Learning	Development of prediction models using labelled data.
Target Leakage	The unwarranted passage of label-associated information to the feature space.
Unsupervised Learning	Development of prediction models using unlabelled data.

Machine Learning (ML) is a field of AI that focuses on training algorithms to learn from data and make predictions or decisions based on learning without being explicitly programmed. The goal of developing a ML model is to create a system that can automatically improve its performance on a specific task over time [131]. ML approaches, however, have roots on a less advanced yet broader, concept – algorithm. An algorithm is a step-by-step procedure required to perform a task or reach a solution/goal to a given problem/question. Although this is a concept deeply associated with computer programming, it existed long before the existence of modern-day computers, dating back to the IX century, from the Persian mathematician whose latinized name originated the term “algorithm” - Muhammad ibn Mūsā al'Khwārizmī [132].

The invention of computers coupled with algorithms boosted the speed and precision at which we can solve some tasks. By enabling the imputation of rules as well as their interpretation and chaining, the concept of software emerged as the set of instructions delivered to the hardware (the computers' physical components) via programming languages. Although the emergence of these concepts can be traced back to the early XIX century, they gained real momentum in the mid-1940s, when computers started to be viable [133]. Only more recently, a new approach named software 2.0 has emerged by leveraging ML to build new software. In contrast to traditional software, which is based on a set of fixed instructions, software 2.0 is built using ML algorithms that can automatically learn and improve their performance over time. This allows software 2.0 to be more adaptable and flexible than traditional software, and to make better predictions and decisions based on data (**Figure 7**) [134]. Besides software development, ML has been a valuable tool for many computational biology fields since it potentiates data analysis, text mining, DDD and many more (specific applications of ML models to biology will be further discussed in section 1.3.).

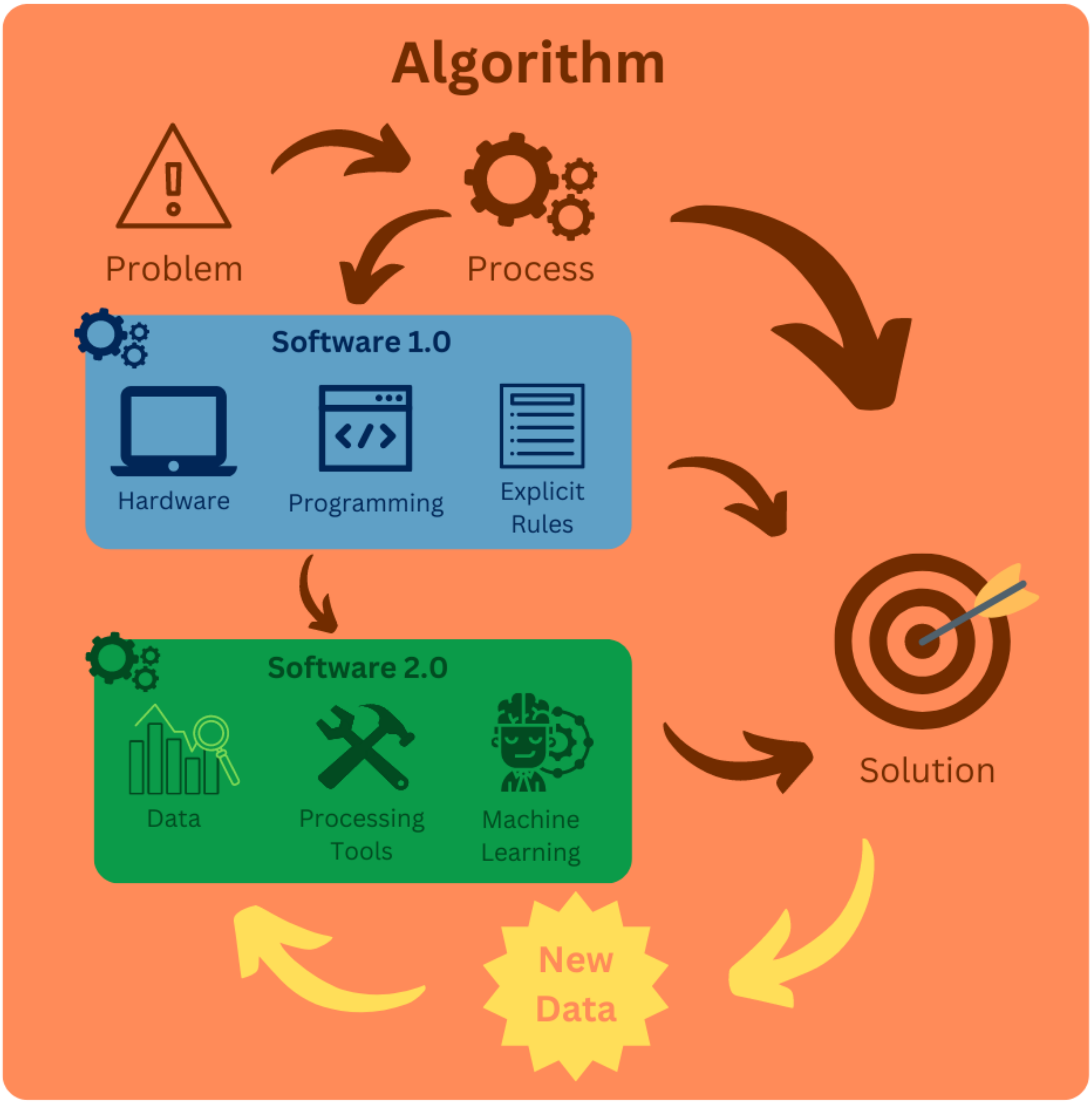


Figure 7: Representation of the relationship between algorithm, software, and Machine Learning [135]

The development of a ML model usually involves some steps that should be followed to assure the quality of the predictions (some of them will be deeply discussed in the next sections). The first step is defining the problem and the goal that the model will solve. This involves understanding the biological context and the desired outcome of the model. ML methods are generally divided into supervised and unsupervised learning [136], the two most used in biological fields (**Figure 8**). There are however other relevant approaches such as reinforcement [137] and semi-supervised learning [138].

Firstly, the data that the model will be trained on must be collected. The data must be expressed in the form of meaningful information that should represent samples according to the main objective [33, 83]. It is important to have a dataset as comprehensive and large as possible, to create the circumstances for the algorithm to find patterns and build models that can recognize the relationship between the data and perform a specific task [139]. After that, pre-processing is a crucial additional step since it involves cleaning the data to remove any errors or inconsistencies and transforming it into a format that the model can use. Once the data is ready, the next step is to select and train the model, something that demands high computational power, even more, if the process must be time effective. This involves choosing a model type and training it on the prepared data.

There are many different types of prediction models, including Decision Trees (DT), Support Vector Machines (SVM), Artificial Neural Networks (ANN) (explored in more detail in section 1.3). Some approaches are displaying consistently positive results for a wide array of problems, such as Deep Learning (DL) [140], which comprises the usage of a wide array of Artificial Neural Networks (ANNs) based methods, has taken the spotlight on several occasions, when discussing ML [141, 142]. Furthermore, prediction models can be combined through ensembles, in which two or more of the best ML algorithms can be merged through a system of voting to form a unique predictor that can, ideally, outperform the individual predictors [143]. However, there is not a perfect model to fit all possible problems. Thorough knowledge of models and data is the best way to maximize the use of both in the development of a good predictor.

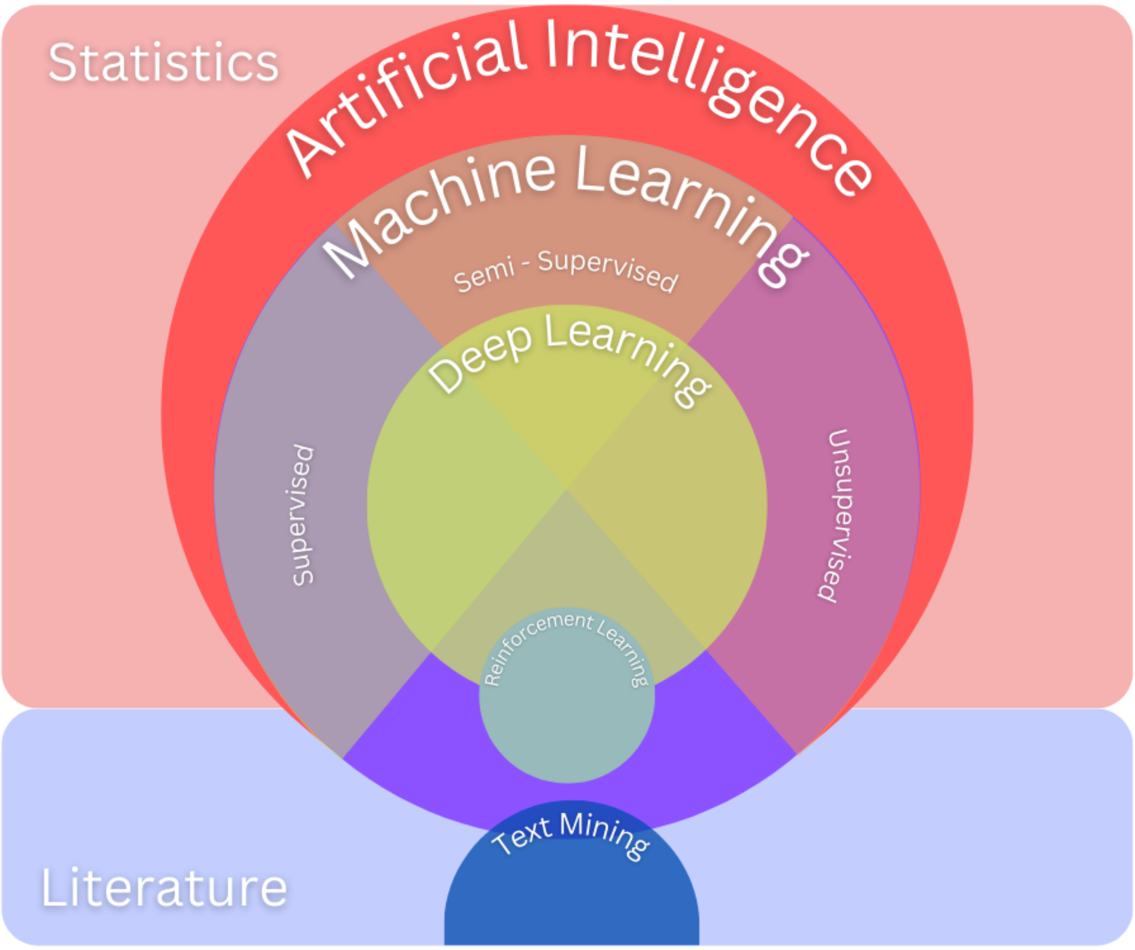


Figure 8: Depiction of ML concepts organization and how they fit into the field of AI [144].

After the model has been trained, it must be evaluated to determine how well it is performing in the presence of unknown outcomes. This involves using metrics to measure the model's performance according to the task at hand [145]. If the model's performance is not satisfactory, the next step is to fine-tune and improve it. This may involve adjusting the model's parameters, changing the training data, or trying a different model altogether. It may be necessary to iteratively repeat the process of fine-tuning and improving the model before the desired performance level is reached. Once the model is performing well, it can be deployed and maintained in a production environment. This involves integrating the model into an application or system and monitoring its performance over time. On a production environment, the model may need to be retrained periodically to ensure that it continues to perform well on new data.

1.2.1. Data collection and pre-processing

Usually, the construction of a dataset is first conducted by gathering instances. Instances are every entry - the available samples - that can be characterized and constitute a data point on a ML deployment pipeline. Gathering more data points will yield more information for the model to learn from. Usually, a dataset with more data points leads to stronger and more generalized models than its smaller counterparts. Regarding the type of data, instances can be many things, if they can be standardized among each other and can yield a pattern that relates towards the target prediction. An example of common standardization can be seen in (**Equation 1**).

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Equation 1: Standardization (z) of a value (x) according to the mean (μ) and standard deviation (σ)

The quality of the data samples determines the possible prediction performance of the models as such it is necessary to filter out irrelevant, faulty or duplicated instances. For all these processes, there are well-developed mathematical approaches that are available in most ML-centred software [146]. The number and quality of instances are determinants for the quality of the upcoming predictions. What is associated or generated from those instances, however, can be equally important. The descriptors that we associate with instances are called features. Features are the characteristics that can be associated to a data point. These features need to be relevant for the output prediction and be independent among each other. One of the simplest ways to assess a feature usefulness is calculating its variance, if a feature has null variance, it will be useless on the scope of most prediction models (**Equation 2**). Conversely, if variance is so high that it means that most samples have unique values, it will also not add much to the model [147].

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2 \tag{2}$$

Equation 2: Feature variance (σ^2) according to the mean (μ) and the average of the sample values for said feature. The number of samples ranges from 1 to n, i being the iterator that represents the current sample. x_i represents the feature value of the current sample i

The usage of inadequate features can introduce biases, noise, or overall weakening of the prediction capability of the model [148, 147]. However, what makes a feature relevant is not always straightforward. Although there are approaches that can test the dataset for the most relevant features, the scientific/technical knowledge of the dataset is certainly an important factor in the selection and analysis of features. Freely available data from databases or, in some cases, data collected by the researchers, does not always contain information adequate to generate robust models.

One of the most common problems faced by researchers is how to deal with missing values. in some cases, several approaches artificially generate values when they are not available [149]. Nevertheless, it is usually preferable to first consider alternative data sources that can yield the corresponding values. Feature extraction is the process in which, for the original raw data or instance data points, alternative features are generated to better describe the entries. Feature extraction is highly dependent on the type of data under focus. After data acquisition and pre-processing, the final dataset should be at least split into training (train the model) and test (validate the model) sets. The test set should not be present in the learning phase, and depending on the context, more subsets can be set aside for validation. This can be performed several times, in a process called cross-validation (CV). The standard approach when deploying CV is to have the computer randomly split the dataset into two sets with a given percentage each time, performing the training on the larger dataset and evaluating the model on the small test set for each case [150, 151]. CV is used to determine how well the ML model can generalize (i.e., if it is able to give accurate predictions about upcoming, unknown, instances) and learn from the data. It also helps to prevent two common problems that occur when training a ML model, overfitting and underfitting. Overfitting is when the prediction model may appear to be highly reliable but when faced with new information, it can output unreliable predictions due to its bias towards the input data [152]. Contrarily, underfitting occurs when the model is not performing properly even on the training data, and as a result, it performs poorly on both the training data and new/unseen data. This is because the model is unable to learn the underlying patterns in the data and cannot make accurate predictions [153].

1.2.2. Feature representation

When developing a ML protocol that leverages computational tools, it is generally necessary to convert or represent the raw data into a form that is suitable for a model to learn from (i.e., feature representation). This typically involves extracting and selecting relevant features from the raw data (i.e., feature extraction) and transforming them into a numerical form that the model can work with.

Characterising data instances through an adequate feature space is pivotal to developing well-performing and useful ML approaches [154]. Like many of the referred concepts, there is no all-encompassing answer regarding the appropriate feature space for each problem. Regarding the set of problems that focuses on drug design and development, it is particularly relevant to address how the biological elements involved are represented. According to the prior exposition, we will highlight more prominently the small ligands and the protein target, as they are the more intervenient elements in drug design. The properties of molecules (e.g., structure, chemical composition) are frequently represented by molecular descriptors. They play a fundamental role in the development of ML models since they can provide features useful for predicting for example the properties and behaviours of a molecule [155]. When considering the unique representation of a molecule through molecular descriptors, it is often mentioned the term molecular fingerprint, which must be exclusive for a single molecule [46]. Although a specific molecular fingerprint should be unique for each molecule, the same molecule can have multiple fingerprints, which might prove useful to improve the performance of ML models, as they often provide unique information [156].

1.2.2.1. Ligands

In ML-based tasks, ligand representation features are commonly used to develop models to tackle issues, such as DTI prediction. Ligands can be represented through atomic or structural data as well as molecular descriptors . In drug design, there are some features recurrently used to study biologically active compound: atom composition, MW, functional groups, bonds connecting different functional groups, distances between different atoms or functional groups and the Polar and Non-Polar Surface Area (PSA and NPSA, respectively) [157, 155]. From a systematic perspective, molecular descriptors can be categorized in 1D, 2D, 3D [157], and, more recently, voxel [48](**Figure 9**).

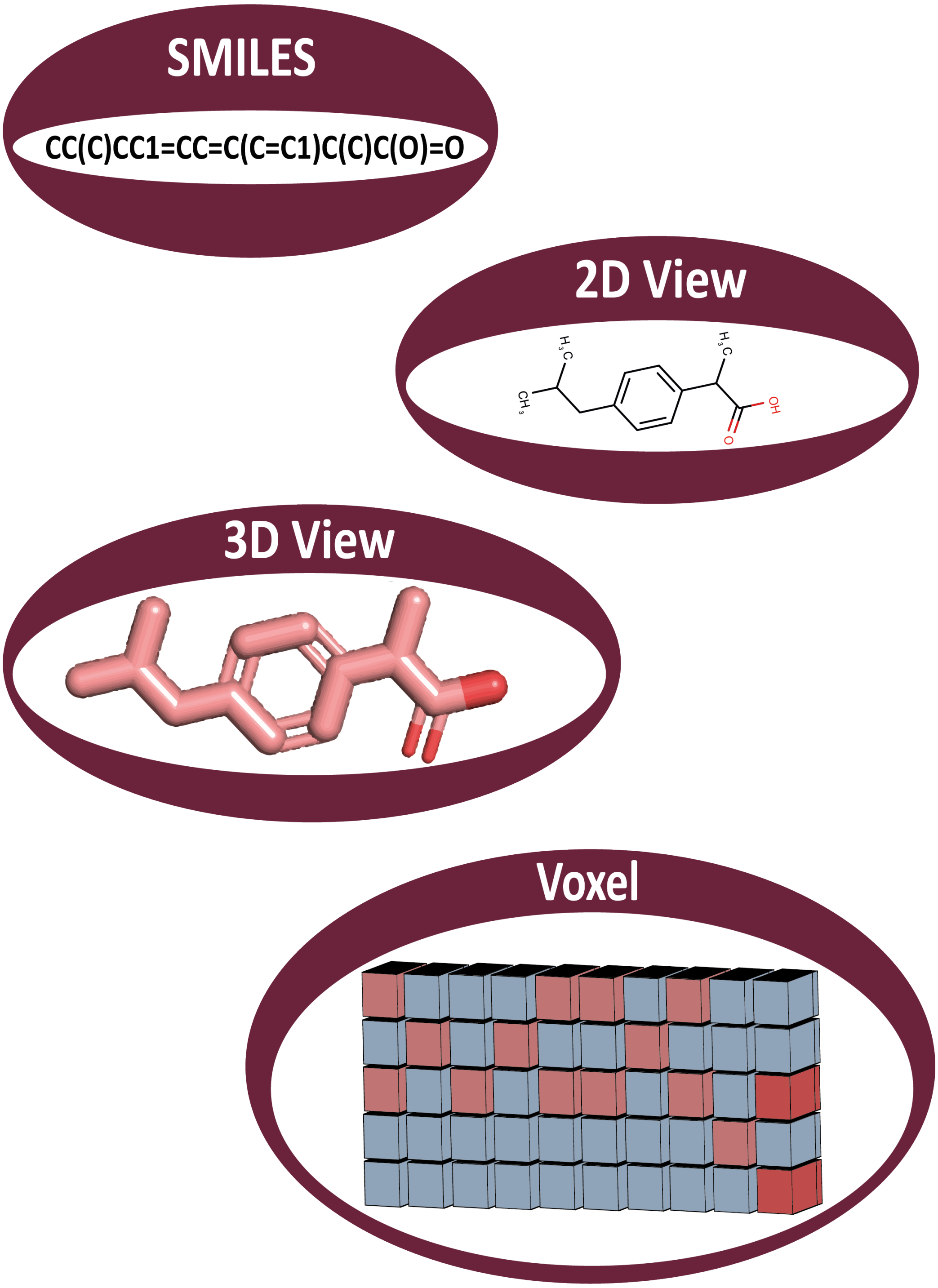


Figure 9: SMILES, 2D, 3D and Voxel representation for the ibuprofen molecule. SMILES and 2D data retrieved from DrugBank [41]. 3D view was made using PyMol [5].

1D descriptors are the simplest type and can be easily calculated using the chemical formula of the ligand. These descriptors consist, for instance, on the frequency of a given atom or functional group, its type, MW and sum or average of atomic properties (e.g., atomic Van der Waals volumes). Most 1D information is narrow and can assume the same values for different molecules, meaning that it is frequently not specific enough. As for 2D descriptors, they are calculated using a representation of the molecule on a plane, where the atoms are laid and connected by bonds, but without the 3D component of space. In this way, 2D descriptors define atoms' connections. Also, this type of representation allows for the calculation of several topological indices which represent properties, like adjacency and connectivity, depending on the size, shape, symmetry, branching and cyclicity of the molecule overcoming some of the 1D descriptors' disadvantages. Lastly, 3D descriptors give information on the molecule's conformation, identifying and quantifying its interaction(s). In addition, PSA and NPSA surface area, intramolecular hydrogen bonding and valence electron distribution are often calculated. To calculate these descriptors, Quantum Mechanics (QM) theory can be a significant contributor since the molecules under scope are often relatively small and cannot be as accurately described with the more standard Newtonian physics [157, 155].

When considering ligands, a fingerprint is often a numerical vector uniquely describing the chemical composition, structural features, and physical properties of a compound [46]. They allow a comparison between different ligands turning the evaluation of molecules similarity into a more straightforward task. Fingerprints can also store 2D information, thus called 2D fingerprints, or 2D and 3D information, in which case they are most known as pharmacophore fingerprints. Cereto-Massagué *et al.* categorized and thoroughly described fingerprints according to the type of information and how it was stored as: substructure keys-based, topological, or path-based, circular pharmacophore, hybrid, and other types of molecular fingerprints [158]. Finally, voxels have been gaining attention as alternative and abstract molecule representations for DTI problem-solving [48], these molecules allow the storage of the same information as 3D, while also enabling the attribution of additional properties, rendering it an n-dimensional representation approachs [159]. Several packages provide the tools to generate ligand representation features, such as Mordred [44], PyDPI [160] and OpenBabel [43].

1.2.2.2. Proteins

On the scope of protein molecular representation, this usually means either sequence or structure-derived features. Sequence-derived features are extracted from the protein sequence [161], and comprise a wide array of information, such as amino acid properties, whole-protein sequence features, and conservation information. When considering amino acid properties for sequence-derived feature extraction, information such as the known composition of the amino acids (e.g., number of sulphur atoms, number of carbon atoms, presence of aromatic rings, etc.) can be used. Experimentally determined values (e.g., pKa values, secondary structure propensity and average accessible area), as those available at the Biological Magnetic Resonance Data Bank [162] are also commonly used. These features characterize each amino acid of the protein individually or when using window-based features an overall environment of each amino acid [163]. Whole-sequence protein features are descriptors common to all amino acids of the system, but that complement the variability introduced by single amino-acid level analysis. Furthermore, these features can be particularly useful to characterize PPI as they provide thorough characterizations of the protein chains [160, 164, 165, 166, 167].

Experimental data availability is a noticeable constraint when addressing protein feature representation. In the case of proteins, it is useful to divide protein features according to their source, whose most prominent two are sequence and structure-based data, depending, respectively, on whether there is knowledge of the amino acid sequence based or the 3D atom spatial distribution. Structural data is less abundant than sequence-based data; however, their use to train ML models' leads to improvements in performance [168].

Evolutionary conservation information has been introduced in several contexts to expand on the more standard sequence-based information [169, 29, 170]. Features encompassing conservation information presume the calculation of a Multiple Sequence Alignment (MSA), which takes the target protein sequence as input and aligns it with other known protein sequences. Several tools were developed and fine-tuned for this purpose, such as Clustal Omega [171], Basic Local Alignment Search Tool (BLAST) and Psi-BLAST [172]. Upon these alignments, a Position-Specific Scoring Matric (PSSM) can be calculated and used to score every amino-acid position according to its conservation, depending on its accordance with the remaining aligned protein sequences. The conservation scores for each amino acid are valuable features, as highly conserved residues tend to be more relevant in both protein structure and function. This information allows the PSSM to represent structural information and as such, albeit being sequence-based, methods, provide meaningful contributions to overall prediction models [170], and to protein-ligand DTI predictors [173, 174, 175].

A more recent approach successfully uses representation learning to automatically extract the most significant characteristics of protein sequences and express them as features [176]. Differently, other approaches focus on minimizing the noise of less relevant features using methods such as wrapper feature selection [177]. However, researchers should consider that if structural data is available and easy to use, it is generally more reliable than sequence-based partly as it also comprises sequence-based information [178, 179]. Some approaches can take the raw atom coordinates and process them inside DL architectures, whereas others can add a prior step in which structural features are abstracted from the coordinates before they are subject to prediction tasks. The construction of feature vectors from contact matrices between the amino acids and physicochemical distance matrices is one of the approaches that was already applied [180].

As previously referred, MPs raise further problems that are not as pronounced when considering soluble proteins. The ability to characterize the structural and physicochemical properties of MPs as well as their interactions and interfaces is essential to develop improved and more targeted therapies as well as to discover new drug targets. Features of proteins, such as electrostatic interactions [181], hydrophobic effects [182] or HS residues [127, 119, 126, 183], were shown to contribute to the affinity and specificity of PPIs. Other well-characterized properties of proteins are the evolutionary conservation and distribution of their amino acids. These two features contribute the most to the prediction of functionally essential residues, as highlighted by several publications [184, 185, 186, 187]. While many studies have dealt with soluble systems, there is a significant lack of in-depth analysis of MP complexes and their interactions.

1.2.3. Supervised and Unsupervised Learning

Supervised learning is a ML technique in which the data fed to the prediction model is characterized by both input and output information. This means that every instance has a label. The labels inform the prediction model of the possible outputs since they are the known values of the target prediction. A supervised learning approach will make use of the labelled instances to build a model able to predict the labels to unknown, previously unseen samples [188]. Although the labels are pivotal for a supervised learning approach they cannot, in any circumstance, be considered alongside the features – such an issue fits into the definition of target leakage [189]. Target leakage can often occur by keeping target-associated variables or by applying pre-processing techniques before dividing the data.

If the model is well-suited to the problem, and the dataset is made up of meaningful and representative data, the model should be able to make predictions close to its real counterparts. Supervised learning can be applied to data with discrete (classification task) or continuous labels (regression task) [190]. When considering classification approaches, there is added attention required when considering the instances. Due to an uneven class population, it is often required to balance the dataset to equilibrate the number of instances in each class. There are several sample balancing processes such as up-sampling (artificially augmenting the lower populated classes) and down-sampling (lowering the number of instances in the overpopulated classes) [191, 146].

Contrarily to supervised learning, unsupervised learning occurs in the absence of labelled data. The algorithm is aimed at apprehending a relationship or pattern between features on the dataset [192]. In some cases, unsupervised learning approaches can be used as intermediary aids to more complex pipelines, for instance, when addressing datasets of large size that might require dimensionality reduction. On this scope, unsupervised learning comprises techniques such as Principal Component Analysis (PCA) and MultiDimensional Scaling (MDS). However, these can be too reductive depending on problem [193]. Autoencoders are more complex unsupervised learning algorithms that can be helpful in this situation [194]. Another well-known application of unsupervised learning is clustering algorithms that compute the similarity between data point pairs. As such, these algorithms can weigh the importance of each feature and reorganize the dataset in clusters of data [138].

1.2.4. Model performance evaluation

The goal of model evaluation is to determine how well the model can make predictions on new, unseen data, and to identify any potential problems or limitations with the model. There are many different metrics and techniques that can be used for model evaluation depending on the specific problem and the type of model being used. In the case of classification models, the most common metrics are derived from a confusion matrix (**Table 3**). On a binary classification problem, this is used to directly compute the results, whereas when there are more than two classes the metrics are iteratively calculated taking each class as positive (P) and the remaining ones as negatives (N), with the final performance metric being averaged out.

Table 3: Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Some frequently used metrics include for example the accuracy (**Equation 3**)., sensitivity or True Positive Rate (TPR, **Equation 4**), specificity, selectivity, or True Negative Rate (TNR, **Equation 5**), precision or Positive Predictive Value (PPV, **Equation 6**), Negative Predictive Value (NPV, **Equation 7**), and F1-score (**Equation 8**) are calculated directly from the values attained from the confusion matrix. The False Discovery Rate (FDR, **Equation 9**), although it can be calculated independently, can also be seen as the inverse of precision. Similarly, the False-Negative Rate (FNR, **Equation 10**) is the inverse of sensitivity. The Area Under the Receiver Operating Characteristic Curve (AUROC, **Equation 11**) depends on the TPR (**Equation 4**) and the FDR (**Equation 9**). By including different metrics on all the evaluated set of data points, AUROC constitutes a good metric for classification models [195]. However, as any other performance metric, should not be considered solely, but rather in conjunction with other metrics, depending on the problem step. The equations listed below (3-11) are all dependent on these values and can be used to address the particularities of a dataset.

$$Accuracy = \frac{TP + TN}{P + N} \tag{3}$$

Equation 3: Accuracy

$$TPR = \frac{TP}{P} \tag{4}$$

Equation 4: Sensitivity, recall or TPR

$$TNR = \frac{TN}{N} \tag{5}$$

Equation 5: Specificity, selectivity, or TNR

$$PPV = \frac{TP}{TP + FP} \tag{6}$$

Equation 6: Precision or PPV

$$NPV = \frac{TN}{TN + FN} \tag{7}$$

Equation 7: NPV

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{8}$$

Equation 8: F1-score

$$FDR = \frac{FP}{FP + TP} \tag{9}$$

Equation 9: FDR

$$FNR = \frac{FN}{P} \tag{10}$$

Equation 10: FNR

$$AUROC = \int_0^1 TPR(FPR), dFPR \tag{11}$$

Equation 11: AUROC

Regarding regression approaches in which the target variable is continuous, the available performance evaluation metrics are entirely different. Nevertheless, there is still no single metric, rather, two groups of performance evaluation metrics must be considered: errors and correlations. Errors are scale dependent, which signifies their meaning depends on the range of values usually associated with the problems at hands. These are often used to optimize ML regression models, through minimization. Errors vary widely, depending on the problem considered, making them less flexible than correlations regarding comparing results across different approaches and datasets. On the other hand, they can give good insights regarding the data under scope. Some of these performance evaluation metrics are Mean Squared Error (MSE, **Equation 12**), Root Mean Squared Error (RMSE, **Equation 13**) and Mean Absolute Error (MAE, **Equation 14**).

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{12}$$

Equation 12: MSE

$$RMSE = \sqrt{MSE} \tag{13}$$

Equation 13: RMSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y_i}|^2 \tag{14}$$

Equation 14: MAE, with n being the number of samples, Y_i the actual values and $\hat{Y_i}$ the predicted values.

Complimentarily to errors, correlations offer an alternative performance evaluation approach to regression-based problems. Correlations are scale independent and, thus, fall inside well-defined ranges, regardless of the problem at hands. Nonetheless, correlations tend to erase the particularities of the data and problem at hands, therefore errors should still be taken into consideration for a complete performance evaluation. Some correlations are R^2 or coefficient of determination (**Equation 15**), Pearson correlation coefficient (Equation 16-18) and Spearman rank correlation coefficient (**Equation 19**).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y})^2}{\sum_{i=1}^{n}(Y_i - \overline{Y})^2} \tag{15}$$

Equation 15: R^2 or coefficient of determination, with n being the number of samples, Y_i the actual values, $\hat{Y_i}$ the predicted values and $\overline{Y}$ the average of the actual values.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(y - \overline{y})^2}{n-1}} \tag{16}$$

Equation 16: Standard deviation (σ), with y being the values and $\overline{y}$ their average.

$$cov(x,y) = \frac{1}{n^2}\sum_{i=1}^{n}(\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)(y_i - y_j)) \tag{17}$$

Equation 17: Covariance (cov), with n being the number of samples, x one characteristic and y the other.

$$Pearson = \frac{cov(Y_i, \hat{Y_i})}{\sigma Y \sigma \hat{Y}} \tag{18}$$

Equation 18: Pearson correlation coefficient, with cov(Y_i,$\hat{Y_i}$) (**Equation 16**) being the covariance between the actual values (Y_i) and the predicted values ($\hat{Y_i}$), σY the standard deviation (**Equation 17**) of the actual values and $\sigma\hat{Y}$ the standard deviation (**Equation 17**) of the predicted values.

$$Spearman = \frac{cov(R(Y_i), R(\hat{Y_i}))}{\sigma R(Y) \sigma R(\hat{Y})} \tag{19}$$

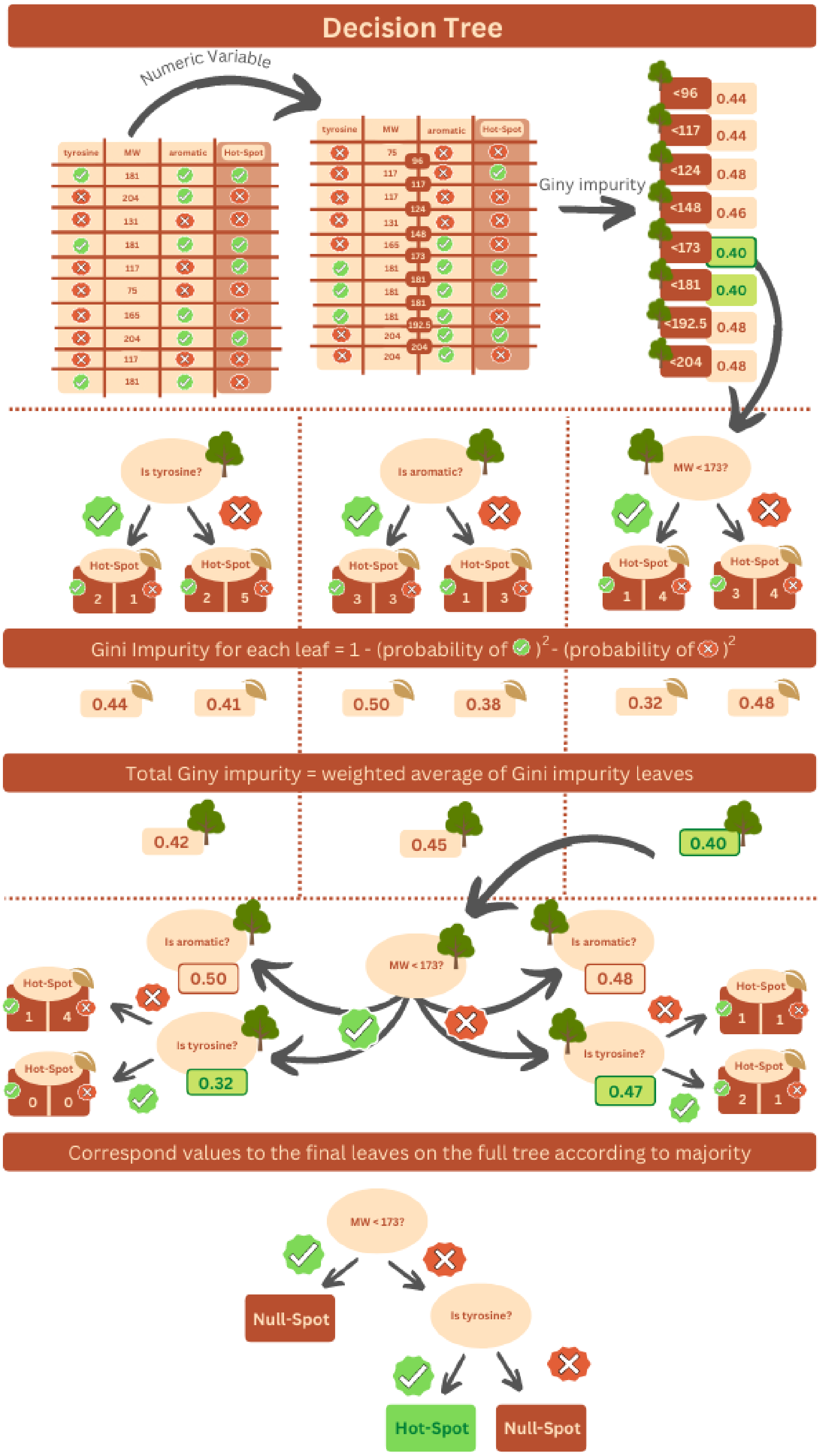
Equation 19: Spearman rank correlation coefficient, with cov(R(Y_i),R($\hat{Y_i}$)) (**Equation 16**) being the covariance between the actual ranked values R(Y_i) and the predicted ranked values R($\hat{Y_i}$), $\sigma R(Y)$ the standard deviation (**Equation 17**) of the actual ranked values and $\sigma R(\hat{Y})$ the standard deviation (**Equation 17**) of the predicted values.

1.3. Prediction Models

Throughout this section, we explore the most commonly applied AI approaches to biological data and we list innovative ones, the ones that were not yet extensively applied in the field. A few of the algorithms comprise simple approaches. However, some of the most recent approaches deal with more complex algorithms, particularly Artificial Neural Networks (ANNs). In this section there will be no major specification between classification or regression approaches as most of the models share most of the components and diverge slightly only at the end, to provide either a continuous or discrete output.

1.3.1. Decision trees, random forests, extreme randomized trees, and extreme gradient boosting

A Decision Tree (DT) algorithm works through nodes, branches, and leaves. When considering a particular dataset, all the features are measured against the target variable to determine the leaf impurity. One common impurity metric is Gini index, although there are others that can be used, such as entropy [196]. The Gini impurity accounts for the probabilities of each classification option given the feature under scope. The feature with the lowest Gini impurity score is defined as the root node. This node ramifies through branches to new nodes (internal nodes) on which the Gini impurity score will be reassessed, excluding the already used feature. The process is repeated until all the features are used, so that their inclusion lowers the Gini impurity score. Thus, if a feature has a Gini impurity score higher than the value achieved without that feature, the DT will stop at that point in what is called a leaf. Leaves are the final stop of a DT, more objectively determined as the point that is connected to prior nodes but is not connected to subsequent nodes. The measurement of a Gini impurity score, for a new sample, at each of the leaves, will yield the final classification [196]. **Figure 10**, below, follows the process of constructing a DT from the ground, with a HS/NS classification example dataset (not based on experimentally determined data).



tyrosine	MW	aromatic	Hot-Spot
✓	181	✓	✓
✗	204	✓	✗
✗	131	✗	✗
✓	181	✓	✓
✗	117	✗	✓
✗	75	✗	✗
✗	165	✓	✗
✗	204	✓	✓
✗	117	✗	✗
✓	181	✓	✗

tyrosine	MW	aromatic	Hot-Spot
✗	75	✗	✗
✗	117	✗	✓
✗	117	✗	✗
✗	131	✗	✗
✗	165	✓	✗
✓	181	✓	✓
✓	181	✓	✓
✓	181	✓	✗
✗	204	✓	✓
✗	204	✓	✗

Figure 10: DT construction from HS/NS example data (not based on experimentally determined data) [197]

Random Forests (RF) are a well-known type of ensemble – meaning their predictions are the result from a combination of predictions from smaller, individual predictions. RFs take multiple DTs as individual predictors, forming an ensemble in which the different attributes are tested in random combinations and the final decision is made taking into consideration the output of the individual DTs. Extreme Randomized Trees (ERT), a variation of RF, were effectively used in several biological problems although still scarcely applied to proteins [198, 199]. This method has increased randomization, compared to RF, picking not only attributes but also samples at random. Furthermore, it chooses node cut-off points fully at random. This means that for continuous variables, which must be split according to a threshold, instead of following the standard approach of common DTs (by calculating Gini impurity scores for the samples until the lowest Gini score is found), ERT uses random cut-off points, which can help eliminate sample dependent bias.

More recently, there has been an increased focus on eXtreme Gradient Boosting (XGB). This method is also an ensemble of DTs, although these trees differ slightly from those previously explored, as they compute similarity scores for each node, instead of Gini impurity scores. Similarity scores make use of the residuals – differences among samples – to assess how well the different tree branches divide the data. The similarity score includes a variable – λ – that should be subject to optimization depending on the problem. Subsequently, the gain is calculated from the similarity scores of each node for each of the possible branch splits. The branch split associated with the highest gain is then kept. This is performed several times from the root node to the full depth of the tree, which can be controlled as a parameter. To prune the XGB tree, there is an additional value – γ- that is picked and subtracted to the gain, if the resulting number is negative, the corresponding branch is removed from the tree; γ is a hyper-parameter that should be subject to optimization depending on each problem. XGB trees predictions are updated according to a learning rate (ϵ) that changes previous predictions. By cumulatively building XGB trees and applying ε, the residuals get lower, until they are either minimal or stop at a set number of trees. This process of improving predictions of the previous trees with the updated ones is what characterizes XGB as a boosting ensemble [200].

1.3.2. Hidden Markov models

When considering a Hidden Markov Model (HMM), there is a non-observable (hidden) variable for which the algorithm will try to solve based on known variables, typically chained sequentially. Thus, HMM is typically appropriate to predict the likelihood of time-dependent events. However, it can also be useful to derive inferences from non-timed yet sequentially chained data, as seen in its usage on biological sequence data, particularly the inclusion of evolutionary data from protein sequences alignments [201]. Based on Bayesian probability, HMM considers, by default, both transition and emission probabilities. The transition probabilities are related to previous samples or states and influence the probability of the current sample prediction introducing a sequential bias. On the other hand, the emission probabilities associate individual events with their probability of occurrence without being influenced by previous events. The collective impact of these probabilities is calculated to estimate the event of the highest likelihood [202].

1.3.3. Support Vector Machines

A Support Vector Machine (SVM) is an algorithm that discriminates samples according to their features. Depending on the features number (n), an n-dimensional space is generated, and samples are assessed in terms of proximity. On a classification approach, the n-dimensional space is then split into k sections, with k being the number of classes that the algorithm is bound to determine. The k sections are split among space by n-dimensional support vectors that define the places that are going to be occupied by the samples (Noble, 2006). First, to define a support vector, an edge is defined as the object between two samples. Secondly, all the samples are evaluated according to this edge, with the possibility of some being misclassified. Next, soft margins are calculated as the distances from the samples used to define the edge and the edge itself. The space occupied between the soft margins (with the edge in between) is called a support vector. Samples outside the support vector should be correctly classified. However, there is a possibility of finding samples inside the soft margins. The hardest and most time and resource consuming task when training a SVM is cross-validation to find the best support vector so that the number of misclassified samples is the lowest while maintaining the performance on new samples, and as such, achieving a generalizable algorithm [203].

1.3.4. Artificial Neural Networks

ANNs were originally inspired by the biological structure of the human brain and how neurons communicate. ANNs can also be interpreted and referred to as Multilayer Perceptron's (MLP). MLPs, as the name indicates, stem from their counterpart, the perceptron. A perceptron is a single input variable that is related to an output variable through a function. As such, a simple linear regression can qualify as a perceptron [141]. A perceptron is usually graphically depicted as a single circle (input node or neuron) connected to another circle (output node or neuron) through a line (edge). This description is also applicable to a simple graph (not to be confused with Graph Neural Networks (GNNs), discussed subsequently) [204]. Upon combining multiple perceptrons, such that a row of input nodes would be connected to the second row of nodes, we have our first MLP. If we add a third layer, we now refer to the first layer as the input layer, the second layer as a hidden layer and the third layer as the output layer. This is now an ANN, with the output layer providing a value that can, depending on the problem, be of a continuous or discrete nature. When we have more than one hidden layer, we can now refer to the network as a Deep Neural Network (DNN), although also still an ANN, it is now entering the realm of Deep Learning (DL) **Figure 11**, below, gives an example of an ANN.

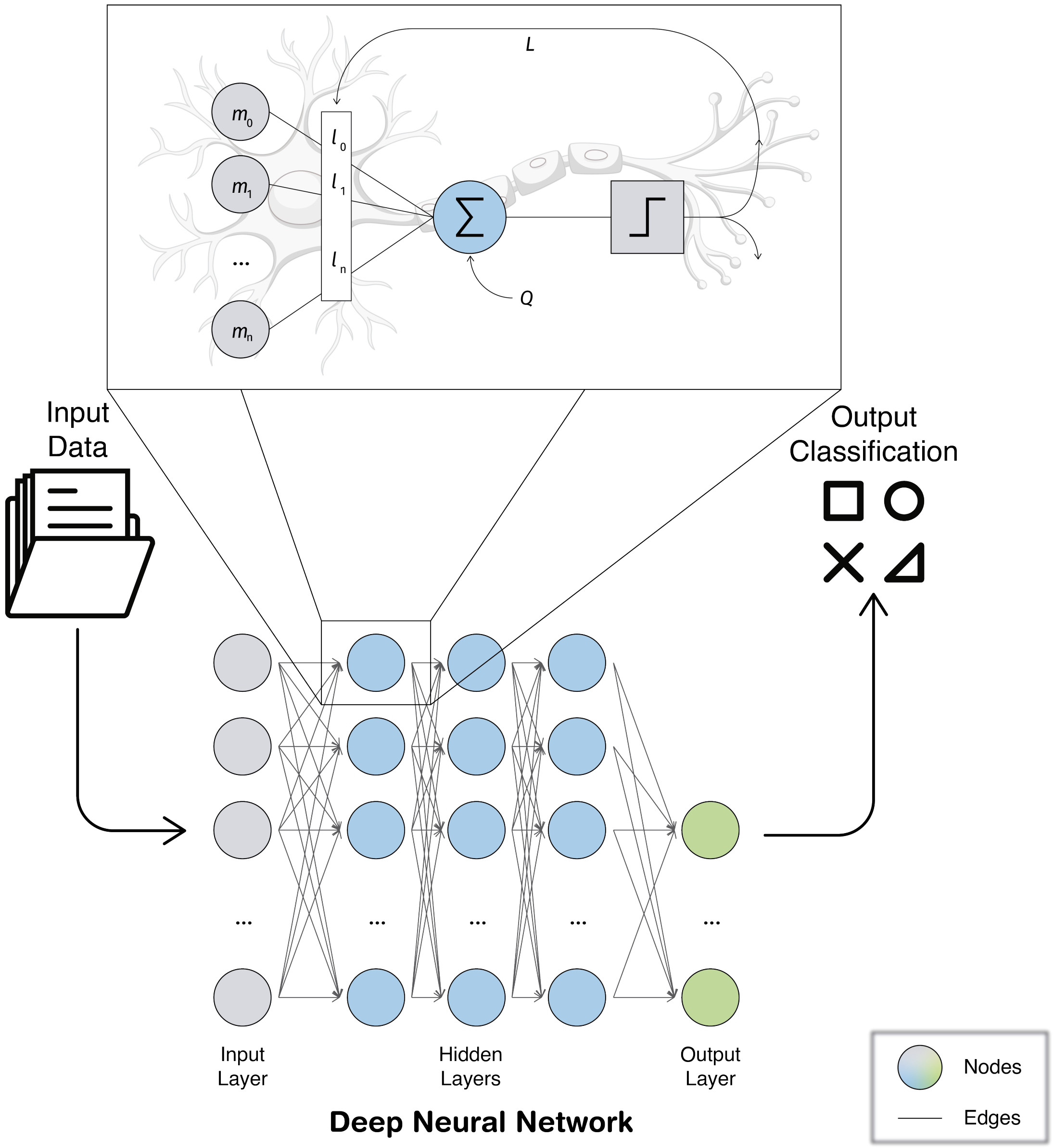


Figure 11: ANN for classification and its biological counterpart.

1.3.4.1. Deep Learning

The ability to arbitrarily add hidden layers can give ANNs the capacity to abstract information and achieve higher performance than other methods, particularly when using increasingly larger amounts of data. Furthermore, this also opens the gates to tackle problems of supervised and unsupervised nature, among others [141]. A DL based model takes the input features, which should have the same size as the input layer and passes them along with the hidden layers by activating nodes (neurons), until it reaches a final vector of values, in the output layer. The activation process depends on the activation function (initially typically sigmoid, currently, more often, ReLU), weights and biases (usually randomly initialized), and the network's architecture. The final output values can then be assessed according to a cost function, against actual values. The algorithm can then be backpropagated to fine-tune the parameters (weights and biases are improved according to a learning rate) until the model has converged and the loss is no longer significantly decreasing [141]. One of the most significant disadvantages of DL is that it is quite demanding in terms of computational resources, especially when dealing with massive datasets. In the last years, DL has proven to effectively perform different computational tasks, mainly of categorical and regression nature [205, 206]. The set of steps described opens a gateway to a whole new family of ML algorithms, nowadays referred to as DL, in which the parameters can be tuned, and the architecture can be twitched to have the best performing algorithm for each task [207]. We elaborate some of these algorithms on the subsections below and provide the most significant examples applied to GPCRs.

1.3.4.2. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are another subtype of ANNs that use a DL architecture. CNNs, unlike other ANNs, do not connect all the neurons in each layer. Instead, they activate subsets of neurons depending on a specific batch of input features and then overlap part of the information not to miss any data. When considering image processing, this would be the equivalent of analyzing subsets of subjacent pixels and finding patterns among and in them. Furthermore, CNN's usually have pool layers that reduce the noise and standardize the information [208].

1.3.4.3. Autoencoder

Autoencoders are unsupervised DL algorithms. However, they have some peculiar characteristics, with three specialized groups of layers on the hidden layers: encoder, bottleneck, and decoder [209]. The encoder layers are the hidden layers situated immediately downstream from the input. The number of layers in this structure varies; however, the number of neurons in each layer will gradually decrease in each layer until the single bottleneck layer is reached. The bottleneck is situated downstream from the encoder and upstream from the decoder. It is usually the middle layer of the network and the layer with fewer neurons on the autoencoder. The decoder layers are added after the bottleneck layer and before the output layer. The size and number of the decoder layers usually mirrors that of the encoder layers, with fewer neurons close to the bottleneck and more neurons on layers closer to the output layer [210]. When data is fed to the input layer, it flows through the encoder section decreasing in dimension. At the bottleneck, information is at its most compressed state. Typically, the bottleneck information represents the input data as it is that data encoded in a smaller state. The decoder segment transforms the bottleneck information on the original data, and so the autoencoder output should be equal to the input data, lending the ability of data reconstruction to autoencoders. This algorithm is commonly applied to dimensionality reduction, denoising and inpainting images, among others [211].

1.3.4.4. Graph Neural Networks

Before explaining what the GNN algorithm is and how it works, it is necessary to introduce the concept of a graph. A graph is a data structure composed of nodes and edges, being the last one responsible for establishing the relationship between nodes. This type of data structure presents a significant difference compared to others, as it does not necessarily consider spatial features and enables the inclusion of data of different sources in a graph representation (the representation of molecules as graphs is one example of GNN usage in biological problems [212]). An example of a graph is the common visual representation of ANNs, such as the one in **Figure 11**. GNN is a DL tool that allows processing, representing, and collecting information from graphs, an example of which are graph embedding techniques [213]. GNNs looks to discover the weight vector called embedding state. Each node has a state in the graph, and both the node and the edges have features. The embedding state is then calculated in an iterative process through a local transition function dependent on the node, edges, state, and neighborhood features. After setting the embedding state, it is possible to determine the output using a local output function that considers this state and the nodes' features. Finally, when obtaining all the nodes' functions, these are stacked, generating a single global function, either for the embedding state or the output [214, 213].

1.4. Machine Learning applications to Drug Design and Development

There has been an increase in the usage of ML in drug design and development. This usage is mainly hinged on the ability of ML to provide data-driven decision-making on drug design pipelines, speeding up the process, reducing the cost and time associated with this process, and diminishing the associated error with human intervention. There are, however, still many challenges associated with the integration of ML in drug design, and the interpretability of these processes and their biological meaning is one of the most promising ones [42, 215].

One subject concerning drug design and development that is pivotal for data-driven approaches is abundant information regarding targets and ligands, the key to attaining automated responses [216]. Subsequently, it can be used in a wide array of tasks such as drug susceptibility prediction [217], drug synergy response prediction [218], protein binding sites prediction [219], ligand VS processing [220], protein structure modulation prediction [221], ligand functional activity prediction [158], drug design for new targets and binding sites [222], DTIs prediction and identification of accurate docking decoys [223, 216]. Some approaches can even aid in selecting the most appropriate drug design protocol, such as PROFILER, designed for polypharmacology prediction, which applies a DT to choose LBDD or SBDD approaches given the amount and quality of available data [224]. Some important ML applications for this thesis will be further discussed in the next sections.

1.4.1. Ligand activity prediction

Ligand activity prediction can engulf several different issues. On top of increasingly comprehensive datasets, there is now an extensive array of features regarding both target and drugs that coupled with complementary tools, are driving DTI prediction [33].

Countless ligands can interact with proteins, making it imperative to improve our understanding and characterization of DTIs [225, 226]. These can be studied through proteome-ligand information to shed light on the binding process and accelerate drug design and development. DTIs depend on several factors that can be leveraged as features, such as binding energy, electrostatic energy, intermolecular energy, the interaction energy of van der Waals or intermolecular forces [227]. Most intermolecular DTIs result from van der Waals forces, weaker than hydrogen bonds or hydrophobic interactions [227, 228]. Throughout the drug design process, compounds are modified to improve properties such as bioactivity and selectivity [229].

Recently, multiple sequence-based ML models with promising accuracy values were developed to predict DTIs, all trained with targets varying from enzymes, ion channels, GPCRs, and nuclear receptors [174, 230, 231, 177, 232, 233, 234, 235]. Different classifiers can make use of a wide array of available algorithms, such as SVM and RF [174, 177, 232, 233]. These classifiers use a broad spectrum of features for the target, such as Position-Specific Scoring Matrix (PSSM) [236, 237, 175, 230], pseudo-position PSSM [175] and BI-Gram Probabilities [238], among others. For the drug features, substructure fingerprints [174, 230, 239, 232] as well as other molecular fingerprinting representations [175], are necessary to attain good-performing DTI predictors. In some cases, data processing tools such as PCA [230], Lasso algorithm [240, 175], or wrapper feature selection [177] are also used to improve the overall model performance. Wang *et al.* described a model with stacked auto-encoders used to extract features from a protein dataset [241]. Those features were then used to predict the DTIs with an RF algorithm that achieved nearly 87% accuracy. In another approach, target bias was included to simulate possible target conformations. Moreover, the authors state that their approach could predict completely new DTIs [231].

Although DTI prediction is a focal point in ligand activity prediction, many other problems can benefit from the deployment of ML approaches. When searching for the biological meaning behind drug activity, it is important to zoom in on several factors such as the knowledge of the functional groups involved in the biological activity [242] and specific protein binding motifs [226]. Protein-ligand binding site classification and dynamic molecular changes are crucial for a more precise ligand activity understanding. Plante *et al.*, by deploying DL upon MD trajectories developed a method able to extract useful information to reveal distinct ligand characteristics and molecular factors and ultimately discriminate protein structure and function with high accuracy on the test set (> 98%) [243]. In another example, an SVM prediction model was developed to estimate if a ligand is an agonist or antagonist showing an accuracy of 86.5% [244]. Another study compared several ML models with other ligand-based VS methods and showed that DNN and RF were able to enhance protein agonists prediction even with fewer compounds in the training set [245].

To generate a method able to extract more precise information than simply agonist/antagonism classification, Wu *et al.* first used DL for molecular fingerprinting representation and then used an

RF algorithm to assess ligand bioactivity [246]. By combining homology modelling, MD, and a VS approach, it was possible to detect allosteric modulator molecules that bind to a protein at a location different to the binding site [247]. Compounds from several libraries were selected based on known positive and negative antagonists and classified. Selected ligands were docked into the protein target binding site and binding modes were calculated and used to improve inhibitory activities [248]. Throughout the current section, we showed how VS is often used in CADD approaches focusing on a single protein target. However, only a few are based on ML methods, which opens new venues of research, such as the automatization of new target identification given a pool of drugs. This has been explored by Ru *et al.*, who developed a model which incorporates an RF classifier to rank putative new drug-target pairs [249].

1.4.2. Boosting target knowledge with Machine Learning

Attaining an MP with good 3D-structure resolution (either by X-ray crystallization or Cryo-EM) is a challenging task [250]. The knowledge about a target structure is, however, crucial for SBDD as well as DTI prediction. Computational applications have helped to accelerate the process of target structure prediction by avoiding or complementing expensive laboratory experiments. The application of ML methods, in particular, has become indispensable over the years and is continuously improving with the increasing number of solved X-ray crystal structures and computational techniques. Many algorithms were already employed to describe and predict the membrane-embedded sections of MPs. Appropriate target features for ML predictions have proven to be useful such as TransMembrane Helix (TMH) domain topology, like inter-TMH residue contacts, TMH-TMH interactions and residue-residue contact patterns (crucial for ab initio protein folding) [251]. Some of these ML model examples include TMHit [252], MemBrain (CMA + ML-based method [251]), PSIPRED [253], MEMSAT3 [254], DeepMetaPSICOV (Kandathil et al., 2019)[255] and GPCR-I-Tasser [256].

As was already highlighted, modelling MPs is still very challenging compared to soluble proteins [101, 257]. Many standard sequence-based methods for model quality evaluation (not based upon ML) were initially developed for water-soluble proteins but can also be applied to MPs [258, 257, 31, 259]. For the best model's discrimination, an ideal scoring function is the output of such methods, measuring the distance between the model and the native structure correlating between the score and quality [257]. Available scoring functions can be split into three categories: physics [260, 261], knowledge [262, 257, 263, 264, 265] and learning-based [257]. The latter, learning-based functions, have recently been highlighted and include methods such as ANNs or SVMs trained to distinguish between correct and incorrect models based on structural features to predict the actual quality of a given model [266, 267, 268, 257, 269].

The increasing number of MPs available structures and the technological advance in computational power allows for bigger systems and longer timescales (microseconds) MD simulations, which created a ”big data” problem in their analysis [270]. The currently reported integration of MD simulations and ML algorithms shows promising results [101, 243]. For example, Plante et al. presented an ML approach to analyse GPCR-ligand MD simulations. The atomic coordinates calculated throughout the simulations were converted into Red Green Blue (RGB) code to form an image that was readable by a DNN-based pipeline. This novel approach successfully classified GPCR conformations by ligand class (full, partial, and inverse agonist), and allowed authors to identify the structural motifs that undergo conformational changes for each type of molecule studied [243].

The Marta Filizola group recently published another ML/MD protocol to better estimate the kinetic properties of (un)binding of a ligand to GPCR. The rate of dissociation of a drug is an important predictor of its *in vivo* efficacy. However, the timescale of drug dissociation is around the minute, which would be computationally inefficient to simulate. Filizola's group reported a possible solution to this problem by using features extracted from a short, unbiased MD as an initial dataset that was fed into a pipeline that used state-of-the-art ML methods for dimensionality reduction. This protocol successfully estimated two prototypical opioid receptor drugs' kinetic rates at a reduced computational cost while granting atomic resolution of transitional structures throughout the unbinding pathway [271].

Most of the cases explored regard single proteins interacting with single ligands. However, on the biological landscape, proteins often group or interact forming dimers, trimers, or higher-order oligomers. The relevance of protein-oligomers has increased over the last few years as more disease-specific heteromers are being identified [272, 273, 274, 275]. Hence, it is now widely accepted that highly dynamic protein networks exist and that the monomer's functions such as ligand binding affinity and signalling may be altered through oligomer formation [272, 276]. This paradigm shifts from basic signal transduction towards a more holistic and multifactorial view on MPs challenge rational drug design [276]. Therefore, computational studies (including AI approaches) on MPs oligomerization are necessary to understand the disease mechanism and support experimental studies to reveal novel pathways for treating MP-linked illnesses [277]. At the plasma membrane, a GPCR-complex can either be a target for dynamic regulation of ligand-binding, promote or inhibit ligand binding cooperativity or potentiate, attenuate downstream signalling or even change G protein selectivity [276]. Several well-established ML-based methods and web servers were already developed for the prediction of their interfaces, such as WHISCY [278] and ISIS [279] (other models/servers were recently reviewed by Barreto *et al.*. However, not all were developed explicitly for protein dimer interface prediction and their modulation. Until today, there are no methods that cover the complexity of oligomeric systems, and as such, these innovative ML-based methods may provide strategic prediction tools [272].

1.4.3. Protein Hot-Spot prediction

Similarly, to other problems, in HS prediction computational methods - particularly ML - have been used in recent years as a viable option to overcome the technical issues (e.g., cost, time, accuracy) concerning experimental techniques, providing thorough insights and a high-throughput HS identification [280]. In HS prediction, a panoply of features (e.g., physicochemical properties, evolutionary scores, solvent-accessible area, binding energy scores) have been extracted and ML algorithms such as SVM [281], ANNs [279], and XGB [241] were deployed.

One of the main reasons ML is a suitable tool to tackle this problem is also its Achille's heel: low amounts of data. Most of the reliable and available experimental data in SPOTON [126] was assembled from several different databases of experimental determined HS and NS: ASEdb [282], BID [283], PINT [284] and SKEMPI [285]. More recently, SKEMPI 2.0 was released, making available a larger amount of experimental information. However, most of the new information does not include mutations to alanine (and the corresponding change in free binding energy), which is necessary for HS prediction [286]. These databases can be used to deploy ML algorithms that take both the positive (HS) and negative (NS) information and construct a binary classifier that should be able to predict, upon previously unforeseen amino acid residues in a protein, its HS/NS status. Based on this problem, and given the available data format, binary classification is the most explored approach.

Several algorithms have been proposed for HS computational predictions, using different ML approaches, features, and datasets [287, 288, 128, 289, 290, 126, 291, 129, 292]. Recently (2017), SPOTON [126], using the information on both the protein sequence and structure, achieved 0.95 accuracy on an independent testing set. Like SPOTON, most of the high-performing HS predictors incorporate structural information. Although yielding robust results, it hinders the possibilities of a broader deployment since there are still fewer proteins for which a 3D structure is available in online repositories [90] compared to the determined and available protein sequences [293].

It is known that sequence-based predictors tend to perform less well in comparison with the ones engulfing structural information. For example, Nguyen *et al.* [291] were able to achieve an accuracy of 0.79 and a precision of 0.75 using sequence-based features, and upon the addition of structure-based features, the accuracy and precision raised to 0.82 and 0.80, respectively. The same pattern is displayed in the results of PredHS, where results with sequence-based features only are significantly worse than those including structure information [294]. A more complete benchmark of the problem can be found in the review by Rosário-Ferreira *et al.* [295].

1.4.4. Synergistic Approaches for Cancer Treatment

Drug resistance in cancer is a multifactorial problem driven by the tumour microenvironment and genetic and nongenetic/epigenetic mechanisms that, along with cell plasticity, contribute to tumour heterogeneity [296]. In clinical settings, this problem is minimized with a combination of drugs administered together or in sequence (i.e., polytherapy). Targeting multiple components of different or interconnected cancer pathways is an efficient strategy to block vital biological processes [297, 298].

In the past years, the development and improvement of high-throughput technologies and computational tools boosted the use of large volumes of multi-omics data (e.g., genomic, transcriptomic, proteomic) essential to dissect and uncover the complex molecular signatures of cancer. Machine learning (ML) algorithms have attracted particular attention for their ability to learn new associations and extract valuable insights from this type of data. A few ML models based on XGB, RF, SVM, and naive Bayes were already developed to predict the best combination of anticancer drugs by the integration of omics data with chemoinformatic properties of drugs or network information of their targets [299, 300, 301, 302] (Table 4). Likewise, DL implemented via DNNs was particularly useful in dealing with the high multidimensionality of omics data in supervised and unsupervised contexts. DL classification and regression models such as AuDNNsynergy [303], DeepDDS [304], DeepSynergy [305], DeepSignalingSynergy [306], Matchmaker [307], TranSynergy [308], or the work by Xia and colleagues [309] were recently developed for drug combination prediction (**Equation 4**). Nearly all the surveyed works developed drug synergy prediction models based upon a single reference model, which is in most cases the Loewe reference model [299, 307, 308, 305, 304, 306]. Currently, there is a wide scope of well-studied available reference models, including the Bliss independence [310], Highest Single Agent (HSA) [311], Loewe additivity [312, 313], and Zero Interaction Potency (ZIP) [314]. Furthermore, recently Malyutina *et al.* [302] developed the Combination Sensitivity Score (CSS), which measures drug combination synergy using their IC50. As such, this led us to the question of whether the development of a novel prediction approach should be based solely upon a single reference model. Besides, most of the available web interfaces such as DECREASE [315] or DrugComb [316] require the upload of a full or partial mandatory dose–response matrix (experimentally determined), which hinders its systematic use by the scientific community and handicaps its usefulness.

Table 4: Summary of drug synergy ML models, with relevant comments on performance metrics, usability, and validity. Performances listed are dependent on the authors' reported metrics, and, sometimes, are incompatible or irrelevant in a biological setting.

Study	**Algorithms**	**Classification**	**Regression**	**GitHub**	**Website**	**New samples**	**Benchmark**	**Cancer**

Chen ***et al.*** **2013** [317]	RF	AUROC = 0.88;
		Accuracy = 0.92;
		Precision = 0.65;
		Specificity = 0.97						

Target: Combined score - chemical similarity, experimental, database and text-mining.								

Sun ***et al.*** **2014** [318]	SVM	Accuracy = 0.68;
		F1-score = 0.67;
		Recall = 0.61;
		Specificity = 0.74						

Target: Positive values from Drug Combination DataBase (DCDB) [319], negatives randomly generated.								
Huang ***et al.*** **2014** [320]	LR	AUROC = 0.92						
Comments: The scope of this predictor is slightly different from the remaining ones. **Target:** FDA approved drug combinations (positive), unsafe drug combinations (negative).								
Li ***et al.*** **2015** [321]	PEA	AUROC = 0.90						
Comments: Website listed but address not available. **Target:** Positives samples come from several databases; negatives are unclear.								
Sun ***et al.*** **2015** [322]	RACS	AUROC = 0.85		✓				✓
Comments: no README on the code, no instructions, no data. **Target:** Drug combination ranking according to similarity with known combinations.								
Wildenhain ***et al.*** **2015** [323]	SONAR	AUROC = 0.91			✓			
Comments: Only data available, not a viable protocol to deploy the predictor. The benchmark presented was not actually performed by the authors, was conducted with challenge results.								
Chen ***et al.*** **2016** [324]	NLSS	AUROC = 0.91						
Target: Literature binary mining (synergistic/non-synergistic).								
Gayvert ***et al.*** **2017** [325]	RF	AUROC = 0.87; Accuracy = 0.82						✓
Target: Chou-Talalay [312] synergy score.								

Study	**Algorithms**	**Classification**	**Regression**	**GitHub**	**Website**	**New samples**	**Benchmark**	**Cancer**
Li ***et al.*** **2017** [47]	SyDRa	AUROC = 0.89						✓
Comments: These results regard the training data. **Target:** Definition from DREAM challenge.								

Xu ***et al.*** **2017** [326]	PDC-SGB	AUROC = 0.95;
		Accuracy = 0.90;
		F1-score = 0.81;
		Recall = 0.93
							✓	

Target: Positives from DCDB [319], negatives randomly generated.								
Shi ***et al.*** **2017** [327]	Ensemble	AUROC = 0.95		✓				
Comments: Github exists, but only shows datasets, no code. **Target:** Positive samples from DCDB [319], unlabelled pairs were considered negative.								

Preuer ***et al.*** **2018** [305]	DeepSynergy	AUROC = 0.90;
		Accuracy = 0.92;
		Recall = 0.56;
		Specificity = 0.51
			MSE = 255.50;
			RMSE = 15.91;
			Spearman = 0.73
				✓	✓			✓

Comments: Github exists, but it is unclear how the code can be deployed on other datasets. Subsequent models claim to benchmark this one, but in fact they simply retrain a neural network with the same architecture. The website does not allow the submission of neither cell lines nor drugs, thus, the user is restricted to the dataset used for model development. **Target:** Loewe additivity model.								
Janizek ***et al.*** **2018** [300]	TreeCombo		MSE = 0.52; Spearman = 0.70				✓	✓
Comments: It is unclear how the benchmark was performed. The presented MSE is highly dubious, as the range is very far out from the usual and incompatible with the Spearman correlation score. **Target:** Loewe additivity model.								
Chen ***et al.*** **2018** [328]	DBN	F1-score = 0.65; Recall = 0.60; Precision = 0.72					✓	✓
Comments: The benchmark presented was not actually performed, it represents challenge results. **Target:** Loewe additivity model.								
Shi ***et al.*** **2019** [233]	TLMCS	AUROC = 0.82						
Target: Only positive samples are clear.								

Study	**Algorithms**	**Classification**	**Regression**	**GitHub**	**Website**	**New samples**	**Benchmark**	**Cancer**

Cheng ***et al.*** **2019** [329]	SynerDrug	AUROC = 0.95;
		F1-score = 0.88;
		Recall = 0.87; Precision = 0.90
				✓				✓

Target: Positive samples DCDB [319], negative samples randomly generated.								

Sidorov ***et al.*** **2019** [330]	RF, XGB		RMSE = 40.40;
			Spearman = 0.60;
			Pearson = 0.65; R^2 = 0,44
					✓			✓

Comments: No GitHub available, but there is a zip file containing the code, however, the link to this file does not work. Website only contains data. **Target:** ComboScore.								
Ianevski ***et al.*** **2019** [315]	DE-CREASE		Pearson = 0.87		✓	✓		✓
Target: Loewe additivity and Bliss independence.								
Zhang ***et al.*** **2019** [331]	SyFFM	AUROC = 0.93; F1-score = 0.76			✓		✓	✓
Comments: The section of the code available is not enough for benchmark. The scope of the authors diverges cancer drug synergy prediction, **Target:** Positives – FDA approved drug combinations; Negatives – randomly generated.								
Jiang ***et al.*** **2020** [332]	GCN	AUROC = 0.89; Accuracy = 0.92					✓	✓
Target: Loewe additivity.								

Julkunen ***et al.*** **2020** [333]	comboFm		RMSE = 11.50;
			Spearman = 0.90;
			Pearson = 0.96
							✓	✓

Class: An inspection of the protocol reveals heavy concerns with data leakage, thus raising doubts about the reported results.**Target:** Claimed to be ComboScore, code inspection also raises doubts on this regard.								

Zhang ***et al.*** **2021** [306]	AuDNNSynergy	AUROC = 0.91;
		Accuracy = 0.93;
		Precision = 0.72
							✓	✓

Target: Unclear, from the available information.								

Liu ***et al.*** **2021** [308]	TranSynergy	AUROC = 0.91	MSE = 232.00;
			Spearman = 0.73;
			Pearson = 0.75
				✓			✓	✓

Target: Loewe additivity.								

Study	Algorithms	Classification	Regression	GitHub	Website	New samples	Benchmark	Cancer
Wang *et al.* 2021 [334]	DeepDDS	AUROC = 0.66; Accuracy = 0.64; Recall = 0.67; Precision = 0.80		✓			✓	✓
Target: Loewe additivity.								
Kuru *et al.* 2022 [307]	Match-Maker		MSE = 267.90; Spearman = 0.69; Pearson = 0.69	✓			✓	✓
Comments: Matchmaker claims to benchmark authors with no code or website available. Github does not allow the full pipeline deployment for prediction of unseen samples. **Target:** Loewe additivity.								

References

[1] Shu Feng Zhou and Wei Zhu Zhong. "Drug Design and Discovery: Principles and Applications". In: *Molecules : A Journal of Synthetic Chemistry and Natural Product Chemistry* 22 (2 Feb. 2017). ISSN: 14203049. DOI: 10.3390/MOLECULES22020279. URL: /pmc/articles/PMC6155886/%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6155886/.

[2] U.S. Food and Drug Administration. "Drugs - FDA Glossary of Terms". In: *https://www.fda.gov/drugs/drug-approvals-and-databases/drugsfda-glossary-terms* (2017).

[3] Sneha Govardhanagiri, Shipra Bethi, and Ganji Purnachandra Nagaraju. "Small Molecules and Pancreatic Cancer Trials and Troubles". In: *Breaking Tolerance to Pancreatic Cancer Unresponsiveness to Chemotherapy* (2019), pp. 117–131. DOI: 10.1016/B978-0-12-817661-0.00008-1.

[4] Dohyun Im et al. "Structure of the dopamine D2 receptor in complex with the antipsychotic drug spiperone". In: *Nature Communications* 11 (1 Dec. 2020). ISSN: 20411723. DOI: 10.1038/S41467-020-20221-0.

[5] LLC and Warren DeLano Schrödinger. "PyMOL". In: (May 2020).

[6] Richard C. Mohs and Nigel H. Greig. "Drug discovery and development: Role of basic biological research". In: *Alzheimer's & Dementia : Translational Research & Clinical Interventions* 3 (4 Nov. 2017), p. 651. ISSN: 23528737. DOI: 10.1016/J.TRCI.2017.10.005. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5725284/.

[7] Alan Talevi. "Computer-Aided Drug Design: An Overview". In: *Computational Drug Discovery and Design* (2018). Ed. by Mohini Gore and Umesh B Jagtap, pp. 1–19. DOI: 10.1007/978-1-4939-7756-7_1. URL: https://doi.org/10.1007/978-1-4939-7756-7_1.

[8] Biswa Mohan Sahoo et al. "Drug Repurposing Strategy (DRS): Emerging Approach to Identify Potential Therapeutics for Treatment of Novel Coronavirus Infection". In: *Frontiers in Molecular Biosciences* 8 (Feb. 2021), p. 35. ISSN: 2296889X. DOI: 10.3389/FMOLB.2021.628144/BIBTEX.

[9] Sudeep Pushpakom et al. "Drug repurposing: progress, challenges and recommendations". In: *Nature Reviews Drug Discovery 2018 18:1* 18 (1 Oct. 2018), pp. 41–58. ISSN: 1474-1784. DOI: 10.1038/nrd.2018.168. URL: https://www.nature.com/articles/nrd.2018.168.

[10] Ciska Verbaanderd, Ilse Rooman, and Isabelle Huys. "Exploring new uses for existing drugs: innovative mechanisms to fund independent clinical research". In: *Trials* 22 (1 Dec. 2021), pp. 1–13. ISSN: 17456215. DOI: 10.1186/S13063-021-05273-X/TABLES/4. URL: https://trialsjournal.biomedcentral.com/articles/10.1186/s13063-021-05273-x.

[11] AJ Preto. “Drug Design and Development”. In: *Canva* (Nov. 2022). URL: https://www.canva.com/design/DAFRjsRJTC8/OrziP1T45oESpUYferzuLw/view.

[12] Bhaven N Sampat and Frank R Lichtenberg. “What are the respective roles of the public and private sectors in pharmaceutical innovation?” In: *Health affairs (Project Hope)* 30 (2 Feb. 2011), pp. 332–339. ISSN: 1544-5208 (Electronic). DOI: 10.1377/hlthaff.2009.0917.

[13] Imogen Fitt and Steve Holloway. *Funding Analysis for AI in Drug Development & Clinical Trials: 2011-2021 - Signify Research (The content of this report should not be used without the appropriate authors’ consent.)* Signify Research, 2021, pp. 1–17.

[14] Hemant Arya and Mohane Selvaraj Coumarb. “Lead identification and optimization”. In: *The Design and Development of Novel Drugs and Vaccines: Principles and Protocols* (Jan. 2021), pp. 31–63. DOI: 10.1016/B978-0-12-821471-8.00004-0.

[15] Fernando D. Prieto-Martínez et al. “Computational Drug Design Methods—Current and Future Perspectives”. In: *In Silico Drug Design: Repurposing Techniques and Methodologies* (Jan. 2019), pp. 19–44. DOI: 10.1016/B978-0-12-816125-8.00002-X.

[16] M. S. Attene-Ramos, C. P. Austin, and M. Xia. “High Throughput Screening”. In: *Encyclopedia of Toxicology: Third Edition* (Jan. 2014), pp. 916–917. DOI: 10.1016/B978-0-12-386454-3.00209-8.

[17] Brijesh Singh Yadav and Vijay Tripathi. “Recent Advances in the System Biology-based Target Identification and Drug Discovery.” In: *Current topics in medicinal chemistry* 18 (20 2018), pp. 1737–1744. ISSN: 1873-4294 (Electronic). DOI: 10.2174/1568026618666181025112344.

[18] A Lavecchia and Di Giovanni C. “Virtual screening strategies in drug discovery: a critical review”. In: *Current medicinal chemistry* 20 (23 June 2013), pp. 2839–2860. ISSN: 1875-533X. DOI: 10.2174/09298673113209990001. URL: https://pubmed.ncbi.nlm.nih.gov/23651302/.

[19] Sangeetha Subramaniam, Monica Mehrotra, and Dinesh Gupta. “Virtual high throughput screening (vHTS) - A perspective”. In: *Bioinformation* 3 (1 Sept. 2008), p. 14. ISSN: 09738894. DOI: 10.6026/97320630003014. URL: /pmc/articles/PMC2586130/%20/pmc/articles/PMC2586130/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2586130/.

[20] Odilia Osakwe. “The Significance of Discovery Screening and Structure Optimization Studies”. In: *Social Aspects of Drug Discovery, Development and Commercialization* (Jan. 2016), pp. 109–128. DOI: 10.1016/B978-0-12-802220-7.00005-3.

[21] Luca Pinzi and Giulio Rastelli. In: *International Journal of Molecular Sciences* 20 (18 2019). ISSN: 14220067. DOI: 10.3390/ijms20184331.

[22] Gregory Sliwoski et al. “Computational Methods in Drug Discovery”. In: *Pharmacological Reviews* 66 (1 Jan. 2014). Ed. by Eric L Barker, 334 LP –395. DOI: 10.1124/pr.112.007336. URL: http://pharmrev.aspetjournals.org/content/66/1/334.abstract.

[23] Ali Ezzat et al. “Computational prediction of drug-target interactions using chemogenomic approaches: an empirical survey.” In: *Briefings in bioinformatics* 20 (4 July 2019), pp. 1337–1357. ISSN: 1477-4054 (Electronic). DOI: 10.1093/bib/bby002.

[24] Mohammad Hassan Baig et al. “Computer Aided Drug Design: Success and Limitations.” In: *Current pharmaceutical design* 22 (5 2016), pp. 572–581. ISSN: 1873-4286 (Electronic). DOI: 10.2174/1381612822666151125000550.

[25] Benjamin Webb and Andrej Sali. “Protein structure modeling with MODELLER”. In: *Methods in Molecular Biology* 1137 (2014). ISSN: 10643745. DOI: 10.1007/978-1-4939-0366-5_1.

[26] Ambrish Roy, Alper Kucukural, and Yang Zhang. “I-TASSER: a unified platform for automated protein structure and function prediction”. In: *Nature protocols* 5 (4 2010), pp. 725–738. ISSN: 1750-2799. DOI: 10.1038/NPROT.2010.5. URL: https://pubmed.ncbi.nlm.nih.gov/20360767/.

[27] Torsten Schwede et al. “SWISS-MODEL: An automated protein homology-modeling server”. In: *Nucleic acids research* 31 (13 July 2003), pp. 3381–3385. ISSN: 1362-4962. DOI: 10.1093/NAR/GKG520. URL: https://pubmed.ncbi.nlm.nih.gov/12824332/.

[28] Muhammed Tilahun Muhammed and Esin Aki-Yalcin. “Homology modeling in drug discovery: Overview, current applications, and future perspectives.” In: *Chemical biology & drug design* 93 (1 Jan. 2019), pp. 12–20. ISSN: 1747-0285 (Electronic). DOI: 10.1111/cbdd.13388.

[29] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (7873 Aug. 2021), pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/S41586-021-03819-2. URL: https://pubmed.ncbi.nlm.nih.gov/34265844/.

[30] Andrew Leaver-Fay et al. “Rosetta3: An object-oriented software suite for the simulation and design of macromolecules”. In: *Methods in Enzymology* 487 (C 2011), pp. 545–574. ISSN: 00766879. DOI: 10.1016/B978-0-12-381270-4.00019-6.

[31] Carol A Rohl et al. “Protein structure prediction using Rosetta.” In: *Methods in enzymology* 383 (2004), pp. 66–93. ISSN: 0076-6879 (Print). DOI: 10.1016/S0076-6879(04)83004-0.

[32] Bian Li, Jeffrey Mendenhall, and Jens Meiler. “Interfaces Between Alpha-helical Integral Membrane Proteins: Characterization, Prediction, and Docking”. In: *Computational and Structural Biotechnology Journal* 17 (Jan. 2019), pp. 699–711. ISSN: 20010370. DOI: 10.1016/J.CSBJ.2019.05.005.

[33] Kanica Sachdev and Manoj Kumar Gupta.
"A comprehensive review of feature based methods for drug target interaction prediction".
In: *Journal of Biomedical Informatics* 93 (May 2019), p. 103159. ISSN: 1532-0464.
DOI: 10.1016/J.JBI.2019.103159.

[34] Maria Kontoyianni. "Docking and Virtual Screening in Drug Discovery".
In: *Methods in molecular biology (Clifton, N.J.)* 1647 (2017), pp. 255–266.
ISSN: 1940-6029. DOI: 10.1007/978-1-4939-7201-2_18.
URL: https://pubmed.ncbi.nlm.nih.gov/28809009/.

[35] Amara Jabeen and Shoba Ranganathan.
"Applications of machine learning in GPCR bioactive ligand discovery".
In: *Current Opinion in Structural Biology* 55 (Apr. 2019), pp. 66–76. ISSN: 0959-440X.
DOI: 10.1016/J.SBI.2019.03.022.

[36] Yijie Ding, Jijun Tang, and Fei Guo.
"The Computational Models of Drug-target Interaction Prediction".
In: *Protein and peptide letters* 27 (5 Apr. 2020), pp. 348–358. ISSN: 1875-5305.
DOI: 10.2174/0929866526666190410124110.
URL: https://pubmed.ncbi.nlm.nih.gov/30968771/.

[37] "Drug-Target Interactions: Prediction Methods and Applications".
In: *Current protein & peptide science* 19 (6 Dec. 2018), pp. 1–1. ISSN: 1875-5550.
URL: https://pubmed.ncbi.nlm.nih.gov/27829350/.

[38] AJ Preto. "Computer-Aided Drug Design". In: *Canva* (2022). URL:
https://www.canva.com/design/DAFRi6YtYx4/5cAZLj5wOgo2-wLZh9DIdQ/view.

[39] Sunghwan Kim et al. "PubChem 2019 update: Improved access to chemical data".
In: *Nucleic Acids Research* 47 (D1 Jan. 2019), pp. D1102–D1109. ISSN: 13624962.
DOI: 10.1093/NAR/GKY1033.

[40] D. S. Wishart.
"DrugBank: a comprehensive resource for in silico drug discovery and exploration".
In: *Nucleic Acids Research* (2006). ISSN: 0305-1048. DOI: 10.1093/nar/gkj067.

[41] David S. Wishart et al.
"DrugBank 5.0: a major update to the DrugBank database for 2018".
In: *Nucleic acids research* 46 (D1 Jan. 2018), pp. D1074–D1082. ISSN: 1362-4962.
DOI: 10.1093/NAR/GKX1037. URL: https://pubmed.ncbi.nlm.nih.gov/29126136/.

[42] Jessica Vamathevan et al.
"Applications of machine learning in drug discovery and development".
In: *Nature Reviews Drug Discovery 2019 18:6* 18 (6 Apr. 2019), pp. 463–477.
ISSN: 1474-1784. DOI: 10.1038/s41573-019-0024-5.
URL: https://www.nature.com/articles/s41573-019-0024-5.

[43] Noel M. O'Boyle et al. "Open Babel: An open chemical toolbox".
In: *Journal of cheminformatics* 3 (10 Oct. 2011). ISSN: 1758-2946.
DOI: 10.1186/1758-2946-3-33.
URL: https://pubmed.ncbi.nlm.nih.gov/21982300/.

[44] Hirotomo Moriwaki et al. “Mordred: A molecular descriptor calculator”. In: *Journal of Cheminformatics* 10 (1 Feb. 2018). ISSN: 17582946. DOI: 10.1186/S13321-018-0258-Y.

[45] Yiqun Cao et al. “ChemmineR: a compound mining framework for R”. In: *Bioinformatics (Oxford, England)* 24 (15 Aug. 2008), pp. 1733–1734. ISSN: 1367-4811. DOI: 10.1093/BIOINFORMATICS/BTN307. URL: https://pubmed.ncbi.nlm.nih.gov/18596077/.

[46] Steven Kearnes et al. “Molecular graph convolutions: moving beyond fingerprints.” In: *Journal of computer-aided molecular design* 30 (8 Aug. 2016), pp. 595–608. ISSN: 1573-4951 (Electronic). DOI: 10.1007/s10822-016-9938-8.

[47] Xiangyi Li et al. “Prediction of synergistic anti-cancer drug combinations based on drug target network and drug induced gene expression profiles”. In: *Artificial Intelligence in Medicine* 83 (Nov. 2017), pp. 35–43. ISSN: 0933-3657. DOI: 10.1016/J.ARTMED.2017.05.008.

[48] Qinqing Liu et al. “OctSurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction”. In: *Journal of molecular graphics & modelling* 105 (June 2021). ISSN: 1873-4243. DOI: 10.1016/J.JMGM.2021.107865. URL: https://pubmed.ncbi.nlm.nih.gov/33640787/.

[49] David Lee Nelson and Michael Cox. “Lehninger Principles of Biochemistry”. In: (2013).

[50] David Sacks et al. “Multisociety Consensus Quality Improvement Revised Consensus Statement for Endovascular Therapy of Acute Ischemic Stroke.” In: *International journal of stroke : official journal of the International Stroke Society* 13 (6 Aug. 2018), pp. 612–632. ISSN: 1747-4949 (Electronic). DOI: 10.1177/1747493018778713.

[51] Artem Lysenko et al. “An integrative machine learning approach for prediction of toxicity-related drug safety.” In: *Life science alliance* 1 (6 Dec. 2018), e201800098. ISSN: 2575-1077 (Electronic). DOI: 10.26508/lsa.201800098.

[52] Fengxu Wu et al. “Computational Approaches in Preclinical Studies on Drug Discovery and Development”. In: *Frontiers in Chemistry* 8 (2020), p. 726. ISSN: 2296-2646. DOI: 10.3389/fchem.2020.00726. URL: https://www.frontiersin.org/article/10.3389/fchem.2020.00726.

[53] Anna O Basile, Alexandre Yahi, and Nicholas P Tatonetti. “Artificial Intelligence for Drug Toxicity and Safety”. In: *Trends in Pharmacological Sciences* 40 (9 2019), pp. 624–635. ISSN: 0165-6147. DOI: https://doi.org/10.1016/j.tips.2019.07.005. URL: http://www.sciencedirect.com/science/article/pii/S0165614719301427.

[54] Jun Shang et al. "Comparative analyses of structural features and scaffold diversity for purchasable compound libraries". In: *Journal of Cheminformatics* 9 (1 2017), p. 25. ISSN: 1758-2946. DOI: 10.1186/s13321-017-0212-4. URL: https://doi.org/10.1186/s13321-017-0212-4.

[55] Ulrike Schmidt et al. "SuperToxic: a comprehensive database of toxic compounds". In: *Nucleic Acids Research* 37 (supplement 1 2008), pp. D295–D299. ISSN: 0305-1048. DOI: 10.1093/nar/gkn850. URL: https://doi.org/10.1093/nar/gkn850.

[56] Dongyue Cao et al. "ADMET Evaluation in Drug Discovery. 11. PharmacoKinetics Knowledge Base (PKKB): A Comprehensive Database of Pharmacokinetic and Toxic Properties for Drugs". In: *Journal of Chemical Information and Modeling* 52 (5 May 2012), pp. 1132–1137. ISSN: 1549-9596. DOI: 10.1021/ci300112j. URL: https://doi.org/10.1021/ci300112j.

[57] Antony J Williams et al. "The CompTox Chemistry Dashboard: a community data resource for environmental chemistry." In: *Journal of cheminformatics* 9 (1 Nov. 2017), p. 61. ISSN: 1758-2946 (Print). DOI: 10.1186/s13321-017-0247-6.

[58] Ann M. Richard and Clar Lynda R. Williams. "Distributed structure-searchable toxicity (DSSTox) public database network: A proposal". In: *Mutation Research - Fundamental and Molecular Mechanisms of Mutagenesis* 499 (1 Jan. 2002), pp. 27–52. ISSN: 00275107. DOI: 10.1016/S0027-5107(01)00289-5. URL: https://pubmed.ncbi.nlm.nih.gov/11804603/.

[59] Eleni E. Litsa, Payel Das, and Lydia E. Kavraki. "Prediction of drug metabolites using neural machine translation". In: *Chemical Science* 11 (47 Dec. 2020), p. 12777. ISSN: 20416539. DOI: 10.1039/D0SC02639E. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8162519/.

[60] Jie Shen et al. "Estimation of ADME Properties with Substructure Pattern Recognition". In: *Journal of Chemical Information and Modeling* 50 (6 June 2010), pp. 1034–1041. ISSN: 1549-9596. DOI: 10.1021/ci100104j. URL: https://doi.org/10.1021/ci100104j.

[61] Antoine Daina, Olivier Michielin, and Vincent Zoete. "SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules". In: *Scientific Reports* 7 (1 2017), p. 42717. ISSN: 2045-2322. DOI: 10.1038/srep42717. URL: https://doi.org/10.1038/srep42717.

[62] Grace Patlewicz and Jeremy M Fitzpatrick. "Current and Future Perspectives on the Development, Evaluation, and Application of in Silico Approaches for Predicting Toxicity." In: *Chemical research in toxicology* 29 (4 Apr. 2016), pp. 438–451. ISSN: 1520-5010 (Electronic). DOI: 10.1021/acs.chemrestox.5b00388.

[63] Kaitlyn M Gayvert, Neel S Madhukar, and Olivier Elemento. “A Data-Driven Approach to Predicting Successes and Failures of Clinical Trials.” In: *Cell chemical biology* 23 (10 Oct. 2016), pp. 1294–1301. ISSN: 2451-9448 (Electronic). DOI: 10.1016/j.chembiol.2016.07.023.

[64] Yannick Djoumbou-Feunang et al. “BioTransformer: a comprehensive computational tool for small molecule metabolism prediction and metabolite identification”. In: *Journal of Cheminformatics* 11 (1 2019), p. 2. ISSN: 1758-2946. DOI: 10.1186/s13321-018-0324-5. URL: https://doi.org/10.1186/s13321-018-0324-5.

[65] Xavier Martinez et al. “MetaTrans: an open-source pipeline for metatranscriptomics”. In: *Scientific reports* 6 (May 2016). ISSN: 2045-2322. DOI: 10.1038/SREP26447. URL: https://pubmed.ncbi.nlm.nih.gov/27211518/.

[66] Lars Ridder and Markus Wagener. “SyGMa: Combining Expert Knowledge and Empirical Scoring in the Prediction of Metabolites”. In: *ChemMedChem* 3 (5 2008), pp. 821–832. DOI: 10.1002/cmdc.200700312. URL: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/cmdc.200700312.

[67] Christina De Bruyn Kops et al. “GLORYx: Prediction of the Metabolites Resulting from Phase 1 and Phase 2 Biotransformations of Xenobiotics”. In: *Chemical research in toxicology* 34 (2 Feb. 2021), pp. 286–299. ISSN: 1520-5010. DOI: 10.1021/ACS.CHEMRESTOX.0C00224. URL: https://pubmed.ncbi.nlm.nih.gov/32786543/.

[68] Mahmoud Ghandi et al. “Next-generation characterization of the Cancer Cell Line Encyclopedia”. In: *Nature* 569 (7757 May 2019), pp. 503–508. ISSN: 1476-4687. DOI: 10.1038/S41586-019-1186-3. URL: https://pubmed.ncbi.nlm.nih.gov/31068700/.

[69] Wanjuan Yang et al. “Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells”. In: *Nucleic acids research* 41 (Database issue Jan. 2013). ISSN: 1362-4962. DOI: 10.1093/NAR/GKS1111. URL: https://pubmed.ncbi.nlm.nih.gov/23180760/.

[70] Susan L. Holbeck et al. “The National Cancer Institute ALMANAC: A comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity”. In: *Cancer Research* 77 (13 July 2017), pp. 3564–3576. ISSN: 15387445. DOI: 10.1158/0008-5472.CAN-17-0489.

[71] D. Amaratunga, H. Göhlmann, and P.J. Peeters. “Microarrays”. In: *Comprehensive Medicinal Chemistry II* (2007), pp. 87–106. DOI: 10.1016/B0-08-045044-X/00078-X. URL: https://linkinghub.elsevier.com/retrieve/pii/B008045044X00078X.

[72] Javier De Las Rivas and Carlos Prieto. “Protein interactions: mapping interactome networks to support drug target discovery and selection”. In: *Methods in Molecular Biology* 912 (2012), pp. 279–296.

[73] K. Luck et al. “Interactomes - Scaffolds of Cellular Systems”. In: *Encyclopedia of Cell Biology* 4 (2016), pp. 187–198. DOI: 10.1016/B978-0-12-394447-4.40037-4.

[74] Suruchi Aggarwal et al. “Posttranslational modifications in systems biology”. In: *Advances in Protein Chemistry and Structural Biology* 127 (Jan. 2021), pp. 93–126. ISSN: 18761631. DOI: 10.1016/bs.apcsb.2021.03.005.

[75] Tiao Lai Huang and Chin Chuen Lin. “Advances in Biomarkers of Major Depressive Disorder”. In: *Advances in Clinical Chemistry* 68 (2015), pp. 177–204. ISSN: 00652423. DOI: 10.1016/bs.acc.2014.11.003.

[76] Ruth Stoney et al. “Mapping biological process relationships and disease perturbations within a pathway network”. In: *npj Systems Biology and Applications 2018 4:1* 4 (1 June 2018), pp. 1–11. ISSN: 2056-7189. DOI: 10.1038/s41540-018-0055-2. URL: https://www.nature.com/articles/s41540-018-0055-2.

[77] Paul Shannon et al. “Cytoscape: a software environment for integrated models of biomolecular interaction networks”. In: *Genome research* 13 (11 Nov. 2003), pp. 2498–2504. ISSN: 1088-9051. DOI: 10.1101/GR.1239303. URL: https://pubmed.ncbi.nlm.nih.gov/14597658/.

[78] Robyn P. Araujo, Lance A. Liotta, and Emanuel F. Petricoin. “Proteins, drug targets and the mechanisms they control: the simple truth about complex networks”. In: *Nature Reviews Drug Discovery 2007 6:11* 6 (11 Nov. 2007), pp. 871–880. ISSN: 1474-1784. DOI: 10.1038/nrd2381. URL: https://www.nature.com/articles/nrd2381.

[79] Woong Hee Shin et al. “Current Challenges and Opportunities in Designing Protein-Protein Interaction Targeted Drugs”. In: *Advances and Applications in Bioinformatics and Chemistry* 13 (Nov. 2020), pp. 11–25. ISSN: 11786949. DOI: 10.2147/AABC.S235542. URL: https://www.dovepress.com/current-challenges-and-opportunities-in-designing-proteinndashprotein--peer-reviewed-fulltext-article-AABC.

[80] Priya Tolani et al. “Big data, integrative omics and network biology”. In: *Advances in Protein Chemistry and Structural Biology* 127 (Jan. 2021), pp. 127–160. ISSN: 18761631. DOI: 10.1016/bs.apcsb.2021.03.006.

[81] Terry Kenakin. “Biased Receptor Signaling in Drug Discovery.” In: *Pharmacological reviews* 71 (2 Apr. 2019), pp. 267–315. ISSN: 1521-0081 (Electronic). DOI: 10.1124/pr.118.016790.

[82] Rebeca Diez-Alarcia et al. “Big Data Challenges Targeting Proteins in GPCR Signaling Pathways; Combining PTML-ChEMBL Models and [35S]GTPγS Binding Assays”. In: *ACS Chemical Neuroscience* 10 (11 2019), pp. 4476–4491. ISSN: 19487193. DOI: 10.1021/acschemneuro.9b00302.

[83] Rita Santos et al. “A comprehensive map of molecular drug targets”. In: *Nature reviews. Drug discovery* 16 (1 Dec. 2017), p. 19. ISSN: 14741784. DOI: 10.1038/NRD.2016.230. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6314433/.

[84] Shailza Singh, Balwant Kumar Malik, and Durlabh Kumar Sharma. “Molecular drug targets and structure based drug design: A holistic approach”. In: *Bioinformation* 1 (8 Dec. 2006), p. 314. ISSN: 09738894. DOI: 10.6026/97320630001314. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1891708/.

[85] Mathias Rask-Andersen, Markus Sällman Almén, and Helgi B. Schiöth. “Trends in the exploitation of novel drug targets”. In: *Nature reviews. Drug discovery* 10 (8 Aug. 2011), pp. 579–590. ISSN: 1474-1784. DOI: 10.1038/NRD3478. URL: https://pubmed.ncbi.nlm.nih.gov/21804595/.

[86] John P. Overington, Bissan Al-Lazikani, and Andrew L. Hopkins. “How many drug targets are there?” In: *Nature reviews. Drug discovery* 5 (12 Dec. 2006), pp. 993–996. ISSN: 1474-1776. DOI: 10.1038/NRD2199. URL: https://pubmed.ncbi.nlm.nih.gov/17139284/.

[87] Simon C. Bull and Andrew J. Doig. “Properties of Protein Drug Target Classes”. In: *PLoS ONE* 10 (3 Mar. 2015). ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0117955. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4379170/.

[88] Ruedi Aebersold et al. “How many human proteoforms are there?” In: *Nature chemical biology* 14 (3 Feb. 2018), p. 206. ISSN: 15524469. DOI: 10.1038/NCHEMBIO.2576. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5837046/.

[89] Malgorzata Skwarczynska and Christian Ottmann. “Protein-protein interactions as drug targets”. In: *Future medicinal chemistry* 7 (16 2015), pp. 2195–2219.

[90] Helen M. Berman et al. “The Protein Data Bank”. In: *Nucleic Acids Research* 28 (1 Jan. 2000), pp. 235–242. ISSN: 03051048. DOI: 10.1093/NAR/28.1.235.

[91] Max Kotlyar et al. “IID 2018 update: context-specific physical protein-protein interactions in human, model organisms and domesticated species”. In: *Nucleic acids research* 47 (D1 Jan. 2019), pp. D581–D589. ISSN: 1362-4962. DOI: 10.1093/NAR/GKY1037. URL: https://pubmed.ncbi.nlm.nih.gov/30407591/.

[92] Kasper Lage. "Protein-protein interactions and genetic diseases: The interactome". In: *Biochimica et biophysica acta* 1842 (10 Oct. 2014), pp. 1971–1980. ISSN: 0006-3002. DOI: 10.1016/J.BBADIS.2014.05.028. URL: https://pubmed.ncbi.nlm.nih.gov/24892209/.

[93] David C. Fry. "Targeting protein-protein interactions for drug discovery". In: *Methods in molecular biology (Clifton, N.J.)* 1278 (2015), pp. 93–106. ISSN: 1940-6029. DOI: 10.1007/978-1-4939-2425-7_6. URL: https://pubmed.ncbi.nlm.nih.gov/25859945/.

[94] Xu Ran and Jason E. Gestwicki. "Inhibitors of Protein-Protein Interactions (PPIs): An Analysis of Scaffold Choices and Buried Surface Area". In: *Current opinion in chemical biology* 44 (June 2018), p. 75. ISSN: 18790402. DOI: 10.1016/J.CBPA.2018.06.004. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6066447/.

[95] J. N. Israelachvili et al. "Physical principles of membrane organization". In: *Quarterly Reviews of Biophysics* 13 (2 1980), pp. 121–200. ISSN: 14698994. DOI: 10.1017/S0033583500001645.

[96] M. Michael Gromiha and Yu Yen Ou. "Bioinformatics approaches for functional annotation of membrane proteins". In: *Briefings in Bioinformatics* 15 (2 2014), pp. 155–168. ISSN: 14774054. DOI: 10.1093/BIB/BBT015.

[97] Markus Sällman Almén et al. "Mapping the human membrane proteome: A majority of the human membrane proteins can be classified according to function and evolutionary origin". In: *BMC Biology* 7 (1 Aug. 2009), p. 50. ISSN: 17417007. DOI: 10.1186/1741-7007-7-50/FIGURES/7. URL: https://bmcbiol.biomedcentral.com/articles/10.1186/1741-7007-7-50.

[98] Mark L. Chiu. "Introduction to membrane proteins". In: *Current protocols in protein science* Chapter 29 (SUPPL.67 Feb. 2012). ISSN: 1934-3663. DOI: 10.1002/0471140864.PS2901S67. URL: https://pubmed.ncbi.nlm.nih.gov/22294326/.

[99] S Tan, HT Tan, and MCM Chung. "Membrane proteins and membrane proteomics". In: *Proteomics* 8 (2008), pp. 3924–3932.

[100] et al. Johnson A Alberts B Lewis J. "Membrane Proteins". In: *Molecular Biology of the Cell* (2002). Ed. by New York: Garland Science. URL: https://www.ncbi.nlm.nih.gov/books/NBK26878/.

[101] Jose G. Almeida et al. "Membrane proteins structures: A review on computational modeling tools". In: *Biochimica et Biophysica Acta - Biomembranes* 1859 (10 Oct. 2017), pp. 2021–2039. ISSN: 18792642. DOI: 10.1016/J.BBAMEM.2017.07.008.

[102] Isabel Moraes et al. "Membrane protein structure determination - The next generation". In: *Biochimica et Biophysica Acta - Biomembranes* 1838 (1 PARTA 2014), pp. 78–87. ISSN: 00052736. DOI: 10.1016/J.BBAMEM.2013.07.010.

[103] Vadim Cherezov et al. "High-resolution crystal structure of an engineered human beta2-adrenergic G protein-coupled receptor". In: *Science (New York, N.Y.)* 318 (5854 Nov. 2007), pp. 1258–1265. ISSN: 1095-9203. DOI: 10.1126/SCIENCE.1150577. URL: https://pubmed.ncbi.nlm.nih.gov/17962520/.

[104] Daniel M Rosenbaum, Søren G F Rasmussen, and Brian K Kobilka. "The structure and function of G-protein-coupled receptors." In: *Nature* 459 (7245 May 2009), pp. 356–363. ISSN: 1476-4687 (Electronic). DOI: 10.1038/nature08144.

[105] Krishna Sriram and Paul A. Insel. "G protein-coupled receptors as targets for approved drugs: How many targets and how many drugs?" In: *Molecular Pharmacology* 93 (4 2018), pp. 251–258. ISSN: 15210111. DOI: 10.1124/mol.117.111062.

[106] Sangmin Seo et al. "Prediction of GPCR-Ligand Binding Using Machine Learning Algorithms." In: *Computational and mathematical methods in medicine* 2018 (2018), p. 6565241. ISSN: 1748-6718 (Electronic). DOI: 10.1155/2018/6565241.

[107] Li-Kun Yang, Zhi-Shuai Hou, and Ya-Xiong Tao. "Biased signaling in naturally occurring mutations of G protein-coupled receptors associated with diverse human diseases." In: *Biochimica et biophysica acta. Molecular basis of disease* 1867 (1 Sept. 2020), p. 165973. ISSN: 1879-260X (Electronic). DOI: 10.1016/j.bbadis.2020.165973.

[108] Mengjie Lu and Beili Wu. "Structural studies of G protein-coupled receptors." In: *IUBMB life* 68 (11 Nov. 2016), pp. 894–903. ISSN: 1521-6551 (Electronic). DOI: 10.1002/iub.1578.

[109] Diana Lindner et al. "Functional role of the extracellular N-terminal domain of neuropeptide Y subfamily receptors in membrane integration and agonist-stimulated internalization." In: *Cellular signalling* 21 (1 Jan. 2009), pp. 61–68. ISSN: 1873-3913 (Electronic). DOI: 10.1016/j.cellsig.2008.09.007.

[110] Daniel Wacker, Raymond C. Stevens, and Bryan L. Roth. "How ligands illuminate GPCR molecular pharmacology". In: *Cell* 170 (3 July 2017), p. 414. ISSN: 10974172. DOI: 10.1016/J.CELL.2017.07.009. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5560499/.

[111] Gáspár Pándy-Szekeres et al. "GPCRdb in 2018: adding GPCR structure models and ligands." In: *Nucleic acids research* 46 (D1 Jan. 2018), pp. D440–D446. ISSN: 1362-4962 (Electronic). DOI: 10.1093/nar/gkx1109.

[112] W Chan et al. “GPCR-EXP: A semi-manually curated database for experimentally-solved and predicted GPCR structures.” In: (2018).
URL: https://zhanglab.ccmb.med.umich.edu/GPCR-EXP/..

[113] Marjorie Damian et al. “GHSR-D2R heteromerization modulates dopamine signaling through an effect on G protein conformation”. In: *Proceedings of the National Academy of Sciences of the United States of America* 115 (17 Apr. 2018), pp. 4501–4506.
ISSN: 1091-6490. DOI: 10.1073/PNAS.1712725115.
URL: https://pubmed.ncbi.nlm.nih.gov/29632174/.

[114] Dimitrios K. Papadopoulos et al. “Dimer formation via the homeodomain is required for function and specificity of Sex combs reduced in Drosophila”.
In: *Developmental Biology* 367 (1 July 2012), pp. 78–89. ISSN: 1095564X.
DOI: 10.1016/J.YDBIO.2012.04.021.

[115] “Anatomy of hot spots in protein interfaces”.
In: *Journal of Molecular Biology* 280 (1 July 1998), pp. 1–9. ISSN: 00222836.
DOI: 10.1006/JMBI.1998.1843.

[116] Tim Clackson and James A. Wells.
“A hot spot of binding energy in a hormone-receptor interface”.
In: *Science* 267 (5196 1995), pp. 383–386. ISSN: 00368075.
DOI: 10.1126/SCIENCE.7529940.

[117] Jinjian Jiang et al. “Prediction of protein hotspots from whole protein sequences by a random projection ensemble system”.
In: *International Journal of Molecular Sciences* 18 (7 July 2017). ISSN: 14220067.
DOI: 10.3390/IJMS18071543.

[118] Ozlem Keskin, Buyong Ma, and Ruth Nussinov.
“Hot regions in protein-protein interactions: The organization and contribution of structurally conserved hot spot residues”.
In: *Journal of Molecular Biology* 345 (5 Feb. 2005), pp. 1281–1294. ISSN: 00222836.
DOI: 10.1016/J.JMB.2004.10.077.

[119] Irina S. Moreira, Pedro A. Fernandes, and Maria J. Ramos.
“Hot spots - A review of the protein-protein interface determinant amino-acid residues”.
In: *Proteins: Structure, Function and Genetics* 68 (4 Sept. 2007), pp. 803–812.
ISSN: 08873585. DOI: 10.1002/PROT.21396.

[120] Yanhua Qiao et al. “Protein-protein interface hot spots prediction based on a hybrid feature selection strategy”. In: *BMC Bioinformatics* 19 (1 Jan. 2018), pp. 1–16.
ISSN: 14712105. DOI: 10.1186/S12859-018-2009-5/FIGURES/8. URL:
https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2009-5.

[121] Mary S. Golden et al.
“Comprehensive experimental and computational analysis of binding energy hot spots at the NF-κB essential modulator/IKKβ protein-protein interface”.

In: *Journal of the American Chemical Society* 135 (16 Apr. 2013), pp. 6242–6256. ISSN: 00027863. DOI: 10.1021/JA400914Z.

[122] "Resolving hot spots in the C-terminal dimerization domain that determine the stability of the molecular chaperone Hsp90". In: *PLoS ONE* 9 (4 Apr. 2014). ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0096031.

[123] Irina Moreira. "The Role of Water Occlusion for the Definition of a Protein Binding Hot-Spot". In: *Current Topics in Medicinal Chemistry* 15 (20 July 2015), pp. 2068–2079. ISSN: 15680266. DOI: 10.2174/1568026615666150519103733.

[124] R. M. Ramos, L. F. Fernandes, and I. S. Moreira. "Extending the applicability of the O-ring theory to protein-DNA complexes". In: *Computational Biology and Chemistry* 44 (2013), pp. 31–39. ISSN: 14769271. DOI: 10.1016/J.COMPBIOLCHEM.2013.02.005.

[125] Outi M.H. Salo-Ahen et al. "Hotspots in an obligate homodimeric anticancer target. Structural and functional effects of interfacial mutations in human thymidylate synthase". In: *Journal of Medicinal Chemistry* 58 (8 Apr. 2015), pp. 3572–3581. ISSN: 15204804. DOI: 10.1021/ACS.JMEDCHEM.5B00137.

[126] Irina S. Moreira et al. "SpotOn: High Accuracy Identification of Protein-Protein Interface Hot-Spots". In: *Scientific Reports* 7 (1 Dec. 2017). ISSN: 20452322. DOI: 10.1038/S41598-017-08321-2.

[127] Steven J. Darnell, Laura LeGault, and Julie C. Mitchell. "KFC Server: interactive forecasting of protein interaction hot spots." In: *Nucleic acids research* 36 (Web Server issue 2008). ISSN: 13624962. DOI: 10.1093/NAR/GKN346.

[128] John Kenneth Morrow and Shuxing Zhang. "Computational Prediction of Protein Hot Spot Residues". In: *Current Pharmaceutical Design* 18 (9 Feb. 2012), pp. 1255–1265. ISSN: 13816128. DOI: 10.2174/138161212799436412.

[129] Nurcan Tuncbag, Ozlem Keskin, and Attila Gursoy. "HotPoint: Hot spot prediction server for protein interfaces". In: *Nucleic Acids Research* 38 (SUPPL. 2 May 2010). ISSN: 03051048. DOI: 10.1093/NAR/GKQ323.

[130] Anastassios C. Papageorgiou, Robert Shapiro, and K. Ravi Acharya. "Molecular recognition of human angiogenin by placental ribonuclease inhibitor—an X-ray crystallographic study at 2.0 Å resolution". In: *The EMBO Journal* 16 (17 Sept. 1997), pp. 5162–5177. ISSN: 1460-2075. DOI: 10.1093/EMBOJ/16.17.5162. URL: https://onlinelibrary.wiley.com/doi/full/10.1093/emboj/16.17.5162.

[131] Miroslav Kubat. *An Introduction to Machine Learning*. Second. Springer, Cham, 2017. ISBN: 978-3-319-63913-0. DOI: https://doi.org/10.1007/978-3-319-63913-0.

[132] Petra Heijden et al. “Khwārizmī: Muhammad ibn Mūsā al-Khwārizmī”. In: *The Biographical Encyclopedia of Astronomers* (2007), pp. 631–633. DOI: 10.1007/978-0-387-30400-7_763. URL: https://link.springer.com/referenceworkentry/10.1007/978-0-387-30400-7_763.

[133] LE Rosenthal. “Computer software”. In: *Dermatologic clinics* 4 (4 1986), pp. 545–551.

[134] Malinda Dilhara, Ameya Ketkar, and Danny Dig. “Understanding Software-2.0”. In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30 (4 July 2021). ISSN: 15577392. DOI: 10.1145/3453478. URL: https://dl.acm.org/doi/10.1145/3453478.

[135] A. J. Preto. “Algorithm”. In: (Dec. 2022). URL: https://www.canva.com/design/DAFSmcEAmI8/JQVy9MJ3e4APeM7cDBtrFg/view?.

[136] Casey S. Greene et al. “Big data bioinformatics”. In: *Journal of cellular physiology* 229 (12 2014), pp. 1896–1900. ISSN: 1097-4652. DOI: 10.1002/JCP.24662. URL: https://pubmed.ncbi.nlm.nih.gov/24799088/.

[137] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. “Deep Reinforcement Learning: An Overview BT - Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016”. In: ed. by Yaxin Bi, Supriya Kapoor, and Rahul Bhatia. Springer International Publishing, 2018, pp. 426–440. ISBN: 978-3-319-56991-8.

[138] Brian Kulis et al. “Semi-supervised graph clustering: a kernel approach”. In: *Machine Learning* 74 (1 2009), pp. 1–22. ISSN: 1573-0565. DOI: 10.1007/s10994-008-5084-4. URL: https://doi.org/10.1007/s10994-008-5084-4.

[139] S Baldi et al. “Multi-model unfalsified switching control of uncertain multivariable systems”. In: *International Journal of Adaptive Control and Signal Processing* 26 (8 Aug. 2012), pp. 705–722. ISSN: 0890-6327. DOI: 10.1002/acs.2310. URL: https://doi.org/10.1002/acs.2310.

[140] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. “Deep learning in bioinformatics”. In: *Briefings in Bioinformatics* 18 (5 Sept. 2017), pp. 851–869. ISSN: 1467-5463. DOI: 10.1093/BIB/BBW068. URL: https://academic.oup.com/bib/article/18/5/851/2562808.

[141] Ian; Goodfellow, Yoshua; Bengio, and Aaron. Courville. *Deep Learning*. MIT Press, 2016.

[142] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature 2015 521:7553* 521 (7553 May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539.

[143] Cha Zhang and Yunqian Ma. *Ensemble Machine Learning*. Springer US, 2012. DOI: 10.1007/978-1-4419-9326-7.

[144] A. J. Preto. “Artificial Intelligence”. In: (Dec. 2022). URL: https://www.canva.com/design/DAFSlcsxL-0/Nvtt7Rhb-TMvDv9NNfjm8g/view?.

[145] Steven A. Hicks et al. “On evaluation metrics for medical applications of artificial intelligence”. In: *Scientific reports* 12 (1 Dec. 2022). ISSN: 2045-2322. DOI: 10.1038/S41598-022-09954-8. URL: https://pubmed.ncbi.nlm.nih.gov/35395867/.

[146] Ramesh A. Gopinath and C. Sidney Burrus. “On Upsampling, Downsampling, and Rational Sampling Rate Filter Banks”. In: *IEEE Transactions on Signal Processing* 42 (4 1994), pp. 812–824. ISSN: 19410476. DOI: 10.1109/78.285645.

[147] Pankaj Mehta et al. “A high-bias, low-variance introduction to Machine Learning for physicists”. In: *Physics reports* 810 (May 2019), p. 1. ISSN: 03701573. DOI: 10.1016/J.PHYSREP.2019.03.001. URL: /pmc/articles/PMC6688775/%20/pmc/articles/PMC6688775/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6688775/.

[148] Jerzy Krawczuk and Tomasz Łukaszuk. “The feature selection bias problem in relation to high-dimensional gene data”. In: *Artificial intelligence in medicine* 66 (Jan. 2016), pp. 63–71. ISSN: 1873-2860. DOI: 10.1016/J.ARTMED.2015.11.001. URL: https://pubmed.ncbi.nlm.nih.gov/26674595/.

[149] A. Rogier T. Donders et al. “Review: a gentle introduction to imputation of missing values”. In: *Journal of clinical epidemiology* 59 (10 Oct. 2006), pp. 1087–1091. ISSN: 0895-4356. DOI: 10.1016/J.JCLINEPI.2006.01.014. URL: https://pubmed.ncbi.nlm.nih.gov/16980149/.

[150] Michael W. Browne. “Cross-validation methods”. In: *Journal of Mathematical Psychology* 44 (1 Mar. 2000), pp. 108–132. ISSN: 00222496. DOI: 10.1006/JMPS.1999.1279.

[151] M. Schumacher, N. Holländer, and W. Sauerbrei. “Resampling and cross-validation techniques: a tool to reduce bias caused by model building?” In: *Statistics in medicine* 16 (24 1997), pp. 2813–2827.

[152] Jyothi Subramanian and Richard Simon. “Overfitting in prediction models - is it a problem only in high dimensions?” In: *Contemporary clinical trials* 36 (2 Nov. 2013), pp. 636–641. ISSN: 1559-2030. DOI: 10.1016/J.CCT.2013.06.011. URL: https://pubmed.ncbi.nlm.nih.gov/23811117/.

[153] Sunmee Kim and Heungsun Hwang. “Evaluation of Prediction-Oriented Model Selection Metrics for Extended Redundancy Analysis”. In: *Frontiers in psychology* 13 (Apr. 2022). ISSN: 1664-1078. DOI: 10.3389/FPSYG.2022.821897.
URL: https://pubmed.ncbi.nlm.nih.gov/35478763/.

[154] Renchu Guan et al. “Feature space learning model”. In: *Journal of ambient intelligence and humanized computing* 10 (5 May 2019), pp. 2029–2040. ISSN: 1868-5137.
DOI: 10.1007/S12652-018-0805-4.
URL: https://pubmed.ncbi.nlm.nih.gov/31068980/.

[155] Francesca Grisoni et al.
“Molecular Descriptors for Structure-Activity Applications: A Hands-On Approach.”
In: *Methods in molecular biology (Clifton, N.J.)* 1800 (2018), pp. 3–53.
ISSN: 1940-6029 (Electronic). DOI: 10.1007/978-1-4939-7899-1_1.

[156] Álmos Orosz, Károly Héberger, and Anita Rácz. “Comparison of Descriptor- and Fingerprint Sets in Machine Learning Models for ADME-Tox Targets”.
In: *Frontiers in Chemistry* 10 (June 2022). ISSN: 22962646.
DOI: 10.3389/FCHEM.2022.852893/FULL.
URL: /pmc/articles/PMC9214226/%20/pmc/articles/PMC9214226/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9214226/.

[157] Balakumar Chandrasekaran et al.
Computer-Aided Prediction of Pharmacokinetic (ADMET) Properties. Vol. 2.
Elsevier Inc., 2018, pp. 731–755. ISBN: 9780128144220.
DOI: 10.1016/B978-0-12-814421-3.00021-X.
URL: http://dx.doi.org/10.1016/B978-0-12-814421-3.00021-X.

[158] Adrià Cereto-Massagué et al.
“Molecular fingerprint similarity search in virtual screening.”
In: *Methods (San Diego, Calif.)* 71 (Jan. 2015), pp. 58–63. ISSN: 1095-9130 (Electronic).
DOI: 10.1016/j.ymeth.2014.08.005.

[159] Limeng Pu et al. “DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network”. In: *PLoS Computational Biology* 15 (2 Feb. 2019).
ISSN: 15537358. DOI: 10.1371/JOURNAL.PCBI.1006718.
URL: /pmc/articles/PMC6375647/%20/pmc/articles/PMC6375647/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6375647/.

[160] Dong-Sheng Cao et al. “PyDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies.”
In: *Journal of chemical information and modeling* 53 (11 Nov. 2013), pp. 3086–3096.
ISSN: 1549-960X (Electronic). DOI: 10.1021/ci400127q.

[161] Mingjian Jiang et al.
“Sequence-based drug-target affinity prediction using weighted graph neural networks”.
In: *BMC Genomics* 23 (1 Dec. 2022), pp. 1–17. ISSN: 14712164.
DOI: 10.1186/S12864-022-08648-9/FIGURES/15. URL: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-022-08648-9.

[162] Eldon L Ulrich et al. "BioMagResBank."
In: *Nucleic acids research* 36 (Database issue Jan. 2008), pp. D402–8.
ISSN: 1362-4962 (Electronic). DOI: 10.1093/nar/gkm957.

[163] Martin Krallinger, Maria Padron, and Alfonso Valencia. "A sentence sliding window approach to extract protein annotations from biomedical articles."
In: *BMC bioinformatics* 6 Suppl 1 (Suppl 1 2005), S19. ISSN: 1471-2105 (Electronic).
DOI: 10.1186/1471-2105-6-S1-S19.

[164] Zhen Chen et al. "iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences."
In: *Bioinformatics (Oxford, England)* 34 (14 July 2018), pp. 2499–2502.
ISSN: 1367-4811 (Electronic). DOI: 10.1093/bioinformatics/bty140.

[165] Zia-Ur Rehman et al. "Predicting G-protein-coupled receptors families using different physiochemical properties and pseudo amino acid composition."
In: *Methods in enzymology* 522 (2013), pp. 61–79. ISSN: 1557-7988 (Electronic).
DOI: 10.1016/B978-0-12-407865-9.00004-2.

[166] Nan Xiao et al. "protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences."
In: *Bioinformatics (Oxford, England)* 31 (11 June 2015), pp. 1857–1859.
ISSN: 1367-4811 (Electronic). DOI: 10.1093/bioinformatics/btv042.

[167] Xuan Xiao et al. "iGPCR-Drug: A Web Server for Predicting Interaction between GPCRs and Drugs in Cellular Networking". In: *PLoS ONE* 8 (8 2013). ISSN: 19326203.
DOI: 10.1371/journal.pone.0072234.

[168] Bart L. Staker, Garry W. Buchko, and Peter J. Myler. "Recent contributions of Structure-Based Drug Design to the development of antibacterial compounds".
In: *Current opinion in microbiology* 27 (Oct. 2015), p. 133. ISSN: 18790364.
DOI: 10.1016/J.MIB.2015.09.003.
URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4659754/.

[169] José Almeida et al. "Co-evolution importance on binding Hot-Spot prediction methods".
In: (Jan. 2017), p. 3889. DOI: 10.3390/MOL2NET-02-03889.

[170] Debora S. Marks et al.
"Protein 3D structure computed from evolutionary sequence variation".
In: *PloS one* 6 (12 Dec. 2011). ISSN: 1932-6203.
DOI: 10.1371/JOURNAL.PONE.0028766.
URL: https://pubmed.ncbi.nlm.nih.gov/22163331/.

[171] Fabian Sievers and Desmond G Higgins.
"Clustal Omega, accurate alignment of very large numbers of sequences."
In: *Methods in molecular biology (Clifton, N.J.)* 1079 (2014), pp. 105–116.
ISSN: 1940-6029 (Electronic). DOI: 10.1007/978-1-62703-646-7_6.

[172] "Database resources of the National Center for Biotechnology Information."
In: *Nucleic acids research* 46 (D1 Jan. 2018), pp. D8–D13. ISSN: 1362-4962 (Electronic).
DOI: 10.1093/nar/gkx1095.

[173] Ben Hu et al. “Three-Dimensional Biologically Relevant Spectrum (BRS-3D): Shape Similarity Profile Based on PDB Ligands as Molecular Descriptors.” In: *Molecules (Basel, Switzerland)* 21 (11 Nov. 2016). ISSN: 1420-3049 (Electronic). DOI: 10.3390/molecules21111554.

[174] Yang Li et al. “Drug-target interaction prediction based on drug fingerprint information and protein sequence”. In: *Molecules* 24 (16 2019). ISSN: 14203049. DOI: 10.3390/molecules24162999.

[175] Han Shi et al. “Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure”. In: *Genomics* 111 (6 2019), pp. 1839–1852. ISSN: 10898646. DOI: 10.1016/j.ygeno.2018.12.007. URL: https://doi.org/10.1016/j.ygeno.2018.12.007.

[176] Raúl Cruz-Barbosa, Erik-German Ramos-Pérez, and Jesús Giraldo. “Representation Learning for Class C G Protein-Coupled Receptors Classification.” In: *Molecules (Basel, Switzerland)* 23 (3 Mar. 2018). ISSN: 1420-3049 (Electronic). DOI: 10.3390/molecules23030690.

[177] Shweta Redkar et al. “A Machine Learning Approach for Drug-target Interaction Prediction using Wrapper Feature Selection and Class Balancing”. In: *Molecular Informatics* 39 (5 2020). ISSN: 18681751. DOI: 10.1002/minf.201900062.

[178] Jason B Cross. “Methods for Virtual Screening of GPCR Targets: Approaches and Challenges.” In: *Methods in molecular biology (Clifton, N.J.)* 1705 (2018), pp. 233–264. ISSN: 1940-6029 (Electronic). DOI: 10.1007/978-1-4939-7465-8_11.

[179] Stanisław Jastrzębski et al. “Three-dimensional descriptors for aminergic GPCRs: dependence on docking conformation and crystal structure.” In: *Molecular diversity* 23 (3 Aug. 2019), pp. 603–613. ISSN: 1573-501X (Electronic). DOI: 10.1007/s11030-018-9894-4.

[180] Hae-Seok Eo et al. “A machine learning based method for the prediction of G protein-coupled receptor-binding PDZ domain proteins.” In: *Molecules and cells* 27 (6 June 2009), pp. 629–634. ISSN: 0219-1032 (Electronic). DOI: 10.1007/s10059-009-0091-2.

[181] Yao Zhang et al. “Experimental and computational evaluation of forces directing the association of transmembrane helices”. In: *Journal of the American Chemical Society* 131 (32 Aug. 2009), pp. 11341–11343. ISSN: 00027863. DOI: 10.1021/JA904625B.

[182] P. Chanphai, L. Bekale, and H. A. Tajmir-Riahi. “Effect of hydrophobicity on protein-protein interactions”. In: *European Polymer Journal* 67 (June 2015), pp. 224–231. ISSN: 00143057. DOI: 10.1016/J.EURPOLYMJ.2015.03.069.

[183] Mireia Rosell and Juan Fernández-Recio. "Hot-spot analysis for drug discovery targeting protein-protein interactions". In: *Expert Opinion on Drug Discovery* 13 (4 Apr. 2018), pp. 327–338. ISSN: 1746045X. DOI: 10.1080/17460441.2018.1430763.

[184] D. R. Caffrey. In: *Protein Science* 13 (1 Jan. 2004), pp. 190–202. ISSN: 0961-8368. DOI: 10.1110/PS.03323604.

[185] John A. Capra and Mona Singh. "Predicting functionally important residues from sequence conservation". In: *Bioinformatics* 23 (15 Aug. 2007), pp. 1875–1882. ISSN: 13674803. DOI: 10.1093/BIOINFORMATICS/BTM270.

[186] Martin B. Ulmschneider and Mark S.P. Sansom. "Amino acid distributions in integral membrane protein structures". In: *Biochimica et Biophysica Acta - Biomembranes* 1512 (1 May 2001), pp. 1–14. ISSN: 00052736. DOI: 10.1016/S0005-2736(01)00299-1.

[187] Qiangfeng Cliff Zhang et al. "Protein interface conservation across structure space". In: *Proceedings of the National Academy of Sciences of the United States of America* 107 (24 June 2010), pp. 10896–10901. ISSN: 00278424. DOI: 10.1073/PNAS.1005894107.

[188] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. "Supervised Learning BT - Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval". In: (2008). Ed. by Matthieu Cord and Pádraig Cunningham, pp. 21–49. DOI: 10.1007/978-3-540-75171-7_2. URL: https://doi.org/10.1007/978-3-540-75171-7_2.

[189] Burak Koçak. "Key concepts, common pitfalls, and best practices in artificial intelligence and machine learning: focus on radiomics". In: *Diagnostic and interventional radiology (Ankara, Turkey)* 28 (5 Sept. 2022), pp. 450–462. ISSN: 1305-3612. DOI: 10.5152/DIR.2022.211297. URL: https://pubmed.ncbi.nlm.nih.gov/36218149/.

[190] Shahadat Uddin et al. "Comparing different supervised machine learning algorithms for disease prediction". In: *BMC Medical Informatics and Decision Making* 19 (1 Dec. 2019), pp. 1–16. ISSN: 14726947. DOI: 10.1186/S12911-019-1004-8/FIGURES/12. URL: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-1004-8.

[191] N. V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002).

[192] Louise Francis. "Unsupervised Learning". In: *Predictive Modeling Applications in Actuarial Science: Volume 1: Predictive Modeling Techniques* 1 (2014). Ed. by Edward W Frees, Glenn Meyers, and Richard A Derrig, pp. 280–312. DOI: DOI:10.1017/CBO9781139342674.012. URL: https://www.cambridge.org/core/books/predictive-modeling-

applications-in-actuarial-science/unsupervised-learning/E12B943F0F44064AEF0120F775C0EF8E.

[193] J B Tenenbaum, V de Silva, and J C Langford. "A global geometric framework for nonlinear dimensionality reduction." In: *Science (New York, N.Y.)* 290 (5500 Dec. 2000), pp. 2319–2323. ISSN: 0036-8075 (Print). DOI: 10.1126/science.290.5500.2319.

[194] Yasi Wang, Hongxun Yao, and Sicheng Zhao. "Auto-encoder based dimensionality reduction". In: *Neurocomputing* 184 (2016), pp. 232–242. ISSN: 18728286. DOI: 10.1016/J.NEUCOM.2015.08.104.

[195] Karimollah Hajian-Tilaki. "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation". In: *Caspian Journal of Internal Medicine* 4 (2 2013), p. 627. ISSN: 20086164. URL: /pmc/articles/PMC3755824/%20/pmc/articles/PMC3755824/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3755824/.

[196] Carl Kingsford and Steven L. Salzberg. "What are decision trees?" In: *Nature biotechnology* 26 (9 Sept. 2008), p. 1011. ISSN: 10870156. DOI: 10.1038/NBT0908-1011. URL: /pmc/articles/PMC2701298/%20/pmc/articles/PMC2701298/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2701298/.

[197] A. J. Preto. "Decision Tree". In: (Dec. 2022). URL: https://www.canva.com/design/DAFT-0Bf_Sw/SA8WhDrcFwnxUb6kB3USsw/view.

[198] Shaherin Basith et al. "iGHBP: Computational identification of growth hormone binding proteins from sequences using extremely randomised tree". In: *Computational and Structural Biotechnology Journal* 16 (Jan. 2018), pp. 412–420. ISSN: 20010370. DOI: 10.1016/J.CSBJ.2018.10.007.

[199] Balachandran Manavalan et al. "AtbPpred: A Robust Sequence-Based Prediction of Anti-Tubercular Peptides Using Extremely Randomized Trees". In: *Computational and Structural Biotechnology Journal* 17 (Jan. 2019), pp. 972–981. ISSN: 20010370. DOI: 10.1016/J.CSBJ.2019.06.024.

[200] Tianqi Chen and Carlos Guestrin. "XGBoost: A scalable tree boosting system". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-August-2016 (Aug. 2016), pp. 785–794. DOI: 10.1145/2939672.2939785.

[201] Lisa Bartoli et al. "CCHMM-PROF: a HMM-based coiled-coil predictor with evolutionary information." In: *Bioinformatics (Oxford, England)* 25 (21 Nov. 2009), pp. 2757–2763. ISSN: 1367-4811 (Electronic). DOI: 10.1093/bioinformatics/btp539.

[202] Daniel Jurafsky and James H Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st. Prentice Hall PTR, 2000. ISBN: 0130950696.

[203] William S. Noble. “What is a support vector machine?” In: *Nature biotechnology* 24 (12 Dec. 2006), pp. 1565–1567. ISSN: 1087-0156. DOI: 10.1038/NBT1206-1565. URL: https://pubmed.ncbi.nlm.nih.gov/17160063/.

[204] E Alpaydin. “Multilayer Perceptrons”. In: *Introduction to Machine Learning* (2014), pp. 267–316. URL: https://ieeexplore.ieee.org/document/6917150.

[205] Jürgen Schmidhuber. “Deep learning in neural networks: an overview.” In: *Neural networks : the official journal of the International Neural Network Society* 61 (Jan. 2015), pp. 85–117. ISSN: 1879-2782 (Electronic). DOI: 10.1016/j.neunet.2014.09.003.

[206] Amirhossein Tavanaei et al. “Deep learning in spiking neural networks.” In: *Neural networks : the official journal of the International Neural Network Society* 111 (Mar. 2019), pp. 47–63. ISSN: 1879-2782 (Electronic). DOI: 10.1016/j.neunet.2018.12.002.

[207] Alexios Koutsoukas et al. “Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data”. In: *Journal of Cheminformatics* 9 (1 2017), p. 42. ISSN: 1758-2946. DOI: 10.1186/s13321-017-0226-y. URL: https://doi.org/10.1186/s13321-017-0226-y.

[208] Man Li, Cheng Ling, and Jingyang Gao. “An efficient CNN-based classification on G-protein Coupled Receptors using TF-IDF and N-gram”. In: *2017 IEEE Symposium on Computers and Communications (ISCC)* (2017), pp. 924–931. DOI: 10.1109/ISCC.2017.8024644.

[209] Cheng-Yuan Liou et al. “Autoencoder for words”. In: *Neurocomputing* 139 (2014), pp. 84–96. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2013.09.055. URL: http://www.sciencedirect.com/science/article/pii/S0925231214003658.

[210] G. E. Hinton and R. R. Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *Science* 313 (5786 July 2006), pp. 504–507. ISSN: 00368075. DOI: 10.1126/SCIENCE.1127647.

[211] Brad Boehmke and Greenwell Brandon M. *Hands-On Machine Learning with R*. 1st ed. Chapman and Hall/CRC, 2019, p. 488. ISBN: 1138495689.

[212] Zeren Shui and George Karypis. “Heterogeneous Molecular Graph Neural Networks for Predicting Molecule Properties”. In: *Proceedings - IEEE International Conference on Data Mining, ICDM* 2020-November (Sept. 2020), pp. 492–500. ISSN: 15504786. DOI: 10.48550/arxiv.2009.12710. URL: https://arxiv.org/abs/2009.12710v1.

[213] Jie Zhou et al. “Graph Neural Networks: A Review of Methods and Applications”. In: (2019).

[214] Franco Scarselli et al. “The graph neural network model.” In: *IEEE transactions on neural networks* 20 (1 Jan. 2009), pp. 61–80. ISSN: 1941-0093 (Electronic). DOI: 10.1109/TNN.2008.2005605.

[215] Giuseppe Zagotto and Marco Bortoli. “Drug Design: Where We Are and Future Prospects”. In: *Molecules* 26 (22 Nov. 2021). ISSN: 14203049. DOI: 10.3390/MOLECULES26227061. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8622624/.

[216] Márton Vass et al. “Molecular interaction fingerprint approaches for GPCR drug discovery”. In: *Current Opinion in Pharmacology* 30 (2016), pp. 59–68. ISSN: 14714973. DOI: 10.1016/j.coph.2016.07.007.

[217] Daniel J. Smit, Klaus Pantel, and Manfred Jücker. “Circulating tumor cells as a promising target for individualized drug susceptibility tests in cancer therapy”. In: *Biochemical pharmacology* 188 (June 2021). ISSN: 1873-2968. DOI: 10.1016/J.BCP.2021.114589. URL: https://pubmed.ncbi.nlm.nih.gov/33932470/.

[218] Vijay Kumar and Nitin Dogra. “A Comprehensive Review on Deep Synergistic Drug Prediction Techniques for Cancer”. In: *Archives of Computational Methods in Engineering* 29 (3 May 2022), pp. 1443–1461. ISSN: 18861784. DOI: 10.1007/S11831-021-09617-3.

[219] Zhan Deng, Claudio Chuaqui, and Juswinder Singh. “Structural interaction fingerprint (SIFt): a novel method for analyzing three-dimensional protein-ligand binding interactions.” In: *Journal of medicinal chemistry* 47 (2 Jan. 2004), pp. 337–344. ISSN: 0022-2623 (Print). DOI: 10.1021/jm030331x.

[220] C Da and D Kireev. “Structural protein-ligand interaction fingerprints (SPLIF) for structure-based virtual screening: method and benchmark study.” In: *Journal of chemical information and modeling* 54 (9 Sept. 2014), pp. 2555–2561. ISSN: 1549-960X (Electronic). DOI: 10.1021/ci500319f.

[221] Andrew C Kruse et al. “Activation and allosteric modulation of a muscarinic acetylcholine receptor.” In: *Nature* 504 (7478 Dec. 2013), pp. 101–106. ISSN: 1476-4687 (Electronic). DOI: 10.1038/nature12735.

[222] Antonio Lavecchia. “Machine-learning approaches in drug discovery: methods and applications.” In: *Drug discovery today* 20 (3 Mar. 2015), pp. 318–331. ISSN: 1878-5832 (Electronic). DOI: 10.1016/j.drudis.2014.10.012.

[223] Vladimir Chupakhin et al. “Predicting ligand binding modes from neural networks trained on protein-ligand interaction fingerprints.” In: *Journal of chemical information and modeling* 53 (4 Apr. 2013), pp. 763–772. ISSN: 1549-960X (Electronic). DOI: 10.1021/ci300200r.

[224] Jamel Meslamani et al. “Computational profiling of bioactive compounds using a target-dependent composite workflow.” In: *Journal of chemical information and modeling* 53 (9 Sept. 2013), pp. 2322–2333. ISSN: 1549-960X (Electronic). DOI: 10.1021/ci400303n.

[225] Beatriz Bueschbell et al. “A Complete Assessment of Dopamine Receptor- Ligand Interactions through Computational Methods”. In: *Molecules 2019, Vol. 24, Page 1196* 24 (7 Mar. 2019), p. 1196. ISSN: 1420-3049. DOI: 10.3390/MOLECULES24071196. URL: https://www.mdpi.com/1420-3049/24/7/1196/htm%20https://www.mdpi.com/1420-3049/24/7/1196.

[226] Akira Shiraishi et al. “Chemical genomics approach for GPCR-ligand interaction prediction and extraction of ligand binding determinants”. In: *Journal of Chemical Information and Modeling* 53 (6 2013), pp. 1253–1262. ISSN: 15499596. DOI: 10.1021/ci300515z.

[227] Chiranjib Chakraborty et al. “Micro-Environmental Signature of The Interactions between Druggable Target Protein, Dipeptidyl Peptidase-IV, and Anti-Diabetic Drugs.” In: *Cell journal* 19 (1 2017), pp. 65–83. ISSN: 2228-5806 (Print). DOI: 10.22074/cellj.2016.4865.

[228] C J Van Oss, R J Good, and M K Chaudhury. “The role of van der Waals forces and hydrogen bonds in “hydrophobic interactions” between biopolymers and low energy surfaces”. In: *Journal of Colloid and Interface Science* 111 (2 1986), pp. 378–390. ISSN: 0021-9797. DOI: https://doi.org/10.1016/0021-9797(86)90041-X. URL: http://www.sciencedirect.com/science/article/pii/002197978690041X.

[229] Thomas Klabunde and Gerhard Hessler. “Drug design strategies for targeting G-protein-coupled receptors.” In: *Chembiochem : a European journal of chemical biology* 3 (10 Oct. 2002), pp. 928–944. ISSN: 1439-4227 (Print). DOI: 10.1002/1439-7633(20021004)3:10<928::AID-CBIC928>3.0.CO;2-5.

[230] Fan Rong Meng et al. “Prediction of drug–target interaction networks from the integration of protein sequences and drug chemical structures”. In: *Molecules* 22 (7 2017). ISSN: 14203049. DOI: 10.3390/molecules22071119.

[231] Ladislav Peska, Krisztian Buza, and Júlia Koller. “Drug-target interaction prediction: A Bayesian ranking approach”. In: *Computer Methods and Programs in Biomedicine* 152 (2017), pp. 15–21. ISSN: 18727565. DOI: 10.1016/j.cmpb.2017.09.003.

[232] Cong Shen et al. “An ameliorated prediction of drug–target interactions based on multi-scale discretewavelet transform and network features”. In: *International Journal of Molecular Sciences* 18 (8 2017). ISSN: 14220067. DOI: 10.3390/ijms18081781.

[233] Jian Yu Shi et al. “Predicting combinative drug pairs via multiple classifier system with positive samples only”. In: *Computer Methods and Programs in Biomedicine* 168 (Jan. 2019), pp. 1–10. ISSN: 0169-2607. DOI: 10.1016/J.CMPB.2018.11.002.

[234] Xuan Xiao et al. “iDrug-Target: predicting the interactions between drug compounds and target proteins in cellular networking via benchmark dataset optimization approach.” In: *Journal of biomolecular structure & dynamics* 33 (10 2015), pp. 2221–2233. ISSN: 1538-0254 (Electronic). DOI: 10.1080/07391102.2014.998710.

[235] Xinke Zhan et al. “Ensemble Learning Prediction of Drug-Target Interactions Using GIST Descriptor Extracted from PSSM-Based Evolutionary Information.” In: *BioMed research international* 2020 (2020), p. 4516250. ISSN: 2314-6141 (Electronic). DOI: 10.1155/2020/4516250.

[236] Ankur Gautam et al. “In silico approaches for designing highly effective cell penetrating peptides”. In: *Journal of translational medicine* 11 (1 Mar. 2013). ISSN: 1479-5876. DOI: 10.1186/1479-5876-11-74. URL: https://pubmed.ncbi.nlm.nih.gov/23517638/.

[237] D T Jones. “Protein secondary structure prediction based on position-specific scoring matrices.” In: *Journal of molecular biology* 292 (2 Sept. 1999), pp. 195–202. ISSN: 0022-2836 (Print). DOI: 10.1006/jmbi.1999.3091.

[238] Alok Sharma et al. “A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition.” In: *Journal of theoretical biology* 320 (Mar. 2013), pp. 41–46. ISSN: 1095-8541 (Electronic). DOI: 10.1016/j.jtbi.2012.12.008.

[239] Ville Ojansivu and Janne Heikkilä. “Blur Insensitive Texture Classification Using Local Phase Quantization”. In: *Image and Signal Processing* (2008). Ed. by Abderrahim Elmoataz et al., pp. 236–243.

[240] Debashis Ghosh and Arul M Chinnaiyan. “Classification and Selection of Biomarkers in Genomic Data Using LASSO”. In: *Journal of Biomedicine and Biotechnology* 2005 (2005), p. 427208. ISSN: 2314-6133. DOI: 10.1155/JBB.2005.147. URL: https://doi.org/10.1155/JBB.2005.147.

[241] Hao Wang, Chuyao Liu, and Lei Deng. “Enhanced Prediction of Hot Spots at Protein-Protein Interfaces Using Extreme Gradient Boosting”. In: *Scientific Reports 2018 8:1* 8 (1 Sept. 2018), pp. 1–13. ISSN: 2045-2322. DOI: 10.1038/s41598-018-32511-1. URL: https://www.nature.com/articles/s41598-018-32511-1.

[242] Sebastian Raschka et al.
"Automated inference of chemical discriminants of biological activity".
In: *Methods in Molecular Biology* 1762 (2018), pp. 307–338. ISSN: 10643745.
DOI: `10.1007/978-1-4939-7756-7_16`.

[243] Ambrose Plante et al. "A machine learning approach for the discovery of ligand-specific functional mechanisms of GPCRs". In: *Molecules* 24 (11 2019). ISSN: 14203049.
DOI: `10.3390/molecules24112097`.

[244] Xue Lian Zhu et al.
"Classification of 5-HT1A receptor agonists and antagonists using GA-SVM method".
In: *Acta Pharmacologica Sinica* 32 (11 2011), pp. 1424–1430. ISSN: 16714083.
DOI: `10.1038/aps.2011.112`. URL: `http://dx.doi.org/10.1038/aps.2011.112`.

[245] Lun K Tsou et al. "Comparative study between deep learning and QSAR classifications for TNBC inhibitors and novel GPCR agonist discovery."
In: *Scientific reports* 10 (1 Oct. 2020), p. 16771. ISSN: 2045-2322 (Electronic).
DOI: `10.1038/s41598-020-73681-1`.

[246] Jiansheng Wu et al. "WDL-RF: Predicting bioactivities of ligand molecules acting with G protein-coupled receptors by combining weighted deep learning and random forest".
In: *Bioinformatics* 34 (13 2018), pp. 2271–2282. ISSN: 14602059.
DOI: `10.1093/bioinformatics/bty070`.

[247] Lauren T. May et al. "Allosteric Modulation of G Protein–Coupled Receptors".
In: *https://doi.org/10.1146/annurev.pharmtox.47.120505.105159* 47 (Jan. 2007), pp. 1–51.
ISSN: 03621642. DOI: `10.1146/ANNUREV.PHARMTOX.47.120505.105159`.
URL: `https://www.annualreviews.org/doi/abs/10.1146/annurev.pharmtox.47.120505.105159`.

[248] Jae Wan Jang et al. "Novel Scaffold Identification of mGlu1 Receptor Negative Allosteric Modulators Using a Hierarchical Virtual Screening Approach".
In: *Chemical Biology and Drug Design* 87 (2 2016), pp. 239–256. ISSN: 17470285.
DOI: `10.1111/cbdd.12654`.

[249] Xiaoqing Ru et al. "Exploration of the correlation between GPCRs and drugs based on a learning to rank algorithm".
In: *Computers in Biology and Medicine* 119 (Apr. 2020), p. 103660. ISSN: 18790534.
DOI: `10.1016/j.compbiomed.2020.103660`.
URL: `https://linkinghub.elsevier.com/retrieve/pii/S0010482520300548`.

[250] Sanychen Muk et al. "Machine Learning for Prioritization of Thermostabilizing Mutations for G-Protein Coupled Receptors."
In: *Biophysical journal* 117 (11 Dec. 2019), pp. 2228–2239.
ISSN: 1542-0086 (Electronic). DOI: `10.1016/j.bpj.2019.10.023`.

[251] Jing Yang et al. "High-accuracy prediction of transmembrane inter-helix contacts and application to GPCR 3D structure modeling."
In: *Bioinformatics (Oxford, England)* 29 (20 Oct. 2013), pp. 2579–2587.
ISSN: 1367-4811 (Electronic). DOI: `10.1093/bioinformatics/btt440`.

[252] Allan Lo et al.
"Predicting helix–helix interactions from residue contacts in membrane proteins".
In: *Bioinformatics* 25 (8 2009), pp. 996–1003. ISSN: 1367-4803.
DOI: 10.1093/bioinformatics/btp114.
URL: https://doi.org/10.1093/bioinformatics/btp114.

[253] Daniel W A Buchan and David T Jones.
"The PSIPRED Protein Analysis Workbench: 20 years on".
In: *Nucleic Acids Research* 47 (W1 2019), W402–W407. ISSN: 0305-1048.
DOI: 10.1093/nar/gkz297. URL: https://doi.org/10.1093/nar/gkz297.

[254] David T Jones. "Improving the accuracy of transmembrane protein topology prediction using evolutionary information."
In: *Bioinformatics (Oxford, England)* 23 (5 Mar. 2007), pp. 538–544.
ISSN: 1367-4811 (Electronic). DOI: 10.1093/bioinformatics/btl677.

[255] Shaun M Kandathil, Joe G Greener, and David T Jones.
"Prediction of interresidue contacts with DeepMetaPSICOV in CASP13".
In: *Proteins: Structure, Function, and Bioinformatics* 87 (12 2019), pp. 1092–1099.
DOI: 10.1002/prot.25779.
URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25779.

[256] Jian Zhang et al. "GPCR-I-TASSER: A Hybrid Approach to G Protein-Coupled Receptor Structure Modeling and the Application to the Human Genome".
In: *Structure* 23 (8 Aug. 2015), pp. 1538–1549. ISSN: 0969-2126.
DOI: 10.1016/j.str.2015.06.007.
URL: https://doi.org/10.1016/j.str.2015.06.007.

[257] Arjun Ray, Erik Lindahl, and Björn Wallner.
"Model quality assessment for membrane proteins."
In: *Bioinformatics (Oxford, England)* 26 (24 Dec. 2010), pp. 3067–3074.
ISSN: 1367-4811 (Electronic). DOI: 10.1093/bioinformatics/btq581.

[258] Lucy R Forrest, Christopher L Tang, and Barry Honig. "On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins."
In: *Biophysical journal* 91 (2 July 2006), pp. 508–517. ISSN: 0006-3495 (Print).
DOI: 10.1529/biophysj.106.082313.

[259] Björn Wallner and Arne Elofsson. "Identification of correct regions in protein models using structural, alignment, and consensus information."
In: *Protein science : a publication of the Protein Society* 15 (4 Apr. 2006), pp. 900–913.
ISSN: 0961-8368 (Print). DOI: 10.1110/ps.051799606.

[260] B. R. Brooks et al. "CHARMM: The biomolecular simulation program".
In: *Journal of Computational Chemistry* 30 (10 July 2009), pp. 1545–1614.
ISSN: 1096987X. DOI: 10.1002/JCC.21287.

[261] Scott J Weiner et al.
"A new force field for molecular mechanical simulation of nucleic acids and proteins".
In: *Journal of the American Chemical Society* 106 (3 Feb. 1984), pp. 765–784.
ISSN: 0002-7863. DOI: 10.1021/ja00315a051.
URL: https://doi.org/10.1021/ja00315a051.

[262] Roland Lüthy, James U Bowie, and David Eisenberg.
"Assessment of protein models with three-dimensional profiles".
In: *Nature* 356 (6364 1992), pp. 83–85. ISSN: 1476-4687. DOI: 10.1038/356083a0.
URL: https://doi.org/10.1038/356083a0.

[263] R Samudrala and J Moult. "An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction."
In: *Journal of molecular biology* 275 (5 Feb. 1998), pp. 895–916.
ISSN: 0022-2836 (Print). DOI: 10.1006/jmbi.1997.1479.

[264] M J Sippl. "Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins."
In: *Journal of molecular biology* 213 (4 June 1990), pp. 859–883.
ISSN: 0022-2836 (Print). DOI: 10.1016/s0022-2836(05)80269-4.

[265] C Zhang and S H Kim. "Environment-dependent residue contact energies for proteins."
In: *Proceedings of the National Academy of Sciences of the United States of America* 97 (6 Mar. 2000), pp. 2550–2555. ISSN: 0027-8424 (Print). DOI: 10.1073/pnas.040573597.

[266] Boris Fain, Yu Xia, and Michael Levitt.
"Design of an optimal Chebyshev-expanded discrimination function for globular proteins."
In: *Protein science : a publication of the Protein Society* 11 (8 Aug. 2002), pp. 2010–2021.
ISSN: 0961-8368 (Print). DOI: 10.1110/ps.0200702.

[267] F J Martinez, J I Couser, and B R Celli.
"Respiratory response to arm elevation in patients with chronic airflow obstruction."
In: *The American review of respiratory disease* 143 (3 1991), pp. 476–480.
ISSN: 0003-0805 (Print). DOI: 10.1164/ajrccm/143.3.476.

[268] Marcin Pawlowski et al.
"MetaMQAP: A meta-server for the quality assessment of protein models".
In: *BMC Bioinformatics* 9 (1 2008), p. 403. ISSN: 1471-2105.
DOI: 10.1186/1471-2105-9-403.
URL: https://doi.org/10.1186/1471-2105-9-403.

[269] Björn Wallner and Arne Elofsson. "Can correct protein models be identified?"
In: *Protein science : a publication of the Protein Society* 12 (5 May 2003), pp. 1073–1086.
ISSN: 0961-8368 (Print). DOI: 10.1110/ps.0236803.

[270] Óscar Díaz, James A R Dalton, and Jesús Giraldo.
"Artificial Intelligence: A Novel Approach for Drug Discovery."
In: *Trends in pharmacological sciences* 40 (8 Aug. 2019), pp. 550–551.
ISSN: 1873-3735 (Electronic). DOI: 10.1016/j.tips.2019.06.005.

[271] João Marcelo Lamim Ribeiro, Davide Provasi, and Marta Filizola. "A combination of machine learning and infrequent metadynamics to efficiently predict kinetic rates, transition states, and molecular determinants of drug dissociation from G protein-coupled receptors." In: *The Journal of chemical physics* 153 (12 Sept. 2020), p. 124105. ISSN: 1089-7690 (Electronic). DOI: 10.1063/5.0019100.

[272] Carlos A.V. Barreto et al. "Prediction and targeting of GPCR oligomer interfaces". In: *Progress in Molecular Biology and Translational Science* 169 (Jan. 2020), pp. 105–149. ISSN: 1877-1173. DOI: 10.1016/BS.PMBTS.2019.11.007.

[273] Ivone Gomes et al. "Disease-specific heteromerization of G-protein-coupled receptors that target drugs of abuse." In: *Progress in molecular biology and translational science* 117 (2013), pp. 207–265. ISSN: 1878-0814 (Electronic). DOI: 10.1016/B978-0-12-386931-9.00009-X.

[274] Achla Gupta et al. "Increased abundance of opioid receptor heteromers after chronic morphine administration." In: *Science signaling* 3 (131 July 2010), ra54. ISSN: 1937-9145 (Electronic). DOI: 10.1126/scisignal.2000807.

[275] Raphael Rozenfeld et al. "AT1R-CB□R heteromerization reveals a new mechanism for the pathogenic properties of angiotensin II." In: *The EMBO journal* 30 (12 May 2011), pp. 2350–2363. ISSN: 1460-2075 (Electronic). DOI: 10.1038/emboj.2011.139.

[276] Sonia Terrillon and Michel Bouvier. "Roles of G-protein-coupled receptor dimerization". In: *EMBO reports* 5 (1 2004), pp. 30–34. DOI: 10.1038/sj.embor.7400052. URL: https://www.embopress.org/doi/abs/10.1038/sj.embor.7400052.

[277] Wataru Nemoto et al. "GGIP: Structure and sequence-based GPCR-GPCR interaction pair predictor." In: *Proteins* 84 (9 Sept. 2016), pp. 1224–1233. ISSN: 1097-0134 (Electronic). DOI: 10.1002/prot.25071.

[278] S. J. de Vries, A. D. van Dijk, and A. M. Bonvin. "WHISCY: what information does surface conservation yield? Application to data-driven docking". In: *Proteins* 63 (3 2006), pp. 479–489.

[279] Yanay Ofran and Burkhard Rost. "ISIS: interaction sites identified from sequence". In: *Bioinformatics (Oxford, England)* 23 (2 2007). ISSN: 1367-4811. DOI: 10.1093/BIOINFORMATICS/BTL303. URL: https://pubmed.ncbi.nlm.nih.gov/17237081/.

[280] Rita Melo et al. "A machine learning approach for hot-spot detection at protein-protein interfaces". In: *International Journal of Molecular Sciences* 17 (8 Aug. 2016). ISSN: 14220067. DOI: 10.3390/IJMS17081215.

[281] Stefano Lise et al. “Predictions of hot spot residues at protein-protein interfaces using support vector machines”. In: *PloS one* 6 (2 2011). ISSN: 1932-6203. DOI: `10.1371/JOURNAL.PONE.0016774`. URL: `https://pubmed.ncbi.nlm.nih.gov/21386962/`.

[282] K. S. Thorn and A. A. Bogan. “ASEdb: A database of alanine mutations and their effects on the free energy of binding in protein interactions”. In: *Bioinformatics* 17 (3 2001), pp. 284–285. ISSN: 13674803. DOI: `10.1093/BIOINFORMATICS/17.3.284`.

[283] T. B. Fischer et al. “The binding inteference database (BID): A compilation of amino acid hot spots in protein interfaces”. In: *Bioinformatics* 19 (11 July 2003), pp. 1453–1454. ISSN: 13674803. DOI: `10.1093/BIOINFORMATICS/BTG163`.

[284] M. D.Shaji Kumar and M. Michael Gromiha. “PINT: Protein-protein Interactions Thermodynamic Database.” In: *Nucleic acids research* 34 (Database issue 2006). ISSN: 13624962. DOI: `10.1093/NAR/GKJ017`.

[285] Iain H. Moal and Juan Fernández-Recio. “SKEMPI: A Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models”. In: *Bioinformatics* 28 (20 Oct. 2012), pp. 2600–2607. ISSN: 13674803. DOI: `10.1093/BIOINFORMATICS/BTS489`.

[286] Justina Jankauskaite et al. “SKEMPI 2.0: An updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation”. In: *Bioinformatics* 35 (3 Feb. 2019), pp. 462–469. ISSN: 14602059. DOI: `10.1093/BIOINFORMATICS/BTY635`.

[287] Engin Cukuroglu et al. “Hot spots in protein-protein interfaces: Towards drug discovery”. In: *Progress in Biophysics and Molecular Biology* 116 (2-3 2014), pp. 165–173. ISSN: 00796107. DOI: `10.1016/J.PBIOMOLBIO.2014.06.003`.

[288] Shan Shan Hu et al. “Protein binding hot spots prediction from sequence only by a new ensemble learning method”. In: *Amino Acids* 49 (10 Oct. 2017), pp. 1773–1785. ISSN: 14382199. DOI: `10.1007/S00726-017-2474-6`.

[289] Quanya Liu et al. “Hot spot prediction in protein-protein interactions by an ensemble system”. In: *BMC Systems Biology* 12 (Dec. 2018). ISSN: 17520509. DOI: `10.1186/S12918-018-0665-8`.

[290] João M. Martins et al. “Solvent-accessible surface area: How well can be applied to hot-spot detection?” In: *Proteins: Structure, Function and Bioinformatics* 82 (3 Mar. 2014), pp. 479–490. ISSN: 10970134. DOI: `10.1002/PROT.24413`.

[291] Quang Thang Nguyen, Ronan Fablet, and Dominique Pastor. “Protein interaction hotspot identification using sequence-based frequency-derived features”. In: *IEEE Transactions on Biomedical Engineering* 60 (11 2013), pp. 2993–3002. ISSN: 00189294. DOI: `10.1109/TBME.2011.2161306`.

[292] Xiaolei Zhu and Julie C. Mitchell. “KFC2: A knowledge-based hot spot prediction method based on interface solvation, atomic density, and plasticity features”. In: *Proteins: Structure, Function and Bioinformatics* 79 (9 Sept. 2011), pp. 2671–2683. ISSN: 08873585. DOI: 10.1002/PROT.23094.

[293] Alex Bateman et al. “UniProt: The universal protein knowledgebase”. In: *Nucleic Acids Research* 45 (D1 Jan. 2017), pp. D158–D169. ISSN: 13624962. DOI: 10.1093/NAR/GKW1099.

[294] Lei Deng et al. “PredHS: a web server for predicting protein-protein interaction hot spots by using structural neighborhood properties”. In: *Nucleic acids research* 42 (Web Server issue July 2014). ISSN: 1362-4962. DOI: 10.1093/NAR/GKU437. URL: https://pubmed.ncbi.nlm.nih.gov/24852252/.

[295] Nícia Rosário-Ferreira, Alexandre M.J.J. Bonvin, and Irina S. Moreira. “Using machine-learning-driven approaches to boost hot-spot’s knowledge”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 12 (5 Sept. 2022), e1602. ISSN: 1759-0884. DOI: 10.1002/WCMS.1602. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/wcms.1602%20https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1602%20https://wires.onlinelibrary.wiley.com/doi/10.1002/wcms.1602.

[296] Neil Vasan, José Baselga, and David M. Hyman. “A view on drug resistance in cancer”. In: *Nature* 575 (7782 Nov. 2019), pp. 299–309. ISSN: 14764687. DOI: 10.1038/S41586-019-1730-1.

[297] Nilanjana Chatterjee and Trever G. Bivona. “Polytherapy and Targeted Cancer Drug Resistance”. In: *Trends in Cancer* 5 (3 Mar. 2019), pp. 170–182. ISSN: 24058033. DOI: 10.1016/J.TRECAN.2019.02.003.

[298] L. F. Piochi et al. “From single-omics to interactomics: How can ligand-induced perturbations modulate single-cell phenotypes?” In: *Advances in Protein Chemistry and Structural Biology* 131 (Jan. 2022), pp. 45–83. ISSN: 18761631. DOI: 10.1016/BS.APCSB.2022.05.006.

[299] Remzi Celebi et al. “In-silico Prediction of Synergistic Anti-Cancer Drug Combinations Using Multi-omics Data”. In: *Scientific Reports* 9 (1 Dec. 2019). ISSN: 20452322. DOI: 10.1038/S41598-019-45236-6.

[300] JD Janizek, S Celik, and S-I Lee. “Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine”. In: *bioRxiv* (2018). DOI: 10.1101/331769.

[301] Hongyang Li et al. “Network propagation predicts drug synergy in cancers”. In: *Cancer Research* 78 (18 Sept. 2018), pp. 5446–5457. ISSN: 15387445. DOI: 10.1158/0008-5472.CAN-18-0740.

[302] Alina Malyutina et al. “Drug combination sensitivity scoring facilitates the discovery of synergistic and efficacious drug combinations in cancer”. In: *PLoS Computational Biology* 15 (5 May 2019). ISSN: 15537358. DOI: 10.1371/JOURNAL.PCBI.1006752.

[303] Jiannan Yang et al. “GraphSynergy: a network-inspired deep learning model for anticancer drug combination prediction”. In: *Journal of the American Medical Informatics Association : JAMIA* 28 (11 Nov. 2021), pp. 2336–2345. ISSN: 1527-974X. DOI: 10.1093/JAMIA/OCAB162. URL: https://pubmed.ncbi.nlm.nih.gov/34472609/.

[304] Jinxian Wang et al. “DeepDDS: deep graph neural network with attention mechanism to predict synergistic drug combinations”. In: *Briefings in Bioinformatics* 23 (1 Jan. 2022). ISSN: 14774054. DOI: 10.1093/BIB/BBAB390. URL: https://academic.oup.com/bib/article/23/1/bbab390/6375262.

[305] Kristina Preuer et al. “DeepSynergy: predicting anti-cancer drug synergy with Deep Learning”. In: *Bioinformatics (Oxford, England)* 34 (9 May 2018), pp. 1538–1546. ISSN: 1367-4811. DOI: 10.1093/BIOINFORMATICS/BTX806. URL: https://pubmed.ncbi.nlm.nih.gov/29253077/.

[306] Tianyu Zhang et al. “Synergistic Drug Combination Prediction by Integrating Multiomics Data in Deep Learning Models”. In: *Methods in molecular biology (Clifton, N.J.)* 2194 (2021), pp. 223–238. ISSN: 1940-6029. DOI: 10.1007/978-1-0716-0849-4_12. URL: https://pubmed.ncbi.nlm.nih.gov/32926369/.

[307] Halil Ibrahim Kuru, Oznur Tastan, and A. Ercument Cicek. “MatchMaker: A Deep Learning Framework for Drug Synergy Prediction”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 19 (4 2022), pp. 2334–2344. ISSN: 1557-9964. DOI: 10.1109/TCBB.2021.3086702. URL: https://pubmed.ncbi.nlm.nih.gov/34086576/.

[308] Qiao Liu and Lei Xie. “TranSynergy: Mechanism-driven interpretable deep neural network for the synergistic prediction and pathway deconvolution of drug combinations”. In: *PLOS Computational Biology* 17 (2 Feb. 2021), e1008653. ISSN: 1553-7358. DOI: 10.1371/JOURNAL.PCBI.1008653. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008653.

[309] Fangfang Xia et al. “Predicting tumor cell line response to drug pairs with deep learning”. In: *BMC Bioinformatics* 19 (Dec. 2018). ISSN: 14712105. DOI: 10.1186/S12859-018-2509-3.

[310] CI BLISS. “THE TOXICITY OF POISONS APPLIED JOINTLY”. In: *Annals of Applied Biology* 26 (3 1939), pp. 585–615. ISSN: 17447348. DOI: 10.1111/J.1744-7348.1939.TB06990.X.

[311] Julie Foucquier and Mickael Guedj. "Analysis of drug combinations: current methodological landscape". In: *Pharmacology Research and Perspectives* 3 (3 June 2015). ISSN: 20521707. DOI: 10.1002/PRP2.149.

[312] TC Chou. "Drug combination studies and their synergy quantification using the chou-talalay method". In: *Cancer Research* 70 (2 Jan. 2010), pp. 440–446. ISSN: 00085472. DOI: 10.1158/0008-5472.CAN-09-1947.

[313] S. T. Loewe and H. Muischnek. "Über kombinationswirkungen". In: *Naunyn-Schmiedebergs Archiv für experimentelle Pathologie und Pharmakologie* 114 (5 1926), pp. 313–326.

[314] Bhagwan Yadav et al. "Searching for Drug Synergy in Complex Dose-Response Landscapes Using an Interaction Potency Model". In: *Computational and Structural Biotechnology Journal* 13 (2015), pp. 504–513. ISSN: 20010370. DOI: 10.1016/J.CSBJ.2015.09.001.

[315] Aleksandr Ianevski et al. "Prediction of drug combination effects with a minimal set of experiments". In: *Nature Machine Intelligence* 1 (12 Dec. 2019), pp. 568–577. DOI: 10.1038/S42256-019-0122-4.

[316] Bulat Zagidullin et al. "DrugComb: An integrative cancer drug combination data portal". In: *Nucleic Acids Research* 47 (W1 July 2019), W43–W51. ISSN: 13624962. DOI: 10.1093/NAR/GKZ337.

[317] Lei Chen et al. "Prediction of effective drug combinations by chemical interaction, protein interaction and target enrichment of KEGG pathways". In: *BioMed Research International* 2013 (2013). ISSN: 23146133. DOI: 10.1155/2013/723780.

[318] Yifan Sun et al. "A hadoop-based method to predict potential effective drug combination". In: *BioMed research international* 2014 (2014). ISSN: 2314-6141. DOI: 10.1155/2014/196858. URL: https://pubmed.ncbi.nlm.nih.gov/25147789/.

[319] Yanbin Liu et al. "DCDB: Drug combination database". In: *Bioinformatics* 26 (4 Feb. 2010), pp. 587–588. ISSN: 1367-4803. DOI: 10.1093/BIOINFORMATICS/BTP697. URL: https://academic.oup.com/bioinformatics/article/26/4/587/243716.

[320] Hui Huang et al. "Systematic prediction of drug combinations based on clinical side-effects". In: *Scientific Reports 2014 4:1* 4 (1 Nov. 2014), pp. 1–7. ISSN: 2045-2322. DOI: 10.1038/srep07160. URL: https://www.nature.com/articles/srep07160.

[321] Peng Li et al. "Large-scale exploration and analysis of drug combinations". In: *Bioinformatics* 31 (12 June 2015), pp. 2007–2016. ISSN: 1367-4803. DOI: 10.1093/BIOINFORMATICS/BTV080. URL: https://academic.oup.com/bioinformatics/article/31/12/2007/214330.

[322] Yi Sun et al. “Combining genomic and network characteristics for extended capability in predicting synergistic drugs for cancer”. In: *Nature Communications 2015 6:1* 6 (1 Sept. 2015), pp. 1–10. ISSN: 2041-1723. DOI: 10.1038/ncomms9481. URL: https://www.nature.com/articles/ncomms9481.

[323] Jan Wildenhain et al. “Prediction of Synergism from Chemical-Genetic Interactions by Machine Learning”. In: *Cell Systems* 1 (6 Dec. 2015), pp. 383–395. ISSN: 2405-4712. DOI: 10.1016/J.CELS.2015.12.003.

[324] Xing Chen et al. “NLLSS: Predicting Synergistic Drug Combinations Based on Semi-supervised Learning”. In: *PLOS Computational Biology* 12 (7 July 2016), e1004975. ISSN: 1553-7358. DOI: 10.1371/JOURNAL.PCBI.1004975. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004975.

[325] Kaitlyn M. Gayvert et al. “A Computational Approach for Identifying Synergistic Drug Combinations”. In: *PLOS Computational Biology* 13 (1 Jan. 2017), e1005308. ISSN: 1553-7358. DOI: 10.1371/JOURNAL.PCBI.1005308. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005308.

[326] Qian Xu et al. “PDC-SGB: Prediction of effective drug combinations using a stochastic gradient boosting algorithm”. In: *Journal of Theoretical Biology* 417 (Mar. 2017), pp. 1–7. ISSN: 0022-5193. DOI: 10.1016/J.JTBI.2017.01.019.

[327] Jian Yu Shi et al. “Predicting combinative drug pairs towards realistic screening via integrating heterogeneous features”. In: *BMC Bioinformatics* 18 (12 Oct. 2017), pp. 1–9. ISSN: 14712105. DOI: 10.1186/S12859-017-1818-2/TABLES/3. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1818-2.

[328] Guocai Chen et al. “Predict effective drug combination by deep belief network and ontology fingerprints”. In: *Journal of biomedical informatics* 85 (Sept. 2018), pp. 149–154. ISSN: 1532-0480. DOI: 10.1016/J.JBI.2018.07.024. URL: https://pubmed.ncbi.nlm.nih.gov/30081101/.

[329] Feixiong Cheng et al. “Network-based prediction of drug combinations”. In: *Nature Communications 2019 10:1* 10 (1 Mar. 2019), pp. 1–11. ISSN: 2041-1723. DOI: 10.1038/s41467-019-09186-x. URL: https://www.nature.com/articles/s41467-019-09186-x.

[330] Pavel Sidorov et al. “Predicting Synergism of Cancer Drug Combinations Using NCI-ALMANAC Data”. In: *Frontiers in Chemistry* 7 (July 2019), p. 509. ISSN: 22962646. DOI: 10.3389/FCHEM.2019.00509/BIBTEX.

[331] Chengzhi Zhang and Guiying Yan. "Synergistic drug combinations prediction by integrating pharmacological data". In: *Synthetic and Systems Biotechnology* 4 (1 Mar. 2019), pp. 67–72. ISSN: 2405-805X. DOI: `10.1016/J.SYNBIO.2018.10.002`.

[332] Peiran Jiang et al. "Deep graph embedding for prioritizing synergistic anticancer drug combinations". In: *Computational and Structural Biotechnology Journal* 18 (Jan. 2020), pp. 427–438. ISSN: 2001-0370. DOI: `10.1016/J.CSBJ.2020.02.006`.

[333] Heli Julkunen et al. "Leveraging multi-way interactions for systematic prediction of pre-clinical drug combination effects". In: *Nature Communications 2020 11:1* 11 (1 Dec. 2020), pp. 1–11. ISSN: 2041-1723. DOI: `10.1038/s41467-020-19950-z`. URL: `https://www.nature.com/articles/s41467-020-19950-z`.

[334] Binliang Wang et al. "ETV4 mediated lncRNA C2CD4D-AS1 overexpression contributes to the malignant phenotype of lung adenocarcinoma cells via miR-3681-3p/NEK2 axis". In: *Cell Cycle* 20 (24 2021), pp. 2607–2618. ISSN: 15514005. DOI: `10.1080/15384101.2021.2005273`.

Chapter 2: Objectives and thesis outline

2.1. Objectives

The main objective of drug design and development is to identify and develop drugs which target specific biological mechanisms or pathways involved in disease processes. By disrupting these mechanisms or pathways, the drugs can help alleviate symptoms of the disease and potentially cure it. This process involves using diverse techniques and approaches, including computer modelling and simulations, laboratory testing, and clinical trials. However, computational tools have a major impact on drug design and development since we can use them to screen large numbers of potential drug compounds and predict which ones are likely to be effective at treating a particular disease quickly and efficiently. This approach is often more cost-effective and time-efficient than performing experimental tests on each compound individually. They are also essential to predict how a potential drug compound is likely to interact with the target biological pathway or mechanism, allowing researchers to identify probable side effects or other potential issues before conducting expensive and time-consuming experimental tests.

On the latest years, with the most up-to-date research and technology, the end goal of attaining patient-specific healthcare has become both more feasible and inevitable. Personalized medicine is still a distant dream, however, step by step, it is getting closer. Thriving for its arrival, however, is a tricky endeavour. It is unreasonable to expect a single work to shift the process of mass-produced chemicals to streamlined personalized drugs. Nevertheless, it is possible to make steady small advances through joint scientific effort.

One of the possible drug design and development strategies could be to break each problem into smaller building blocks. It means looking separately at the drugs and the targets and building up from there to their interactions, their location and roles in the cells and organism and their biological significance. This approach can lead to the understanding of complex biological networks and the ability to modulate them through specific drugs.

To address the **target** it is necessary to characterize proteins on multiple levels. This will open the gates to create target feature spaces that are usable by ML approaches. Another goal was MP oligomer characterization, as they pose the most considerable challenge among the candidates while also being the most promising. In parallel, I was focused on the usability of the information, and as such, the generation of sequence-based features was privileged, due to the superior abundance of protein sequence data in comparison to structural data.

Target protein objectives

Aim: Characterize protein targets as thoroughly as possible to increase the understanding and generate ML-appropriate feature spaces on amino acid, single protein, oligomeric protein and, ultimately, MP levels:

- **Objective 1:** Address the feasibility of designing protein sequence-only features and hinging HS/NS prediction on these.
- **Objective 2:** Deploy and optimize an HS/NS ML predictor using sequence information only.
- **Objective 3:** Create a web-based easily accessible computational tool for HS/NS prediction available for users with any type of background.
- **Objective 4:** Gather, explore, and characterize MP dimers in a systematic approach.
- **Objective 5:** Create a database that allows easy access to curated data on MP dimers and inquiry of their main features.

Regarding the **drugs**, the approach must be slightly different. While proteins can build up to be some of the largest biological molecules, drugs tend to be small molecules that bind to these. However, the involvement of drugs does not come without hindrances. Their smaller size demands a more detailed characterization, usually at an atomic level.

Drug objectives

Aim: Characterise drugs, with a focus on small molecules, considering the biological meaning and ML-purposeful usage.

- **Objective 1:** Attain from the literature a broad knowledge of the current state of AI-driven research related to drugs and the most viable targets (with a high focus on GPCRs).
- **Objective 2:** Explore the available information for drug-related feature space generation.
- **Objective 3:** Create an easy-to-use and fast tool that can be used to generate meaningful and interpretable drug features while also providing bulk analysis insights.

The final section of this work will address the competitive landscape of drug combination synergy prediction on cancer cell lines. Cancer is one of the leading causes of death worldwide and the most difficult-to-treat disease (or set of diseases) that afflict humankind. Thus, any advance regarding its treatment is a gateway to further research and drug development since it helps scientists and physicians determine and find the most effective treatment options for cancer patients. Combination synergistic therapy has proven to be more effective in treating cancer than using a single drug alone, leading to improved patient outcomes and reducing the risk of resistance development. Since it can manifest itself in several ways and individuals, tackling it demands a wide understanding of the organs, cell tissues and molecules involved. This unfolds unto a multipronged problem, that requires multipronged solutions. Thus, drug combinations synergy prediction on cancer cell lines has become a quintessential cornerstone on drug design and development, as it is the crossroad for many health, research and technological paths. These are the aim and objectives of this work regarding this problem:

Synergy drug ccombination prediction in cancer objectives

Aim: Create a model that is able to accurately predict the synergy of different drug combinations in cancer cell lines.

- **Objective 1:** Address the definition and ambiguity of the definition of synergy.
- **Objective 2:** Identify available and viable data to use regarding the problem, both regarding the samples and the features – particularly on the scope of omics data, that still has a lot of promising venues to be explored.
- **Objective 3:** Develop and deploy a thorough protocol that leverages the available data and tools to arrive at the most optimized solutions for feature selection, data preprocessing, and are able to select the best approaches out of a battery of ML prediction models' testing and optimization.
- **Objective 4:** Create a platform that provides physicians and researchers with a tool that can help them quickly and accurately predict the potential effectiveness of previously unseen drug pairs for treating cancer, so that they can choose the best treatment options for their patients.

2.2. Thesis outline

The work conducted under the scope of the proposed objectives generated an output of several publications in peer-reviewed journals (n=8, 5 original research papers, 1 indexed database and 2 reviews in the form of book chapters). Under the scope of computational results output, it is also relevant to mention that 5 GitHub repositories, 3 websites and 1 freely distributed Python package are directly associated with the research conducted.

I **Protein understanding through HS prediction and MP features characterization.** As the most relevant drug targets, proteins are irrevocable when considering a work centred on drug design and development. Under this scope, three publications are most relevant for deepening the understanding of proteins and providing tools for subsequent work development.

  i **HS Detection with DL**

  - *Original Research Article 1:* **Preto, A.J.,** Matos-Filipe, P., de Almeida, J.G., Mourão, J., Moreira, I.S. (2021). Predicting Hot Spots Using a Deep Neural Network Approach. In: Cartwright, H. (eds) Artificial Neural Networks. *Methods in Molecular Biology*, 2190. Humana, New York, NY. `https://doi.org/10.1007/978-1-0716-0826-5_13`
  - *GitHub repository 1:* HS Detection with DL

  ii **SPOTONE**

  - *Original Research Article 2:* **Preto, A.J.**; Moreira, I.S. (2020). SPOTONE: Hot Spots on Protein Complexes with Extremely Randomized Trees via Sequence-Only Features. International Journal of Molecular Sciences, 21, 7281. `https://doi.org/10.3390/ijms21197281`
  - *GitHub repository 2:* SPOTONE
  - *Website 1:* `https://moreiralab.com/resources/spotone`

  iii **Membrane protein dimer characterization**

  - *Review in the form of book chapter 1:* **António J. Preto ***, Preto Matos-Filipe *, Panagiotis I. Koukos, Pedro Renault, Sérgio F. Sousa, Irina S. Moreira (2020). Structural Characterization of Membrane Protein Dimers. *Methods in Molecular Biology*. Chapter 21 (1958). `https://doi.org/10.1007/978-1-4939-9161-7_21`. *António J. Preto and Pedro Matos-Filipe contributed equally with all other contributors.

  iv **MensaDB**

  - *Original Research Article 3:* Matos-Filipe, P.*, **Preto, A. J.***, Koukos, P. I., Mourão, J., Bonvin, A. M. J. J., Moreira, I. S. (2021). MENSAdb: a thorough structural analysis of membrane protein dimers. *Database: the journal of biological databases and curation*, baab013. `https://doi.org/10.1093/database/baab013` *Co-first authors
  - *GitHub repository 3:* https://github.com/MoreiraLAB/mensadb-open

- *Website 2:* `http://www.moreiralab.com/resources/mensadb/`

II **Small molecule explainable representation as key to drug understanding**. This section focuses on ML explainability and answers it by providing a package that takes small molecules as input and categorises them taxonomically according to their chemical properties.

i **Drugs in AI-driven research**

- *Review in the form of book chapter 2:* **AJ Preto**, C Marques-Pereira, Salete J Baptista, B Bueschbell, Carlos AV Barreto, AT Gaspar, I Pinheiro, N Pereira, M Pires, D Ramalhão, D Silvério, N Rosário-Ferreira, R Melo, J Mourão, IS Moreira (2022). Targeting GPCRs Via Multi-Platforms Arrays and AI. Comprehensive Pharmacology, 2.08.
`https://doi.org/10.1016/B978-0-12-820472-6.00048-7`

ii **DrugTax**

- *Original Research Article 4:* **Preto, A.J.**, Correia, P.C., Moreira, I.S. (2022) DrugTax: package for drug taxonomy identification and explainable feature extraction. *Journal of Cheminformatics* 14, 73.
`https://doi.org/10.1186/s13321-022-00649-w`
- *GitHub repository 4:* DrugTax
- *Python Package 1:* `https://pypi.org/project/drugtax/`

III **Drug synergy prediction of cancer cell lines.** The final topic of this work condenses many previous concepts of feature representation, data pre-processing, ML, and model evaluation, by tackling the problem of drug combination prediction of cancer cell lines. By building six different final ML predictors, it is possible to understand the complexity of the problem, and how a single answer - according to the current state of the literature – is unlikely to solve the problem.

i **SynPred**

- *Original Research Article 5:* **António J Preto**, Pedro Matos-Filipe, Joana Mourão, Irina S Moreira, (2022). SYNPRED: prediction of drug combination effects in cancer using different synergy metrics and ensemble learning, *GigaScience*, 11, giac087,
`https://doi.org/10.1093/gigascience/giac087`
- *Indexed database:* **Preto AJ,** Matos-Filipe P, Mourão J, Moreira IS (2022). Supporting data for “SYNPRED: Prediction of Drug Combination Effects in Cancer using Different Synergy Metrics and Ensemble Learning” *GigaScience Database*.
`http://dx.doi.org/10.5524/102255`
- *GitHub repository 5:* SynPred
- *Website 3:* `http://www.moreiralab.com/resources/synpred/`

Chapter 3: Results and discussion

3.1. Protein understanding through HS prediction and MP features characterization

3.1.1. Predicting Hot Spots Using a Deep Neural Network Approach

Chapter 13

Predicting Hot Spots Using a Deep Neural Network Approach

António J. Preto, Pedro Matos-Filipe, José G. de Almeida, Joana Mourão, and Irina S. Moreira

Abstract

Targeting protein–protein interactions is a challenge and crucial task of the drug discovery process. A good starting point for rational drug design is the identification of hot spots (HS) at protein–protein interfaces, typically conserved residues that contribute most significantly to the binding. In this chapter, we depict point-by-point an in-house pipeline used for HS prediction using only sequence-based features from the well-known SpotOn dataset of soluble proteins (Moreira et al., Sci Rep 7:8007, 2017), through the implementation of a deep neural network. The presented pipeline is divided into three steps: (1) feature extraction, (2) deep learning classification, and (3) model evaluation. We present all the available resources, including code snippets, the main dataset, and the free and open-source modules/packages necessary for full replication of the protocol. The users should be able to develop an HS prediction model with accuracy, precision, recall, and AUROC of 0.96, 0.93, 0.91, and 0.86, respectively.

Key words Protein–protein interactions, Hot spots, Machine learning, Neural networks, Python, TensorFlow

Abbreviations

FN	False negatives
FP	False positives
TN	True negatives
TP	True positives

1 Introduction

The human interactome is composed of approximately 650,000 protein–protein interactions (PPIs), which dynamically contribute to the understanding of cellular function and organization [1]. Detailed characterization of PPIs is key, as their dysregulation

Hugh Cartwright (ed.), *Artificial Neural Networks*, Methods in Molecular Biology, vol. 2190,
https://doi.org/10.1007/978-1-0716-0826-5_13, © Springer Science+Business Media, LLC, part of Springer Nature 2021

is often involved in several diseases such as cancer, neurological disorders, metabolic diseases, and others [2]. As such, PPIs involved in disease pathways have become popular targets for the development of new diagnostic and therapeutic strategies [3, 4].

In PPIs, not all the residues contribute equally to the binding free energy and Hot-Spots (HS) are one of these cases. HS were defined as those residues that, upon alanine mutation, generate a variation of the free binding energy ($\Delta\Delta G_{\text{binding}}$) of at least 2.0 kcal/mol [5, 6]. These are typically conserved residues and have been identified as crucial for the tight binding and stability of proteins to their partners [6]. Computational methods, in particular machine learning (ML), have been used in recent years as a viable option to overcome the technical issues (e.g., cost, time, accuracy) concerning experimental techniques, providing thorough insights and a high-throughput HS identification [7]. The key principle of this approach is that it can provide answers based on a mathematical representation by recognizing patterns within data [8, 9], avoiding the need of being explicitly programmed to achieve its goal. In HS prediction, a panoply of features (e.g., physicochemical properties, evolutionary scores, solvent-accessible area, binding energy scores) were extracted from protein interactions, and ML algorithms such as support vector machines—SVM [10], neural networks [11], and extreme gradient boosting [12] were applied to develop prediction models. Moreira et al. [5] proposed SpotOn, a model that uses sequence—and structure-based features in an ML ensemble method to predict Hot-Spots and non-Hot-Spots.

In this chapter, we depict point-by-point, an in-house pipeline used for HS prediction and based on a deep neural network (DNN) in the same dataset published in SpotOn [5]. This dataset continues to be the most relevant collection of important biological HS. Furthermore, its size is adequate to highlight the importance of being able to handle small datasets, a recurrent problem in the overlap between the biological sciences and data analysis. Tensor Flow in a familiar python-based fashion was the chosen platform for this analysis.

DNNs are a complex type of artificial neural network (ANN). Overall, DNNs assume a graph-based architecture [13], where mathematical operations, updated according to a loss function (L), are performed in nodes connected by directed edges that carry weights (w_i) conditioning those mathematical operations [14]. The nodes can be organized in layers, where the first layer accepts the inputs, and the last layer returns the outputs. In between, secondary operations are executed in hidden layers. In each layer, extra information can be added in the form of biases (b) to enhance the final output of the DNN (these concepts are shown schematically in Fig. 1). While simpler ANNs comprise only one hidden layer, DNNs typically include more [15]. DNNs, like other

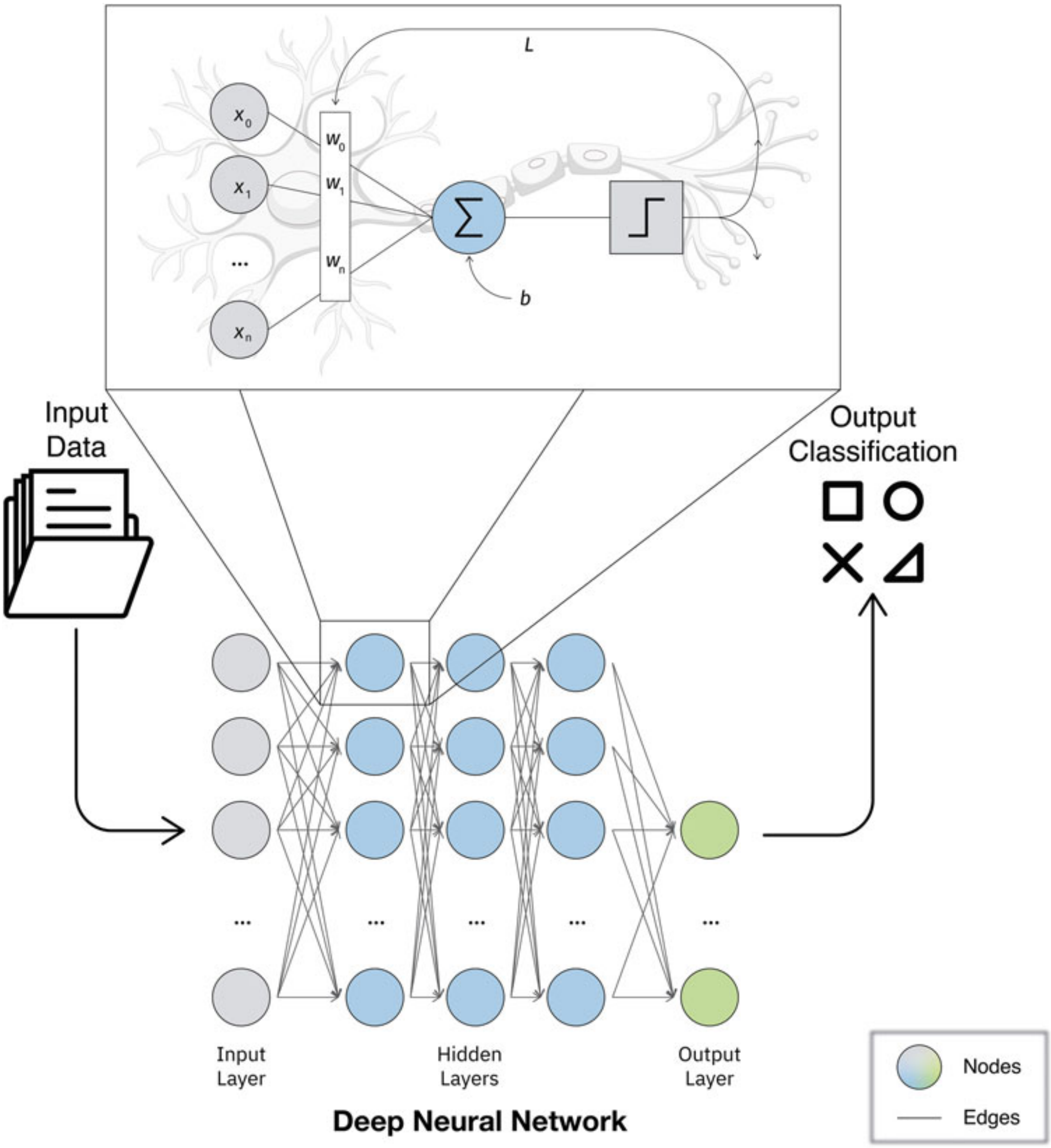


Fig. 1 Representation of a DNN. Weights are represented by w_i, the input of the loss function is represented by L and the bias to the node is represented by b

deep learning (DL) algorithms, allow for a greater understanding of abstract patterns within input data in comparison to simpler ML models [16].

2 Materials

All the materials used in this chapter are freely accessible through the Web. The provided code was tested in a 64-bit version of Linux Ubuntu 18.04 (Intel Xeon 40 Core 2.2 GHz, 126 GB RAM) and uses python version 3.7 and the associated free and open-source packages (*see* Subheading 2.2).

Table 1
Location of the SpotOn dataset and the one-hot encoded amino acid table

Description	Location	Reference
SpotOn (spoton.Csv) *The SpotOn dataset table has suffered minimal transformation.*	https://github.com/MoreiraLAB/Deep-Neural-Networks-for-Hot-Spots-prediction	[5]
Amino acid identification (one-hot encoded)		–

Table 2
Information regarding Python and the associated free and open-source packages as well as TensorFlow, the deep-learning library for designing, building, and training ML models

Name	Version	References
Biopython	1.74	[17]
NumPy	1.17	[18]
Pandas	0.25.1	[19]
Python	3.7.4	[20]
Scikit-learn	0.21.0	[21]
TensorFlow	1.14	[22]

2.1 Data

The SpotOn dataset (Table 1) used herein is constituted by 482 amino acids from 47 complexes. It was curated to ensure that every user can fully replicate the pipeline, solely from the code presented here and the tools and data available online.

2.2 Tools

The basic ML and python tools necessary to perform this tutorial are listed in Table 2.

3 Methods

This section will cover our approach to predict HS in step-by-step fashion, with easily recognizable sequence-based features, through the implementation of a simple neural network. This tutorial makes use of the SpotOn dataset of soluble proteins, in which several amino acids are classified as Hot-spots or non-Hot-spots (Table 3). The information for the features will be acquired through the corresponding files attained from Protein Data Bank (PDB). These files have tridimensional information of the proteins shown by indicating the space coordinates of each atom. To show how it is possible to collect the data, we will also be displaying the

Table 3
First four rows of the "spoton.csv" file

CPX	PDBChain	PDBResNo	PDBResName	Class
1A4Y	A	261	TRP	NS
1A4Y	A	263	TRP	NS
1A4Y	A	289	SER	NS
1A4Y	A	318	TRP	NS

The columns represent the PDBid (**CPX**), protein chain (**PDBChain**), residue number (**PDBResNo**), residue name (**PDBResName**), and hot-spot (**HS**) or non-hot-spots/null-spot (**NS**) labels (**Class**)

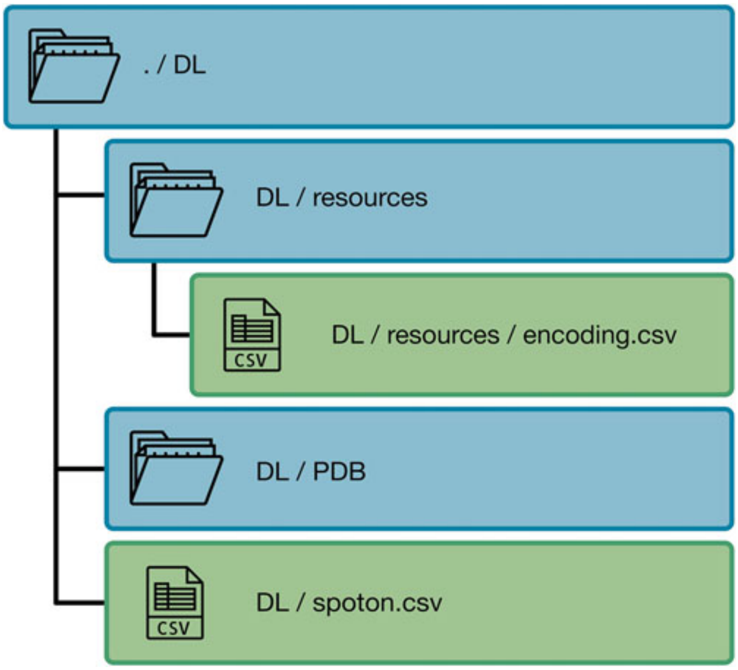


Fig. 2 Folder structure to deploy the protocol. The python scripts should be added inside the "DL" folder. Blue boxes represent folders and green boxes represent ".csv" files

steps necessary to download the ".pdb" files. This chapter assumes some familiarity with Python.

The process will be split into three main steps with Code Snippets (C.S.) provided for full replication of the protocol: (Subheading 3.1) feature extraction, (Subheading 3.2) deep learning classification, and (Subheading 3.3) model evaluation. The folder structure (Fig. 2), depicts the organization required to run the code smoothly, as well as where the dataset should be located. All the forthcoming code should be integrated into scripts and run from the terminal/command line with ≫*python script.py*. To run the code as it was originally run, you should have two scripts, the first containing the code from Subheading 3.1 and the second containing the code from Subheadings 3.2 and 3.3.

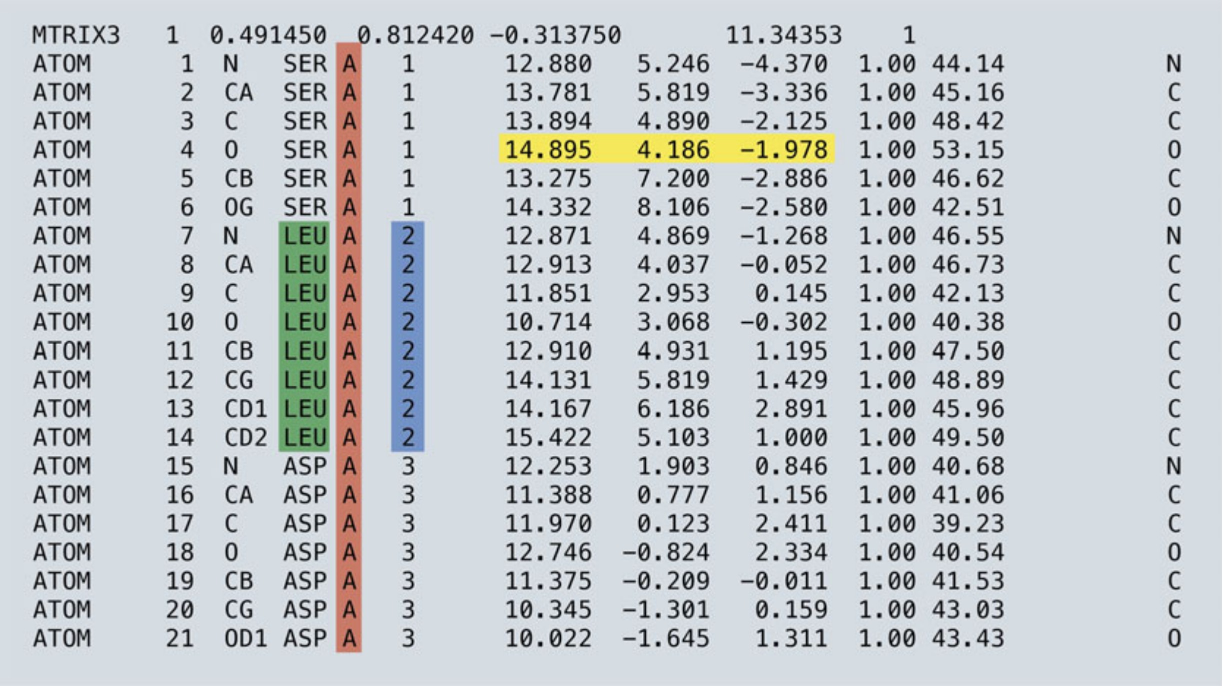
3.1 Feature Extraction

The first step of feature extraction is the acquisition of protein data from the ".pdb" files. In particular, for this protocol, we only used the amino acid number and the full sequence, which can be fetched from the column with the residue names of the ".pdb" files (Fig. 3).

In order to open and process the ".pdb" files we need several python packages (**C.S.1**): (1) to interact with the computer's folder structure in order to access the files with the operative system package (**os**), (2) to use *biopython* for easily download and manipulation of ".pdb" files (**Bio**), and (3) to effortlessly manipulate tables (***pandas***).

```python
import os
import Bio
from Bio.PDB import *
import pandas as pd
```

C.S.1: Importation of feature extraction packages.



```
MTRIX3    1  0.491450  0.812420 -0.313750       11.34353    1
ATOM      1  N    SER A   1      12.880   5.246  -4.370  1.00 44.14           N
ATOM      2  CA   SER A   1      13.781   5.819  -3.336  1.00 45.16           C
ATOM      3  C    SER A   1      13.894   4.890  -2.125  1.00 48.42           C
ATOM      4  O    SER A   1      14.895   4.186  -1.978  1.00 53.15           O
ATOM      5  CB   SER A   1      13.275   7.200  -2.886  1.00 46.62           C
ATOM      6  OG   SER A   1      14.332   8.106  -2.580  1.00 42.51           O
ATOM      7  N    LEU A   2      12.871   4.869  -1.268  1.00 46.55           N
ATOM      8  CA   LEU A   2      12.913   4.037  -0.052  1.00 46.73           C
ATOM      9  C    LEU A   2      11.851   2.953   0.145  1.00 42.13           C
ATOM     10  O    LEU A   2      10.714   3.068  -0.302  1.00 40.38           O
ATOM     11  CB   LEU A   2      12.910   4.931   1.195  1.00 47.50           C
ATOM     12  CG   LEU A   2      14.131   5.819   1.429  1.00 48.89           C
ATOM     13  CD1  LEU A   2      14.167   6.186   2.891  1.00 45.96           C
ATOM     14  CD2  LEU A   2      15.422   5.103   1.000  1.00 49.50           C
ATOM     15  N    ASP A   3      12.253   1.903   0.846  1.00 40.68           N
ATOM     16  CA   ASP A   3      11.388   0.777   1.156  1.00 41.06           C
ATOM     17  C    ASP A   3      11.970   0.123   2.411  1.00 39.23           C
ATOM     18  O    ASP A   3      12.746  -0.824   2.334  1.00 40.54           O
ATOM     19  CB   ASP A   3      11.375  -0.209  -0.011  1.00 41.53           C
ATOM     20  CG   ASP A   3      10.345  -1.301   0.159  1.00 43.03           C
ATOM     21  OD1  ASP A   3      10.022  -1.645   1.311  1.00 43.43           O
```

Fig. 3 Representation of a ".pdb" file attained by opening it with a text editor. This file lists the residue name (green), the chain name (red), the residue number (blue) as well as the atom coordinates (yellow)

Having imported the necessary tools, we need to write a function to automatically download the ".pdb" files (**C.S.2**).

```
def get_unique(input_df):

    from Bio.PDB import PDBList
    unique_pdbs = input_df.CPX.unique()
    pdbl = PDBList()
    for single_pdb in unique_pdbs:
        pdbl.retrieve_pdb_file(single_pdb, pdir='PDB', file_format = "pdb")
```

C.S.2: Use of *biopython* to download the ".pdb" files by iterating over a column ("CPX" corresponding to the complexes' PDBid) with the ".pdb" file code identifiers.

To prepare in advance the extraction of protein sequences from the ".pdb" files and the opening of tables in comma separated values (.csv) format, we developed a "utilities" class, which comprises a set of helper functions, that will be useful throughout the remainder of the section (**C.S.3**). This step requires the user to have a "PDB" folder inside the same folder where the code is run (Fig. 2).

```
class utilities:

    def __init__(self):

        self.amino_acids = ['CYS', 'ASP', 'SER', 'GLN', 'LYS',
                            'ILE', 'PRO', 'THR', 'PHE', 'ASN',
                            'GLY', 'HIS', 'LEU', 'ARG', 'TRP',
                            'ALA', 'VAL', 'GLU', 'TYR', 'MET']
       self.converter = {'CYS': 'C', 'ASP': 'D', 'SER': 'S', 'GLN': 'Q',
                         'LYS': 'K', 'ILE': 'I', 'PRO': 'P', 'THR': 'T',
                         'PHE': 'F', 'ASN': 'N', 'GLY': 'G', 'HIS': 'H',
                         'LEU': 'L', 'ARG': 'R', 'TRP': 'W', 'ALA': 'A',
                         'VAL':'V', 'GLU': 'E', 'TYR': 'Y', 'MET': 'M'}

    def table_opener(self, file_path, sep = ","):

        opened_decoder = pd.read_csv(file_path, sep = sep, header = 0)
        return opened_decoder
```

C.S.3: The "utilities" class with its "amino_acids" and "converter" functions allow us to treat protein sequences and easily convert between the single letter and the three-letter amino acid codes, necessary to process the full sequence. The "table_opener" function allows us to open a simple table easily and automatically generate a *pandas* data frame.

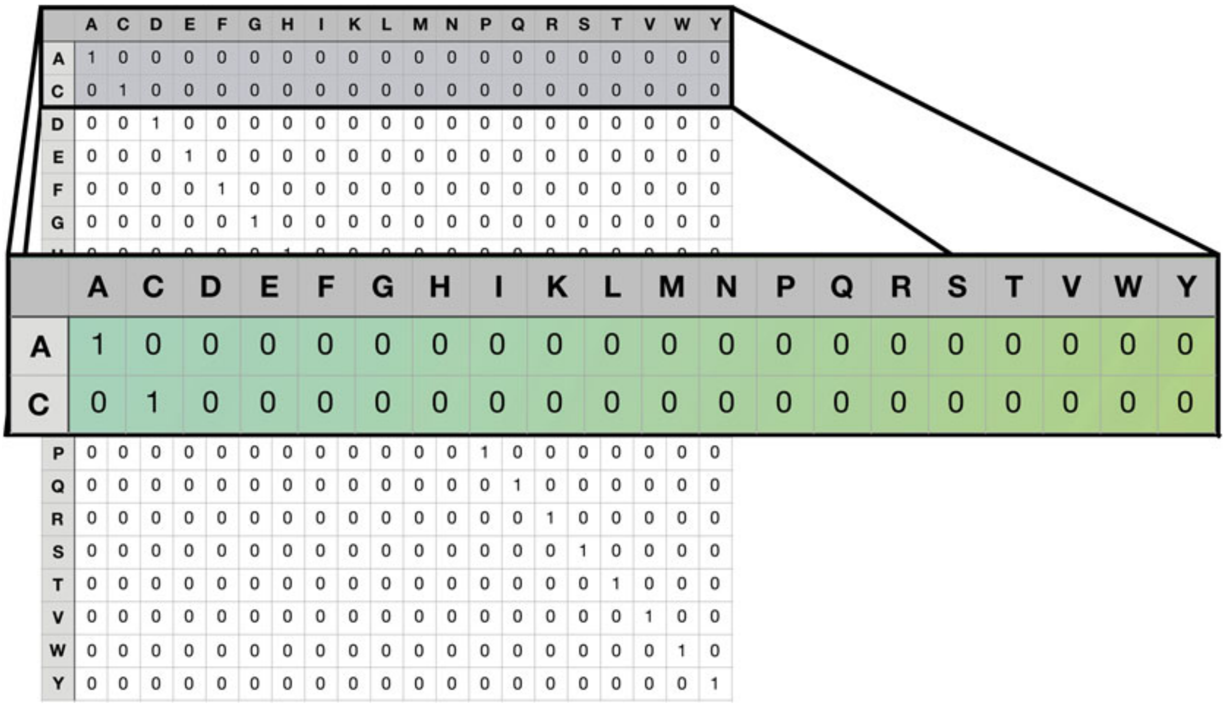
To systematically manipulate proteins, we need to store their sequence in a dictionary, that holds the proteins' information, particularly, residue number and name (**C.S.4**).

```
def retrieve_sequence_raw(input_folder = "PDB", system_sep = "/"):

    target_folder = os.getcwd() + system_sep + input_folder
    output_dict = {}
    for files in os.listdir(target_folder):
        parser = PDBParser()
        target_file = os.getcwd() + system_sep + input_folder +
            system_sep + files
        structure = parser.get_structure(files[0:-4], target_file)
        pdb_id, pdb_dict = structure.id[3:], {}
        for model in structure:
            for chain in model:
                chain_dict, chain_name, sequence = {}, chain.id, ""
                for residue in chain:
                    res_number, res_name = residue.get_full_id()[-1][1],
                   residue.resname
                if res_name in utilities().amino_acids:
                    single_letter = utilities().converter[res_name]
                    sequence += single_letter
                    chain_dict[res_number] = res_name
            pdb_dict[chain_name] = chain_dict
        output_dict[pdb_id] = pdb_dict
    return output_dict
```

C.S.4: The "retrieve_sequence_raw" function mines the "PDB" folder in order to construct a dictionary that holds all the proteins' numbered sequences.

To develop the proposed method, we also perform feature extraction. This is based upon straightforward features that are obtained from sequence alone. We built twenty features for each amino acid residue (**C.S.5**). These twenty features are a one-hot encoded version [23] of the target amino acid residue. So, they represent the twenty non-exotic amino acids and in only one of the columns a positive value could be found, while the remaining columns are filled with zero. For this, we built our one-hot amino acid encoding table (Fig. 4) in the form of a ".csv" file, stored in the "resources" folder (Fig. 2), which the user should add on the same folder of the script (*see* **Note 1** for additional remarks on this topic).

```
def generate_encoded(input_sequence):

    output_table = []
    encoded_table = utilities().table_opener(encoder_path)
    for residue_number in input_sequence.keys():
        residue_letter = utilities().converter[input_sequence[residue_number]]
        encoded_residue = encoded_table.loc[encoded_table[class_id_name] ==
                                         residue_letter].iloc[:,1:]
        proper_row = [residue_number] + list(encoded_residue.values[0])
        output_table.append(proper_row)
    header = [class_id_output] + list(encoded_residue)
    return pd.DataFrame(output_table, columns = header)
```



	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 4 One-hot encoded amino acid table, considering the single-letter amino acid codes (*A* alanine, *C* cysteine, *D* aspartic acid, *E* glutamic acid, *F* phenylalanine, *G* glycine, *H* histidine, *I* isoleucine, *K* lysine, *L* leucine, *M* methionine, *N* asparagine, *P* proline, *Q* glutamine, *R* arginine, *S* serine, *T* threonine, *V* valine, *W* tryptophan, *Y* tyrosine)

C.S.5: The "generate_encoded" scans a residue numbered dictionary from a protein chain and yields a table with their one-hot encoded version.

Iterating over the SpotOn table, upon use of the "converter" attribute from the "utilities" class, we can extract the one-hot encoded version of the amino acid, which is now able to be easily used as a feature.

Three more sequence-based features are also crucial for the fulfillment of this protocol. These features are simply the relative distance of every amino acid in the sequence to both the N- and C-termini of the protein and the relative distance to the target residue. Since, in this case, we only use the target residue, the last feature value is always zero. However, as such this does not affect the upcoming steps, and could be useful to the reader for other purposes, we kept this chunk of code. Thus, the "features" class (**C.S.6**), when iterating over the table from SpotOn, retrieves the one-hot encoded version of the sequence. This is performed by the "retrieve_sequence" function which needs to be adapted depending on the dataset table, namely, by changing the names of the column tables. Finally, this Python class also has the "location_features" function that takes the full sequence associated with the row in the target table and calculates the aforementioned relative positions for all the amino acids belonging to the sequence of the protein.

```
class features:

    def __init__(self, row):
        self.row = row

    def retrieve_sequence(self, input_sequences):

        self.complex_name = self.row["CPX"].lower()
        self.chain = self.row["PDBChain"]
        self.res_number = self.row["PDBResNo"]
        chain_sequence = input_sequences[self.complex_name][self.chain]
        encoded_sequence = generate_encoded(chain_sequence)
        return encoded_sequence

    def location_features(self, input_sequences, target_residue):

        self.sequence_table = self.retrieve_sequence(input_sequences)
        order_list = list(range(0,self.sequence_table.shape[0]))
        ordering = pd.DataFrame(order_list, columns = ["order"])
        inverse_ordering = pd.DataFrame(order_list[::-1], columns =
                                ["reverse_order"])
        pseudo_distance = pd.DataFrame(list(range(0,target_residue –
            self.sequence_table[class_id_output].iloc[0] + 1))[::-1]
            + list(range(1, self.sequence_table[class_id_output].iloc[-1] –
            target_residue + 1)), columns = ["pseudo_distance"])
        return pd.concat([self.sequence_table, ordering / ordering.max(),
            inverse_ordering / inverse_ordering.max(), pseudo_distance /
            pseudo_distance.max()], axis = 1)
```

C.S.6: The "features" class merges the previous functions into a tool that takes as input a row from the target table and extracts twenty-three sequence-based features from the protein sequence to which the target residue belongs.

To iterate over the "spoton.csv" file, we run the "generate_file" (**C.S.7**) function that takes as input the location of the referred table and the dictionary with the features. When iterating over the table, this function matches the target residues with the corresponding features and transforms the target label into a binary format. This function outputs two *pandas* data frames, one containing the original row identifiers from the SpotON table and the other containing the processed labels.

```
def generate_file(input_file, residues_features):

    prepared_table, classes = [], []
    for row in range(input_file.shape[0]):
        current_row = input_file.iloc[row]
        current_properties =pd.DataFrame(features(current_row)
                            .location_features(residues_features,
                            current_row[class_id_original]))
        writeable_row = list(current_row.values) + \
                    current_properties.loc[current_properties[class_id_output]
                    == current_row[class_id_original]].values.tolist()[0]
        if current_properties.isnull().any().any() == True: continue
        prepared_table.append(writeable_row)
        if current_row[class_name] == NS: classe = 0
        elif current_row[class_name] == HS: classe = 1
        classes.append(classe)

    return pd.DataFrame(prepared_table),
       pd.DataFrame(classes, columns = [class_name])
```

C.S.7: The "generate_file" function iterates over the target table, matches the amino acid residues with their corresponding features, transforms the label into a binary form and outputs two tables with the identifiers and features as well as the processed labels.

Before deploying the code, we defined a set of variables (**C.S.8**) containing most of the static information to be used. This set includes some of the file paths, usable variable strings as well as the names for the output files.

```
encoder_path = os.getcwd() + "/resources/encoding.csv"
target_table = "spoton.csv"
output_features_name = "spoton_clean.csv"
output_class_name = "class_clean.csv"
class_id_original = "PDBResNo"
class_id_output = "res_number"
class_id_name = "res_letter"
class_name = "Classe"
NS, HS = "NS", "HS"
```

C.S.8: Static variables to be used throughout the script.

Finally, we deploy the previous code to open the target file as a *pandas* data frame and download the ".pdb" files for all the proteins present in the SpotOn Table (**C.S.9**). Furthermore, we retrieve the dictionary containing the numbered sequences from the proteins present in SpotOn and use the target file and the processed sequences to extract the output data frames with the features and the label. Finally, we write these data frames into .csv files. These files will be used to perform deep learning classification in the following section.

```
opened_file = utilities().table_opener(target_table)
get_unique(opened_file)
residues_dict = retrieve_sequence_raw()
novel_features, classes = generate_file(opened_file, residues_dict)
novel_features.to_csv(output_features_name, sep = ",", index = False)
classes.to_csv(output_class_name, sep = ",", index = False)
```

C.S.9: Call the previous functions to use the whole pipeline and attain the files ready for Deep Learning deployment.

3.2 Deep Learning Classification

In the previous section, we finished our feature extraction deployment with two files: “spoton_clean.csv”, containing the original identifiers and the extracted features, and “class_clean.csv”, with the corresponding label for each residue in a processed format. This section makes use of these files as well as TensorFlow to construct a DNN that can classify the residues as hot spots or null spots. The code in this subsection has purposely been left unprocessed and more drawn out state, to allow the reader a clear understanding of the steps required to follow this part of the protocol.

Firstly, we import the needed packages (**C.S.10**). We use scikit-learn (***sklearn***) to perform random train-test split (*see* **Note 2** for details) and **TensorFlow** to perform the learning associated tasks. Finally, we also import numeric python (***numpy***) to handle different types of variables.

```
import pandas as pd
from sklearn.model_selection import train_test_split
import tensorflow as tf
import numpy as np
```

C.S.10: The packages imported for the Deep Learning classification script.

We then write the function “encode_binary” (**C.S.11**) to split our single label column from the previous steps into two columns, hence constructing a one-hot encoded version of the data that will make the upcoming steps simpler.

```
def encode_binary(input_col):

    HS_col, NS_col = [], []
    for class_value in input_col.values:
        if class_value == 1: HS_col.append(1), NS_col.append(0)
        elif class_value == 0: NS_col.append(1), HS_col.append(0)
    output_df = pd.DataFrame()
    output_df[NS], output_df[HS] = HS_col, NS_col
    return output_df
```

C.S.11: The "encode_binary" function takes a single column data frame and yields a two column one-hot encoded data frame version of the label.

Next, we construct the "neural_network" function (**C.S.12**) that sets up a five-layered neural network (*see* **Note 3** for further details) in which each layer is built by multiplying the weights and adding the bias factors to the result of the activation of the previous layer. The activation was either performed with ReLU or the sigmoid function.

```python
def neural_network(features):

    layer_1 = tf.add(tf.matmul(features,
                    weights['hidden_1']), biases['bias_1'])
    layer_2 = tf.add(tf.matmul(tf.nn.relu(layer_1),
                    weights['hidden_2']), biases['bias_2'])
    layer_3 = tf.add(tf.matmul(tf.nn.relu(layer_2),
                    weights['hidden_3']), biases['bias_3'])
    layer_4 = tf.add(tf.matmul(tf.nn.relu(layer_3),
                    weights['hidden_4']), biases['bias_4'])
    layer_5 = tf.add(tf.matmul(tf.nn.relu(layer_4),
                    weights['hidden_5']), biases['bias_5'])
    out_layer = tf.matmul(tf.nn.relu(layer_5),
                    weights['output'])
    return out_layer
```

C.S.12: The "neural_network" function sets up the input, the hidden and the output layers of the neural network to be later used.

To use the "neural_network" function, we need to set up the starting weights and biases (**C.S.13**).

```
weights = {
    'hidden_1': tf.Variable(tf.random_normal([num_input, num_hidden_1])),
    'hidden_2': tf.Variable(tf.random_normal([num_hidden_1, num_hidden_2])),
    'hidden_3': tf.Variable(tf.random_normal([num_hidden_2, num_hidden_3])),
    'hidden_4': tf.Variable(tf.random_normal([num_hidden_3, num_hidden_4])),
    'hidden_5': tf.Variable(tf.random_normal([num_hidden_4, num_hidden_5])),
    'output': tf.Variable(tf.random_normal([num_hidden_5, num_classes])),
    }

biases = {
    'bias_1': tf.Variable(tf.random_normal([num_hidden_1])),
    'bias_2': tf.Variable(tf.random_normal([num_hidden_2])),
    'bias_3': tf.Variable(tf.random_normal([num_hidden_3])),
    'bias_4': tf.Variable(tf.random_normal([num_hidden_4])),
    'bias_5': tf.Variable(tf.random_normal([num_hidden_5])),
    }
```

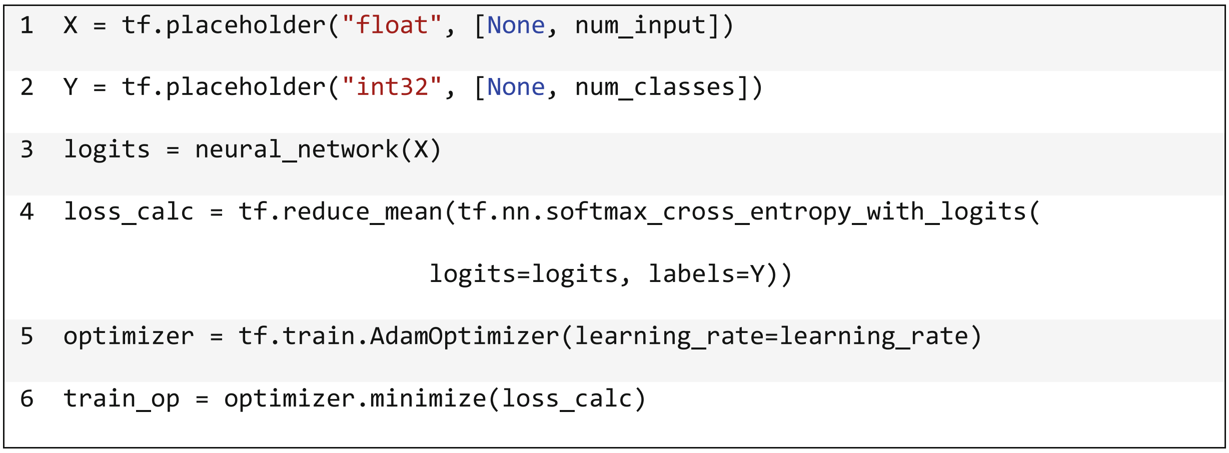
C.S.13: Set up the weights and biases of the neural network by yielding random values in a normal distribution.

We also need to set up some static neural network parameters for the neural network (**C.S.14**).

```
num_hidden_1 = 100
num_hidden_2 = 100
num_hidden_3 = 100
num_hidden_4 = 100
num_hidden_5 = 100
num_input = 23
num_classes = 2
display_step = 1
num_steps = 100000
learning_rate = 0.001
```

C.S.14: Set up the layer sizes, the number of steps, the learning rate and the batch size.

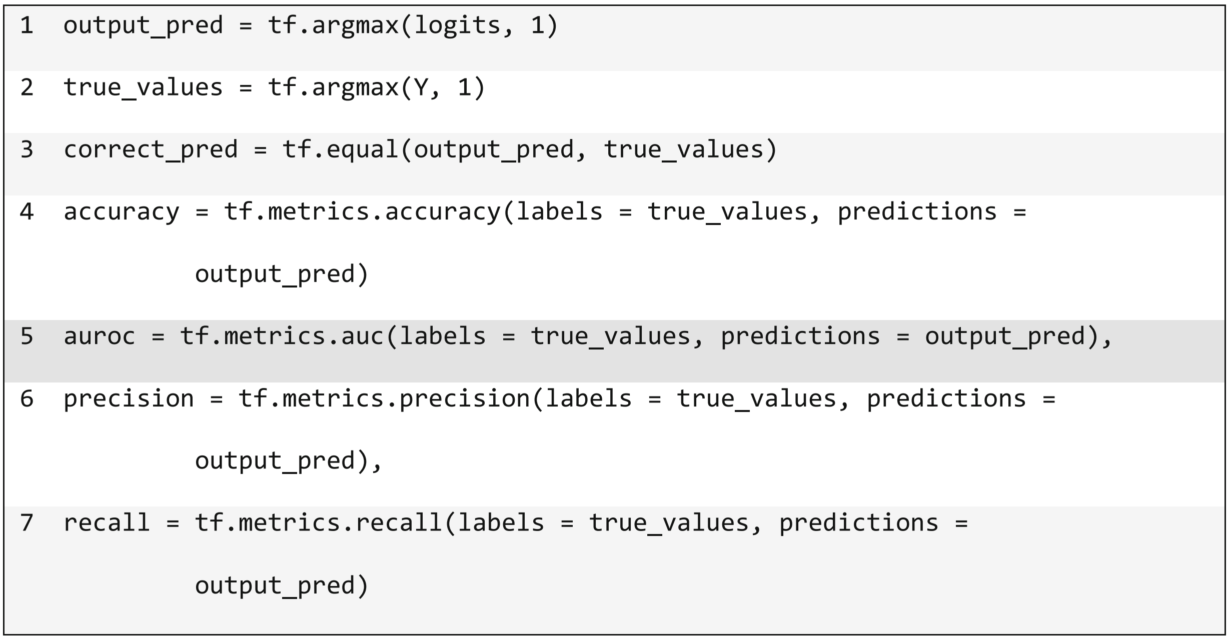
We set up TensorFlow placeholders to which the training data is fed (*X*, *Υ* variables). Furthermore, we also need to set up the "logits" variable, that can transform the output layer information into label prediction values. Next, we can calculate the loss with "softmax_cross_entropy_with_logits" and minimize this loss using an optimizer, that generally leads to good results (AdamOptimizer) [24], taking into consideration the learning rate (**C.S.15**). These are crucial steps to ensure the neural network can minimize the loss along the epochs (*see* **Note 4** for further details).



```
X = tf.placeholder("float", [None, num_input])
Y = tf.placeholder("int32", [None, num_classes])
logits = neural_network(X)
loss_calc = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(
                         logits=logits, labels=Y))
optimizer = tf.train.AdamOptimizer(learning_rate=learning_rate)
train_op = optimizer.minimize(loss_calc)
```

C.S.15: Prepare the neural network output processing and loss minimization.

Similarly, to the training data, we need to prepare variables that can compare the predictions with the labeled values and evaluate the model (accuracy, AUROC, precision, and recall) (**C.S.16**).



```
output_pred = tf.argmax(logits, 1)
true_values = tf.argmax(Y, 1)
correct_pred = tf.equal(output_pred, true_values)
accuracy = tf.metrics.accuracy(labels = true_values, predictions =
         output_pred)
auroc = tf.metrics.auc(labels = true_values, predictions = output_pred),
precision = tf.metrics.precision(labels = true_values, predictions =
         output_pred),
recall = tf.metrics.recall(labels = true_values, predictions =
         output_pred)
```

C.S.16: Prepare the comparison of the predictions and labels and the output evaluation.

Having set up everything, we need to load our data and randomly split into train and test set, in this case, using a 70:30 ratio (**C.S.17**) (*see* **Note 2** for details).

```
identifiers = pd.read_csv("spoton_clean.csv", sep = ",").iloc[:,0:4]
features = pd.read_csv("spoton_clean.csv", sep = ",").iloc[:,6:]
classes = pd.read_csv("class_clean.csv", sep = ",")
encoded_classes = encode_binary(classes)
X_train, X_test, y_train, y_test = train_test_split(features, encoded_classes,
test_size=0.3, random_state = 42)
```

C.S.17: Load the data and split it into a training and a test set.

Finally, we can initiate a TensorFlow session and deploy our neural network on the training set, printing out the loss and accuracy values of each epoch, which allows for the monitorization of the process. In the end, the model prints out the evaluation results for both the training and the test set (**C.S.18**).

```
init = tf.global_variables_initializer()
with tf.Session(config=tf.ConfigProto(log_device_placement=True)) as sess:
    sess.run(init)
    sess.run(tf.initialize_local_variables())
    for step in range(1, num_steps + 1):
        sess.run(train_op, feed_dict={X: batch_x, Y: batch_y})
        if step % display_step == 0 or step == 1:
            loss, acc = sess.run([loss_calc, accuracy],
               feed_dict={X:batch_x, Y: batch_y})
            print("Row:", row ,"Step " + str(step) + ", Loss= " + \
                      str(float(loss)) + ", Training Accuracy= " + \
                      str(float(acc)))
    print("Optimization Finished!")
    print("Training Accuracy:", \
        sess.run(accuracy, feed_dict={X: X_train, Y: y_train}))
    print("Training AUROC:", \
        sess.run(auroc, feed_dict={X: X_train, Y: y_train}))
    print("Training Precision:", \
        sess.run(precision, feed_dict={X: X_train, Y: y_train}))
    print("Training Recall:", \
        sess.run(recall, feed_dict={X: X_train, Y: y_train}))

    print("Testing Accuracy:", \
        sess.run(accuracy, feed_dict={X: X_test, Y: y_test}))
    print("Testing AUROC:", \
        sess.run(auroc, feed_dict={X: X_test, Y: y_test}))
    print("Testing Precision:", \
        sess.run(precision, feed_dict={X: X_test, Y: y_test}))
    print("Testing Recall:", \
        sess.run(recall, feed_dict={X: X_test, Y: y_test}))
```

C.S.18: Deploy the TensorFlow model and evaluate the results.

3.3 Metrics Used for Evaluating Model Performance

After deploying the pipeline indicated in the METHODS section, for the SpotOn dataset, we achieved the results presented in Table 4. The different metrics shown are (1) accuracy: represents the fraction of correct predictions by our model (Eq. 1); (2) precision: attempts to answer what fraction of positive identifications are actually correct (Eq. 2); (3) recall: represents the fraction of actual positives that were identified correctly by our algorithm (Eq. 3); and (4) AUROC: measures the two-dimensional area underneath the entire receiver operating characteristic (ROC) curve (Eq. 4). On a ROC curve, recall (*rec*) (Eq. 3) is plotted on the *y* axis and selectivity (*sel*) (Eq. 5) is plotted on the *x* axis.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Table 4
Results of the deployment of the METHODS section in the SpotOn dataset

	Training-set	Test-set
Accuracy	0.95	0.96
Precision	0.99	0.93
Recall	0.96	0.91
AUROC	0.97	0.86

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{AUROC} = \int_0^1 \text{rec}(\text{sel}(x))\,\text{d}x \tag{4}$$

$$\text{selectivity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5}$$

4 Notes

1. Although we built our own table, it is also possible to generate an internal one-hot encoded version of the amino acids. A simple approach to do this would be to use the function LabelEncoder, from the *sklearn* package.
2. In ML, datasets are usually split into training (for adjusting the model's weights and biases) and test (for evaluation) sets. The fraction of the whole dataset reserved for training and testing stages is usually stapled at 70% and 30% respectively; however, this ratio may vary depending on the characteristics of the original dataset and the model itself [25].
3. The number of hidden layers and nodes included within them is often obtained via a grid-search procedure where various architectures are tested. The user then selects the best hyperparameter set to attain a high-performance algorithm.
4. The number of epochs corresponds to a hyperparameter that defines the number of times an entire dataset passed through the learning algorithm. An epoch that is usually too big to feed the network is divided into smaller batches, containing a lesser number of samples.

Acknowledgments

This work was supported by the European Regional Development Fund (ERDF), through the Centro 2020 Regional Operational Programme under project CENTRO-01-0145-FEDER-000008: BrainHealth 2020 and through the COMPETE 2020—Operational Programme for Competitiveness and Internationalisation and Portuguese national funds via FCT—Fundação para a Ciência e a Tecnologia, under project[s] POCI-01-0145-FEDER-031356, PTDC/QUI-OUT/32243/2017, and UIDB/04539/2020. A. J. Preto was also supported by FCT through PhD scholarship SFRH/BD/144966/2019. I. S. Moreira was funded by the FCT Investigator Programme—IF/00578/2014 (co-financed by European Social Fund and Programa Operacional Potencial Humano). The authors would like also to acknowledge ERNEST—European Research Network on Signal Transduction, CA18133, and STRATAGEM—New diagnostic and therapeutic tools against multidrug-resistant tumors, CA17104.

References

1. Kotlyar M, Pastrello C, Malik Z et al (2019) IID 2018 update: context-specific physical protein–protein interactions in human, model organisms and domesticated species. Nucleic Acids Res 47:D581–D589
2. Lage K (2014) Protein–protein interactions and genetic diseases: the interactome. Biochim Biophys Acta Mol basis Dis 1842:1971–1980
3. Ran X, Gestwicki JE (2018) Inhibitors of protein–protein interactions (PPIs): an analysis of scaffold choices and buried surface area. Curr Opin Chem Biol 44:75–86
4. Fry DC (2015) Targeting protein-protein interactions for drug discovery. Protein-protein interactions. Methods Mol Biol 1278:93–106
5. Moreira IS, Koukos PI, Melo R et al (2017) SpotOn: high accuracy identification of protein-protein Interface hot-spots. Sci Rep 7:8007
6. Moreira IS, Fernandes PA, Ramos MJ (2007) Hot spots-a review of the protein-protein interface determinant amino-acid residues. Proteins 68:803–812
7. Melo R, Fieldhouse R, Melo A et al (2016) A machine learning approach for hot-spot detection at protein-protein interfaces. Int J Mol Sci 17:1215
8. Sommer C, Gerlich DW (2013) Machine learning in cell biology—teaching computers to recognize phenotypes. J Cell Sci 126:5529–5539
9. Libbrecht MW, Noble WS (2015) Machine learning applications in genetics and genomics. Nat Rev Genet 16:321–332
10. Lise S, Buchan D, Pontil M et al (2011) Predictions of hot spot residues at protein-protein interfaces using support vector machines. PLoS One 6:e16774
11. Ofran Y, Rost B (2007) ISIS: interaction sites identified from sequence. Bioinformatics 23: e13–e16
12. Wang H, Liu C, Deng L (2018) Enhanced prediction of hot spots at protein-protein interfaces using extreme gradient boosting. Sci Rep 8:14285
13. Jain AK, Jianchang M, Mohiuddin KM (1996) Artificial neural networks: a tutorial. Computer (Long Beach Calif) 29:31–44
14. Gonzalez RC (2018) Deep convolutional neural networks [lecture notes]. IEEE Signal Process Mag 35:79–87
15. Bengio Y (2009) Learning deep architectures for AI. Found trends®. Mach Learn 2:1–127
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
17. Cock PJA, Antao T, Chang JT et al (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25:1422–1423
18. van der Walt S, Colbert SC, Varoquaux G (2011) The NumPy Array: a structure for

efficient numerical computation. Comput Sci Eng 13:22–30

19. McKinney W (2010) Data structures for statistical computing in python, in: proceeding of the 9th python in science Conf (SciPy 2010), Austin, Texas
20. Rossum G van, Boer J de (1991) Linking a stub generator (AIL) to a prototyping language (python), In: EurOpen Conference Proceedings, Tromso, Norway
21. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
22. Abadi M, Agarwal A, Barham P et al (2015) TensorFlow: large-scale machine learning on heterogeneous distributed systems, *preprint* available at arXiv:1603.04467
23. Buckman J, Roy A, Raffel C et al (2018), Thermometer encoding: one hot way to resist adversarial examples. In: 6th international conference on learning representations (ICLR 2018), Vancouver, Canada
24. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *Preprint* available at arXiv:1412.6980
25. Crowther PS, Cox RJ (2005) A method for optimal division of data sets for use in neural networks, presented at the knowledge-based intelligent information and engineering systems. KES 2005. In: Lecture notes in computer science, vol 3684. Springer, Berlin, Heidelberg

3.1.2. SPOTONE: Hot Spots on Protein Complexes with Extremely Randomized Trees via Sequence-Only Features

International Journal of
Molecular Sciences

Article

SPOTONE: Hot Spots on Protein Complexes with Extremely Randomized Trees via Sequence-Only Features

A. J. Preto [1] and Irina S. Moreira [2,*]

[1] CNC—Center for Neuroscience and Cell Biology, University of Coimbra, 3004-504 Coimbra, Portugal; martinsgomes.jose@gmail.com

[2] Department of Life Sciences, Center for Neuroscience and Cell Biology, Coimbra University, 3000-456 Coimbra, Portugal

* Correspondence: irina.moreira@cnc.uc.pt

Received: 10 August 2020; Accepted: 30 September 2020; Published: 1 October 2020

Abstract: Protein Hot-Spots (HS) are experimentally determined amino acids, key to small ligand binding and tend to be structural landmarks on protein–protein interactions. As such, they were extensively approached by structure-based Machine Learning (ML) prediction methods. However, the availability of a much larger array of protein sequences in comparison to determined tree-dimensional structures indicates that a sequence-based HS predictor has the potential to be more useful for the scientific community. Herein, we present SPOTONE, a new ML predictor able to accurately classify protein HS via sequence-only features. This algorithm shows accuracy, AUROC, precision, recall and F1-score of 0.82, 0.83, 0.91, 0.82 and 0.85, respectively, on an independent testing set. The algorithm is deployed within a free-to-use webserver, only requiring the user to submit a FASTA file with one or more protein sequences.

Keywords: big-data; hot-spots; machine learning; protein–protein complexes; structural biology

1. Introduction

Hot-Spots (HS) can be defined as amino acid residues that upon alanine mutation generate a change in binding free energy ($\Delta\Delta G_{binding}$) higher than 2.0 kcal mol^{-1}, in opposition to Null-Spots (NS), which are unable to meet this threshold. Although the threshold of 2.0 kcal mol^{-1} can vary in the definition of HS, a representative amount of studies on the subject typically use this cut-off [1–6]. HS are key elements in Protein–Protein Interactions (PPIs) and, as such, fundamental for a variety of biochemical functions. The disruption of these interactions can alter entire pathways and is of interest to therapeutic approaches [1,7]. These residues are also known to be important for protein dimerization [8] and ligand binding [9]. Indeed, HS tend to be associated with the binding of small ligands, hence becoming ideal subjects of study on target proteins for drug design approaches [9–11].

Databases of experimental determined HS and NS can be found in the literature: ASEdb [12], BID [13], PIN [14] and SKEMPI [15]. More recently, SKEMPI 2.0 was released, making available a larger amount of experimental information. However, most of the new information does not include mutations to alanine (and the corresponding change in free binding energy), which is the material under scope in the present work [16]. These databases can be used to deploy Machine-Learning (ML) algorithms that take both the positive (HS) and negative (NS) information and construct a binary classifier that should be able to predict, upon previously unforeseen amino acid residues in a protein, its HS/NS status. Although ML is not limited to binary classification, on this problem and given the available data format, binary classification was the most explored approach until now. Several algorithms have been proposed for HS computational predictions, using different ML approaches, features and datasets [17–25]. Recently (2017), SPOTON [22], using information on both the protein sequence and structure, achieved results of 0.95 accuracy on an independent testing set, making it the best performing HS predictor at the time. Most of the high-performing HS predictors incorporate structural information. Although yielding clearly robust results, it hinders the possibilities of a broader deployment, since there are still fewer proteins for which a three-dimensional (3D) structure is available in online repositories [26] compared to the determined and available protein sequences [27]. It is known that sequence-based predictors tend to perform more poorly, in comparison with the ones engulfing structural information. For example, Nguyen et al. (2013) [19] were able to achieve an accuracy of 0.79 and a precision of 0.75 using sequence-based frequency-derived features. More recently, Hu et al. (2017) [20] achieved an F1-score of 0.80 using only sequence-based features while Liu et al. (2018) [21] achieved an F1-score of 0.86 using sequence-based features and amino acid relative Solvent Accessible Surface Area (SASA). The problem of HS computational determination is usually riddled with class imbalance, as there are commonly more experimentally determine residues as NS than HS due to the nature of PPIs. Conversely, the size of the dataset is usually not large enough to dilute this discrepancy. As such, problems emerge on the dataset training, but, more importantly, on the analysis of the results. We developed SPOTONE (hot SPOTs ON protein complexes with Extremely randomized trees), a HS predictor that only makes use of protein sequence-based features, all of which were calculated with an in-house Python pipeline. To avoid protein-centered overfitting, features concerning the whole protein were not applied to the classification problem. This allowed us to avoid the predictor from learning HS/NS only on a specific subset of proteins and be able to correctly classify even for unforeseen subtypes of biological machineries. Furthermore, we deployed a rigorous train–test split that ensured equality among classes, not only in the training and testing datasets, but also regarding the amino acid types. The resulting platform and predictor are available at: http://moreiralab/resources/spotone.

2. Results

The results presented herein were attained following a ML pipeline, depicted in Figure 1, which lays the overall steps involved in dataset preparation and prediction model training and refinement. The detailed version of each step is further explored in the Material and Methods Section.

2.1. Dataset

We began by analyzing our dataset, the same previously mined and cleaned for SPOTON [22], composed by 534 amino acid residues, of which 127 are HS and 407 are NS, from 53 protein–protein complexes. Figure 2A shows the class distribution by amino acid type. Clearly, TYR, one of the most common HS in nature, is an outlier. Secondly, it should be noted that MET and CYS have no registered HS. Finally, it should also be noted that, due to the nature of the method used for HS experimental determination, there are no ALA residues in either the HS or NS class (as already explained). Figure 2B shows the split of the protein primary sequences into four equally long quartiles, which allowed us to analyze the HS/NS distribution along these ordered sections. It should be noted that, in the first quartile of the protein, the number of HS is at its highest value, although the number of NS is not equally as high. In the last quartile of the protein sequences, the number of overall registered HS/NS is the lowest; however, the proportion in which they stand favors the existence of HS rather than NS, in comparison with the remaining quartiles. The comparison with the literature-based features can be consulted at the landing page of our website. These features include secondary structure propensity, pKa associated values, number of atoms of each type and standard area and mass associated values. Their analyses can show tendencies of these features that correlate to their usefulness to the ML deployment.

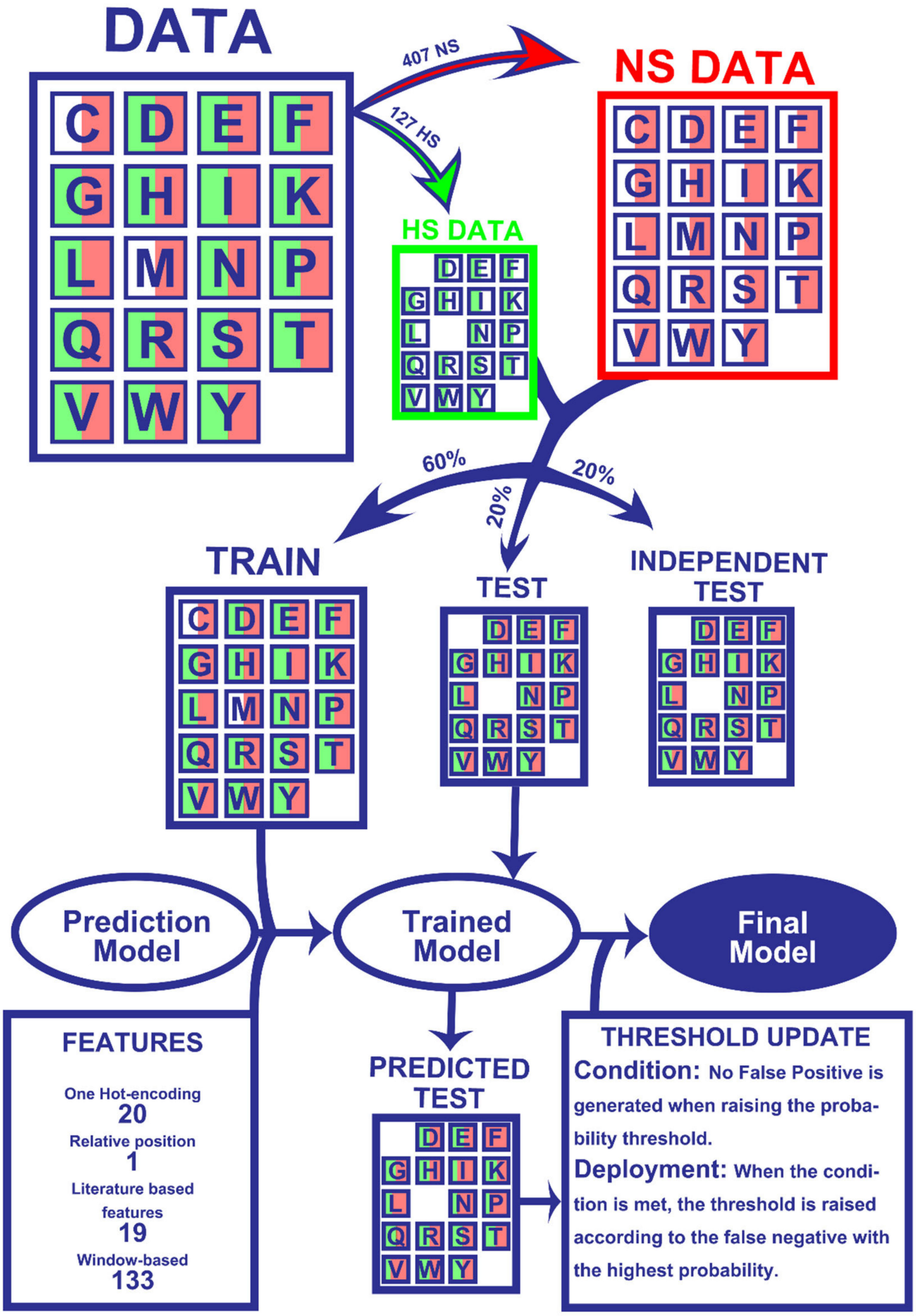

DATA
C D E F
G H I K
L M N P
Q R S T
V W Y
407 NS
127 HS
NS DATA
HS DATA
60%
20%
20%
TRAIN
TEST
INDEPENDENT TEST
Prediction Model
Trained Model
Final Model
FEATURES
One Hot-encoding
20
Relative position
1
Literature based features
19
Window-based
133
PREDICTED TEST
THRESHOLD UPDATE
Condition: No False Positive is generated when raising the probability threshold.
Deployment: When the condition is met, the threshold is raised according to the false negative with the highest probability.

Figure 1. Workflow of the Machine Learning pipeline. Firstly, the 534 amino acids were split into experimentally determined HS (127) and NS (407). Secondly, 60% of the entries of both classes were randomly picked for the train dataset while the remaining 40% were not used for the training phase (20% for test and 20% for an independent test). All datasets were matched with their corresponding 173 features. The training data were used to train the models, which were tested on the test set to yield HS/NS predictions. The predictions were then used to update probability thresholds and generate the final model, which basically consists of the trained model with subsequent HS probability correction. The final model was then applied to an independent test, which did not influence any step of the process, in order to be evaluated. More details on the used method can be found in the Materials and Methods Section.

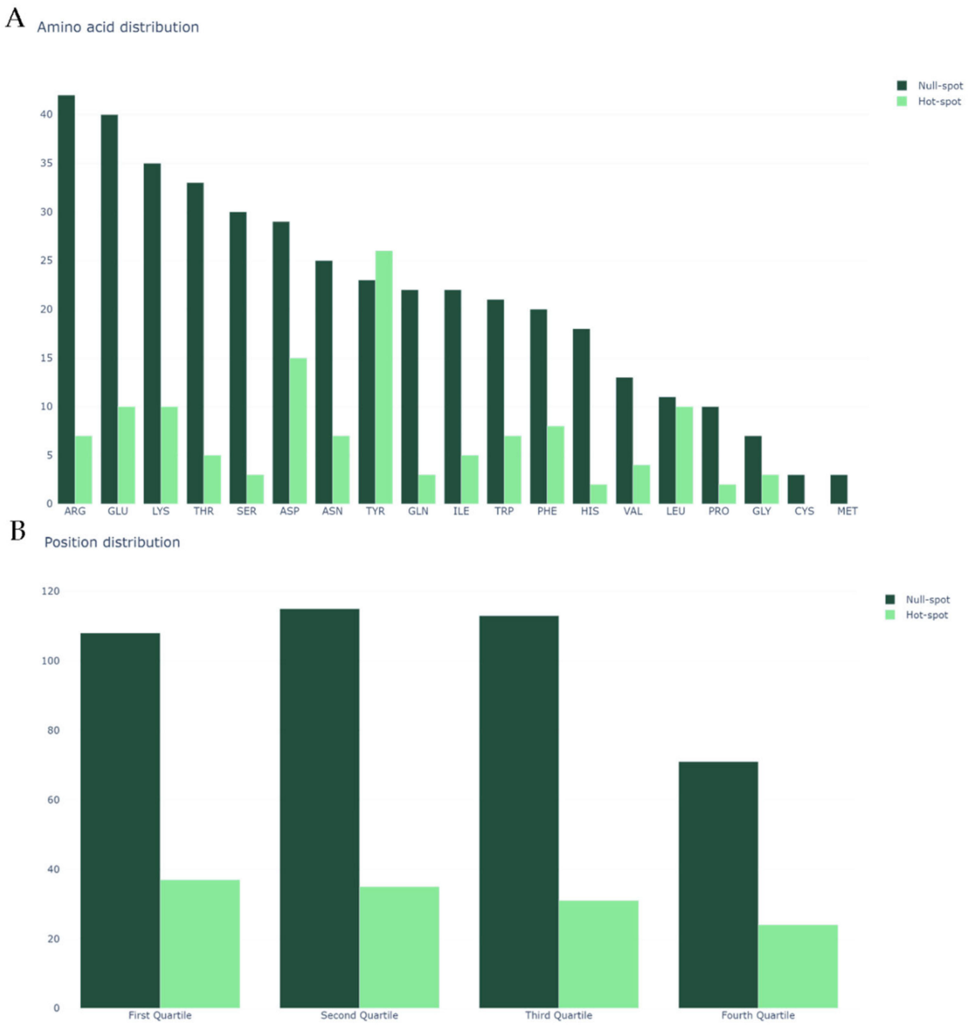


Figure 2. (**A**) Class distribution by amino acid type; and (**B**) class distribution by relative position in the protein sequence. In both plots, the y-axis represents the amino acid count.

2.2. Machine-Learning Algorithms

Tables A1 and A2 in the Appendix A list the full results attained for the various algorithms and methods. Table A1 shows that the in-house built features subset displayed one of the highest performance metrics in comparison with any of the other features alone. It can be noticed that PSSM led to a slight improvement, but the small difference of performance does not compensate the larger amount of time needed for this feature calculation. The introduction of iFeatures, concerning the whole protein, did not increase significantly the performance and introduced concerns related to protein-centered overfitting, and as such was discarded of further studies.

The extremely randomized trees took the lead in most performance metrics, and it is clearly more robust in what concerns the identification of HS, as denoted by the high recall score. It should be noted that neither grid search parameter tuning nor prediction probability tuning according to amino acid type performance was used before method selection to keep the independent test unbiased (further explained in the Material and Methods Section). As such, all values presented in Table 1 concern default settings. This allowed the selection of extremely randomized trees algorithm for parameter tuning, as well as subsequent required alterations.

Table 1. ML results in the training and testing sets using 5 different algorithms and evaluated using the metrics accuracy (Acc), AUROC, precision (Prec), recall (Rec) and F1-score (F1).

Method	Data	Acc	AUROC	Prec	Rec	F1
Neural network	Train	0.81	0.73	0.81	0.81	0.81
	Test	0.69	0.56	0.72	0.69	0.71
AdaBoost	Train	0.98	0.98	0.98	0.98	0.98
	Test	0.71	0.56	0.77	0.71	0.74
Support Vector Machine	Train	0.77	0.00	1.00	0.77	0.87
	Test	0.76	0.00	1.00	0.76	0.86
Extremely Randomized Trees	Train	0.99	1.00	0.99	0.99	0.99
	Test	0.81	0.77	0.88	0.81	0.83

To avoid the adaptation introduced and displayed in Table A3 leading to the generation of false positives, we set half of the testing set aside, comprising 20% of the whole dataset. Table 2, which lists the performance metrics of the parameter-tuned adapted model for both the training and the testing set, shows a significant increase in the testing performance, while the training scores remain unchanged. This trend was further validated by deploying the model in the independent testing set.

Table 2. Performance metrics on the same training and testing sets after updating the prediction thresholds, and evaluated using the metrics accuracy (Acc), AUROC, precision (Prec), recall (Rec) and F1-score (F1).

Data	Acc	AUROC	Prec	Rec	F1
Training after threshold adaptation	0.99	0.99	0.99	0.99	0.99
Testing after threshold adaptation	0.85	0.88	0.93	0.85	0.87
Independent Testing after threshold adaptation	0.82	0.83	0.91	0.82	0.85

It should be noted that the “class_weight” parameter, available on the deployment of the extremely randomized trees used was particularly relevant in tackling class imbalance, since, by setting it to “balanced_subsample”, it generates and updates class weights based on the samples. A full comparison with state-of-the-art predictions is shown in Table 3. Apart from SPOTON [22], two values for each performance metric are listed: on the left is the value assessed with the dataset used on SPOTONE and on the right are the values presented in the corresponding scientific papers for each method. These values were attained from the pipeline used in SPOTON [22]; since the dataset is the same, the performance comparison also stands equal. In the case of the sequence-based methods that are not SPOTONE, we were not able to deploy our dataset as the webservers indicated were not active or available; this applies to the methods of Nguyen et al. (2013) [19] (reported metrics in their dataset: accuracy of 0.79, recall of 0.59, F1-score of 0.66 and precision of 0.75), Hu et al. (2017) [20] (reported metrics in their dataset: recall of 0.67, F1-score of 0.80 and precision of 1.00) and Liu et al. (2018) [21] (reported F1-score of 0.86 in their dataset).

Table 3. Structure-based HS prediction performances.

Metrics for Testing-set Evaluation	Structure-Based Methods			
	SPOTON [22]	**SBHD2 [23]**	**KFC-A [24]**	**KFC-B [25]**
AUROC	0.91	0.69/0.69	0.66/–	0.67/-
Recall	0.98	0.70/0.77	0.53/0.85	0.28/0.62
F1-score	0.96	0.62/0.86	0.56/-	0.42/-

3. Discussion

This work presents a significant improvement in HS prediction at the interface of protein–protein complexes. However, more than the high performing metrics, the robustness of this model emerges from a thorough treatment and splitting of the dataset, as well as from the exclusion of whole protein sequence features, leaving only residue specific sequence-based features. Figures A1–A3 display the performance of SPOTONE upon being applied to three different complexes (PDB ids: 1a4y, 1jck and 3sak), with insights on all the residues experimentally determined for these complexes and comparison of this information to our HS/NS SPOTONE prediction. These three examples clearly show how well the predictor works on a point-by-point example. Our final accuracy (0.82), recall (0.82) and precision (0.91) highlight the existence of a very low number of falsely predicted HS as well as NS. Its closeness in performance to the best structural based predictor is complemented with the high versatility of using only sequence-based features prediction, which allows a much wider application in a variety of biological problems.

Finally, all the work is available in a free-to-use platform that allows the user to input one or more protein sequences in FASTA format (Box 1) and attain a detailed HS/NS prediction with corresponding graphical interface. The platform is available at http://moreiralab.com/resources/spotone.

Box 1. Example FASTA file, with the different proteins' chains separated by paragraphs and clear identifiers initiated with ">", separated from the single letter amino acid code chain with a paragraph. This needs to be stored in a ".fasta" file to be submitted to SPOTONE.

```
>6Q1G:H|PDBID|CHAIN|SEQUENCE
ASQVQLQESGPGLVKPSGTLSLTCAISGGSISSSNWWTWVRQPPGKGLQWIGEIQHGGGTNYNPS
LKSRATIFVDVSKNHFSLRLSSVTAADTAVYYCAKVPPYCTSASCPDDYYYHYMDVWGKGTTVTV
SGASTKGPSVFPLAPSSKSTSGGTAALGCLVKDYFPEPVTVSWNSGALTSGVHTFPAVLQSSGLYSL
SSVVTVPSSSLGTQTYICNVNHKPSNTKVDKRVEPKSCDKHHHHHH
>6Q1G:L|PDBID|CHAIN|SEQUENCE
ASSSELTQDPAVSVALGQTVRITCQGDSLRGYSASWYQLKPGQAPVLVIYGKNNRPSGIPDRFSGST
SGNRASLIITGTQAEDEADYYCNSRDTNGYRPVLFGGGTKLTVLGQPKGAPSVTLFPPSSEELQAN
KATLVCLISDFYPGAVTVAWKADSSPVKAGVETTTPSKQSNNKYAASSYLSLTPEQWKSHRSYSCQ
VTHEGSTVEKTVAPTECS
```

4. Materials and Methods

The dataset used here was retrieved from our previous method, SPOTON [22], and is comprised of 534 amino acid residues (127 positive-HS and 407 negative-NS). This dataset was constructed of data merged from the experimental databases ASEdb [12], BID [13], PINT [14] and SKEMPI [15], and as such comprises all literature available experimental data coming from alanine scanning mutagenesis. We also highlight that sequence redundancy was already eliminated in our previous work. To address this particular problem, we did not simply split the 534 samples into training and testing sets. Firstly, we split all the samples into two datasets containing either HS or NS. Of these datasets, we extracted 20 different subsets from each (corresponding to the 20 possible amino acids). We randomly split these 40 sets (20 HS subsets and 20 NS subsets) in a 60:40 ratio, using "train_test_split" from scikit-learn [28]. Finally, we stitched the tables corresponding to the training set and the testing set back together. Our process was devised to ensure that HS and NS were equally represented for each residue in both the training set and the testing set. Unfortunately, ALA entries were completely absent from the dataset (due to the experimental detection method typically used in wet labs) and CYS and MET only had NS entries (as these residues have a lower/null incidence as key in PPIs). For the latter two cases, we included them in the training set, as it would not be possible to assay their presence in the testing set. Following this procedure, we ended up with a training set containing 312 residues and a testing set containing 222 residues. We randomly split the final testing set in two, with 111 residues each; half the testing set was used to fine-tune probability thresholds (see Prediction Probability Tuning), while the other half was set aside for fully independent test analysis, only having been used after selecting the ML model and performing all parameter tuning.

4.1. Features

The following section reports the calculation of 173 features with an in-house Python pipeline and literature-based information on amino acid characteristics. All the extracted features can be calculated simply using the input sequence of a FASTA file. It should be noted that we only used sequence-based features and, furthermore, we did not add any sequence feature about the protein as a "whole", which might have, due to the size of the dataset, promoted overfitting on a protein level. As shown in Tables A1 and A2, pre-constructed whole-sequence based features and Position-Specific Scoring Matrix (PSSM) were also tested. For the first, we used iFeature [29] and attained 14.056 whole sequence-based features, for each of the chains. For PSSM, we used an in-house psiblast [30] deployment to extract 42 position conservation features.

4.2. One-hot Encoding (20 Features)

The first twenty features extracted for each amino acid residue were simply a one-hot encoded representation of the amino acid; thus, for each amino acid, nineteen columns were filled with "0", and only one (with the corresponding value), was filled with "1".

4.3. Relative Position Feature (1 Feature)

In Figure 2B, we display the abundance of NS/HS on the protein sequence quartiles. The quartiles were defined by splitting the proteins' length by four and analyzing the residues present in each of the sections. As such, we used the numbering 1–4 (representing its relative position in the sequence) as a feature that indicates the quartile in which each amino acid is present.

4.4. Literature-Based Features (19 Features)

Several amino acid properties are constantly determined, updated and made available online. We downloaded 19 amino acid properties from the BioMagResBank [31] and associated each of them with each of the amino acids; the features and corresponding values per amino acid used are listed in Tables A4 and A5. Please note that this database is regularly updated to improve the reliability of the experimental data. The statistical distribution of these properties regarding their HS/NS on the dataset used are available in form of violin, scatter and boxplots on the landing page (http://www.moreira.com/resources/spotone).

4.5. Window-Based Features (133 Features)

Window-based features were described with a "sliding windows" that stopped on the target residue and considered the residues that stand close to it, sequence wise. We considered window sizes of 2, 5, 7, 10, 25, 50 and 75 amino acid residues, and, for each target residue, averaged the values corresponding to the features of in the Literature-Based Features Section on the residues comprised in the windows. Thus, if we multiply the number of raw features (19) by the number of windows (7), we added 133 features.

4.6. Machine-Learning Models Deployment

We exploited different algorithms: Neural Networks ("MLPClassifier") [32], Random Forest ("RandomForestClassifier") [33], AdaBoost ("AdaBoostClassifier"), Support Vector Machine ("SVC") [34] and Extremely Randomized Trees ("ExtraTreesClassifier") [35]. All of the algorithms were used from their scikit-learn [28] deployment. The extremely randomized trees algorithm, similar to a random forest, is based on decision trees. From the training set, the algorithm picks attributes at random and generates subsets; by training these on the decision trees that comprise the model, an ensemble model is built by majority vote. However, one of the main differences to other algorithms is that it chooses node cut-points (the bifurcation points' thresholds in a decision tree) fully at random; another significant difference is that the full training set is used, instead of a bootstrap replica, for each of the decision trees that comprise the ensemble model. This additional randomization is ideal in small datasets, in which overfitting is more likely to occur on the training set without a proper test evaluation of robustness. This method has proven to have successful results in solving other biological based problems [36,37]. After running all the methods in default scikit-learn [28] settings, we fine-tuned some parameters of the extremely randomized trees [35] with a grid search ("GridSearchCV", scikit-learn [28]), and the following parameters were updated: "n_estimators": 500; "bootstrap": True; and class_weight: "balanced_subsample". The full set of parameters can be consulted in Table A6, the parameters not referred were kept as default. Grid search was performed with 10-fold cross-validation.

4.7. Model Evaluation

To evaluate the models, we subjected both the training and the testing set to confusion matrix analysis. This table relates the actual and the predicted instances (sample) and compares them by their binary status of Negative (N) or Positive (P) in the prediction to their actual class of True (T) or False (F). It further relates these in four different possible combination states: True Negative (TN) is when the prediction is N and the actual is F; True Positive (TP) is when the prediction is P and the actual is T; False Negative (FN) is when the prediction is N and the actual is T; and False Positive (FP) is when the prediction is P and the actual is F.

The confusion matrix allows the calculation of several metrics, such as accuracy (Equation (1)); precision (Equation (2)); sensitivity, recall or True Positive Rate (recall, Equation (3)); False Positive Rate (FPR, Equation (4)); F1-score (Equation (5)); and Area Under the Receiver Operating Characteristic curve (AUROC, Equation (6)). All these metrics were used from the scikit-learn package ^{20}.

$$\mathbf{accuracy} = \frac{\mathbf{TP{+}TN}}{\mathbf{TP + TN + FP + FN}} \tag{1}$$

$$\mathbf{precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}} \tag{2}$$

$$\mathbf{recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}} \tag{3}$$

$$\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{FP + TN}} \tag{4}$$

$$\mathbf{F1 - score} = \frac{\mathbf{2 * (precision * recall)}}{\mathbf{(precision + recall)}} \tag{5}$$

$$\mathbf{AUROC} = \int_{\mathbf{x=0}}^{\mathbf{1}} \mathbf{TPR}\Big(\mathbf{FPR^{-1}(x)}\Big)\mathbf{dx} \tag{6}$$

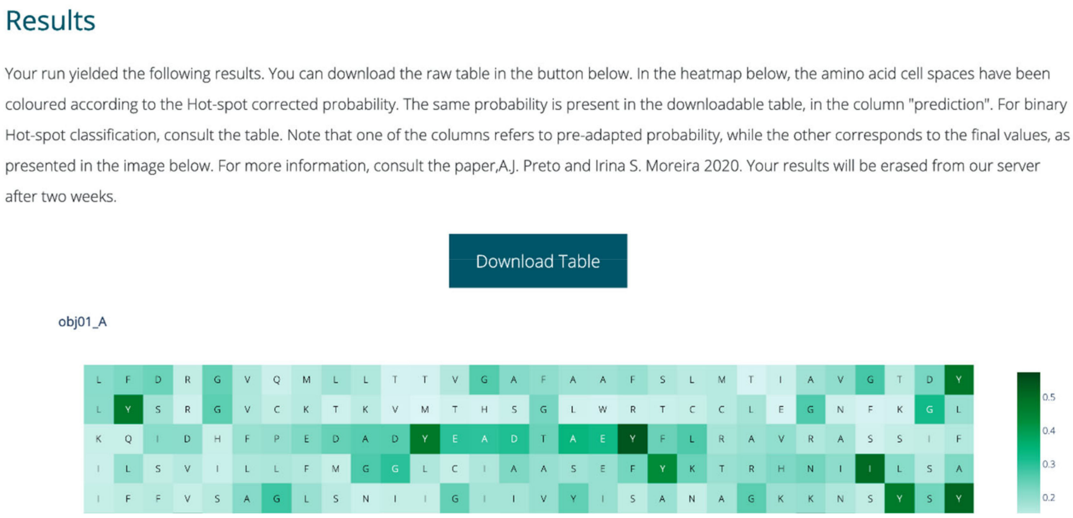
4.8. Prediction Probability Tuning

We performed further inspection of the HS/NS prediction by amino acid, in addition to the whole dataset, as can be seen in the "original" rows in Table A3. This inspection led us to notice that the HS/NS ratio had a significant toll in model performance. For example, TYR had a robust prediction of HS/NS; however, residues which had not such a balanced HS/NS ratio performed more poorly. Although this is a classification problem, most classification methods calculate class probability before yielding the predicted class, which is determined according to the higher probable class. As such, we examined the probability associated to the positive class (HS). Upon inspection of classification probabilities of the actual residues, it was noticed that, although not classified as HS, most of these amino acids still had a higher probability of HS classification than NS. The adaptation value displayed in Table A3 is the increase in probability of the HS class, added post-training, that allows higher HS probability amino acids to reach the HS class (above 50%). This value was implemented following the condition that it should not generate FP while increasing the amount of TP. As such, when, for each amino acid, the maximum false negative HS probability was higher than the maximum true negative HS probability, the HS probability (for that amino acid) was updated (Equation (7)). CYS, MET and ALA were not displayed in Table A3 due to their absence from the testing set.

$$\text{Correction factor} = 0.50 - \text{Maximum False Negative HS probability} \tag{7}$$

4.9. Webserver Implementation

The webserver was fully implemented with Python. Plotly [38] was used for dynamic graphical representations; scikit-learn [28] was used to perform user submission treatment, analysis and prediction; and in-house Python scripts were used to perform all feature extraction and intermediate steps. Flask was used for overall server set-up and visual layout construction [39]. The output each run includes a dynamic heatmap displaying the probability of HS, for each amino acid in the single or more chains submitted by the user. The full table with the classification probabilities as well as binary class before and after class probability tuning are also available for the user to download. A snapshot of the webserver output is displayed in Figure 3.



Results

Your run yielded the following results. You can download the raw table in the button below. In the heatmap below, the amino acid cell spaces have been coloured according to the Hot-spot corrected probability. The same probability is present in the downloadable table, in the column "prediction". For binary Hot-spot classification, consult the table. Note that one of the columns refers to pre-adapted probability, while the other corresponds to the final values, as presented in the image below. For more information, consult the paper,A.J. Preto and Irina S. Moreira 2020. Your results will be erased from our server after two weeks.

Download Table
obj01_A
L F D R G V Q M L L T T V G A F A A F S L M T I A V G T D Y
L Y S R G V C K T K V M T H S G L W R T C C L E G N F K G L
K Q I D H F P E D A D Y E A D T A E Y F L R A V R A S S I F
I L S V I L L F M G G L C I A A S E F Y K T R H N I I L S A
I F F V S A G L S N I I G I I V Y I S A N A G K K N S Y S Y
0.5
0.4
0.3
0.2

Figure 3. Sample of the output page of SPOTONE.

5. Conclusions

SPOTONE is a thorough prediction algorithm that tackles HS classification in a problem-tailored protocol. The pre-processing and ML steps can be the framework for further protein-based structural biology problems, as are innovating in several processes: (1) by highlighting the importance of protein-based overfitting versus amino acid based features; (2) by providing an answer with a set of simple, replicable, in-house features that make use of freely available information and amino acid position; (3) by considering the evaluation of the amino acid prediction capabilities instead of simply the target features at hand; (4) by attributing specific weights to amino acid types as a way to underline that these are not only features but also subsample spaces of the dataset; (5) by introducing a viable sequence-based HS predictor; and (6) by providing an intuitive and biologically relevant data interpretation tool (HS probability maps). Furthermore, SPOTONE as a webserver (http://moreiralab.com/resources/spotone) is easily usable by non-proficient researchers, with an intuitive framework.

Author Contributions: Data curation and research, A.J.P.; webserver assembling, A.J.P.; supervision, I.S.M.; writing, A.J.P. and I.S.M.; and fund raising, I.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by COMPETE 2020—Operational Programme for Competitiveness and Internationalisation and Portuguese national funds via FCT—Fundação para a Ciência e a Tecnologia, under projects POCI-01-0145-FEDER-031356 and UIDB/04539/2020. António Preto was supported by FCT through Ph.D. scholarship SFRH/BD/144966/2019. The authors also acknowledge STRATAGEM—New diagnostic and therapeutic tools against multidrug-resistant tumors, CA17104.

Acknowledgments: A. J. Preto acknowledges José G. Almeida for thorough discussions on the proper approach towards the problem at hands that yielded fruitful protocol improvements.

Conflicts of Interest: The authors declare no conflict of interest.

Data and Code Availability: All data and code used to perform the described experiences are available at https://github.com/MoreiraLAB/spotone.

Appendix A

Table A1. Performance metrics (training and testing datasets) for the three studied subsets: with only the in-house features (one-hot encoding, relative position, literature based and window-based features), using only PSSM features and the joint dataset with both in-house and PSSM features.

Dataset	Classifier Name	Subset	Accuracy	AUC	Precision	Recall	F1-Score
In-house features	Extremely Randomized Trees	Train	0.99	1.00	0.99	0.99	0.99
		Test	0.81	0.77	0.88	0.81	0.83
	Neural Network	Train	0.81	0.73	0.81	0.81	0.81
		Test	0.69	0.56	0.72	0.69	0.71
	AdaBoost	Train	0.98	0.98	0.98	0.98	0.98
		Test	0.71	0.56	0.77	0.71	0.74
	Support Vector Machine	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86
PSSM features	Extremely Randomized Trees	Train	0.96	0.98	0.97	0.96	0.97
		Test	0.72	0.55	0.82	0.72	0.76
	Neural Network	Train	0.96	0.97	0.96	0.96	0.96
		Test	0.70	0.57	0.74	0.70	0.72
	AdaBoost	Train	0.91	0.92	0.93	0.91	0.91
		Test	0.73	0.60	0.79	0.73	0.75
	Support Vector Machine	Train	0.80	0.86	0.96	0.8	0.86
		Test	0.76	0.64	0.92	0.76	0.82
In-house + PSSM	Extremely Randomized Trees	Train	1.00	1.00	1.00	1.00	1.00
		Test	0.83	0.86	0.93	0.83	0.86
	Neural Network	Train	0.83	0.78	0.85	0.83	0.82
		Test	0.56	0.50	0.52	0.56	0.53
	AdaBoost	Train	0.98	0.98	0.98	0.98	0.98
		Test	0.72	0.60	0.74	0.72	0.73
	Support Vector Machine	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86

Table A2. Performance metrics for the subsets: The joint dataset with both in-house (one-hot encoding, relative position, literature based and window-based features) and iFeature features (full sequence features); the dataset with in-house, PSSM and iFeature features; and the dataset with only iFeatures.

Dataset	Classifier Name	Subset	Accuracy	AUC	Precision	Recall	F1-Score
In-house + iFeatures	Extremely Randomized Trees	Train	1.00	1.00	1.00	1.00	1.00
		Test	0.83	0.77	0.85	0.83	0.84
	Neural Network	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86
	AdaBoost	Train	0.99	0.99	0.99	0.99	0.99
		Test	0.81	0.75	0.83	0.81	0.82
	Support Vector Machine	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86
In-house + PSSM + iFeatures	Extremely Randomized Trees	Train	1.00	1.00	1.00	1.00	1.00
		Test	0.83	0.77	0.84	0.83	0.83
	Neural Network	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86
	AdaBoost	Train	0.99	0.99	0.99	0.99	0.99
		Test	0.77	0.69	0.79	0.77	0.78
	Support Vector Machine	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86
iFeatures	Extremely Randomized Trees	Train	0.83	0.80	0.90	0.83	0.85
		Test	0.77	0.67	0.82	0.77	0.79
	Neural Network	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86
	AdaBoost	Train	0.83	0.76	0.86	0.83	0.84
		Test	0.79	0.72	0.80	0.79	0.79
	Support Vector Machine	Train	0.77	0.00	1.00	0.77	0.87
		Test	0.76	0.00	1.00	0.76	0.86

Table A3. Extremely randomized trees algorithm scores, by amino acid, in the testing set.

Amino Acid		Adaptation Value	Accuracy	Precision	Recall	Amount Used for Threshold Adaptation
ASP	Original	-	0.71	0.00	1.00	11
	Adapted		-	-	-	
SER	Original	-	1.00	0.00	0.00	4
	Adapted		-	-	-	
GLN	Original	-	0.67	0.00	0.00	6
	Adapted		-	-	-	
LYS	Original	-	1.00	1.00	1.00	12
	Adapted		-	-	-	
ILE	Original	+0.15	0.80	0.00	0.00	5
	Adapted		1.00	1.00	1.00	
PRO	Original	+0.15	0.50	0.00	0.00	2
	Adapted		1.00	1.00	1.00	

Table A3. *Cont.*

Amino Acid		Adaptation Value	Accuracy	Precision	Recall	Amount Used for Threshold Adaptation
THR	Original	-	1.00	1.00	1.00	8
	Adapted		-	-	-	
PHE	Original	+0.25	0.75	0.00	0.00	4
	Adapted		1.00	1.00	1.00	
ASN	Original	+0.15	0.50	0.00	0.00	6
	Adapted		0.83	0.67	1.00	
GLY	Original	-	1.00	0.00	0.00	1
	Adapted		-	-	-	
HIS	Original	-	0.80	0.00	0.00	5
	Adapted		-	-	-	
LEU	Original	+0.06	0.50	0.00	0.00	4
	Adapted		1.00	1.00	1.00	
ARG	Original	-	1.00	0.00	0.00	9
	Adapted		-	-	-	
TRP	Original	-	0.71	0.00	0.00	7
	Adapted		-	-	-	
VAL	Original	+0.25	0.67	0.00	0.00	3
	Adapted		0.67	0.00	0.00	
GLU	Original	-	0.85	0.00	0.00	13
	Adapted		-	-	-	
TYR	Original	-	0.55	0.33	0.67	11
	Adapted		-	-	-	

Table A4. Literature-based amino acid features, such as secondary structure propensity, pKa associated values, number of atoms of each type and standard area and mass associated values, attained from BioMagResBank [31].

Amino Acid	Helix Propensity	Sheet Propensity	Helix Propensity Values	Sheet Propensity Values	Molecular Weight	pKa Carboxylate	pKa Amine	pKa Side Chain	Number of Carbons
ALA	1	1	1.45	0.97	89.09	2.30	9.90	0.00	3
CYS	2	2	0.77	1.30	121.16	1.80	10.80	8.65	3
ASP	2	3	0.98	0.80	133.10	2.00	10.00	4.04	4
GLU	1	4	1.53	0.26	147.13	2.20	9.70	4.39	5
PHE	3	2	1.12	1.28	165.19	1.80	9.10	0.00	9
GLY	4	3	0.53	0.81	75.07	2.40	9.80	0.00	2
HIS	3	5	1.24	0.71	155.16	1.80	9.20	6.75	6
ILE	5	6	1.00	1.60	131.17	2.40	9.70	0.00	6
LYS	5	5	1.07	0.74	146.19	2.20	9.20	11.00	6
LEU	1	2	1.34	1.22	131.17	2.40	9.60	0.00	6
MET	3	6	1.20	1.67	149.21	2.30	9.20	0.00	5
ASN	6	5	0.73	0.65	132.12	2.00	8.80	0.00	4
PRO	4	5	0.59	0.62	115.13	2.00	10.60	0.00	5
GLN	3	2	1.17	1.23	146.15	2.20	9.10	0.00	5
ARG	2	3	0.79	0.90	174.20	1.80	9.00	12.50	6
SER	2	5	0.79	0.72	105.09	2.10	9.20	0.00	3
THR	2	2	0.82	1.20	119.12	2.60	10.40	0.00	4
VAL	3	6	1.14	1.65	117.15	2.30	9.60	0.00	5
TRP	3	2	1.14	1.19	204.22	2.40	9.40	0.00	11
TYR	6	2	0.61	1.29	181.19	2.20	9.10	9.75	9

Table A5. Literature-based amino acid features, such as secondary structure propensity, pKa associated values, number of atoms of each type and standard area and mass associated values, attained from BioMagResBank [31].

Amino Acid	Number of Hydrogens	Number of Nitrogen Atoms	Number of Oxygens	Number of Sulphur	Standard Free Area	Protein Standard Area	Folded Buried Area	Mean Fractional Area	Residue Mass	Monoisotopic Mass
ALA	7	1	2	0	118.10	31.50	86.60	0.74	71.08	71.04
CYS	7	1	2	1	146.10	13.90	132.30	0.91	103.14	103.01
ASP	7	1	4	0	158.70	60.90	97.80	0.62	115.09	115.03
GLU	9	1	4	0	186.20	72.30	113.90	0.62	129.12	129.04
PHE	11	1	2	0	222.80	28.70	194.10	0.88	147.18	147.07
GLY	5	1	2	0	88.10	25.20	62.90	0.72	57.05	57.02
HIS	9	3	2	0	202.50	46.70	155.80	0.78	137.14	137.06
ILE	13	1	2	0	181.00	23.00	158.00	0.88	113.16	113.08
LYS	14	2	2	0	225.80	110.30	115.50	0.52	128.17	128.10
LEU	13	1	2	0	193.10	29.00	164.10	0.85	113.16	113.08
MET	11	1	2	1	203.40	30.50	172.90	0.85	131.19	131.04
ASN	8	2	3	0	165.50	62.20	103.30	0.63	114.10	114.04
PRO	9	1	2	0	146.80	53.70	92.90	0.64	97.12	97.05
GLN	10	2	3	0	193.20	74.00	119.20	0.62	128.13	128.06
ARG	14	4	2	0	256.00	93.80	162.20	0.64	156.19	156.10
SER	7	1	3	0	129.80	44.20	85.60	0.66	87.08	87.03
THR	9	1	3	0	152.50	46.00	106.50	0.70	101.11	101.05
VAL	11	1	2	0	164.50	23.50	141.00	0.86	99.13	99.07
TRP	12	2	2	0	266.30	41.70	224.60	0.85	186.21	186.08
TYR	11	1	3	0	236.80	59.10	177.70	0.76	163.18	163.06

Table A6. Extreme randomized trees parameters tested in the Grid Search.

Parameter	Default Value	Tested Values
n_estimators	100	(50,100,250,500,1000)
boostrap	False	(True, False)
class_weight	None	(None,"balanced_subsample","balanced")
criterion	"gini"	("gini","entropy")
max_depth	None	(None,1,2,3)
min_samples_split	2	(2,3,4,5)
min_samples_leaf	1	(1,2,3)
max_leaf_nodes	None	(None,1,2,3)
max_samples	None	(None,1,2,5,10)
max_features	"auto"	("auto","sqrt","log2")
min_impurity_decrease	0.0	(0.0, 0.01, 0.001)
min_weight_fraction_leaf	0.0	(0.0, 0.01, 0.001)

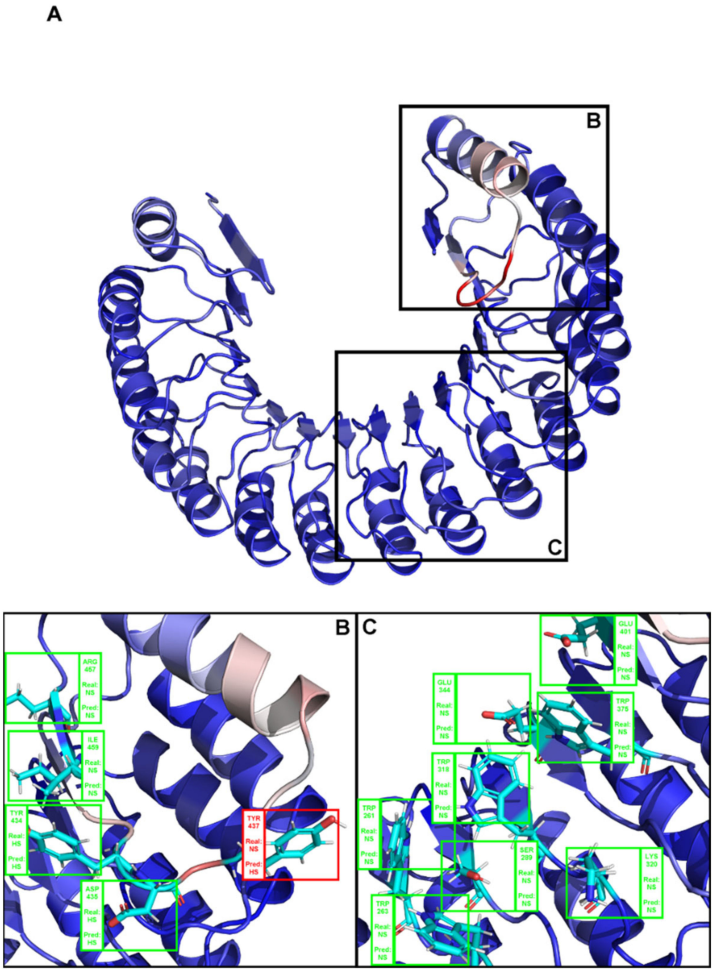

A
B
C
B
ARG 457 Real: NS Pred: NS
ILE 459 Real: NS Pred: NS
TYR 434 Real: HS Pred: HS
ASP 435 Real: HS Pred: HS
TYR 437 Real: NS Pred: HS
C
GLU 401 Real: NS Pred: NS
GLU 344 Real: NS Pred: NS
TRP 375 Real: NS Pred: NS
TRP 318 Real: NS Pred: NS
TRP 261 Real: NS Pred: NS
SER 289 Real: NS Pred: NS
LYS 320 Real: NS Pred: NS
TRP 263 Real: NS Pred: NS

Figure A1. (**A**) Structural representation of the complex between angiogenin and a ribonuclease inhibitor: PDB ID 1a4y. Brighter red colors were attributed to residues with a higher probability of being classified as HS. (**B**,**C**) Close-ins of all interfacial residues for which there is an experimental $\Delta\Delta G_{binding}$ value, and as such a HS/NS classification. Green boxes represent correctly predicted residues, whereas red boxes represent incorrectly classified residues.

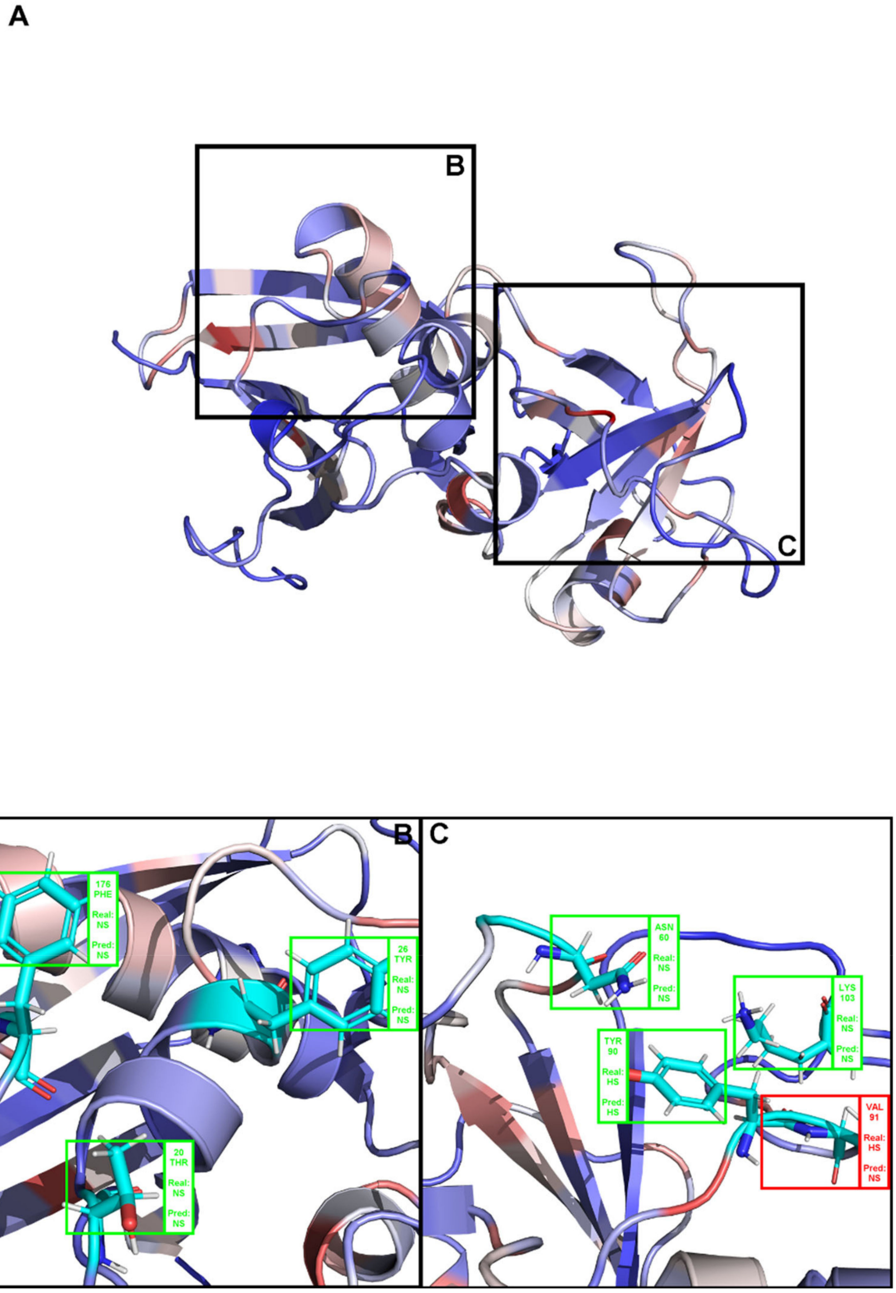


Figure A2. (**A**). Depiction the complex between a T-Cell receptor beta chain and SEC3 superantigen: PDB ID 1jck. Brighter red colors were attributed to residues with a higher probability of being classified as HS. (**B**,**C**) Close-ins of all interfacial residues for which there is an experimental $\Delta\Delta G_{binding}$ value, and as such a HS/NS classification. Green boxes represent correctly predicted residues, whereas red boxes represent incorrectly classified residues.

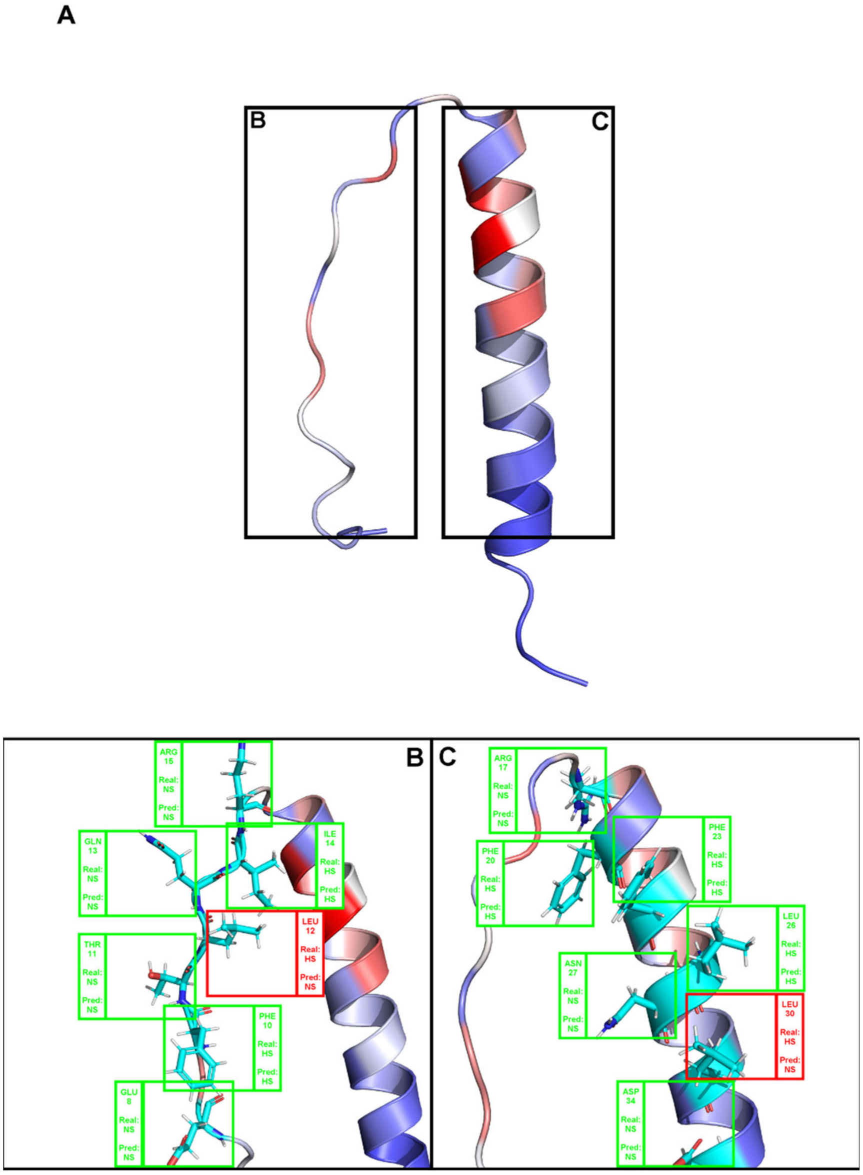


Figure A3. (**A**). Depiction of chain C of the complex PDB ID 3sak. Brighter red colors were attributed to residues with a higher probability of being classified as HS. (**B**,**C**) Close-ins of all interfacial residues for which there is an experimental $\Delta\Delta G_{binding}$ value, and as such a HS/NS classification. Green boxes represent correctly predicted residues, whereas red boxes represent incorrectly classified residues.

References

1. Moreira, I.S.; Fernandes, P.A.; Ramos, M.J. Hot Spots—A Review of the Protein-Protein Interface Determinant Amino-Acid Residues. *Proteins Struct. Funct. Genet.* **2007**, *68*, 803–812. [CrossRef] [PubMed]
2. Bogan, A.A.; Thorn, K.S. Anatomy of hot spots in protein interfaces. *J. Mol. Biol.* **1998**, *280*, 1–9. [CrossRef] [PubMed]
3. Keskin, O.; Ma, B.; Nussinov, R. Hot Regions in Protein-Protein Interactions: The Organization and Contribution of Structurally Conserved Hot Spot Residues. *J. Mol. Biol.* **2005**, *345*, 1281–1294. [CrossRef] [PubMed]
4. Jiang, J.; Wang, N.; Chen, P.; Zheng, C.; Wang, B. Prediction of Protein Hotspots from Whole Protein Sequences by a Random Projection Ensemble System. *Int. J. Mol. Sci.* **2017**, *18*, 1543. [CrossRef] [PubMed]
5. Qiao, Y.; Xiong, Y.; Gao, H.; Zhu, X.; Chen, P. Protein-protein interface hot spots prediction based on hybrid feature selection strategy. *BMC Bioinform.* **2018**, *19*. [CrossRef]

6. Clackson, T.; Wells, J.A. A hot spot of binding energy in a hormone-receptor interface. *Science* **1995**, *267*, 383–386. [CrossRef]
7. Golden, M.S.; Cote, S.M.; Sayeg, M.; Zerbe, B.S.; Villar, E.A.; Beglov, D.; Sazinsky, S.L.; Georgiadis, R.M.; Vajda, S.; Kozakov, D.; et al. Comprehensive Experimental and Computational Analysis of Binding Energy Hot Spots at the NF-KB Essential Modulator/IKKβ Protein-Protein Interface. *J. Am. Chem. Soc.* **2013**, *135*, 6242–6256. [CrossRef]
8. Ciglia, E.; Vergin, J.; Reimann, S.; Smits, S.H.J.; Schmitt, L.; Groth, G.; Gohlke, H. Resolving Hot Spots in the C-Terminal Dimerization Domain That Determine the Stability of the Molecular Chaperone Hsp90. *PLoS ONE* **2014**, *9*, e96031. [CrossRef]
9. Salo-Ahen, O.M.H.; Tochowicz, A.; Pozzi, C.; Cardinale, D.; Ferrari, S.; Boum, Y.; Mangani, S.; Stroud, R.M.; Saxena, P.; Myllykallio, H.; et al. Hotspots in an Obligate Homodimeric Anticancer Target. Structural and Functional Effects of Interfacial Mutations in Human Thymidylate Synthase. *J. Med. Chem.* **2015**, *58*, 3572–3581. [CrossRef]
10. Moreira, I.S. The Role of Water Occlusion for the Definition of a Protein Binding Hot-Spot. *Curr. Top. Med. Chem.* **2015**, *15*, 2068–2079. [CrossRef]
11. Ramos, R.M.; Fernandes, L.F.; Moreira, I.S. Extending the applicability of the O-ring theory to protein-DNA complexes. *Comput. Biol. Chem.* **2013**, *44*, 31–39. [CrossRef] [PubMed]
12. Thorn, K.S.; Bogan, A.A. ASEdb: A Database of Alanine Mutations and Their Effects on the Free Energy of Binding in Protein Interactions. *Bioinformatics* **2001**, *17*, 284–285. [CrossRef] [PubMed]
13. Fischer, T.B.; Arunachalam, K.V.; Bailey, D.; Mangual, V.; Bakhru, S.; Russo, R.; Huang, D.; Paczkowski, M.; Lalchandani, V.; Ramachandra, C.; et al. The Binding Interface Database (BID): A Compilation of Amino Acid Hot Spots in Protein Interfaces. *Bioinformatics* **2003**, *19*, 1453–1454. [CrossRef] [PubMed]
14. Kumar, M.D.S.; Gromiha, M.M. PINT: Protein-Protein Interactions Thermodynamic Database. *Nucleic Acids Res.* **2006**, *34*, D195–D198. [CrossRef]
15. Moal, I.H.; Fernandez-Recio, J. SKEMPI: A Structural Kinetic and Energetic Database of Mutant Protein Interactions and Its Use in Empirical Models. *Bioinformatics* **2012**, *28*, 2600–2607. [CrossRef]
16. Jankauskaite, J.; Jiménez-García, B.; Dapkunas, J.; Fernandéz-Recio, J.; Moal, I.H. SKEMPI 2.0: And updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* **2019**, *35*, 462–469. [CrossRef]
17. Cukuroglu, E.; Engin, H.B.; Gursoy, A.; Keskin, O. Hot Spots in Protein–Protein Interfaces: Towards Drug Discovery. *Prog. Biophys. Mol. Biol.* **2014**, *116*, 165–173. [CrossRef]
18. Morrow, J.K.; Zhang, S. Computational Prediction of Protein Hot Spot Residues. *Curr. Pharm. Des.* **2012**, *18*, 1255–1265. [CrossRef]
19. Nguyen, Q.; Fablet, R.; Pastor, D. Protein Interaction Hotspot Identification Using Sequence-Based Frequency-Derived Features. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 2993–3002. [CrossRef]
20. Hu, S.-S.; Chen, P.; Wang, B.; Li, J. Protein Binding Hot Spots Prediction from Sequence Only by a New Ensemble Learning Method. *Amino Acids* **2017**, *49*, 1773–1785. [CrossRef]
21. Liu, Q.; Chen, P.; Wang, B.; Zhang, J.; Li, J. Hot Spot Prediction in Protein-Protein Interactions by an Ensemble System. *BMC Syst. Biol.* **2018**, *12* (Suppl. 9), 132. [CrossRef] [PubMed]
22. Moreira, I.S.; Koukos, P.I.; Melo, R.; Almeida, J.G.; Preto, A.J.; Schaarschmidt, J.; Trellet, M.; Gümüş, Z.H.; Costa, J.; Bonvin, A.M.J.J. SpotOn: High Accuracy Identification of Protein-Protein Interface Hot-Spots. *Sci. Rep.* **2017**. [CrossRef] [PubMed]
23. Martins, J.M.; Ramos, R.M.; Pimenta, A.C.; Moreira, I.S. Solvent-Accessible Surface Area: How Well Can Be Applied to Hot-Spot Detection? *Proteins Struct. Funct. Bioinforma.* **2014**, *82*. [CrossRef] [PubMed]
24. Zhu, X.; Mitchell, J.C. KFC2: A Knowledge-Based Hot Spot Prediction Method Based on Interface Solvation, Atomic Density, and Plasticity Features. *Proteins* **2011**, *79*, 2671–2683. [CrossRef]
25. Tuncbag, N.; Keskin, O.; Gursoy, A. HotPoint: Hot Spot Prediction Server for Protein Interfaces. *Nucleic Acids Res.* **2010**, *38*, W402–W406. [CrossRef] [PubMed]
26. Berman, H.M.; Battistuz, T.; Bhat, T.N.; Bluhm, W.F.; Bourne, P.E.; Burkhardt, K.; Feng, Z.; Gilliland, G.L.; Iype, L.; Jain, S.; et al. The Protein Data Bank. *Acta Cryst. Sect. D Biol. Cryst.* **2002**, *28*, 235–242. [CrossRef]
27. The UniProt Consortium. UniProt: The Universal Protein Knowledgebase. *Nucleic Acids Res.* **2017**, *45*, D158–D169. [CrossRef]

28. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
29. Chen, Z.; Zhao, P.; Li, F.; Leier, A.; Marquez-Lago, T.T.; Wang, Y.; Webb, G.I.; Smith, A.I.; Daly, R.J.; Chou, K.-C.; et al. IFeature: A Python Package and Web Server for Features Extraction and Selection from Protein and Peptide Sequences. *Bioinformatics* **2018**, *34*, 2499–2502. [CrossRef]
30. Madeira, F.; Park, Y.M.; Lee, J.; Buso, N.; Gur, T.; Madhusoodanan, N.; Basutkar, P.; Tivey, A.R.N.; Potter, S.C.; Finn, R.D.; et al. The EMBL-EBI Search and Sequence Analysis Tools APIs in 2019. *Nucleic Acids Res.* **2019**, *47*, W636–W641. [CrossRef]
31. Ulrich, E.L.; Akutsu, H.; Doreleijers, J.F.; Harano, Y.; Ioannidis, Y.E.; Lin, J.; Livny, M.; Mading, S.; Maziuk, D.; Miller, Z.; et al. BioMagResBank. *Nucleic Acids Res.* **2008**, *36* (Suppl. 1), D402–D408. [CrossRef] [PubMed]
32. Hinton, G.E. Connectionist Learning Procedures. *Artif. Intell.* **1989**, *40*, 185–234. [CrossRef]
33. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
34. Wu, T.-F.; Lin, C.-J.; Weng, R.C. Probability Estimates for Multi-Class Classification by Pairwise Coupling. *J. Mach. Learn. Res.* **2004**, *5*, 975–1005.
35. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
36. Basith, S.; Manavalan, B.; Shin, T.H.; Lee, G. iGHBP: Computational identification of growth hormone binding proteins from sequences using extremely randomised tree. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 412–420. [CrossRef]
37. Manavalan, B.; Basith, S.; Shin, T.H.; Wei, L.; Lee, G. AtbPpred: A Robust Sequence-Based Prediction of Anti-Tubercular Peptides Using Extremely Randomized Trees. *Comput. Struct. Biotechnol. J.* **2019**, *17*, 972–981. [CrossRef]
38. Plotly Technologies Inc. *Collaborative Data Science*; Plotly Technologies Inc.: Montreal, QC, Canada, 2015.
39. Grinberg, M. *Flask Web Development: Developing Web Applications with Python*, 1st ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2014.

3.1.3. Membrane protein dimer characterization

Chapter 21

Structural Characterization of Membrane Protein Dimers

António J. Preto, Pedro Matos-Filipe, Panagiotis I. Koukos, Pedro Renault, Sérgio F. Sousa, and Irina S. Moreira

Abstract

Membrane proteins are essential vessels for cell communication both with other cells and noncellular structures. They modulate environment responses and mediate a myriad of biological processes. Dimerization and multimerization processes have been shown to further increase the already high specificity of these processes. Due to their central role in various cell and organism functions, these multimers are often associated with health conditions, such as Alzheimer's disease (AD), Parkinson's disease (PD), and diabetes, among others.

Understanding the membrane protein dimers' interface takes advantage of the specificity of the structure, for which we must pinpoint the most relevant interfacial residues, since they are extremely likely to be crucial for complex formation. Here, we describe step by step our own in silico protocol to characterize these residues, making use of known experimental structures. We detail the computational pipeline from data acquisition and pre-processing to feature extraction. A molecular dynamics simulation protocol to further study membrane dimer proteins and their interfaces is also illustrated.

Key words Membrane protein dimers, Machine learning, Feature extraction, Interfacial residues, Protein-protein interaction, Molecular dynamics

1 Introduction

Membranes are essential structures for life assuming many functions within cells, such as mobility and nutrient intake. To add to these, energy transduction, biosynthesis, and immunologic and nerve response are displayed in higher organisms [1]. These actions are often controlled by membrane proteins (MPs), which play essential roles such as ion and nutrient transport, communication with the extracellular environment, and signal transduction [2]. These proteins are also ubiquitous: 20–30% of genes of most organisms code for MPs [3]. Understanding these cellular

António J. Preto and Pedro Matos-Filipe contributed equally with all other contributors.

Alexander E. Kister (ed.), *Protein Supersecondary Structures: Methods and Protocols*, Methods in Molecular Biology, vol. 1958, https://doi.org/10.1007/978-1-4939-9161-7_21, © Springer Science+Business Media, LLC, part of Springer Nature 2019

functions requires detailed knowledge of MPs' 3D structures and interactions. MPs frequently assemble as dimers or even higher-order oligomers. These higher-order assemblies can have specific roles that not necessarily coincide with those of their monomeric constituents [4, 5]. This makes the structural biology of MPs even more complex and demands the development of new experimental and theoretical methods to elucidate their function.

Experimental characterization of MPs is difficult as the membrane imposes obstacles to its manipulation, notably its purification and crystallization. As such MP structural studies can greatly benefit from in silico tools, since they provide useful approaches that complement, make use of, and add to the experimental results. In spite of the difficulties mentioned above, progress in experimental techniques has generated a growing body of structural information. For instance, the *mpstruc*—Membrane Proteins of Known 3D Structure—database from the Stephen White Laboratory at UC Irvine (available at http://blanco.biomol.uci.edu/mpstruc/) [6] now lists 817 unique membrane proteins whose 3D structures are known (as of August 30, 2018).

Dimers or higher-order assemblies of MPs are often the subject of computational studies [7]. These typically aim at predicting protein-protein interactions (PPIs) or hot spots (HS), interfacial residues that upon alanine mutation generate a binding free energy difference of 2.0 kcal/mol. We have recently developed a web server, SpotOn, for the prediction of HS in a soluble complex [8, 9]. However, we are still lacking reliable computational approaches that target the understanding of multimeric MPs. Here, we describe in detail our protocol to analyze a variety of biological and physic-chemical characteristics of interfacial interactions within MPs. By following this protocol, the reader has access to a comprehensive set of tools that target the understanding of MP dimerization and that can be used to construct any possible database regarding these biological systems. In the interest of readers, we also revise and explain the basics of machine learning (ML) and molecular dynamics (MD) techniques, which could potentially be used to further describe the MP interfacial residues. This tool contributes to further understanding the interaction of MP complexes and should be a valuable addition to the repertoire of methods/tools that aim to elucidate MP structure and function.

2 Materials

The goal of this chapter is to introduce a variety of tools to help readers characterize interfacial residues between two transmembrane monomers of a MP system. To achieve this, we built a pipeline of different scripts and tools able to process protein databank files (.PDB) containing the two monomeric chains. We also provide tools to retrieve a large amount of key features. In the end, the users can apply ML to this database to attain a predictive algorithm or MD if their main interest is to depict the mechanism of a particular system.

2.1 Machine Learning

ML has been defined in several different ways, yet there is a common concept of ML as the science of "getting computers to act without being explicitly told how to do so." This means that ML is appropriate to solve real life problems in which there is no tool to deduct an answer as ML focuses on learning from experience. Usually, this means that the predictions from a ML model are not absolute; they improve when gathering more data [10]. Recently, many fields have experienced an increase in the accessible data. In particular, in the realm of biological problems where scarce data is many times a big obstacle, this was also true [11].

When referring to data, we also use the term instances, the available "samples" that we can feed to a predictor. Each of these instances is associated to the characteristic we want to predict and to the descriptors (features) that are associated or can be extracted from it. Furthermore, the features are components of the dataset instances that can be used to predict the target characteristic.

2.1.1 Supervised and Unsupervised Learning

ML is typically divided in two subfields, depending on its relation to the data: supervised and unsupervised learning (although there can also be the concept of semi-supervised learning, which can make use of both the previous approaches). This partition implies that the methods used to construct the prediction models are usually distinct for each different type of learning [12]. Supervised learning is the case in which the data fed to the prediction model is constituted of both input and output information. This means that every instance has a label. The labels inform the prediction model of the possible outputs, since they are the known values of the target prediction. A supervised learning model will make use of the labeled instances to predict cases in which the entries have unknown output values. The input information is constituted by all the features that characterize the instance, not including the output information. The output information on the data of a supervised learning model can lead to classification or regression models. A classification model is generated when the output is limited to a discrete number of possible values (classes). When there are only two possible classes, the problem is referred to as a binary problem. Regression models allow an infinite number of possible values. Unsupervised learning models do not have associated output values. This means that it is impossible to label an unknown new entry according to the starting data. However, these models are useful to identify patterns, since they can group instances according to their input information (features).

2.1.2 Dataset Construction: Instances

From a general point of view, the construction of a dataset is firstly conducted by gathering instances. Overall, instances are every entry that can be characterized and constitute a data point on a ML deployment pipeline. Gathering more data points will yield more information for the model to learn from. Usually, a dataset with more data points leads to stronger and more generalized models than its smaller counterparts. Regarding the type of data, instances can be many things, as long as they are able to be standardized along each other and can yield a pattern that relates towards the target prediction. In the case of classification models, it is preferable if the number of instances for each class is similar. This sometimes requires that the dataset is balanced to equilibrate the number of instances in each class. There are several sample (instances) selection processes such as up-sampling (artificially augmenting the lower populated classes) and down-sampling (lowering number of instances in the overpopulated classes). Another possibility is filtering out the irrelevant instances. For all these processes, there are well-developed mathematical approaches that are available in most ML-centered software [13].

2.1.3 Dataset Construction: Features

The number and quality of instances are certainly determinants for the quality of the upcoming predictions. What is associated or generated from those instances, however, can be equally important. The descriptors that we associate with instances are called features. Features are all the characteristics that can be associated to a data point. These features need to be relevant for the output prediction and be independent among each other. If this relationship is missing, the features can introduce biases, noise or overall weakening of the prediction capability of the model. What makes a feature relevant, however, is not always straightforward. Although there are approaches that can test the dataset for the most relevant features, the scientific/technical knowledge on the dataset is certainly an important factor in the selection and analysis of features. Freely available data from databases or, in some cases, data collected by the researchers, does not always comprise all the necessary information to generate strong models. For example, sometimes the problem of missing values must be addressed; in some cases, several approaches artificially generate values where they are not available. Nevertheless, it is always preferable to first mine alternative data sources that can yield the corresponding values. Feature extraction is the process in which, for the original raw data or instance data points, alternative features are generated to better describe the entries. Feature extraction is highly dependent on the type of data under focus.

2.1.4 Splitting the Dataset

"Generalized" is a term that has been mentioned several times to address the quality of a model. A ML model is said to be generalized if it can give accurate predictions about upcoming, unknown, instances. Indeed, a classification model may appear to be highly reliable and yield good accuracy, but when faced with new information, it can output unreliable predictions due to its bias towards the input data. In order to overcome these issues, the data should be split into training and test sets. The training set should then be used to train the model, while the test set should not be present in the learning phase. This is usually performed several times, in a process called cross-validation (CV). When deploying CV, the computer is not informed of the specific instances that will be used for the training and test sets. Rather, it is told to split the dataset into two sets with a given percentage each time (commonly 70–30%), performing the training on the larger dataset and testing the model on the small test set for each case [14]. CV is usually performed several times for each run. Each time, the data undergoes randomized resampling, which leads to different training and test sets, in order to achieve an unbiased and generalized model [15].

2.1.5 Predictive Model Deployment

The application of predictive models on the dataset is probably the step for which ML is more commonly identified. A ML model engulfs a predictive approach that makes use of a mathematical or logical model to predict an outcome with a given degree of certainty. The specific model, however, differs for classification, regression, or clustering (unsupervised). Although some approaches are displaying consistently positive results for a wide array of problems, such as deep learning [16], there is not a perfect model to fit all possible problems. A thorough knowledge on the models and the data is the best way to maximize the use of both on the construction of a good predictor. Furthermore, there are approaches that allow the combination of several models, which are referred to as ensemble models. Ensemble models have been displaying competitive results in comparison to single complex models, even if sometimes the models that make up ensemble models are themselves simple. Such is the case of our own SpotOn predictor [8, 9].

2.1.6 Model Evaluation

The final evaluation of the models is one of the most important steps, since it can lead to the drawback of the process until the very start. Evaluating a ML model means assessing its validity upon unknown outcomes. There are many available metrics, but most supervised learning approaches rely on the relation between the predicted outcome and the actual outcome. This ratio is yielded from the test and validation sets when in comparison with the outcome predicted by the trained model. In the case of classification models, several common metrics derive from a confusion matrix (Table 1). Sensitivity (Eq. 1), specificity (Eq. 2), precision (Eq. 3), negative predictive value (NPV, Eq. 4), and F1-score (Eq. 7) are calculated directly from the values attained from the

Table 1
Confusion matrix

	Predicted: no	Predicted: yes
Actual: no	True negative (TN)	False positive (FP)
Actual: yes	False negative (FN)	True positive (TP)

confusion matrix. The false discovery rate (FDR, Eq. 5), although it can be calculated independently, can also be seen as the inverse of precision. Similarly, the false-negative rate (FNR, Eq. 6) is the inverse of sensitivity. The area under the receiver operating characteristic curve (AUROC, Eq. 8) depends on the true-positive rate (TPR, Eq. 1) and the false discovery rate (FDR, Eq. 5). By including different metrics on all the evaluated set of data points, AUROC constitutes a good metric for binary classification models [17]. All this can be attained by computing a confusion matrix (Table 1). The equations listed below (1)–(8) are all dependent on these values and can be used to address the particularities of a dataset.

Sensitivity formula

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1}$$

Specificity formula

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \tag{2}$$

Precision formula

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

Negative predictive value

$$\mathrm{NPV} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{4}$$

False discovery rate

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV} \tag{5}$$

False-negative rate

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR} \tag{6}$$

F1-score

$$\mathrm{F1\text{-}score} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{7}$$

AUROC

$$\text{AUROC} = \int_{-\infty}^{\infty} \text{TPR}(T)\text{FDR}'(T)dT \tag{8}$$

2.2 Molecular Dynamics Simulations

Molecular dynamics (MD) simulations are now a standard tool in the study of biomolecules. Following the publication of the first study describing an MD simulation of a protein (the bovine pancreatic trypsin inhibitor) already 40 years ago [18], this field has been one of the strong and enthusiastic developments, taking advantage of the astonishing computational progress that has characterized the past decades and of the good parallelization efficiency of most modern MD algorithms and more recently of the efficient use of GPUs. The dynamic properties of a protein have a profound effect upon its functional behavior. This is even more important when dealing with protein-protein interfaces. MD simulations allow the study of the dynamic properties of a system. They enable the complex and dynamic processes that take place in biological systems to be analyzed and provide atomistic detail concerning the individual particle motion as a function of time. Typical examples of application include the study of phenomena such as protein stability, molecular recognition, conformational changes, protein folding, and ion transport in biological systems.

MD methods normally used when dealing with such systems are based on the classic equations of motion, which are at the cornerstone molecular mechanics (MM). MM methods represent the energy of a system as a parametric function of the nuclear coordinates. These methods neglect both electrons and the quantum aspects of the nuclear motion and are based on classical Newtonian mechanics. They typically consider a rather simplified scheme of the interactions within a system. A "ball and spring" model is usually employed, in which the atoms are described as charged spheres of different sizes, whereas the bonds are described as springs with different degrees of stiffness. The van der Waals interactions are also an important step in the interaction between the modeled biomolecules and the solvent [19]. The neglect of the concept of electron forecloses any direct study of processes involving the formation or breaking of chemical bonds.

The energy of the system is split into a sum of contributions from different processes, including the stretching of bonds, the opening and closing of angles, rotations around simple bonds, etc. Each of these contributing processes is described by an individual expression, parameterized for a given set of standard atom types. Hence, a MM method is characterized not only by its functional form but also by the corresponding parameters, the two of which form a single entity termed force field. The parameters involved are typically derived from experimental data and/or from calculations with higher-level methods (e.g., density functional theory (DFT)) for small molecules. The accuracy of the parameterization protocol is of paramount importance to the reliability of the force field, and special care should be taken when calculating properties other than those included in the parameterization process.

2.2.1 Biomolecular Force Fields

When preparing an MD simulation, one of the most critical choices to be made is the selection of a force field. As previously mentioned, the term force field encompasses the functional form, and the parameter sets are used to calculate the potential energy of a system of atoms or coarse-grained particles in molecular mechanics and molecular dynamics simulations. The development of a general force field, able to yield accurate results for a plethora of chemically different compounds, is a particularly hard, complex, and ungrateful task. To obtain high-accuracy calculations on such chemically different compounds, a careful parameterization of an extremely diverse and complete set of reference molecules is required. This is, in practice, an impossible mission. Hence, it is not surprising that currently available general force fields had to sacrifice accuracy for a wider applicability. Improved quality is normally achieved by developing specialized force fields, ensuring accurate calculations to be performed, albeit in a much more limited class of compounds. The limited structural diversity, in terms of building blocks, that characterizes most biological systems of relevance, including proteins, lipids, carbohydrates, and nucleotides, renders the development of specialized force fields for each one of these large and important classes of biological macromolecules a particular interesting and valuable strategy, with an almost infinite number of applications, given the large number of combinations of the correspondent biological structural basic elements that can be found in nature [20–24].

Different levels of detail can also be achieved using different types of force fields, including coarse-grained, united-atom, and all-atom force fields. All-atom (i.e., atomistic) force fields have explicit parameters for all the atoms in a system, including hydrogen atoms. United-atom force fields treat the hydrogen and carbon atoms in each methyl group (terminal methyl) and each methylene bridge as one interaction center, providing a cruder representation. Coarse-grained force fields, which are often used in long-time simulations of macromolecules such as lipids, proteins, nucleic acids, and multi-component complexes, provide even cruder representations for higher computing efficiency.

All-atom force fields are generally the most accurate, as they retain virtually all atomic-level interactions and can use time steps in the femtosecond range. While this makes them quite slow and computationally expensive compared with the other alternatives, the wide range of carefully tested parameters available for these models, including proteins, lipids, nucleic acids, and small organic molecules, makes them reliable when it comes to quantitative prediction of properties such as motional time scales or interaction strengths, showing that this type of simulations has advantages, over those with a lower level of detail. They are also the most appropriate type of force field for simulating the interactions involving membrane proteins, as they provide an explicit representation of all the atoms and interactions at the interface, including those involving hydrogen atoms.

2.2.2 Simulating Biomembranes

In the particular case of membrane proteins, MD simulations offer an unparalleled way to analyze from a dynamic perspective the interactions established between MPs when inserted in the membrane, taking also into account the particularities of the water/membrane interface. Performing MD on membrane proteins requires the use of force fields for the representation of the protein, the water, and a model of the biomembrane.

AMBER, CHARMM, GROMOS, and OPLS are the most popular molecular mechanics force field families devised to describe biomolecular systems [25]. A common characteristic to these force fields is that the potential energy function is a function of pairs of atoms (it is two-body additive). Most force fields used in biological simulations apply the same form for the energy function, with harmonic terms for bonds and angles, Fourier series for torsions, and pairwise van der Waals and Coulombic interactions between atoms that are separated by three or more bonds. However, they are parameterized in conceptually different ways. Hence, individual parameters from different force fields should not be compared, as the parameterization scheme varies from force field to force field. Comparisons have to focus on the ability to reproduce observable data for a given system. Each of these force field families has specialized versions for the treatment of proteins and lipids. However, while for the treatment of proteins, accurate atomistic force field variations have been commonly in use with great success in a wide range of problems for a couple of decades, options to simulate lipids have remained for many years some steps behind. Furthermore, when combining membranes and proteins, it is important to take into account that the parameters used should be consistent, which means that the same general protocol should have been followed in the parameterization of all the associated molecules. This is especially important in the treatment of the non-bonded interactions (particularly charges, which decay slower with the distance), as the interactions between atoms within the protein or within the lipid bilayers have to be handled in the same fashion, and so should be the ones involving atoms in the protein with those in the bilayer. Such requirement is critical for an accurate representation of the interaction between the different partners.

More recently, dedicated force field extensions for the treatment of lipids have also been made available within all the major biomolecular force fields, levelling both fields and contributing to accurate representations of both the protein and the biomembrane [24, 26]. Within GROMOS, a number of variations have been made available through the years [27], including the parameter sets 45A3 [28], G53A [29], and G54A [30] and the popular Berger lipid FF [31], based on the original GROMOS non-bonded parameters and adopting a united-atom representation. More recent and improved versions include the 43A1-S3 [32] and the G53A6 [33] parameter sets. CHARMM [34, 35] included several parameter sets for atomistic simulations of lipids, including the CHARMM22 set (C22) [36], CHARMM27 (C27 and C27r) [17, 18, 37, 38], and the more recent CHARMM36 (C36) parameter set [39]. An extension for cholesterol has also been made available (C36c) [40]. Within AMBER, lipid simulations were done through many years with sets of lipid parameters based on re-parameterizations of the general AMBER force field [21, 41–43]. A specialized AMBER parameter set for lipids, called LIPID11 [44], was reported in 2012, followed by LIPID14 [45]. OPLS-AA also included parameters for lipids containing the DPPC bilayer [46]. Other common force field examples include MARTINI [47, 48], a coarse-grained force field, and Slipids [49].

The other critical partner is water, which also plays a fundamental role in mediating the interactions between different proteins and of these with the biomembrane. For the representation of the water molecules [50], common choices include the TIP3P (transferable intermolecular potential 3P) [51, 52], SPC (simple point charge) [53], and the SPC/E (extended simple point charge) [54] water models.

In spite of the increase in computational power that has characterized the past decades and advance in the technical sophistication of the software packages and force fields available, knowledge by the user still represents the single most determinant factor for a detailed simulation [55], particular of a complex problem such as this which involves the interaction between the protein, the membrane, and the water molecules. It is also important to consider that the length of the simulation is always a critical issue when discussing an MD simulation. Different chemical phenomena involve different time scales, and even when considering only proteins, it is important to keep in mind that their various characteristic types of motion have very different time scales, ranging from the fast and localized motion characteristic of atomic fluctuations to the slow and large-scale motions that involve rearrangements on the full protein. The length of the simulation should therefore be adequate to the type of motion under study. In addition, it is also important to retain that the different types of motion are interdependent and coupled to one another, although for some practical applications, some types may be regarded as independent. In general, these motions can span over 20 orders of magnitude in terms of time scale, from femtoseconds (e.g., vibrations of bonds) to several seconds and even hours. Membrane protein recognition and membrane interaction in particular normally require a minimum simulation length from 20 to 100 ns for proper sampling of the properties associated [56].

3 Methods

All the python-associated methods of the work pipeline for this protocol are based on Python version 3.6 and its respective packages. Manually curated changes and visualization were performed with PyMOL [57] unless otherwise indicated. The methods and databases referred along the text can be consulted in Table 2, in Subheading 4. The overall workflow is depicted in Fig. 1.

3.1 Dataset

The final biological dataset should be made of protein dimers that obey a predetermined set of criteria and for which a variety of features can be calculated. Residues should also be labeled as interfacial and non-interfacial, a binary positive and negative class, to be used by the reader to effectively train a ML model.

3.1.1 Raw Data

We began by accessing the *mpstruc* [6] database in which all MPs are associated to a .PDB file corresponding to the experimentally determined structure, mostly through X-ray crystallography and more rarely by nuclear magnetic resonance (NMR). The list of MP protein identification codes is made available at http://blanco.biomol.uci.edu/mpstruc/, by means of Extensible Markup Language (.xml) files. The .xml files retrieved are available on the “XML representations” section of the website. We downloaded “XML for the β-barrel proteins” and “XML for the α-helical membrane proteins,” since the only remaining structures (monotopic MP) do not comply to one of the requirements for this database: constituting a transmembrane protein. The files were read with the python package *ElementTree*, and the final structures were retrieved from PDB [58] with an inbuilt method that employs Biopython [63] through a python pipeline. The structures were downloaded with the code below:

Table 2
List of methods or databases referred along the text

Method or database	URL	Description	Ref.
Mpstruc	http://blanco.biomol.uci.edu/mpstruc/	Known membrane protein structures	[6]
SpotOn	http://milou.science.uu.nl/cgi/services/SPOTON/spoton/	Soluble protein complexes, hot spot detection	[9]
Protein data bank	https://www.rcsb.org/	Known protein structures	[58]
AMBER	http://ambermd.org/	Biomolecular molecular dynamics simulation software	[59]
CHARMM	https://www.charmm.org/charmm/	Biomolecular molecular dynamics simulation software	[60]
GROMOS	http://www.gromos.net/	Biomolecular molecular dynamics simulation software	[61]
OPLS		Molecular dynamics force field for liquid simulations	[62]
PyMOL	https://pymol.org/2/	Molecular visualization software	[57]
ElementTree	https://pypi.org/project/elementtree/	Python package for XML files handling	
Biopython	https://biopython.org/	Python-based biological computational tools	[63]
MODELLER	https://salilab.org/modeller/	Protein structures homology modeling tool	[64]
VMD	https://www.ks.uiuc.edu/Research/vmd/	Molecular modeling and visualization tool	[65]
PyDPI	https://pypi.org/project/pydpi/	Python-based chemoinformatics and bioinformatics package	[66]
iFeature	http://ifeature.erc.monash.edu/	Python-based package for protein feature extraction	[67]
DSSP	https://swift.cmbi.umcn.nl/gv/dssp/index.html	Protein secondary structure dictionary. Can be accessed via Biopython	[68]
Rosetta	https://www.rosettacommons.org/software	Molecular modeling program	[69]
Psiblast	https://www.ebi.ac.uk/Tools/sss/psiblast/	Protein sequence alignment tool	[70]
LipidBuilder	http://lipidbuilder.epfl.ch/home	Lipid creation, storage, and sharing	[37]
MemBuilder	http://bioinf.modares.ac.ir/software/mb2/	Membrane model initial configuration tool	[71]

(continued)

Table 2
(continued)

Method or database	URL	Description	Ref.
Insane	http://www.cgmartini.nl/index.php/insane	Lipid bilayer system setup tool	[39]
Packmol	http://m3g.iqm.unicamp.br/packmol/home.shtml	Molecular dynamics simulations initial configuration tool	[40]
InflateGRO	https://github.com/fuentesdt/MembraneProtein/blob/master/inflategro.pl	Biomembrane lipid simulation tool	[21]
Griffin		AMBER force field development	[41]
Alchembed	https://github.com/philipwfowler/alchembed-tutorial	Tool for incorporating multiple proteins into lipids	[72]
SHAKE		Molecular dynamics simulation box	[73]
LINCS		Molecular simulation constraint solver	[74]

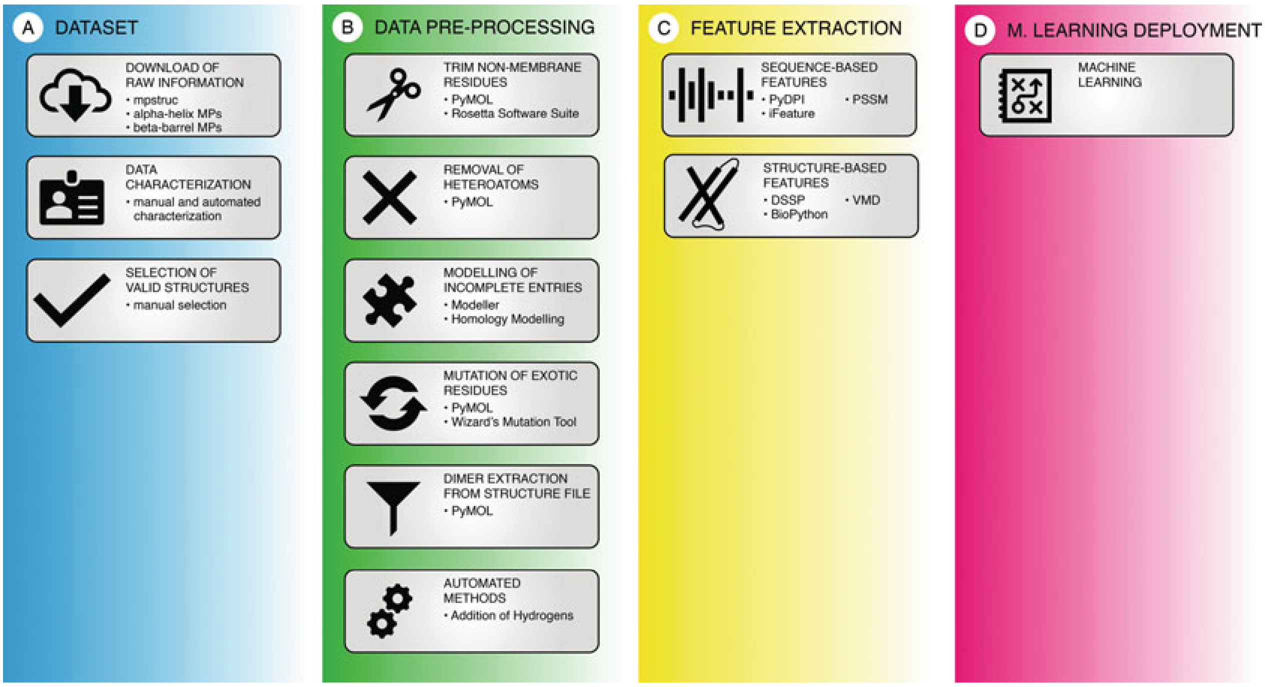


Fig. 1 Overall graphical depiction of the work pipeline for in silico characterization of MP interfacial residues database

```
import Bio
from Bio.PDB import *
import os

pdbl = PDBList()

###Append the PDB files to the following list before continuing
PDBlist2=[]

for pdb_entry in PDBlist2:
        try:
                print pdb_entry
                pdbl.retrieve_pdb_file(pdb_entry, pdir='PDB')
        except:
                continue
```

3.1.2 Data Characterization

To more easily and thoroughly select and manipulate the structures for the database, we performed an initial analysis. From the .xml files, by once again using the *ElementTree* package in python, we organized tables that characterized each structure with its PDB identification code, protein name, organism species and taxonomic domain, resolution of the structure, digital object identifier (DOI), protein subgroup name, and description. When the information was not available for all the referred fields, we attempted to retrieve it manually. In addition to the previous information, the number of chains, chains' biological names, in-file chain names (according to alphabetical labeling), and number of non-repeated chains were retrieved by analyzing the complexes with the Biopython [63] package. Stoichiometry was retrieved from PDB [58] with python through the selenium package for web automation. However, due to a high number of failing processes, this information was also partially manually retrieved and fully manually confirmed. Finally, all structures were manually analyzed to determine the complex class and the oligomer state. Regarding the complex class, there were the following possibilities: single chain, protein-ligand, protein-antibody, protein-protein, protein-peptide, and pore. When considering oligomer states, the options considered were single chain, multimer, multimer with at least one soluble protein (multimer*), membrane protein dimer (m–m), membrane protein-soluble protein dimer (m–s), membrane protein and soluble protein dimer (m–m*), or membrane protein dimer with soluble protein (both). All these characteristics were documented in separate tables for β-barrel and α-helical MPs. We also constructed a joint table to document all the complexes adding a descriptor to characterize them as β-barrel or α-helical. From the final characteristics, some were especially determinant on the selection of the structures: number of chains, complex class, and oligomer state.

3.1.3 Selection of Valid Dimeric Structures

At this point, we had a fully characterized set of MP structures, which needed to be further analyzed to ensure that the protein structures selected contributed positively to the purpose of this

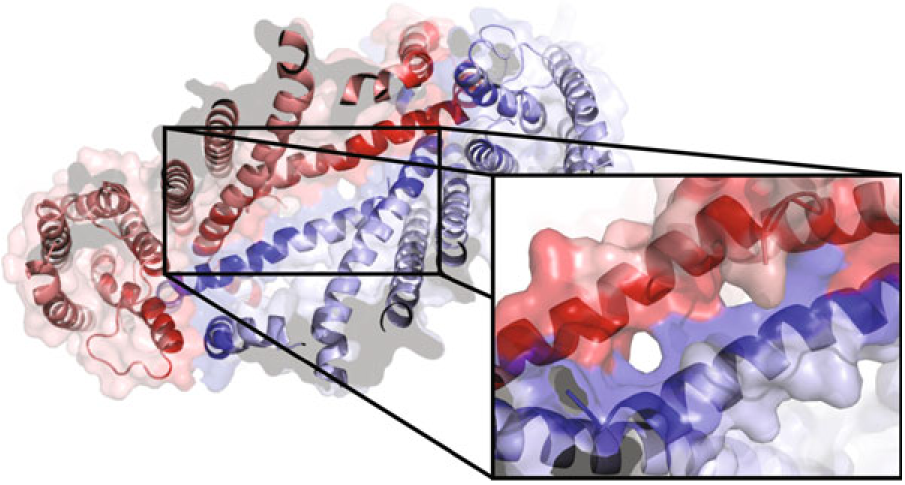


Fig. 2 Representation of an MP example (PDBid 5sy1 [75]) with the interface illustrated in the close-up

dataset. Although not referred extensively at the time, the choice of not using the monotopic membrane proteins was already under the scope of selection criteria. First of all, as said before, there are basic characteristics that this dataset must comply to: the structures must be MP and must contain more than one transmembrane chain. By choosing the *mpstruc* database, we ensured only MP would be present, and by excluding monotopic MP, we already excluded some non-transmembrane or single-chained proteins.

The characterization of the downloaded structures allowed us to automatically exclude, from the complex class and oligomer state, single chains, dimers in which one of the chains was a soluble protein (m–s), single MPs interacting with soluble small peptides (protein-peptide), pores, protein-antibodies (since antibodies are soluble proteins), and proteins with small organic or nonorganic ligands (protein-ligand). None of these structures had truly a PPI between two transmembrane MPs that contributed decisively to the upcoming prediction of interfacial residues. Regarding the remaining structures, not all were considered. The second round of selection excluded structures in which an excessive number of residues were unknown or incomplete. Furthermore, although some structures had two or more transmembrane chains, there was no clear PPI, in most cases due to a significantly large distance between them (much higher than the typical 5 Å). Finally, a very high amount of lipids between the chains was in some cases determinant for the exclusion, since it would clearly interfere in the interface. Figure 2 illustrates a typical MP dimer.

3.1.4 Data Pre-processing

Pre-processing treatment of any dataset is necessary and mandatory to ensure the success of ML application. First, since the quality of the data is vital to the training of the model, we describe the steps employed to guarantee the viability of the models from the dataset built (3.1.4.i–3.1.4.v). Next, an additional pre-processing step from our automated pipeline is also described (3.1.4.vi).

(i) Trim Non-transmembrane Residues

The α-helical and β-barrel structures attained for the dataset were representative of transmembrane dimers but had nevertheless non-transmembrane residues or motifs. Visual inspection of the .PDB as well as information contained in these files was used to identify the transmembrane domain in order to posteriorly remove residues outside these regions. The process of identifying non-transmembrane residues is, on the automated part of this pipeline, underlined by the use of an inbuilt deployment of Rosetta [69].

(ii) Remove Heteroatoms

The downloaded structures were experimentally determined and, in many cases, present structural water molecules, metal ions, or small molecules. Also, since these are MP proteins, lipids are often found, even if not in sufficient amounts to exclude the structure for interfacial interference. For the purpose of this pipeline, these atoms would introduce unnecessary error and often intervene negatively with the measurements from the upcoming steps. Hence, a very simple but yet important step was to remove these atoms. They are listed as heteroatoms in the .PDB files, and PyMOL [57] scripts allow their easy removal. The PyMOL code snippet below displays a possible protocol for PyMOL visualization and heteroatom removal after loading all structures:

```
load pdb_file.pdb
bg_color white
set depth_cue, off
set fog, off
hide lines
show cartoon
util.cbc(selection='(all)',first_color=7,quiet=1,legacy=0,_self=cmd)
remove hetatm
```

(iii) Mutate Exotic Amino Acids

Besides heteroatoms, there were still other atoms capable of introducing error or nullifying some of our methods. In particular, many feature extraction methods are not prepared to deal with amino acids out of the ordinary set of 20. Selenomethionine residues are an example of amino acids that raise this problem. Furthermore, since these amino acids stand in the backbone of the protein, they could not be simply erased, as the heteroatoms were. To avoid this, such residues were mutated to their more usual counterparts (selenomethionines, e.g., were mutated to methionine) by using the PyMOL [57] "Mutagenesis" tool, available at the “Wizard” section, and choosing the rotamer with the lowest number of crashes.

(iv) Model Incomplete Structures

Structures from the raw dataset that contained residues with many missing atoms were excluded, as referred before. However, in some cases, only a few residues were incomplete, and so the structure was kept. This being the case, homology modelling was used to rebuild the full residues. Following the extraction of the sequence of the original structure in the form of a FASTA file, we employed MODELLER [76], to generate an alignment file with both the original and the target sequences, which are the same in this case.

```
from modeller import *
import os

def align(input_pdb):

        pdb_name = input_pdb[0:-4]
        env = environ()
        aln = alignment(env)
        first_chain = "FIRST:A"
        last_chain = "LAST:B"
        fasta_name = pdb_name + ".fasta"
        mdl = model(env, file=pdb_name, model_segment=(first_chain,last_chain))
        aln.append_model(mdl, align_codes=pdb_name, atom_files=input_pdb)
        aln.append(file=fasta_name, align_codes=pdb_name)
        aln.align2d()
        q_ali_name = pdb_name + ".ali"
        aln.write(file=q_ali_name, alignment_format='PIR')
```

Also using MODELLER [76] with the previously generated alignment and the original structure as template, we generated models of the structure with the full residues (where they were previously incomplete). These models, since the template was the protein itself, had very little difference in structure, but they were complete and apt to be properly included in the dataset.

```
from modeller import *
from modeller.automodel import *
from modeller import soap_protein_od

def generate_model(input_ali):

    input_name = input_ali[0:-4]
    env = environ()
    a = automodel(env, alnfile=input_ali,
                  knowns=input_name, sequence=input_name,
                  assess_methods=(assess.DOPE,
                                  assess.GA341))
    a.starting_model = 1
    a.ending_model = 5
    a.make()
```

(v) Dimer Extraction from the Structure Files

Having standardized all the models to meet the criteria for the dataset purpose, the final step was the extraction of the dimers from the structural files in which there were more than two valid possible dimer options. This was performed by visual inspection with PyMOL [57]. The following steps of the protocol are fully automated.

(vi) Add Hydrogens

Most structures available do not include hydrogen atoms, but their explicit representation is important since they can be involved in hydrogen bonds that contribute to stabilize the structure and, in particular, the interfaces. To add them, we employed the visual molecular dynamics (VMD) [65] software in a fully automated manner. Using a template (.tpl) file which stores the necessary commands that VMD [65] needs to properly add hydrogens to a .PDB file with two chains (see the code snippet below), a new file was generated specifying the commands for the specific structure under scope. This output file was then run, also from inside python, employing the python "os.system" in-built function.

```
package require psfgen
topology top_na.inp
alias residue HIS HSD
alias residue HOH TIP3
alias residue ZN ZN2
alias atom ILE CD1 CD
alias atom HOH O OH2
pdbalias residue DG GUA
pdbalias residue DC CYT
pdbalias residue DA ADE
pdbalias residue DT THY
foreach bp { GUA CYT ADE THY URA } {
     pdbalias atom $bp "O5\*" O5'
     pdbalias atom $bp "C5\*" C5'
     pdbalias atom $bp "O4\*" O4'
     pdbalias atom $bp "C4\*" C4'
     pdbalias atom $bp "C3\*" C3'
     pdbalias atom $bp "O3\*" O3'
     pdbalias atom $bp "C2\*" C2'
     pdbalias atom $bp "O2\*" O2'
     pdbalias atom $bp "C1\*" C1'
}

segment A {
     pdb chain_A.pdb
     first none
     last none
}
segment B {
     pdb chain_B.pdb
     first none
     last none
}

coordpdb chain_A.pdb A
coordpdb chain_B.pdb B

guesscoord
writepdb final_file_HS.pdb
quit
exit
```

3.2 Feature Extraction

In order to perform ML on a given dataset, more than the instances (in these cases the MP structures' residues), there is a need to associate them with features, which, as already mentioned, are descriptors that characterize the instances. For this dataset, we engulf as many features as possible, provided they are reliable and their extraction/calculation can be automated. In the case of proteins, many of the features relate to their different hierarchical structures: primary (or sequence-based), secondary, and tertiary features. Furthermore, other features can be associated with the interfacial interaction or the proteins' evolutionary profile.

3.2.1 Sequence-Based Features

i) PyDPI

PyDPI [66] is a python package developed toward chemoinformatics, bioinformatics, and chemogenomics studies. We focused on PyPro, a PyDPI [66] sub-module that mines protein structural files in order to retrieve sequence-based features. To perform this, the sequences of both chains of each dimer were retrieved from the files with the aid of Biopython [63]. A "PyPro()" object was initialized, allowing for the next steps. This object read the sequences employing the *ReadProteinSequence()* method. Finally, we employed the *GetALL()* method on the same object, to retrieve a python dictionary in which the keys were the name of the feature and the values were the computed results. Notice that these features are not residue specific but rather associated to the whole sequence; hence, all the residues in one chain will have the same associated score. The features provided by this method are:

- 20 Amino Acid Composition (AAC) descriptors—the amount of each amino acid residue in the sequence.
- 400 Dipeptide Composition (DPC) descriptors—the amount of possible combinations of two subsequent amino acids.
- 240 Moreau-Broto autocorrelation (MBauto) descriptors.
- 240 Moran autocorrelation (Moranauto) descriptors.
- 240 Geary autocorrelation (Gearyauto) descriptors.
- 21 Composition descriptors.
- 21 Transition descriptors.
- 105 Distribution descriptors.
- 100 Quasi-Sequence Order (QSO) descriptors.
- 777 Pseudo Amino Acid Composition (PAAC), Amphiphilic Pseudo Amino Acid Composition (APAAC) and Conjoint Triad (CT) descriptors.

```
from pydpi import pypro
from pydpi import protein
from pydpi.pypro import GetAAIndex1, GetAAIndex23
from pydpi.pypro import PyPro
from pydpi.protein import getpdb, AAComposition
from pydpi.pypro import CTD

def amino_sequence_pypro(input_pdb):

    ###Retrieves amino acid sequence from bioython structure object
    structure = pdb_parser(input_pdb)[0]
    seq_type = "ATOM "
    for model in structure:
        for chain in model:
            seq = ""
            for residue in chain:
                ## The test below checks if the amino acid
                ## is one of the 20 standard amino acids
                ## Some proteins have "UNK" or "XXX", or other symbols
                ## for missing or unknown residues
                if is_aa(residue.get_resname(), standard=True):
                    seq = seq + (str(three_to_one(residue.get_resname())))
                else:
                    continue
    return seq,seq_type

def pypro_features(input_pdb, chain_name):

    ###All the sequence based features come from here
    path_to_pdb = main + "/" + input_pdb + "_HS_" + chain_name + ".pdb"
    res1, res2 = amino_sequence_pypro(path_to_pdb)
    protein = PyPro()
    protein.ReadProteinSequence(res1)
    all_features = protein.GetALL()
    key_list = []
    value_list = []
    for key, value in all_features.items():
        key_list.append(round_number(key))
        value_list.append(round_number(value))
    return key_list,value_list
```

i) iFeature

iFeature [67] is another package developed for python applications which encompasses several tools for bioinformatics deployment. Namely, it allows feature extraction. Similar to PyDPI [66], we used iFeature [67] to extract sequence-based features. iFeature [67] was called from inside the main script. To use it, the features must be called separately and computed from the sequence (FASTA file), since there is no unified function. Similar to PyDPI [66], the scores do not characterize specific residues, but rather the proteins' chains. Hence, all the residues from the same chain have the same associated score. The features retrieved from iFeature are:

- 240 Normalized Moreau-Broto (NMBroto) descriptors.
- 240 Moran descriptors.
- 39 Composition features.
- 60 Sequence-Order-Coupling Numbers (SOCNumber).
- 100 Quasi-Sequence-Order (QSOrder) descriptors.
- 50 Pseudo Amino Acid Composition (PAAC) descriptors.
- 80 Amphiphilic Pseudo Amino Acid Composition (APAAC) descriptors.

Even considering that some features are similar between PyDPI [66] and iFeature [67], we kept all of them since posterior ML methods chosen by readers will be able to rule out or ignore redundant features.

```
import os

executeCommands = True
def RunInOS(command):
    if executeCommands:
        os.system(command)

def write_iFeature(pdb_input, iFeature_path, feature_type):

    ###Write a .txt file with
    out_name = pdb_input[0:-4] + "_" + feature_type + ".txt"
    fasta_path = pdb_input[0:-4] + ".fasta"
    new_command = "python " + '"' + iFeature_path + '"' + " --file " + '"' + fasta_path + '"' + " --type "
+ feature_type + " --out " + out_name
    RunInOS(new_command)

def read_iFeature(input_feature_txt, feature_type):

    ###Read the previously written iFeature .txt
    read_file = open(input_feature_txt, "r").readlines()
    count = 0
    header = []
    chain_A = []
    chain_B = []
    for row in read_file:
        row = row.split()
        for cell in row[1:]:
            if count == 0:
                feature_name = feature_type + "_" + cell
                header.append(feature_name)
            if count == 1:
                chain_A.append(round_number(cell))
            if count == 2:
                chain_B.append(round_number(cell))
        count = count + 1
    return header, chain_A, chain_B
```

3.2.2 Secondary and Tertiary Features

In order to predict secondary structure, and later secondary and tertiary structure derived features, we employed DSSP (Database of Secondary Structure assignments for all Protein entries), from PDB [58]. The approach that we developed was to use the “sys” package from python to call DSSP from the shell, using the command:

```
dssp -i input_pdb > input_pdb_name_dssp.txt
```

This command generates a text file from which several features can be obtained. Before extracting the features, however, we used DSSP to attain the secondary structure prediction, which could then be manipulated for obtaining amino acid propensity in secondary structure motifs, as explained below. Regarding the features extracted with DSSP, they are residue specific and are:

- Relative accessible surface area (ASA).
- Phi angle.

- Psi angle.
- NH–O1 energy and relaxation (2 features).
- O–NH1 energy and relaxation (2 features).
- NH–O2 energy and relaxation (2 features).
- O–NH2 energy and relaxation (2 features).

```
def DSSP_features(input_pdb, feature_number):

    ###Retrieves the features 0-13 described bellow, from bioython structure object
    ###0            DSSP index
    ###1            Amino acid
    ###2            Secondary structure
    ###3            Relative ASA
    ###4            Phi
    ###5            Psi
    ###6            NH-->O_1_relidx
    ###7            NH-->O_1_energy
    ###8            O-->NH_1_relidx
    ###9            O-->NH_1_energy
    ###10           NH-->O_2_relidx
    ###11           NH-->O_2_energy
    ###12           O-->NH_2_relidx
    ###13           O-->NH_2_energy
    to_break = [7,8,9,10]
    structure = pdb_parser(input_pdb)[0]
    dssp_name = input_pdb[0:-4] + "_dssp.txt"
    opened_file = open(dssp_name, "r").readlines()
    chain_SS_sequences = []
    useful = False
    feature_residues_A = {}
    feature_residues_B = {}
    residues_A_count = 0
    residues_B_count = 0
    feature_gaps = {"0":[0,5], "1":[5,10], "2":[10,12], "3": [12,14], "4":[14,22], "5":[22,33], "6":[34,38],"7":[38,50], "8":[50,61], "9":[61,72], "10":[72,83], "11":[83,91], "12":[91,97], "13":[97,103], "14":[103,109], "15":[109,115], "16":[115,122], "17":[122,129], "18":[129, 136], "19":[136, 150]}
    for row in opened_file:

        if useful == True:
            if row[feature_gaps["2"][0]:feature_gaps["2"][-1]].replace(" ","") == "A":
                residues_A_count = residues_A_count + 1
                if feature_number in to_break:
                    feature_to_store = row[feature_gaps[str(feature_number)][0]:feature_gaps[str(feature_number)][1]].replace(" ","").split(",")
                    feature_value = round_number(feature_to_store[-1])
                    feature_residues_A[residues_A_count] = feature_value
                else:
                    feature_value = round_number(row[feature_gaps[str(feature_number)][0]:feature_gaps[str(feature_number)][1]].replace(" ",""))
                    feature_residues_A[residues_A_count] = feature_value
            if row[feature_gaps["2"][0]:feature_gaps["2"][-1]].replace(" ","") == "B":
                residues_B_count = residues_B_count + 1
                if feature_number in to_break:
                    feature_to_store = row[feature_gaps[str(feature_number)][0]:feature_gaps[str(feature_number)][1]].replace(" ","").split(",")
                    feature_value = round_number(feature_to_store[-1])
                    feature_residues_B[residues_B_count] = feature_value
                else:
                    feature_value = round_number(row[feature_gaps[str(feature_number)][0]:feature_gaps[str(feature_number)][1]].replace(" ",""))
                    feature_residues_B[residues_B_count] = feature_value
        if row[feature_gaps["0"][0]:feature_gaps["0"][-1]].replace(" ","") == "#":
            useful = True
    chain_SS_sequences.append(feature_residues_A)
    chain_SS_sequences.append(feature_residues_B)
    return chain_SS_sequences
```

Using the Biopython [63] module, we extracted the B-factor values from the complexes and constructed a windowed function that averages, for each residue, its values in a radius of 5 residues, generating a new feature (please check **Note 2** for further information).

```
import Bio
from Bio.PDB import *

def pdb_parser(input_pdb):

    ###Parses pdb from .pdb file
    parser = PDBParser()
    pdb_name = input_pdb[0:-4]
    structure = parser.get_structure(pdb_name, input_pdb)
    return structure, pdb_name

def b_factor(input_pdb, input_atom = "CA"):

    ###Returns b-factor for the input .pdb atoms, alpha carbon as default
    structure = pdb_parser(input_pdb)[0]
    chain_tagger = []
    for model in structure:
        for chain in model:
            count = 0
            b_factors = {}
            for residue in chain:
                count = count + 1
                for atom in residue:
                    if atom.get_name() == input_atom:
                        B = atom.get_bfactor()
                        feature_value = round_number(B)
                        b_factors[count] = feature_value
            chain_tagger.append(b_factors)
    return chain_tagger

def window(input_dicts, user_function, window_size = 5):

    ###Iterates over previously achieved scores and builds new values by sliding a window of twice the
size of the argument
    chain_storer = []
    for chain in input_dicts:
        output_dict = {}
        for entry in chain.keys():
            value_list = []
            current_value = chain[entry]
            value_list.append(current_value)
```

```
            for new_value in range(1, window_size):
                try:
                    value_list.append(chain[int(entry) + new_value])
                except:
                    continue
                try:
                    value_list.append(chain[int(entry) - new_value])
                except:
                    continue
            final_value = user_function(value_list)
            output_dict[entry] = final_value
        chain_storer.append(output_dict)
    return chain_storer
```

Using the secondary structure predicted with DSSP, we created an amino acid propensity feature that associates to each of the possible secondary structural motifs the frequency of occurrence of each amino acid.

```
def amino_acid_propensity(sequence, secondary_sequence, secondary_structure_tag):

    ###Fetches the amino acid counts by secondary_structure:
    all_chains = []
    for chain_simple, chain_second in zip(sequence, secondary_sequence):
        PC_dict = {'G': 0, 'A': 0,'V': 0,'L': 0,'M': 0,'I': 0,'F': 0,
               'Y': 0,'W': 0,'S': 0,'T': 0,'C': 0,'P': 0,'N': 0,
               'Q': 0,'K': 0,'R': 0,'H': 0,'D': 0,'E': 0, 'X' : 0}
        count = 0
        for residue, residue_TM in zip(chain_simple, chain_second):
            if residue_TM == secondary_structure_tag:
                count = count + 1
                PC_dict[residue] = PC_dict[residue] + 1
        for entry in PC_dict:
            try:
                PC_dict[entry] = float(PC_dict[entry])/float(count)
            except:
                continue
        new_PC_dict = {}
        new_count = 0
        for new_residue in chain_simple:
            new_count = new_count + 1
            feature_value = round_number(PC_dict[new_residue])
            new_PC_dict[new_count] = feature_value

        all_chains.append(new_PC_dict)
    return all_chains
```

Furthermore, we used VMD [65] to find surface residues using the code below, in which the individual solvent-accessible surface area values considered were the ones described in Miller et al. [77].

```
mol new file_name.pdb
set allsel [atomselect top "all and chain name"]
set chain A
set tot_sasa [dict create ARG 241 TRP 259 TYR 229 LYS 211 PHE 218 MET 204 GLN 189 HIS 194 GLU 183 LEU 180
ILE 182 ASN 158 ASP 151 CYS 140 VAL 160 THR 146 PRO 143 SER 122 ALA 113 GLY 85]
set residlist [lsort -unique [$allsel get resid]]
set surf_list [list]
foreach r $residlist {
        set sel [atomselect top "resid $r and chain $chain"]
        set temp_rsasa [measure sasa 1.4 $allsel -restrict $sel]
        set temp_name [lsort -unique [$sel get resname]]
        set temp_id [lsort -unique [$sel get resid]]
        set temp_tot [dict get $tot_sasa $temp_name]
        set rsasa [expr $temp_rsasa/$temp_tot]
        if {$rsasa > 0.2} {lappend surf_list "$temp_id $temp_name"}
}


set filename "residues_surface_chain_name"
set fileId [open $filename "w"]
puts $fileId $surf_list
close $fileId
exit
```

This code was then run on VMD, from the python main frame by issuing the command:

```
vmd -dispdev text -e get_surf_residues_chain_name.tcl
```

The surface residues are then associated with 1, while the non-surface residues are associated with 0, in the data table for ML deployment. Similar to surface residues, we calculated the interface residues, also using VMD [65], with specific model scripts. The interfacial characterization (also binary) was the class we used to perform the ML deployment. The code used to perform this is displayed below.

```
mol file_name.pdb
set outfile [open "residues_number_according_to_chain" w]
set sel1 [atomselect top "protein and (chain A) and within 5 of (chain B)"]
$sel1 get {resid resname}
set sel2 [lsort -unique [$sel1 get {resid resname}]]
puts $outfile "$sel_number_according_to_chain"
close $outfile
quit
exit
```

In addition to the individual solvent-accessible surface area, lipid accessibility by residue was extracted as well. For this procedure, we used the “mp_lipid_acc”—an application included in the Rosetta Software Suite [69]. The following command was used, creating a new model .PDB file from information in the original .PDB file in addition to the span file. The obtained model contains a binary score column with the values 0 and 50, depending in the lipid accessibility—0 corresponding to inaccessible and 50 to accessible.

```
./rosetta_bin_linux_2018.09.60072_bundle/main/source/bin/mp_lipid_acc.static.linuxgccrelease -database
./rosetta_bin_linux_2018.09.60072_bundle/main/database -in:file:s [input_pdb_file] -mp:setup:spanfiles
[general span file, built in the previous command] -ignore_unrecognized_res
```

Like stated in section 3.1.4, it is essential to restrict the analysis exclusively to residues in the transmembrane region. This procedure was performed using the “mp_span_from_pdb” application, from the Rosetta Software Suite [69]. The following commands, having downloaded and built the Rosetta Software Suit in the same directory as the input files, output a span file, which lists the transmembrane regions of the chain under analysis. Below, is highlighted the command needed to attain this output which will generate a set of span files, one for each chain in the input .PDB file.

```
./rosetta_bin_linux_2018.09.60072_bundle/main/source/bin/mp_span_from_pdb.static.linuxgccrelease -
in:file:s [input_pdb_file] -ignore_unrecognized_res
```

An example of the output file, obtained from the sucrose-specific porin (PDBid: 1A0T) [78], is displayed below:

```
Rosetta-generated spanfile from SpanningTopology object
19 413
antiparallel
n2c
 3 11
 49 58
 64 72
 91 98
 111 118
```

```
123 128
139 147
151 157
178 183
186 190
221 228
234 241
270 276
281 287
308 314
319 324
352 357
369 373
407 412
```

This procedure was performed using a Linux operating system. The Rosetta build may vary depending on the operating system in use.

3.2.3 Sequence Comparison Features

In this pipeline, there is also the possibility of adding position-specific scoring matrix (PSSM) features. To do this, the option must be specifically selected, since these features increase significantly the time of the process. While without PSSM features the in-house process can run within 3 min, for one dimer, it can take up to a few hours if this option is chosen, due to the alignment process. This happens because of the use of the "psiblast" [70] alignment for PSSM extraction. Please confer **Note 3** to learn how to employ "psiblast" to extract PSSM features.

3.3 Molecular Dynamics Simulations

3.3.1 Building the Protein-Membrane Model

A critical step before initiating any MD simulation involving a membrane protein is the creation of reasonable conformation containing the protein and the biomembrane model [21, 79]. Such structure has to represent or enable within a reasonable simulation time a realistic packing of the protein and lipids [35].

The choice of the biomembrane model represents an obvious approximation into the biology of the problem. Biomembranes are complex systems comprised by a wide range of different molecules, the balance of which determines its physical properties [80]. The relative composition of different molecular types in biomembranes can vary significantly between different organelles and cell types, ranging from 20% to 60% proteins, 30% to 80% lipids, and up to 10% carbohydrates. Among lipids, phospholipids, sphingolipids, and sterols are the major components present, in a ratio that determines a variety of properties, including surface charge, thickness, packing order, curvature, etc. [81]. However, most simulations represent the biomembrane as small bilayer patches containing just a few lipid moieties, often manually prepared.

Sometimes, a limited number of cholesterol molecules are introduced into the simulation to partially account for the heterogeneity of the bilayer. However, when preparing the biomembrane model, it is also important to take into account that building the biomembrane model as a simple assembly of the selected lipids is not enough, as extensive sampling would be required to transform the modeled bilayer structure into a reasonable biomembrane model [35]. Rather, a variety of solutions are currently available in current MD packages which contain ensembles of pre-equilibrated lipids [45, 82]. This approach is, however, not flexible, as it enables the simulation of only the limited set of alternatives explicitly present.

More recently a variety of membrane builder applications have been made available, enabling customization by the user in building biomembranes with custom compositions. These can be grouped into two main categories: web servers and distributed software. Web server membrane builder applications are normally user-friendly and relatively fast in distributing different components along the model generated. Examples include CHARMM-GUI [36], LipidBuilder [37], and MemBuilder [38]. However, the membranes generated are normally a long way from equilibrium, as optimizing biomembrane distribution and interactions can be an intensive process. Also, these servers are normally limited to specific molecular components and lipid types. Software-based application includes programs such as Insane [39] and Packmol [40] that can be downloaded and installed locally. They can generate any kind of densely packed structures, with the components and physical requirement imposed by the user and taking into account density optimization. However, such alternatives are still in their infancy, and generally consider only a crude description of the properties of the individual molecules in the density optimization.

i) Insertion of the Protein in the Membrane

Choosing the exact orientation and position of the membrane protein within the bilayer model can be a challenging task, as there is often no experimental data to guide how the protein should be placed relative to the bilayer. Nevertheless, an analysis on the amino acid residues defining the surface of the membrane protein can provide valuable clues, taking into account the amphipathic nature of the biomembrane. In fact, for many membrane proteins, there is a clear distinction between the type of amino acid residues located in the surface interacting the hydrophobic core of the biomembranes and those that interact with the lipid polar heads or with the solvent. The surface of the membrane protein interacting with the solvent or with the polar heads results from a careful balance between hydrophilic and hydrophobic amino acid side chains, with the first being prevalent [83]. However, the surface of the membrane protein that interacts with the hydrophobic core of the biomembrane presents an almost total absence of polar or charged amino acid residues, being composed almost entirely by amino acid residues with hydrophobic side chains. Taken together, these tendencies normally help to define a perpendicular axis for membrane protein insertion in the biomembrane, which will be clearer for membrane proteins with a high degree of insertion in the biomembrane or for transmembrane proteins. These observations also help to identify an axis along the membrane protein that will be parallel to the biomembrane surface and that differentiates the predominately hydrophilic from the predominant hydrophobic surface regions of the protein and that will tend to be aligned with hydrophilic/hydrophobic regions of the membrane.

Once an orientation for the membrane protein is selected, the next step is its insertion into the membrane. Early alternatives relied on building the membrane around the protein or deleting a certain amount of lipids from a pre-equilibrated bilayer, creating a void into which the membrane protein could be placed. Both approaches tend to lead to excessive perturbations into the overall structure of the biomembrane and require careful equilibration. More recently, several specialized methods have been developed to ensure a smoother insertion of the membrane protein. Examples include the InflateGRO methods [21, 41], GRIFFIN [42], and GROMACS-based approaches such as *Mdrun_hole* [43] and *G_membed* [44]. Alchembed [72] is another popular alternative for membrane protein insertion, making use of soft-core potentials to slowly push the lipids away from the membrane protein during insertion. CHARMM-GUI also contains a functionality that enables the inclusion of one membrane protein per biomembrane. Its wide range of functionalities coupled to the ability to generate membranes with different compositions make CHARMM-GUI a popular starting point for generating custom biomembranes and inserting the protein.

ii) Selection of Force Fields

As described above, several force fields are currently available for the simulation of membrane proteins inserted in a bilayer. The study of membrane protein interactions with detail greatly encourages the use of atomistic force fields, and these enable the inclusion of all the main interactions formed along the protein interface with detail, enabling also the inclusion of the effect of the membrane and water. As reviewed previously, the major limitation in terms of atomistic force fields has been lipid representation. Presently, several atomic-level force fields able to describe a variety of lipid molecular types with accuracy have been made available, including CHARMM36, Lipid14, and SLipids, just to cite some of the most popular and recent. Although the level of accuracy of different alternatives can differ when performing extensive MD simulation on lipid properties, for the interactions between membrane proteins, alternatives as the ones mentioned would provide excellent results. Consistency in the force field selection for protein and biomembrane is therefore presently the main issue to take into account, as the parameters used to describe the amino acid residues and lipid molecules should have been developed using a similar approach, based on the same type of overall principles. Hence, mixing different force field families and classes for protein and membrane is highly discouraged.

Final choice often emerges from the specific software package available to the user and his working knowledge. While some software packages like GROMACS and NAMD offer the user the possibility to choose from several different force fields, others like CHARMMM and AMBER initially only supported their own specific force fields. Alternatives to convert topology files and input parameters from one software package to another have been increasingly made available, making it possible to use specific force fields in other software packages. However, this process is often difficult to master for the non-expert user, often still limiting the final choice.

iii) Simulating the Protein by Molecular Dynamics

Once the model system is prepared and force field and software are selected, the molecular dynamics simulation can be performed. This is normally run in a cuboid box, with periodic boundary conditions along the biomembrane plane (*xy*), typically with an integration time step of 1 or 2 fs (if bonds involving hydrogen are kept constrained with specialized algorithms as SHAKE [73] or LINCS [74]), and with a cutoff of 10–12 Å for the treatment of the non-bonded interactions. Given the complexity of the model, and the two phases that it comprises (water and biomembrane), special care must be taken when starting the simulation. First, to prevent disruption of the model system in the initial stages of the simulation, a set of MM minimizations are normally recommended. These normally start with a preliminary MM minimization in which all heavy atoms are frozen and only hydrogen atoms are allowed to optimize. Typically, in a second stage, the water molecules are optimized, while the gross of the biomembrane and protein is kept frozen. Subsequent steps involve the progressive release of the constraints imposed in the system (e.g., protein side chains, biomembrane tails, protein backbone, biomembrane heads, etc.), ending in a fully free MM minimization of the full system. Only after this stage is the system ready for MD simulation.

This normally starts with a stage in an NVT ensemble starting a 0 K, during which the temperature of the system is gradually increased up to the desired simulation temperature (typically 298 or 310 K). The densities of the water and biomembrane are evaluated through time, until equilibrated. The system is then switched to an NPT ensemble (or a variation), and the simulation continues at the desired temperature and pressure. The structural stability of the components analyzed is monitored through time through a RMSd analysis. Total simulation lengths of 20–500 ns after equilibration are normally pursued.

3.3.2 Analyzing the Interactions

Depending on the specific software chosen, a variety of tools can be used to analyze membrane protein interactions through MD simulations. Common examples include the analysis of hydrogen bonds, distances, radial distribution function, and solvent-accessible surface areas. An important feature is the analysis of the hydrogen interactions along the protein interface. Contrary to the static representation of systems, MD simulations offer the opportunity to assess the prevalence of specific hydrogen bonds during an entire simulation, enabling the determination of dynamic properties including average length and angle and their standard deviation, average time during which the hydrogen bond is kept, maximum occupancy, alternative hydrogen bonds involving the same group, etc. Similar procedures can be used to analyze other interactions or lengths, including the overall length and width of the protein, difference between centers of mass of different proteins, differences between average axis of α-helices, etc. [46].

Radial distribution function analysis is often applied to sample the accessibility of specific functional groups along the interface to solvent molecules, or the atoms from other molecular components. In a radial distribution analysis, a number of increasingly larger circles are traced around atoms or groups of reference, with increasing size (typically by 0.1 Å) typically covering a range of different radius from as much as 0–10 Å. Within each increasing circle, the number of interacting molecules (e.g., water) is determined for each recorded conformation of the simulation trajectory. From these analyses, a probability density for the type of interactions evaluated with distance emerges.

Another common property is the solvent-accessible surface area. This property can be used to analyze a simulation of a protein-protein biomembrane complex and determine the area of a given amino acid residue that is in contact with the solvent or with the biomembrane. SASA tools can also normally be adjusted to estimate the area of a specific amino acid residue that is in contact with other protein, the potential SASA lost upon protein-protein interaction, or the percentage of surface of an amino acid residue that is employed in the interaction, always from a dynamic perspective, as these quantities oscillate during a simulation. VMD [65] is a popular molecular visualization tool used to analyze molecular dynamics simulations. It contains a selection of built-in tools for automated analysis of these and other properties. AMBER, CHARMM, GROMACS also contain specific commands to analyze these properties.

Here, we described two identifiable processes. The first, assembling the dataset prepared to characterize a MP database and to potentially train an interfacial residues predictor. The processed forms of the original .PDB files, the final dimer database, and the description of the used structures from their original files will be available for use and constitute a landmark for protein dimer study.

The automated pipeline for the study itself is hereby explained in its individual steps and will be made fully available for any user to access it in an easy way. The second process was focused on special techniques and advices when applying MD to extra characterization of structural and mechanistic features of membrane proteins of particular interest for the user.

4 Notes

1. When adding hydrogens bonds, PyMOL [57] can also be used in a simpler manner. To do this the PDB file must be loaded, and the method "add_h" is called, adding the hydrogens. This method was not employed due to not being as thorough as VMD [65] and being of difficult employment on a Python-integrated pipeline; however, it can be used for simpler modifications.
2. Regarding the windowed function used to compute *B*-factors with the influence of surrounding residues, it can also be used for other purposes. The feature under scope can be different from the *B*-factor. The window radius of residues and the function that is employed on the values can be changed. This aims at reproducing the influence of other residues on a given residue.
3. The following command, having installed psiblast and downloaded the "non-redundant" proteins dataset, outputs a PSSM for one chain.

```
 psiblast_path -query file.fasta -evalue 0.001 -num_iterations
2 -db nr -outfmt 5 -out pssm_file_ chain_name.txt -out_-
ascii_pssm pssm_file__chain_name.pssm -num_threads 6
```

The output file can then be read to retrieve 42 PSSM-derived features.

Acknowledgments

Irina S. Moreira acknowledges support by the Fundação para a Ciência e a Tecnologia (FCT) Investigator programme—IF/00578/2014 (co-financed by European Social Fund and Programa Operacional Potencial Humano). This work was also financed by the European Regional Development Fund (ERDF), through the Centro 2020 Regional Operational Programme under project CENTRO-01-0145-FEDER-000008: BrainHealth 2020. We also acknowledge the grants POCI-01-0145-FEDER-031356 and

PTDC/QUI-OUT/32243/2017 financed by national funds through the FCT/MCTES and co-financed by the European Regional Development Fund (ERDF), namely, under the following frameworks: “Projetos de Desenvolvimento e Implementação de Infraestruturas de Investigaçãoinseridas no RNIE”; “Programa Operacional Competitividade e Internacionalização—POCI”; “Programa Operacional Centro2020”; and/or State Budget.

References

1. Israelachvili JN, Marcelja S, Horn RG (1980) Physical principles of membrane organization. Q Rev Biophys 13(2):121–200
2. Chiu ML 2012 Introduction to membrane proteins. Curr Protoc Protein Sci Chapter 29: Unit 29.1
3. Gromiha MM, Ou YY (2014) Bioinformatics approaches for functional annotation of membrane proteins. Brief Bioinform 15 (2):155–168
4. Papadopoulos DK et al (2012) Dimer formation via the homeodomain is required for function and specificity of Sex combs reduced in Drosophila. Dev Biol 367(1):78–89
5. Damian M et al (2018) GHSR-D2R heteromerization modulates dopamine signaling through an effect on G protein conformation. In: Proceedings of the National Academy of Sciences
6. Moraes I et al (2014) Membrane protein structure determination - the next generation. Biochim Biophys Acta 1838(1 Pt A):78–87
7. Almeida JG et al (2017) Membrane proteins structures: a review on computational modeling tools. Biochim Biophys Acta 1859 (10):2021–2039
8. Melo R et al (2016) A machine learning approach for hot-spot detection at protein-protein interfaces. Int J Mol Sci 17(8):1215
9. Moreira IS et al (2017) SpotOn: high accuracy identification of protein-protein interface hot-spots. Sci Rep 7(1):8007
10. Bastanlar Y, Ozuysal M (2014) Introduction to machine learning. Methods Mol Biol 1107:105–128
11. Cook CE et al (2016) The European Bioinformatics Institute in 2016: data growth and integration. Nucleic Acids Res 44(Database issue): D20–D26
12. Greene CS et al (2016) Big data bioinformatics. Methods (San Diego, CA) 111:1–2
13. Gopinath RA, Burrus CS (1994) On upsampling, downsampling, and rational sampling rate filter banks. IEEE Trans Signal Process 42(4):812–824
14. Browne MW (2000) Cross-validation methods. J Math Psychol 44(1):108–132
15. Schumacher M, Hollander N, Sauerbrei W (1997) Resampling and cross-validation techniques: a tool to reduce bias caused by model building? Stat Med 16(24):2813–2827
16. Min S, Lee B, Yoon S (2017) Deep learning in bioinformatics. Brief Bioinform 18 (5):851–869
17. Hajian-Tilaki K (2013) Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. Caspian J Intern Med 4(2):627–635
18. McCammon JA, Gelin BR, Karplus M (1977) Dynamics of folded proteins. Nature 267:585
19. Mori T et al (2016) Molecular dynamics simulations of biological membranes and membrane proteins using enhanced conformational sampling algorithms. Biochim Biophys Acta Biomembr 1858(7, Part B):1635–1651
20. Neves RPP et al (2013) Parameters for molecular dynamics simulations of manganese-containing metalloproteins. J Chem Theory Comput 9(6):2718–2732
21. Coimbra JT et al (2014) Biomembrane simulations of 12 lipid types using the general Amber force field in a tensionless ensemble. J Biomol Struct Dyn 32(1):88–103
22. Sousa SF, Fernandes PA, Ramos MJ (2007) General performance of density functionals. J Phys Chem A 111(42):10439–10452
23. Comba P, Remenyi R (2003) Inorganic and bioinorganic molecular mechanics modeling—the problem of the force field parameterization. Coord Chem Rev 238–239:9–20
24. Nerenberg PS, Head-Gordon T (2018) New developments in force fields for biomolecular simulations. Curr Opin Struct Biol 49:129–138
25. Lopes PEM, Guvench O, MacKerell AD (2015) Current status of protein force fields

for molecular dynamics. Methods Mol Biol (Clifton, NJ) 1215:47–71

26. Lyubartsev AP, Rabinovich AL (2016) Force field development for lipid membrane simulations. Biochim Biophys Acta 1858 (10):2483–2497
27. Eichenberger AP et al (2011) GROMOS++ software for the analysis of biomolecular simulation trajectories. J Chem Theory Comput 7 (10):3379–3390
28. Chandrasekhar I et al (2003) A consistent potential energy parameter set for lipids: dipalmitoylphosphatidylcholine as a benchmark of the GROMOS96 45A3 force field. Eur Biophys J 32(1):67–77
29. Oostenbrink C et al (2004) A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. J Comput Chem 25(13):1656–1676
30. Poger D, Van Gunsteren Wilfred F, Mark Alan E (2009) A new force field for simulating phosphatidylcholine bilayers. J Comput Chem 31 (6):1117–1125
31. Berger O, Edholm O, Jähnig F (1997) Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure, and constant temperature. Biophys J 72(5):2002–2013
32. Chiu S-W et al (2009) An improved united atom force field for simulation of mixed lipid bilayers. J Phys Chem B 113(9):2748–2763
33. Jämbeck JP, Lyubartsev AP (2012) Derivation and systematic validation of a refined all-atom force field for phosphatidylcholine lipids. J Phys Chem B 116(10):3164–3179
34. Pastor RW, MacKerell AD (2011) Development of the CHARMM force field for lipids. J Phys Chem Lett 2(13):1526–1532
35. Zhu X, Lopes PEM, Mackerell AD (2012) Recent developments and applications of the CHARMM force fields. Wiley Interdiscip Rev Comput Mol Sci 2(1):167–185
36. Feller SE et al (1997) Molecular dynamics simulation of unsaturated lipid bilayers at low hydration: Parameterization and comparison with diffraction studies. Biophys J 73 (5):2269–2279
37. Feller SE, MacKerell AD Jr (2000) An improved empirical potential energy function for molecular simulations of phospholipids. J Phys Chem B 104(31):7510–7515
38. Klauda JB et al (2005) An ab initio study on the torsional surface of alkanes and its effect on molecular simulations of alkanes and a DPPC bilayer. J Phys Chem B 109(11):5300–5311
39. Klauda JB et al (2010) Update of the CHARMM all-atom additive force field for lipids: validation on six lipid types. J Phys Chem B 114(23):7830–7843
40. Lim JB, Rogaski B, Klauda JB (2012) Update of the cholesterol force field parameters in CHARMM. J Phys Chem B 116(1):203–210
41. Wang J et al (2004) Development and testing of a general Amber force field. J Comput Chem 25(9):1157–1174
42. Dickson CJ et al (2012) GAFFlipid: a General Amber Force Field for the accurate molecular dynamics simulation of phospholipid. Soft Matter 8(37):9617–9627
43. Ogata K, Nakamura S (2015) Improvement of parameters of the AMBER potential force field for phospholipids for description of thermal phase transitions. J Phys Chem B 119 (30):9726–9739
44. Skjevik AA et al (2012) LIPID11: a modular framework for lipid simulations using amber. J Phys Chem B 116(36):11124–11136
45. Dickson CJ et al (2014) Lipid14: the amber lipid force field. J Chem Theory Comput 10 (2):865–879
46. Maciejewski A et al (2014) Refined OPLS all-atom force field for saturated phosphatidylcholine bilayers at full hydration. J Phys Chem B 118(17):4571–4581
47. Marrink SJ et al (2007) The MARTINI force field: coarse grained model for biomolecular simulations. J Phys Chem B 111 (27):7812–7824
48. Marrink SJ, De Vries AH, Mark AE (2004) Coarse grained model for semiquantitative lipid simulations. J Phys Chem B 108 (2):750–760
49. Jämbeck JPM, Lyubartsev AP (2012) Derivation and systematic validation of a refined all-atom force field for phosphatidylcholine lipids. J Phys Chem B 116(10):3164–3179
50. Demerdash O, Wang LP, Head-Gordon T (2018) Advanced models for water simulations. Wiley Interdiscip Rev Comput Mol Sci 8(1):e1355
51. Jorgensen WL et al (1983) Comparison of simple potential functions for simulating liquid water. J Chem Phys 79(2):926–935
52. Neria E, Fischer S, Karplus M (1996) Simulation of activation free energies in molecular systems. J Chem Phys 105(5):1902–1921
53. Berweger CD, van Gunsteren WF, Müller-Plathe F (1995) Force field parametrization by weak coupling. Re-engineering SPC water. Chem Phys Lett 232(5–6):429–436

54. Berendsen HJC, Grigera JR, Straatsma TP (1987) The missing term in effective pair potentials. J Phys Chem 91(24):6269–6271
55. Wong-Ekkabut J, Karttunen M (2016) The good, the bad and the user in soft matter simulations. Biochim Biophys Acta Biomembr 1858 (10):2529–2538
56. Khalili-Araghi F et al (2013) Molecular dynamics simulations of membrane proteins under asymmetric ionic concentrations. J Gen Physiol 142(4):465–475
57. DeLano WL (2002) The PyMOL molecular graphics system. Delano Scientific, San Carlos, CA
58. Berman HM et al (2000) The protein data bank. Nucleic Acids Res 28(1):235–242
59. Case DA et al (2005) The Amber biomolecular simulation programs. J Comput Chem 26 (16):1668–1688
60. Brooks BR et al (2009) CHARMM: the biomolecular simulation program. J Comput Chem 30(10):1545–1614
61. Christen M et al (2005) The GROMOS software for biomolecular simulation: GROMOS05. J Comput Chem 26(16):1719–1751
62. Das A, Ali SM (2018) Molecular dynamics simulation for the test of calibrated OPLS-AA force field for binary liquid mixture of tri-isoamyl phosphate and n-dodecane. J Chem Phys 148(7):074502
63. Cock PJA et al (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25 (11):1422–1423
64. Webb B, Sali A (2014) Protein structure modeling with MODELLER. Methods Mol Biol 1137:1–15
65. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38
66. Cao DS et al (2013) PyDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. J Chem Inf Model 53(11):3086–3096
67. Chen Z et al (2018) iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics 34(14):2499–2502
68. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22(12):2577–2637
69. Leaver-Fay A et al (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487:545–574
70. Altschul SF et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25 (17):3389–3402
71. Ghahremanpour MM et al (2014) MemBuilder: a web-based graphical interface to build heterogeneously mixed membrane bilayers for the GROMACS biomolecular simulation program. Bioinformatics 30 (3):439–441
72. Jefferys E et al (2015) Alchembed: a computational method for incorporating multiple proteins into complex lipid geometries. J Chem Theory Comput 11(6):2743–2754
73. Ruymgaart AP, Elber R (2012) Revisiting molecular dynamics on a CPU/GPU system: Water Kernel and SHAKE parallelization. J Chem Theory Comput 8(11):4624–4636
74. Hess B, Bekker H, Berendsen HJC, Fraaije JG (1997) LINCS: a linear constraint solver for molecular simulations. J Comput Chem 18:1463–1472
75. Chen Y et al (2016) Structure of the STRA6 receptor for retinol uptake. Science 353 (6302):aad8266
76. Eswar N et al (2006) Comparative protein structure modeling using modeller. Curr Protoc Bioinformatics Chapter 5:Unit 5.6
77. Miller S et al (1987) Interior and surface of monomeric proteins. J Mol Biol 196 (3):641–656
78. Forst D et al (1998) Structure of the sucrose-specific porin ScrY from Salmonella typhimurium and its complex with sucrose. Nat Struct Biol 5:37
79. Chavent M, Duncan AL, Sansom MSP (2016) Molecular dynamics simulations of membrane proteins and their interactions: from nanoscale to mesoscale. Curr Opin Struct Biol 40:8–16
80. Goñi FM (2014) The basic structure and dynamics of cell membranes: an update of the Singer–Nicolson model. Biochim Biophys Acta Biomembr 1838(6):1467–1476
81. van Meer G, Voelker DR, Feigenson GW (2008) Membrane lipids: where they are and how they behave. Nat Rev Mol Cell Biol 9 (2):112–124
82. Kulig W, Pasenkiewicz-Gierula M, Rog T (2015) Topologies, structures and parameter files for lipid simulations in GROMACS with the OPLS-aa force field: DPPC, POPC, DOPC, PEPC, and cholesterol. Data Brief 5:333–336
83. Lee AG (2005) How lipids and proteins interact in a membrane: a molecular approach. Mol BioSyst 1(3):203–212

3.1.4. MENSAdb: a thorough structural analysis of membrane protein dimers

Database, 2021, 1–10
doi:10.1093/database/baab013
Database tool

Database tool

MENSAdb: a thorough structural analysis of membrane protein dimers

Pedro Matos-Filipe[1,†], António J. Preto[1,2,†], Panagiotis I. Koukos[3], Joana Mourão[1], Alexandre M.J.J. Bonvin[3] and Irina S. Moreira[4,5,*]

[1]Center for Neuroscience and Cell Biology, University of Coimbra, Coimbra 3005-504, Portugal, [2]PhD Programme in Experimental Biology and Biomedicine, Institute for Interdisciplinary Research, University of Coimbra, Coimbra, 3030-789, Portugal, [3]Bijvoet Centre for Biomolecular Research, Faculty of Science—Chemistry, Utrecht University, Utrecht, 3584, CH, Netherlands, [4]Department of Life Sciences, University of Coimbra, Coimbra, 3000-456, Portugal and [5]Center for Neuroscience and Cell Biology, Center for Innovative Biomedicine and Biotechnology, University of Coimbra, Coimbra, Portugal

*Corresponding author: Tel: +351-239240227 or +351- 231 249 170; Fax: +351- 231 249 179; Email: irina.moreira@cnc.uc.pt

†Co-first authors.

Citation details: Matos-Filipe, P., Preto, A.J., Koukos, P.I. *et al.* MENSAdb: a thorough structural analysis of membrane protein dimers. *Database* (2021) Vol. 2021: article ID baab013; doi:10.1093/database/baab013

Received 8 May 2020; Revised 19 January 2021; Accepted 1 March 2021

Abstract

Membrane proteins (MPs) are key players in a variety of different cellular processes and constitute the target of around 60% of all Food and Drug Administration–approved drugs. Despite their importance, there is still a massive lack of relevant structural, biochemical and mechanistic information mainly due to their localization within the lipid bilayer. To help fulfil this gap, we developed the MEmbrane protein dimer Novel Structure Analyser database (MENSAdb). This interactive web application summarizes the evolutionary and physicochemical properties of dimeric MPs to expand the available knowledge on the fundamental principles underlying their formation. Currently, MENSAdb contains features of 167 unique MPs (63% homo- and 37% heterodimers) and brings insights into the conservation of residues, accessible solvent area descriptors, average *B*-factors, intermolecular contacts at 2.5 Å and 4.0 Å distance cut-offs, hydrophobic contacts, hydrogen bonds, salt bridges, π–π stacking, T-stacking and cation–π interactions. The regular update and organization of all these data into a unique platform will allow a broad community of researchers to collect and analyse a large number of features efficiently, thus facilitating their use in the development of prediction models associated with MPs.

Database URL: http://www.moreiralab.com/resources/mensadb.

Introduction

Membrane proteins (MPs) account for around 15–39% of the human proteome (1, 2). They assume a critical role in a vast set of cellular and physiological mechanisms, including molecular transport, nutrient uptake, toxin and waste product clearance, respiration and signalling (3).

While roughly 60% of all Food and Drug Administration (FDA)–approved drugs target MPs, there is a shortage of structural and biochemical data about them mainly due to their localization in the lipid bilayer (4, 5). In the last years, a daunting challenge of drug discovery has been the development of compounds that can target the 'undruggable' regions of MPs, enabling the modulation of protein–lipid, protein–nucleic acid and protein–protein interactions (PPIs) (6, 7). In this respect, being able to characterize the structural and physicochemical properties of MPs as well as their interactions and interfaces is essential to develop improved and more targeted therapies as well as to discover new drug targets. Particular features of proteins, such as electrostatic interactions (8), hydrophobic effects (9) or 'hot-spot' residues (10–13), were shown to contribute to the affinity and specificity of PPIs. Other well-characterized properties of proteins are the evolutionary conservation and distribution of their amino acids. These two features contribute the most to the prediction of functionally essential residues, as highlighted by several publications (14–17). While many studies have dealt with soluble systems, there is a significant lack of in-depth analysis of MP complexes and their interactions.

We present here the MEmbrane protein dimer Novel Structure Analyser database (MENSAdb), the first interactive web application exposing a comprehensive and thorough array of fundamental features of dimer surfaces of MPs and their interfacial regions. Users can easily access a thorough, systematic analysis of sequence–structure relationships (Figure 1) based on a curated database of 201 protein dimers obtained from the Membrane Proteins of Known 3D structure (MPSTRUC) (18). MENSAdb delivers tabular and graphical data formats that can be visually explored for a large number of MP features based on conservation, accessible solvent area (ASA) descriptors, average *B*-factors, intermolecular contacts at 2.5 Å and 4.0 Å distance cut-offs, hydrophobic contacts, hydrogen bonds, salt bridges, π–π stacking, T-stacking and cation–π interactions. Additionally, users can inspect differences in these features between three distinctive residue classes: (i) non-surface, (ii) surface and non-interfacial and (iii) surface and interfacial. The web server relies on a custom front-end application that provides the results to the user. The resulting knowledge and full database can be easily assessed and downloaded.

Our main goal with the integration of these features into a single platform is to assist the development of prediction models associated with MPs, either for classification or for regression tasks, as well as to help researchers to better understand MP interfacial characteristics. Our database is freely available at www.moreiralab.com/resources/mensadb.

Materials and methods

Data collection and pre-processing

Experimental structures of 167 unique transmembrane (TM) proteins that included β-barrel TMs and α-helix TMs were obtained from MPSTRUC (http://blanco.biomol.uci.edu/mpstruc/) (18). These correspond to structures achieved mainly from X-ray crystallography (91%) or electron microscopy (4%), with a resolution below or equal to 4.50 Å, and less frequently from nuclear magnetic resonance (5%). We discarded all non-TM, monomeric and monotopic (not embedded in the lipid bilayer) proteins. Pre-processing of the database was performed by excluding dimers in which one of the chains was a soluble protein, single MPs interacting with small soluble peptides (protein–peptide), pores, protein–antibodies (since antibodies are soluble proteins) and proteins with small organic or non-organic ligands (protein–ligand). In the previous case, the complex was maintained if the presence of more than one MPs chain was observed. Additionally, structures with unknown residues or with many incomplete amino acids were also excluded, as were structures with interfaces interacting highly with lipids. Sequences were filtered to ensure at most 35% sequence redundancy in each interface by using the PISCES web server (19). The final database was composed of 63% ($n = 105/167$) homodimers and 37% ($n = 62/167$) heterodimers. From the Protein Data Bank (PDB) files, all possible dimer combinations were extracted for the structures in which the number of chains was higher than two (functional high-order oligomers) and it is constituted by 201 protein dimer combinations (Supplementary File 1). The selected structures were then subjected to further processing. In particular, we (i) identified and removed residues outside the TM domain according to the MPSTRUC (18) annotation of α-helix and β-barrel amino acids available in the PDB (20) in conjunction with visual inspection; (ii) removed unnecessary heteroatoms; (iii) reversed mutated non-standard amino acids (e.g. selenomethionine was mutated to methionine); and (iv) added hydrogens to the structures. In-house PyMOL (20) and Visual Molecular Dynamics (VMD) scripts (21) were used to perform these pre-processing steps.

Definition of interfacial and non-interfacial residues

The relative solvent accessibility (RSA) defined as the ratio between an amino acid ASA value and its corresponding area in a Gly-X-Gly peptide was calculated using an in-house pipeline with Database of Secondary Structure

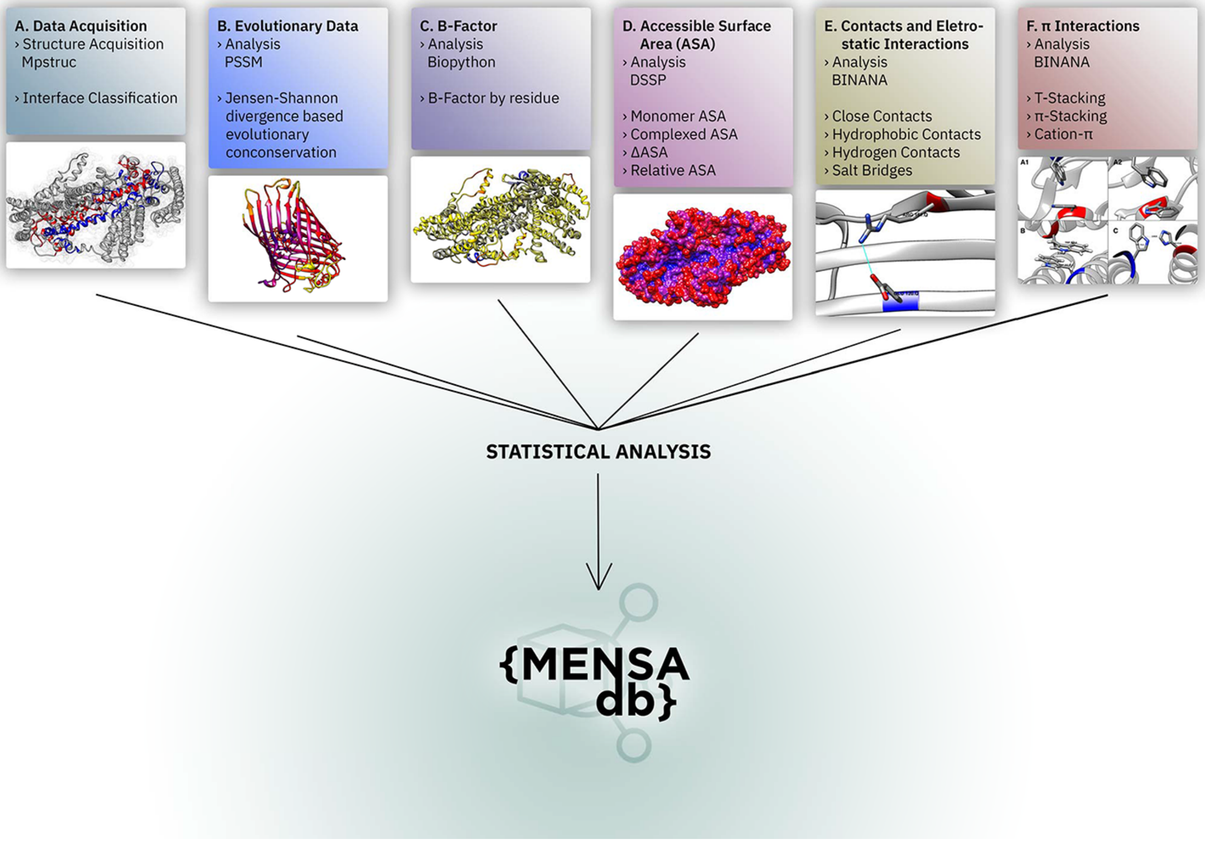


Figure 1. Overall representation of MENSAdb. Boxes A–F illustrate the steps involving the data collection, evolutionary conservation, *B*-factor, accessible surface and PPI analysis. Each box contains an example of the proteinic motifs under the scope of this work. (A) Interface between chains A and B of the STRA6 receptor for retinol uptake in *Danio rerio* (PDBid: 5SY1) (55). (B) Representation of evolutionary conservation of protein motifs (purple being more conserved and yellow less conserved) in the chain P of a hedgehog auto-processing domain in *Drosophila melanogaster* (PDBid: 1AT0) (56). (C) and (D) Average *B*-factor and complexed accessible surface area, respectively, of the chains A and B of 5SY1 (55). (E) Salt bridge between GLU120 and ARG161 of the chain Q of the sucrose-specific porin (PDBid: 1A0T) of *Salmonella typhimurium* (57). (F) The spectrum of π systems predicted: (A1 and A2) T-stacking motif between TRP25 (chain L) and TRP255 (chain M) from *Rattus norvegicus* S100B protein (PDBid: 1XYD) (58) is represented from two perspectives; (B) illustration of a π–π stacking structure between TRP262 (chain A) and TRP262 (chain B) from *Archaeoglobus fulgidus* CDP-alcohol phosphotransferase (PDBid: 4O6M) (59) and (C) cation–π interaction between HIS275 (chain B) and TRP175 (chain C) from *Escherichia coli* formate dehydrogenase-N (PDNid: 1KQF) (60).

assignments for all Proteins entries (DSSP) (22). Residues above a 0.20 RSA cut-off were considered as surface residues (23). We obtained 55 008 possible surface residues from a total of 91 861, while the remaining ones were considered core residues. Secondly, we considered those for which the pairwise distance between any atom of chain A and any atom of chain B was below 5 Å as interfacial residues, splitting surface residues into two classes: interfacial (15 277 residues) and non-interfacial ones (39 731 residues).

Determination of sequence and structural features of all residues

Evolutionary conservation of all sites was calculated using the Jensen–Shannon divergence (JSD) measure, a symmetrized and smoothed version of the Kullback–Leibler divergence (24), of the Position-Specific Scoring Matrix (PSSM), which itself was calculated with a local deployment of PSI-BLAST against the NCBI non-redundant database with parameters num_iterations = 3 and evalue = 0.001 (25). Equation 1 was used to quantify the similarity between two probability distributions and compares the amino acid distribution observed in PSSM p_{ia} with a background distribution f_a.

$$JSD = H\left(\frac{p_{ia}+p_a}{2}\right) - \frac{1}{2}H(p_{ia}) - \frac{1}{2}H(f_a) \qquad (1)$$

$H(\cdot)$ denotes the entropy of amino acid distribution. The code provided by Capra *et al.* was introduced into the pipeline due to its high performance in comparison with other methods (16). This metric works on the premise that the highest JSD value corresponds to a more conserved residue. We tested three different background distributions, BLOSUM62 (the PSI-BLAST default one), SLIM (26) and bbTM (27) to assess which one of them was the most suitable for MPs interface prediction. SLIM is a non-symmetric matrix optimized for TM protein segments, whereas bbTM is a set of matrices optimized for β-barrel proteins that uses three different matrices (one for intracellular segments, one for extracellular segments and another for TM residues). Herein, we only used the matrix developed for TM segments, since the remaining residues were already excluded from the analysis. We also generated a new column named 'appropriate JSD' in which we selected SLIM and bbTM depending on the presence and absence of an α-helix or β-barrel protein, respectively.

DSSP was used to calculate the RSA not only in the complexed form but also in the monomeric form, which were then multiplied by Sander and Rost amino acid constants (ALA: 106, ARG: 248, ASN: 157, ASP: 163, CYS:135, GLN: 198, GLU: 194, GLY: 84, HIS: 184, ILE: 169, LEU: 165, LYS: 205, MET: 188, PHE: 197, PRO: 136, SER: 130, THR: 142, TRP: 227, TYR: 222 and VAL: 142) (28) to calculate ASA of each amino acid, 'i', in the complexed ($_{\mathrm{comp}}\mathrm{ASA}_{\mathrm{i}}$) and monomeric ($_{\mathrm{mon}}\mathrm{ASA}_{\mathrm{i}}$) systems, respectively. These values were also used to calculate $\Delta\mathrm{ASA}_{\mathrm{i}}$ (Equation 2).

$$\Delta ASA_i = compASA_i - mon\ ASA_i \tag{2}$$

For further clarification, we also listed all $_{\mathrm{rel}}\mathrm{ASA}_{\mathrm{i}}$ values (Equation 3), which allows the differentiation of residues with equal $\Delta\mathrm{ASA}_{\mathrm{i}}$ but with different absolute monomer ASA values (29–31).

$$rel\ ASA_i = \frac{\Delta ASA_i}{mon\ ASA_i} \tag{3}$$

We also extracted the temperature factor (*B*-factor) value for each residue from the PDB file of the analysed structures (obtained directly from MPSTRUC) using Biopython (32).

Determination of structural descriptors of MP–protein interface

Close and hydrophobic contacts, hydrogen bonds, salt bridges and π-interactions (π–π stacking, T-stacking and cation–π interactions) were described using BINANA—Binding Analyzer, a Python-implemented algorithm that characterizes protein complexes (33). Close contacts correspond to the number of pairs of atoms formed within 2.5 and 4.0 Å radius.

Data treatment

Since the composition of the database was not equally distributed across the three classes of MPs presented here, we defined a correction factor (C_{factor}), Equation 4, based on the concept of propensity score calculation, as shown by Huang (34). This factor is defined as the ratio between the frequency of occurrence of residue *i* in each one of the classes (fi_{CLAS}) and the frequency of occurrence of the total number of amino acids in that class (fi_{TOT}). The obtained MP-class-specific C_{factor} was used to correct the various metrics described in the 'Results' section by multiplying them by their respective C_{factor} except that of $_{\mathrm{rel}}\mathrm{ASA}$.

$$C_{factor} = \frac{fi_{CLAS}}{fi_{TOT}} \tag{4}$$

Statistics

For all plots, residues are ordered by increasing hydrophobicity based on the Kyte and Doolittle hydropathy index (35). Descriptive statistics such as three quartiles (Q1, Q2 and Q3), average and standard deviation were obtained using Pandas, a Python library (36). *P*-values were calculated through SciPy (https://docs.scipy.org/) with the independent *t*-test and one-way ANOVA. Further statistics were calculated for amino acids sets split according to the hydrophilic and hydrophobic potential as (i) charged—Asp, Glu, Lys and Arg; (ii) positively charged—Lys and Arg; (iii) negatively charged—Asp and Glu; (iv) polar—Ser, Thr, Asn, Gln, Tyr and His; (v) non-polar—Ala, Val, Ile, Leu, Met, Phe and Trp; aromatic—Phe, Trp and Tyr. Cys, Gly and Pro were not included in those subsets.

Code availability

MENSAdb code used for all the structural and physicochemical analyses of MP dimers is freely distributed as a GitHub repository at https://github.com/MoreiraLAB/mensadb-open. The available Python code allows users to perform feature extraction using a pre-processed PDB file easily. For detailed information on all the pre-processing steps (trimming of non-TM residues, removal of heteroatoms, mutation of exotic residues, modelling of incomplete structures and dimer extraction from the structure files), please see Preto *et al.* (37). The addition of hydrogens was implemented within the pipeline available in the GitHub repository. The original code was tested in a 64-bit version of Linux Ubuntu 18.04 (Intel Xeon 40 Core 2.2 GHz, 126 GB RAM) and required the installation of Python version 3.7.2 with the following free and open-source packages: NumPy ≥ 1.16.1, pandas ≥ 0.23.4, vmd-python ≥ 3.0.6, dit ≥ 1.2.3, Biopython ≥ 1.7.3 and standalone software: BLAST+ ≥ 2.9.0, BINANA ≥ 1.2.0, DSSP ≥ 3.0.7, MGTools ≥ 1.5.6 and AutoDock ≥ 3.0.7. The JSD measure we determined using a non-redundant protein database for comparison (for download options, please see https://ftp.ncbi.nlm.nih.gov/blast/db/).

Database development

Data resulting from this work are available through MENSAdb (www.moreiralab.com/resources/mensadb), without the need for login, registration or license, a rich data visualization web application built using Python's 'Flask'-based 'Dash' visualization framework (by 'Plotly'). MENSAdb's real-time query features are supported by a MongoDB back end, which enables the application to query, filter and aggregate the data in multiple meaningful ways. To boost performance, a 'Flask' caching layer is applied to support the complex queries required for visualization. To further ensure performance and security and support high-availability scenarios, all HTTP traffic directed at MENSAdb is served by the NGINX high-performance webserver and load balancer, which then routes it to multiple MENSAdb application instances. The final database of MENSAdb containing all the raw data of structural and physicochemical properties of MPs is publicly available from Figshare (Data Citation 1; dx.doi.org/10.6084/m9.figshare.7808909), and the full membrane dimer structures listed according to PDB code can be found in Supplementary File 1.

Results and discussion

MP dimer composition and characteristics

The overall residue distribution in Figure 2A and B shows that MPs have a higher content of hydrophobic and aromatic residues, such as leucine (13.2%), alanine (9.4%), valine (8.6%), glycine (8.4%), isoleucine (8.3%) and phenylalanine (6.9%) that account for 54.8% of all detected residues. For a better clarification the percentages presented in this sub-section, oppositely to remaining sub-sections are listed without correction factor. Indeed, these residues were shown to contribute the most to the accuracy of machine learning (ML) models developed for predicting protein–protein binding sites (38). This high content in hydrophobic residues, also previously reported in other studies (14, 38–43), is essential since it favours the thermodynamic interactions with the lipid bilayer. Figure 2A and B also show that GAS residues are significantly enriched at the MPs core (12.3%) and non-interfacial surface locations (8.5%), in comparison to interfacial surface (3.0%). These small residues are the strong driving force for membrane folding (44, 45). As expected, charged residues (arginine, aspartate, glutamate and lysine) are typically excluded from the MPs interface (surface: 7.4%; core: 2.6%; interface: 2.3%).

Evolutionary conservation of protein sequences is a key feature to better understand and characterize the functionally and structurally important residues at PPIs. Herein, we used three different background matrices to calculate conservation, namely BLOSUM62 (PSSM_JSD), SLIM and bbTM as well as the 'appropriate_JSD'. Figure 3 illustrates their distribution split into three different protein regions: core/non-surface, interfacial surface and non-interfacial

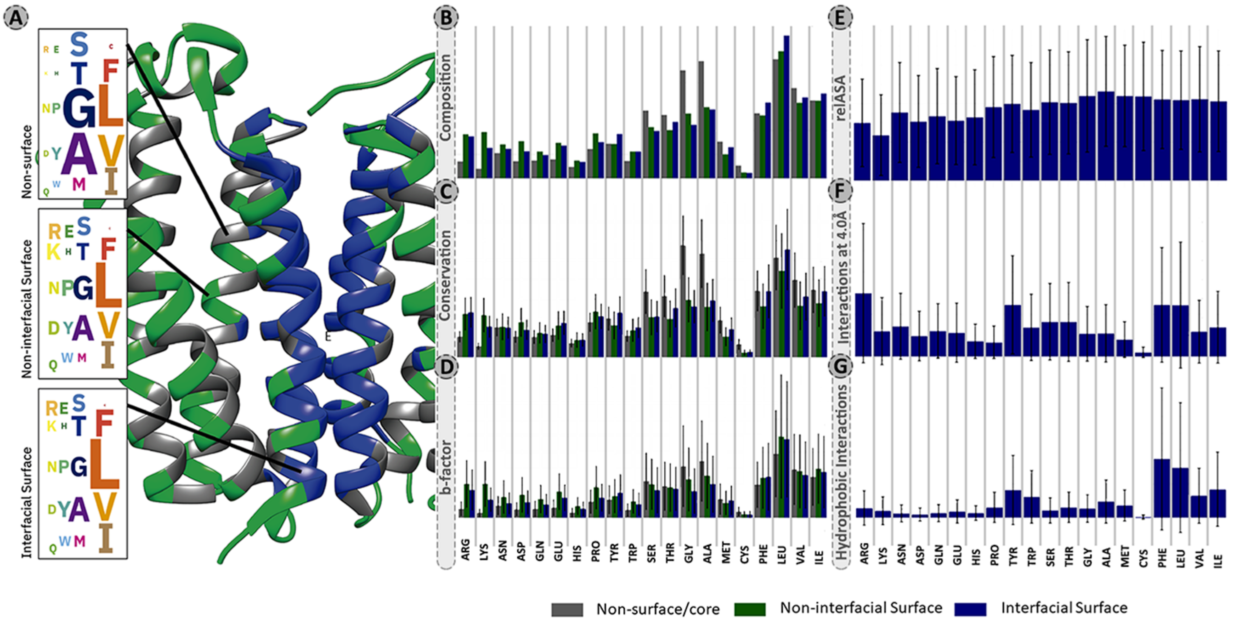


Figure 2. Panel of selected structural and physicochemical properties of MPs and their interactions. (A)—residue distribution of the translocator membrane protein (PDBid: 4UC1) from *Rhodobacter sphaeroides* (61). Amino acids are coloured according to the protein region within which they are embedded: grey—non-surface residues; green—non-interfacial surface residues; blue—interfacial surface residues. (B)—residue composition of the database. The correction factor described in section "Data treatment" of Material and methods was not applied here. (C)—normalized evolutionary conservation scores. (D)—normalized B-factor scores. (E)—normalized $_{rel}$ASA. (F)—normalized intermolecular contacts at 4 Å. (G)—normalized hydrophobic contacts.

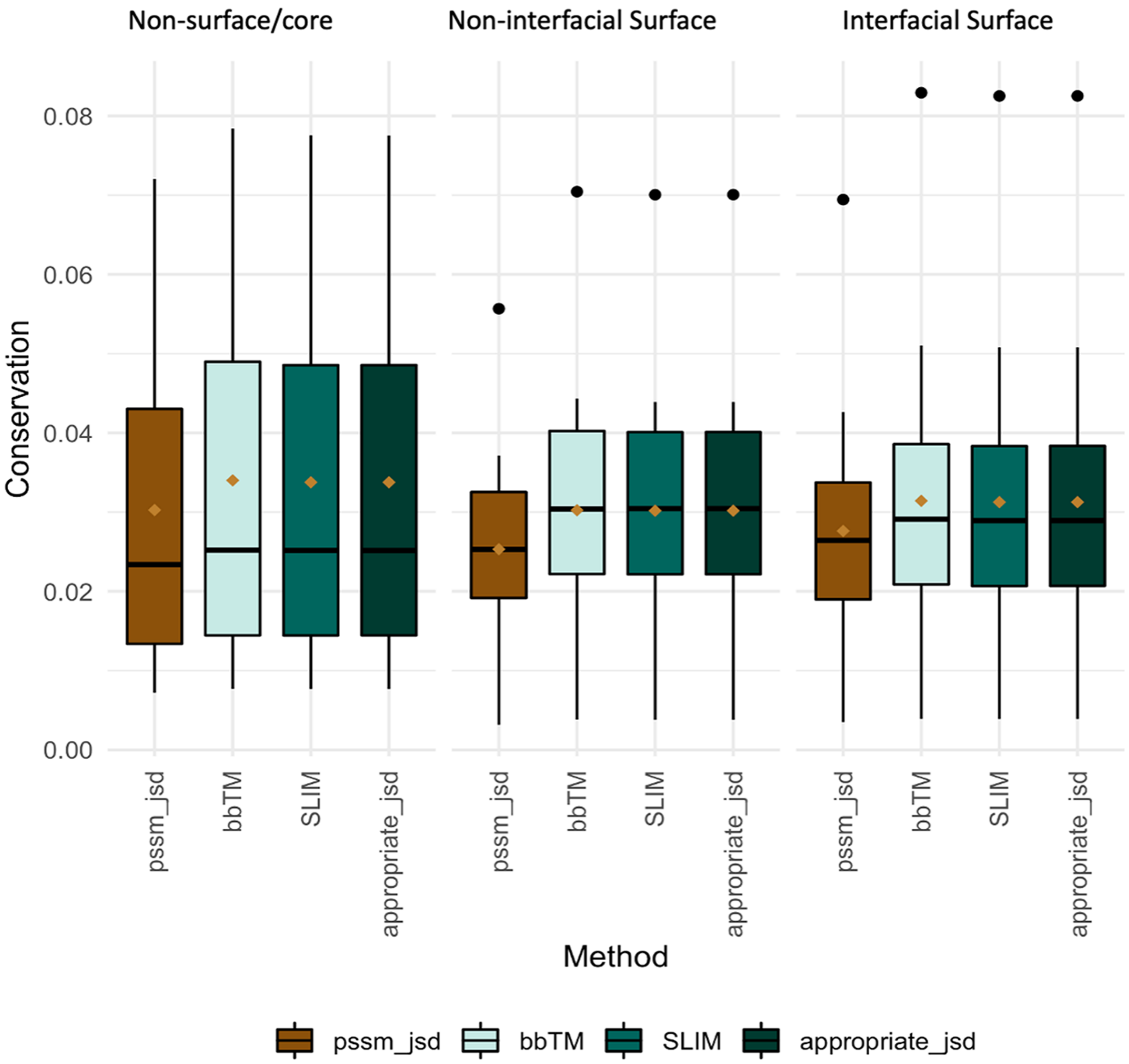


Figure 3. Conservation JSD distribution using BLOSUM62, SLIM, bbTM and the appropriate JSD background matrices (SLIM and bbTM were considered for α-helix and β-barrel proteins, respectively). Mean values are represented as a brown diamond. The results from the multiple pairwise test against all three background matrices yielded non-significant.

surface. The three different background matrices yielded similar results, which were non-significant according to multiple pairwise test. The same pattern was observed for all, with conservation being lower for surface, followed by interface and then protein core. As the used background matrix does not change the main conclusions about conservation at a MP dimer, we decided to follow up with the BLOSUM62 matrix for an easier implementation by the reader. Figure 2C reveals that for MPs, the more conserved JSD normalized values were found in the non-surface (0.05 ± 0.03) and in the interface (interface: 0.04 ± 0.02, surface non-interfacial 0.03 ± 0.02). The highest differences were for the GAS residues of the core region (core: 0.06 ± 0.03, surface: 0.03 ± 0.02; interface: 0.03 ± 0.02) and for the non-polar residues at protein core (core: 0.05 ± 0.02; surface: 0.04 ± 0.02; interface: 0.05 ± 0.03). These results, albeit not remarked different, support that the core and the interface are the most conserved regions, granting the necessary structural stability at specific PPIs, as previously observed (46). Additional results are available in the 'Conservation' option in the MENSAdb webserver.

B-factor (Figure 2D), related to the displacement of an atom from its reference position due to thermal motion and positional disorder (47), is typically used in a variety of applications including as a measure of atoms mobility for

PPIs prediction (48, 49). We observed a decrease in normalized average *B*-factor values of the interfacial residues compared to the non-interfacial surface ones (5.71 ± 6.10 Å^2 vs 6.25 ± 6.16 Å^2), putting their average closer to the non-surface MP residues (6.02 ± 5.69 Å^2). Also, positively charged residues are one of the most dissimilar ones (*B*-factor core: 1.19 ± 0.96 Å^2; *B*-factor surface: 5.34 ± 3.72 Å^2; *B*-factor interface: 3.74 ± 2.86 Å^2). This is in agreement with the fact that residues participating in PPIs are usually less flexible in comparison with the ones from the surface, which is reflected in lower *B*-factor values (49–51). Leucine, very conserved at the interface, seems to also have a higher mobility at PPI-associated locations (surface: 12.66 ± 9.86 Å^2; interface: 12.25 ± 9.52 Å^2 vs core: 9.88 ± 6.83 Å^2). Previous studies have suggested that leucine and isoleucine have an important role in flexible loop-mediated PPIs (52). Users can find illustrative plots of average *B*-factor values (by residue) in the 'Average *B*-factor' option in the MENSAdb web server.

The ASA descriptors detect protein regions that, when interacting or aggregating, lose solvent accessible area, while relASA indicates the relative exposed solvent surface area. MENSAdb and Figure 2E show that $_{\text{rel}}$ASA, which is the fraction of ΔASA by $_{\text{mon}}$ASA, is increased upon complex formation. These seem to be particularly relevant for non-polar residues (core: 5.27 ± 19.78 Å^2; surface: 0.00 ± 0.09 Å^2; interface: 52.01 ± 32.49 Å^2). Additional and detailed information about 'Monomer Accessible Surface Area' ($_{\text{mon}}$ASA), 'Complex Accessible Surface Area' ($_{\text{comp}}$ASA), 'Delta Accessible Surface Area' (ΔASA) and 'Relative Accessible Surface Area' ($_{\text{rel}}$ASA) can be viewed in MENSAdb web server options.

Characteristics of interfacial residues

Identification and characterization of critical features of membrane PPI dimers can provide important clues to pinpoint residues or interactions, important for drug development. For this, additional interfacial structural characteristics were quantified to better understand MP dimers. Concerning the intermolecular atomic contacts per amino acid type, we observed that the aromatic residues (Figure 2F, corrected contacts at 4 Å: 0.56 ± 0.61) are much more prone to establish close contacts at short distance than other residues. Arg was also highlighted in our results (corrected contacts at 4 Å: 0.75 ± 0.82). For further information, check the 'Interactions at 2.5 Angstroms' and 'Interactions at 4.0 Angstroms' options in the MENSAdb web server.

Hydrophobicity involving large aromatic residues is key in MP dimers and aromatic residues, and non-polar residues show a high number of hydrophobic contacts (Figure 2G, aromatic: 0.25 ± 0.34 and non-polar: 0.23 ± 0.32). In particular, Phe and Tyr establish π–π stacking, T-stacking and cation–π interactions in different dimers. Cation–π interactions are also particularly relevant for Arg (for a closer detailed view, please see the 'Hydrophobic Interactions', 'Pi–Pi Interactions', 'T-Stacking Interactions' and 'Cation–Pi Interactions' options in the MENSAdb).

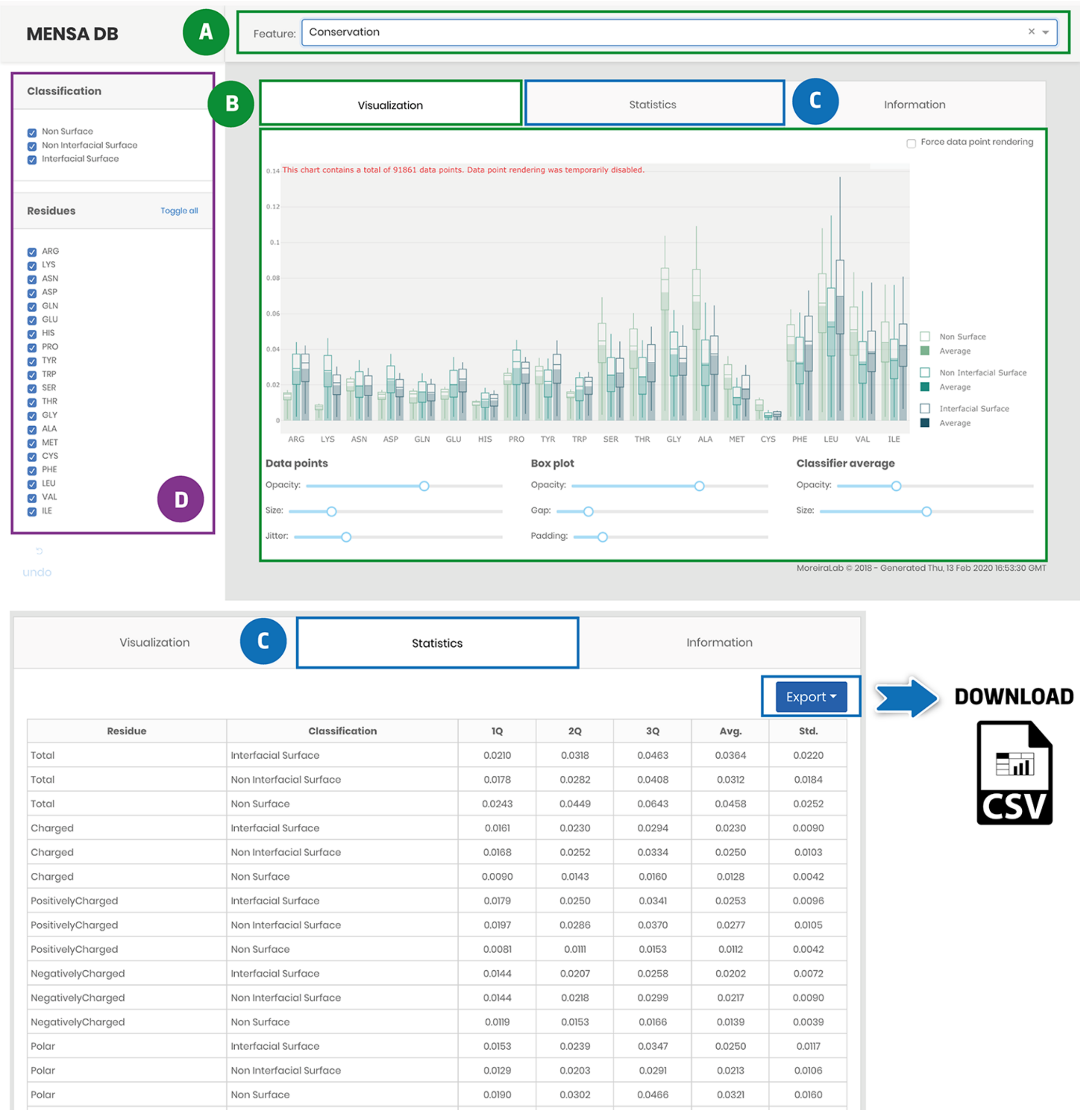
Additionally, although MP residues reside in a non-polar (low dielectric) environment (8, 53), both salt bridges between charged residues and hydrogen bonds through almost all amino acids are common to stabilize the interface and promote complex formation. Hydrogen bonds measured here involving both side chains and backbone are particularly important not only for charged residues (0.01 ± 0.03) but also for aromatic ones (0.01 ± 0.02), in particular tyrosine (0.01 ± 0.03) and tryptophan (0.01 ± 0.01). For a closer detailed view, please see the 'Salt-bridge Interactions' and 'Hydrogen-bond Interactions' options in the MENSAdb web server.

All different values presented herein showed statistical relevance.

All the results presented herein were obtained under the assumption that the interfaces in this study were biologically relevant, and utmost care was taken to ensure this (Supplementary File 1). Further limitations could arise from possible crystallographic artefacts.

MENSAdb interface and usability

The developed application enables users to explore the MP-dimer database (Figure 4). Access to evolutionary and physicochemical features is provided through a drop-down menu on the main page (Figure 4A). The data are presented in downloadable box plots for visual inspection that can be easily changed, for example, by filtering, zooming or panning (Figure 4B). Besides, data-associated statistics are also accessible in a tabular format [Q1, Q2, Q3, Average (Avg.) and Standard Deviation (Std.)] (Figure 4C). Stats and raw data can be downloaded as a .csv file using the export button for further reuse and integration in other studies. Users can also filter data for each selected feature by classification (non-surface, non-interfacial surface and interfacial surface) or residue type (Figure 4D). The database also has an 'Information' tab with general information for each included feature and a brief description of the underlying methods for their acquisition and pre-processing to help first-time users. MENSAdb will continue to be updated at least annually, and we expect, shortly, to integrate a new model for the prediction of MP interfaces.



Feature: Conservation

This chart contains a total of 91861 data points. Data point rendering was temporarily disabled.

MoreiraLab © 2018 - Generated Thu, 13 Feb 2020 16:53:30 GMT

Residue	Classification	1Q	2Q	3Q	Avg.	Std.
Total	Interfacial Surface	0.0210	0.0318	0.0463	0.0364	0.0220
Total	Non Interfacial Surface	0.0178	0.0282	0.0408	0.0312	0.0184
Total	Non Surface	0.0243	0.0449	0.0643	0.0458	0.0252
Charged	Interfacial Surface	0.0161	0.0230	0.0294	0.0230	0.0090
Charged	Non Interfacial Surface	0.0168	0.0252	0.0334	0.0250	0.0103
Charged	Non Surface	0.0090	0.0143	0.0160	0.0128	0.0042
PositivelyCharged	Interfacial Surface	0.0179	0.0250	0.0341	0.0253	0.0096
PositivelyCharged	Non Interfacial Surface	0.0197	0.0286	0.0370	0.0277	0.0105
PositivelyCharged	Non Surface	0.0081	0.0111	0.0153	0.0112	0.0042
NegativelyCharged	Interfacial Surface	0.0144	0.0207	0.0258	0.0202	0.0072
NegativelyCharged	Non Interfacial Surface	0.0144	0.0218	0.0299	0.0217	0.0090
NegativelyCharged	Non Surface	0.0119	0.0153	0.0166	0.0139	0.0039
Polar	Interfacial Surface	0.0153	0.0239	0.0347	0.0250	0.0117
Polar	Non Interfacial Surface	0.0129	0.0203	0.0291	0.0213	0.0106
Polar	Non Surface	0.0190	0.0302	0.0466	0.0321	0.0160

Figure 4. Main landing page of MENSAdb web server. Screenshot of the home page (A)—quickly query by evolutionary or physicochemical features. (B)—In the visualization tab, the results are shown in a graphical format. Users can easily change visual properties (opacity, size, jitter, gap and padding) by interacting with the lower panel. (C)—Statistics tab displays the data in a tabular format with associated metrics (Q1, Q2, Q3, Average-Avg. and Standard Deviation-Std.). Stats and raw data can be downloaded using the Export button in the top right corner, as a .csv file. (D)—In the left panel, users can filter graphic data by classification and residue type.

MENSAdb is the first comprehensive resource dedicated explicitly to exposing the evolutionary and physicochemical features of dimeric MP structures. Our main goal with the integration of these features into a single platform is to assist the development of experimental and computational assays, relevant for a better understanding of dimeric MP interactions and interfaces of this largest but poorly studied type of proteins. In the last years, some studies used evolutionary and physicochemical properties similar to the ones provided in our database to train ML for the prediction of MP complex binding sites (38, 46, 54). Nevertheless, as far as we know, herein we offer original features such as the ones from membrane PPI analysis not yet used or provided by other databases more dedicated to MP structures (PDBTM, OPM, MemProtMD and, MPSTRUC) or classification (TCDB).

Supplementary data

Supplementary data are available at *Database* online.

Acknowledgements

A.J.P. was also supported by FCT through PhD scholarship SFRH/BD/144966/2019. I.S.M. was funded by the FCT Investigator programme—IF/00578/2014 (co-financed by the European Social Fund and Programa Operacional Potencial Humano). A.M.J. and P.I.K acknowledge funding from the Dutch Foundation for Scientific Research (NWO) (TOP-PUNT grant 718.015.001). The authors would also like to acknowledge ERNEST (European Research Network on Signal Transduction, CA18133) and STRATAGEM (New diagnostic and therapeutic tools against multidrug-resistant tumours, CA17104).

Funding

COMPETE 2020—Operational Programme for Competitiveness and Internationalisation—and Portuguese national funds via FCT—Fundação para a Ciência e a Tecnologia—under projects POCI-01-0145-FEDER-031356, PTDC/QUI-OUT/32243/2017 and UIDB/04539/2020.

Author contributions

P.M.F., A.J.P, P.I.K. and I.S.M performed the acquisition of data. P.M.F., A.J.P, J.M. and I.S.M. processed the data. I.S.M. and A.M.J. conceived the study. All authors wrote and approved the final version of the manuscript.

Conflict of interest. None declared.

Data citation

Matos-Filipe,P., Preto,A.J., Koukos,P.I., Mourão,J., Bonvin,A.M.J.J., Moreira,I.S. *figshare* dx.doi.org/10.6084/m9.figshare.7808909.

References

1. Almén,M.S., Nordström,K.J.V., Fredriksson,R. *et al.* (2009) Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biol.*, **7**, 50.
2. Michael Gromiha,M. and Ou,Y.Y. (2014) Bioinformatics approaches for functional annotation of membrane proteins. *Brief. Bioinformatics*, **15**, 155–168.
3. Tan,S., Tan,H.T. and Chung,M.C.M. (2008) Membrane proteins and membrane proteomics. *Proteomics*, **8**, 3924–3932.
4. Overington,J.P., Al-Lazikani,B. and Hopkins,A.L. (2006) How many drug targets are there? *Nat. Rev. Drug Discov.*, **5**, 993–996.
5. Yıldırım,M.A., Goh,K.-I., Cusick,M.E. *et al.* (2007) Drug-target network. *Nat. Biotechnol.*, **25**, 1119–1126.
6. Yin,H. and Flynn,A.D. (2016) Drugging membrane protein interactions. *Annu. Rev. Biomed. Eng.*, **18**, 51–76.
7. Feng,Y., Wang,Q. and Wang,T. (2017) Drug target protein-protein interaction networks: a systematic perspective. *Biomed. Res. Int.*, **2017**, 1289259.
8. Zhang,Z., Witham,S. and Alexov,E. (2011) On the role of electrostatics on protein-protein interactions. *Phys. Biol.*, **8**, 035001.
9. Chanphai,P., Bekale,L. and Tajmir-Riahi,H.A. (2015) Effect of hydrophobicity on protein-protein interactions. *Eur. Polym. J.*, **67**, 224–231.
10. Darnell,S.J., LeGault,L. and Mitchell,J.C. (2008) KFC server: interactive forecasting of protein interaction hot spots. *Nucleic Acids Res.*, **36**, 265–269.
11. Moreira,I.S., Fernandes,P.A. and Ramos,M.J. (2007) Hot spots–a review of the protein-protein interface determinant amino-acid residues. *Proteins Struct. Funct. Bioinform.*, **68**, 803–812.
12. Moreira,I.S., Koukos,P.I., Melo,R. *et al.* (2017) SpotOn: high accuracy identification of protein-protein interface hot-spots. *Sci. Rep.*, **7**, 1–11.
13. Rosell,M. and Fernández-Recio,J. (2018) Hot-spot analysis for drug discovery targeting protein-protein interactions. *Expert. Opin. Drug. Discov.*, **13**, 327–338.
14. Ulmschneider,M.B. and Sansom,M.S.P. (2001) Amino acid distributions in integral membrane protein structures. *Biochimica Et Biophysica Acta Biomembranes*, **1512**, 1–14.
15. Caffrey,D.R. (2004) Are protein-protein interfaces more conserved in sequence than the rest of the protein surface? *Protein Sci.*, **13**, 190–202.
16. Capra,J.A. and Singh,M. (2007) Predicting functionally important residues from sequence conservation. *Bioinformatics*, **23**, 1875–1882.
17. Zhang,Q.C., Petrey,D., Norel,R. *et al.* (2010) Protein interface conservation across structure space. *Proc. Natl. Acad. Sci.*, **107**, 10896–10901.
18. White,S.H. (2009) Biophysical dissection of membrane proteins. *Nature*, **459**, 344–346.
19. Wang,G. and Dunbrack,R.L. (2003) PISCES: a protein sequence culling server. *Bioinformatics*, **19**, 1589–1591.
20. DeLano,W.L. (2015) The PyMOL molecular graphics system. Version 2.2 Schrödinger, LLC.
21. Humphrey,W., Dalke,A. and Schulten,K. (1996) VMD: visual molecular dynamics. *J. Mol. Graph.*, **14**, 33–38.
22. Touw,W.G., Baakman,C., Black,J. *et al.* (2015) A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.*, **43**, D364–D368.
23. Lins,L., Thomas,A. and Brasseur,R. (2003) Analysis of accessible surface of residues in proteins. *Protein Sci.*, **12**, 1406–1417.
24. Lin,J. (1991) Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory*, **37**, 145–151.
25. Altschul,S.F., Madden,T.L., Schäffer,A.A. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
26. Müller,T., Rahmann,S. and Rehmsmeier,M. (2001) Non-symmetric score matrices and the detection of homologous transmembrane proteins. *Bioinformatics (Oxford, England)*, **17**, S182–S189.
27. Jimenez-Morales,D., Adamian,L. and Liang,J. (2008) Detecting remote homologues using scoring matrices calculated from the estimation of amino acid substitution rates of beta-barrel membrane proteins. *Annu. Int. Conf. IEEE Eng. Med. Biol Soc.*, **2008**, 1347–1350.

28. Rost,B. and Sander,C. (1994) Conservation and prediction of solvent accessibility in protein families. *Proteins Struct. Funct. Bioinform.*, **20**, 216–226.
29. Melo,R., Fieldhouse,R., Melo,A. *et al.* (2016) A machine learning approach for hot-spot detection at protein-protein interfaces. *Int. J. Mol. Sci.*, **17**, 1–14.
30. Munteanu,C.R., Pimenta,A.C., Fernandez-Lozano,C. *et al.* (2015) Solvent accessible surface area-based hot-spot detection methods for protein-protein and protein-nucleic acid interfaces. *J. Chem. Inf. Model*, **55**, 1077–1086.
31. Martins,J.M., Ramos,R.M., Pimenta,A.C. *et al.* (2014) Solvent-accessible surface area: how well can be applied to hot-spot detection? *Proteins Struct. Funct. Bioinform.*, **82**, 479–490.
32. Cock,P.J.A., Antao,T., Chang,J.T. *et al.* (2009) Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.
33. Durrant,J.D. and McCammon,J.A. (2011) BINANA: a novel algorithm for ligand-binding characterization. *J. Mol. Graph. Model.*, **29**, 888–893.
34. Huang,H.L. (2014) Propensity scores for prediction and characterization of bioluminescent proteins from sequences. *PLoS ONE*, **9**, e97158.
35. Kyte,J., Doolittle,R.F., Diego,S. *et al.* (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.
36. McKinney,W. (2010) Data structures for statistical computing in python. In: Van Der Walt S, Millman J (eds). *Proceedings of the 9th Python in Science Conference*. Austin, Texas, USA, pp. 51–56.
37. Preto,A., Matos-Filipe,P., Koukos,P. *et al.* (2019) Structural characterization of membrane protein dimers. In: Kister AE (ed). *Protein Supersecondary Structures. Methods in Molecular Biology*. Humana Press, New York, NY, pp. 403–436.
38. Bordner,A.J. (2009) Predicting protein-protein binding sites in membrane proteins. *BMC Bioinform.*, **10**, 312.
39. Eilers,M., Patel,A.B., Liu,W. *et al.* (2002) Comparison of helix interactions in membrane and soluble alpha-bundle proteins. *Biophys. J.*, **82**, 2720–2736.
40. Saidijam,M., Azizpour,S. and Patching,S.G. (2018) Comprehensive analysis of the numbers, lengths and amino acid compositions of transmembrane helices in prokaryotic, eukaryotic and viral integral membrane proteins of high-resolution structure. *J. Biomol. Struct. Dyn.*, **36**, 443–464.
41. Mbaye,M.N., Hou,Q., Basu,S. *et al.* (2019) A comprehensive computational study of amino acid interactions in membrane proteins. *Sci. Rep.*, **9**, 12043.
42. Duarte,J.M., Biyani,N., Baskaran,K. *et al.* (2013) An analysis of oligomerization interfaces in transmembrane proteins. *BMC Struct. Biol.*, **13**, 1–11.
43. Yan,C., Wu,F., Jernigan,R.L. *et al.* (2008) Characterization of protein-protein interfaces. *Protein J.*, **27**, 59–70.
44. Zhang,S.Q., Kulp,D.W., Schramm,C.A. *et al.* (2015) The membrane- and soluble-protein helix-helix interactome: similar geometry via different interactions. *Structure*, **23**, 527–541.
45. Zhang,Y., Kulp,D.W., Lear,J.D. *et al.* (2009) Experimental and computational evaluation of forces directing the association of transmembrane helices. *J. Am. Chem. Soc.*, **131**, 11341–11343.
46. Li,B., Mendenhall,J. and Meiler,J. (2019) Interfaces between alpha-helical integral membrane proteins: characterization, prediction, and docking. *Comput. Struct. Biotechnol. J.*, **17**, 699–711.
47. Trueblood,K.N., Bürgi,H.B., Burzlaff,H. *et al.* (1996) Atomic dispacement parameter nomenclature. Report of a subcommittee on atomic displacement parameter nomenclature. *Acta. Crystallogr. A Found Crystallogr.*, **52**, 770–781.
48. Chung,J.-L., Wang,W. and Bourne,P.E. (2005) Exploiting sequence and structure homologs to identify protein-protein binding sites. *Proteins*, **62**, 630–640.
49. Liu,R., Jiang,W. and Zhou,Y. (2010) Identifying protein–protein interaction sites in transient complexes with temperature factor, sequence profile and accessible surface area. *Amino Acids*, **38**, 263–270.
50. Chakravarty,D., Janin,J., Robert,C.H. *et al.* (2015) Changes in protein structure at the interface accompanying complex formation. *IUCrJ*, **2**, 643–652.
51. Jones,S. and Thornton,J.M. (1995) Protein-protein interactions: a review of protein dimer structures. *Prog. Biophys. Mol. Biol.*, **63**, 31–65.
52. Gavenonis,J., Sheneman,B.A., Siegert,T.R. *et al.* (2014) Comprehensive analysis of loops at protein-protein interfaces for macrocycle design. *Nat. Chem. Biol.*, **10**, 716–722.
53. Lomize,A.L., Pogozheva,I.D., Lomize,M.A. *et al.* (2007) The role of hydrophobic interactions in positioning of peripheral proteins in membranes. *BMC Struct. Biol.*, **7**, 1–30.
54. Zeng,B., Hönigschmid,P. and Frishman,D. (2019) Residue co-evolution helps predict interaction sites in α-helical membrane proteins. *J. Struct. Biol.*, **206**, 156–169.
55. Chen,Y., Clarke,O.B., Kim,J. *et al.* (2016) Structure of the STRA6 receptor for retinol uptake. *Science*, **353**, aad8266.
56. Hall,T.M.T., Porter,J.A., Young,K.E. *et al.* (1997) Crystal structure of a hedgehog autoprocessing domain: homology between hedgehog and self-splicing proteins. *Cell*, **91**, 85–97.
57. Forst,D., Welte,W., Wacker,T. *et al.* (1998) Structure of the sucrose-specific porin ScrY from *Salmonella typhimurium* and its complex with sucrose. *Nat. Struct. Biol.*, **5**, 37–46.
58. Wilder,P.T., Varney,K.M., Weiss,M.B. *et al.* (2005) Solution structure of zinc- and calcium-bound rat S100B as determined by nuclear magnetic resonance spectroscopy. *Biochemistry*, **44**, 5690–5702.
59. Sciara,G., Clarke,O.B., Tomasek,D. *et al.* (2014) Structural basis for catalysis in a CDP-alcohol phosphotransferase. *Nat. Commun.*, **5**, 4068.
60. Jormakka,M., Törnroth,S., Byrne,B. *et al.* (2002) Molecular basis of proton motive force generation: structure of formate dehydrogenase-N. *Science*, **295**, 1863–1868.
61. Li,F., Liu,J., Zheng,Y. *et al.* (2015) Crystal structures of translocator protein (TSPO) and mutant mimic of a human polymorphism. *Science*, **347**, 555–558.

3.2. Small molecule explainable representation as key to drug understanding

3.2.1. Drugs in Artificial Intelligence-driven research

Targeting GPCRs Via Multi-Platforms Arrays and AI

AJ Preto[a, b, e], **C Marques-Pereira**[a, b], **Salete J Baptista**[a, b, c], **B Bueschbell**[a, b, e], **Carlos AV Barreto**[a, b, e], **AT Gaspar**[a, b], **I Pinheiro**[a, b], **N Pereira**[a, b], **M Pires**[a, b], **D Ramalhão**[a, b], **D Silvério**[a, b], **N Rosário-Ferreira**[a, b, d], **R Melo**[a, c], **J Mourão**[a, b, e], **and IS Moreira**[b, f], [a] Center for Neuroscience and Cell Biology (CNC), University of Coimbra, Coimbra, Portugal; [b] Center for Innovative Biomedicine and Biotechnology (CIBB), University of Coimbra, Coimbra, Portugal; [c] Centro de Ciências e Tecnologias Nucleares, Instituto Superior Técnico, Universidade de Lisboa, Bobadela, Portugal; [d] Department of Chemistry, Coimbra Chemistry Center, University of Coimbra, Coimbra, Portugal; [e] Institute for Interdisciplinary Research, University of Coimbra, Coimbra, Portugal and [f] Department of Life Sciences, Center for Neuroscience and Cell Biology, Coimbra University, Coimbra, Portugal

Email address: irina.moreira@cnc.uc.pt (I.S. Moreira)

Nomenclature

5-HT Neurotransmitter serotonin
5-HT1AR 5-HT1A receptor
5-HT2AR Serotonin 5-HT2A receptor
ADMET Absorption, distribution, metabolism, excretion and toxicity
AI Artificial intelligence
AMINO Automatic mutual information noise omission
ANN Artificial neural network
AOP Adverse outcome pathways
BIGP Bi-gram probabilities
BLAST Basic local alignment search tool
BoW Bag of words
BRDTI Bayesian ranking prediction of drug-target interactions
CADD Computer aided drug design
CMA Correlated mutation analysis
CNN Convolutional neural network
CRD: Cysteine-rich-domain
Cryo-EM cryo-Electron Microscopy
DA Drug abuse
DKPES 3,12-Diketo-4,6-petromyzonene-24-sulfate
DL Deep learning
DNN Deep neural network
DT Decision tree
DTI Drug-target interaction
ECD Extracellular domain
ECL1-3 Extracellular loops 1-3
ERT Extreme randomized trees
FDA Food and drug administration
FTT Failure to thrive
FZD Frizzled
GAFS Genetic algorithm-based feature selection
GAIN G-protein-coupled-receptor Autoproteolysis-INducing
GNN Graph neural network
GPCRdb G protein-coupled receptors database
GPCRs G protein-coupled receptors
HMM Hidden Markov model
HTS High-throughput screening
ICL(1–3) Intracellular loops 1–3
IFP Interaction fingerprints
J48 Decision tree
kNN k-nearest neighbor
LBDD Ligand-based drug design
LD50 Lethal dose 50%
MD Molecular dynamics
MDS Multidimensional scaling
MIE Molecular initiating events
ML Machine learning
MLP Multilayer perceptron
MP Membrane protein
MSA Multiple sequence alignment
NB Naïve bayes
nDCG Normalized discounted cumulative gain
NLP Natural language processing
NMR Nuclear magnetic resonance
NPSA Non-polar surface area
PCA Principal component analysis
PDB Protein data bank
PPI Protein-protein interactions
PSA Polar surface area
PsePSSM Pseudo-position PSSM
PSSM Position-specific scoring matrix
PT Perturbation theory
PTM Post-translational modifications
QSAR Quantitative structure-activity relationship
RAVE Reweighted autoencoded variational bayes for enhanced sampling
RC Reaction coordinate
RF Random forest
RL Reinforcement learning
RVM Relevance vector machine
SBDD Structure-based drug design
SMO Smoothened
SVM Support vector machine
TF-IDF Term frequency-inverse document frequency
TM Text mining
TM1–7 Transmembrane segments 1–7
TMH Transmembrane helix domain
VFT Venus flytrap
VS Virtual screening

1 Introduction

1.1 GPCRs structural characterization

G protein-coupled receptors (GPCRs) represent the largest family of Membrane Proteins (MPs) (**Rosenbaum et al., 2009**) and play an intracellular coupling to G proteins and Arrestins, induces the activation of various downstream signaling pathways, making them attractable drug targets (**Sangmin et al., 2018**).

GPCRs are composed by seven TransMembrane α-helices (TM1–7), separated by three loops on each side of the cellular membrane, three ExtraCellular Loops (ECL1–3) and three Intracellular Loops (ICL1–3) (**Lu and Wu, 2016**). Additionally, GPCRs possess both an extracellular N-terminus and a C-terminus located in the intracellular space (**Lindner et al., 2009**) as well as two cysteine residues (one on ECL1 and one on ECL2) that form a disulfide link responsible for conformational stabilization (**Bockaert and Pin, 1999**). According to the Glutamate, Rhodopsin, Adhesion, Frizzled/Taste, Secretion system (**Schiöth and Fredriksson, 2005**), this protein superfamily can be subdivided into five subfamilies: class A (rhodopsin-like receptors), class B1 (secretin receptors), class B2 (adhesion receptors), class C (glutamate receptors) and class F (Frizzled (FZD) receptors) (**Lu and Wu, 2016**). Despite their common architecture, major structural differences exist between the subfamilies and therefore they do not share any overall sequence homology (**Lindner et al., 2009**; **Rosenbaum et al., 2009**; **Katritch et al., 2013**). For example, the N-terminus is variable in terms of length, sequence and shape (**Lindner et al., 2009**; **Unal and Karnik, 2012**; **Venkatakrishnan et al., 2013**). For class A GPCRs it is usually shorter (e.g., only seven residues for adenosine A2A receptor, (**Lindner et al., 2009**)) whereas for class B receptor and class C receptors it can be very long (ranging from 100 to 4000 residues) (**Krishnan et al., 2016**). Another prominent feature of the class B receptor N-terminus is the formation of a network of disulfide bridges (**Gether, 2000**). The adhesion receptor family additionally contains a proteolytic domain (so-called G-protein-coupled-receptor Autoproteolysis-INducing (GAIN)-domain) for autocleavage (**Lagerström and Schiöth, 2008**; **Krishnan et al., 2016**), which is unique among the GPCR superfamily. Class F receptors are characterized by Cysteine-Rich-Domain (CRD) in their extracellular space (**Zhang et al., 2018**; **Wright et al., 2019**).

GPCRs react to very diverse stimuli spanning a multitude of molecules such as photons, ions, odorants, nucleotides, amino-acids, peptides and even other proteins (**Bockaert and Pin, 1999**; **Coleman et al., 2017**). As such, the classical ligand-binding domain (often also called orthosteric binding pocket) must be structurally diverse among them and requires several factors to provide recognition of specific ligands (**Katritch et al., 2012**). For class A GPCRs, ligands are recognized by a binding pocket, located in the transmembrane region near the intracellular space, while for class B, ligands are recognized by transmembrane domains and extracellular domains (ECDs) (secretin-like receptors) (**Coleman et al., 2017**). For class C receptors the ligand-binding pocket is located on the ECD that contains the Venus FlyTrap (VFT) motif (**Neumann et al., 2008**; **Basith et al., 2018a**). For class F GPCRs, both subgroups Smoothened (SMO) and Frizzled (FZD), the CRD and a linker domain on the ECD are involved in ligand-recognition (**Basith et al., 2018a**).

The majority of approved drugs target mainly the orthosteric binding site. This is due to the fact that orthosteric sites usually are wide open and easily accessible from the extracellular region, without needing to penetrate the membrane (**Chan et al., 2019**). However, recently, additional ligand-receptor interactions were revealed such as positive and negative allosterism, inverse agonism, biased signaling and multimeric receptor interactions which complicates the traditional ligand-binding principle (**Lane et al., 2017**; **Chan et al., 2019**). The traditional orthosteric site of GPCRs is located near the extracellular region between the ECL2 and a highly conserved W6.48 on TM6 (**Chan et al., 2019**). The ECL2 plays a critical role in ligand recognition, access and selectivity (**Dror et al., 2011**; **Zhang et al., 2015a**). For class A GPCRs, lipophilic ligands get in on the orthosteric site through the "lid" formed by the N-terminus and ECL2 (**Basith et al., 2018a**), while for class B receptors, which are mainly targeted by peptides, a more solvent-accessible binding pocket is required in order to provide enough space and flexibility for their modulators (**Liang et al., 2017**).

Furthermore, for class A GPCRs, several structural motifs were also identified, such as the "ionic lock," an interaction between R3.50 of the DRY motif on TM3 with D/E3.49 and D/E6.30 (**Ballesteros et al., 1998**; **Moreira, 2014**), a hydrophobic arginine cage on TM3 (positions 3.46 and 6.37) that restrains the absolute inactive conformation of R3.50 (**Prioleau et al., 2002**; **Moreira, 2014**), the NPxxYxF motif on TM7 (**Prioleau et al., 2002**; **Moreira, 2014**) and the rotamer toggle switch (**Venkatakrishnan et al., 2013**; **Moreira, 2014**). Upon ligand activation, aromatic residues on TM6, which detect the binding of a ligand, undergo spatial movements in rotamer angles triggering the cleavage of the ionic lock (**Preininger et al., 2013**; **Lu and Wu, 2016**; **Manglik and Kruse, 2017**). The cluster of aromatic residues on TM6 around W6.48 are part of the CWxP motif that undergoes a conformational rearrangement pointing from towards TM7 (inactive state) towards TM5 (active state), (**Visiers et al., 2002**; **Moreira, 2014**). For some ligands, extracellular loops are also relevant for ligand-binding, especially ECL2, which is the most structurally variable loop in the extracellular region (**Flood, 1990**; **Karnik et al., 2003**; **Katritch et al., 2012**). Residues on TM2 were also reported to interact with orthosteric ligands (**Chan et al., 2019**).

While X-ray crystallization of many class A receptors has significantly increased the understanding of the structural mechanisms of receptor activation, for class B receptors there is still a lack of solved structures due to the large N-terminal ECD (**Krumm and Roth, 2020**). Many of these structural microdomains already described for class A are absent for class B receptors. However, it is known that peptide binding to class B receptors causes rearrangement of the ECL2 (**Krumm and Roth, 2020**). As reported for class B1 GPCRs, peptide-ligands are located above a central polar network in the presence of waters and interacting with a conserved residue with TM5 and TM6 (**Krumm and Roth, 2020**; **Liang et al., 2020**; **Ma et al., 2020**). The large ECD possesses a three-layer α-β-β/α-fold, which is also involved in peptide binding (**Krumm and Roth, 2020**). It can be assumed that similarly to class A receptors, class B receptor activation leads to an outward movement of TM5 and TM6 (**Krumm and Roth, 2020**). Class C receptors possess, besides VFT on the ECD and CRD (except for the GABAB receptor), the further unique characteristic to mandatory form homo- or heterodimers with the VFT upon their activation (**Chun et al., 2012**). In the apo-state the lobes of the VFT oscillate between an open and closed conformation. As soon as a ligand (e.g., glutamate) binds to one lobe, it stabilizes the closed conformation by interactions with the second (**Chun et al., 2012**). The molecular mechanisms of activation of class F receptors was also not yet studied in detail but in a recent study a molecular switch was identified, consisting of basic amino-acids on TM6, similar to what can be observed for class A receptors (**Wright et al., 2019**). Depending on the effect provoked on the basal activity of receptors and the active conformation that underlies the functional activity of the GPCRs, a drug can be defined as an agonist (a drug capable of enhancing the activation of GPCRs), an inverse agonist (a drug described as a ligand that changes the equilibrium towards the inactive state and thereby inhibits the basal activity) or an antagonist (a drug that blocks the activation of GPCRs) (**Wootten et al., 2018**; **Diez-Alarcia et al., 2019**).

1.2 Medical and biological importance of GPCRs

Over- or malfunctioning of GPCRs leads to severe pathologies such as retinitis pigmentosa (rhodopsin receptors), cardiac diseases (beta-adrenergic receptor polymorphisms), nephrogenic diabetes insipidus (arginine vasopressin receptor 2), familial hypocalciuric hypercalcemia (calcium-sensing receptor), atopic asthma (cysteinyl leukotriene receptors), neuropsychiatric and neurodegenerative diseases such as Parkinson's disease, Huntington's disease and schizophrenia (**Jaber et al., 1996**), and many more (**Zalewska et al., 2014**). Mostly, these are caused by mutations in the coding sequence that lead to either gain-of-function or loss-of-function of the receptors. Over 700 mutations are known to change GPCR function and subsequent signaling (**Zalewska et al., 2014**).

GPCRs are the largest druggable class of biomolecules, with over 35% of the United States Food and Drug Administration (FDA)-approved drugs (**Jabeen and Ranganathan, 2019**). In 2016, 460 GPCR-targeting drugs of a total of 1286 were approved by the FDA (**Chan et al., 2019**). Of these drugs 460 drugs, approximately 94% target class A GPCRs, 4% class B, 2% class C and F from a total of 108 targeted-GPCRs (**Chan et al., 2019**). The most frequent categories of GPCR-drugs are the ones used as analgesic, schizophrenia, antiallergic and antihypertensive (**Chan et al., 2019**). As such, it is also not surprising that GPCRs possess a broad and diverse function in cancer and its metastatic progress (e.g., protease-activated receptors) (**Arakaki et al., 2018**). Nevertheless, despite the success of GPCRs as therapeutic targets, there are still currently only nine antibodies/anticancer drugs on the market that specifically target GPCRs (**Usman et al., 2020**). Moreover, the discovery of GPCRs as a highly complex network of interactions (either as dimeric or oligomeric structures), has led to a paradigm shift in the last few years, since it unfolds the already large array of targets into a much broader landscape. By including their potential combinations in multimeric forms, GPCRs can behave differently towards ligands, in comparison to the individual counterparts (Barreto et al., 2020a,b). From the over 1000 estimated genes coding for unique GPCRs in the human genome (**Nemoto et al., 2016**), 108 receptors are FDA approved drug targets while for around 100 orphan receptors neither their physiological function nor endogenous ligand is yet known (**Gloriam et al., 2007**; **Chung et al., 2008**; **Fang et al., 2015**; **Hauser et al., 2017**; **Jabeen and Ranganathan, 2019**). The main tasks of drug development today are associated with the deorphanization of GPCRs and GPCR-multimeric structures, understanding the pathophysiology of known GPCR-related diseases and the application of drugs not only to monomeric but also to dimeric or oligomeric structures.

1.3 GPCRs structures available

The number of experimentally solved three-dimensional (3D) GPCR structures has increased in the last few years due to technological improvements both in the crystallography and Cryo-Electron Microscopy (Cryo-EM) fields. Since a large number of in silico methods require prior 3D knowledge of protein and/or drug, this increment boosted not only the number of structural and dynamical studies but also broader approaches that use Artificial Intelligence (AI) algorithms to extract and process all the information available in these structures.

Known 3D structures are generally deposited in Protein Data Bank (PDB), two other GPCR-specific databases are also available: the G Protein-Coupled Receptor database (GPCRdb) (**Pándy-Szekeres et al., 2018**) and the GPCR-EXP (**Chan et al., 2018**), providing a more organized and detailed information about the current available structures. Both GPCRdb and GPCR-EXP provide predicted structures from homology model protocols for receptors, or different activation states of receptors, that are still not available (**Chan et al., 2018**; **Pándy-Szekeres et al., 2018**). In September of 2020, about 500 GPCR structures were available in PDB, compared to the 154 structures available in 2015. Considering all five subfamilies: there are ~ 80 unique receptor structures, ~ 300 unique receptor-ligand complexes, 35 unique receptor-G-Protein complexes and 4 unique receptor-Arrestin complexes. Most of these structures are from Class A receptors, where structures from Secretin, Glutamate and FZD subfamilies start to appear. Experimentally determined structures are not yet available for classes B2, D1 and Taste.

1.4 Why is AI appropriate to tackle GPCRs under the scope of finding new therapies?

AI is not the first candidate when considering computational approaches to tackle GPCRs, since Molecular Dynamic (MD) simulations and/or docking protocols are widely explored in the literature (**Velgy et al., 2018**; **Ribeiro and Filizola, 2019**; **Zou et al., 2019**) (**Fig. 1**). Although effective and highly detail-oriented, these approaches may suffer from problems similar to those of non-in silico methods as these techniques generally focus on understanding a single GPCR or GPCR-ligand interaction. Regarding these approaches, it is often difficult to generalize a protocol that can, although unchanged, be quickly used to test other GPCRs and ligands of interest. In this respect, one of the strongest traits of AI is its ability to handle a massive amount of data.

AI is a field that started with the goal of programming a machine to mimic human intelligence. To accomplish this goal, AI uses algorithms that allow the machine to perform human-like tasks like learning, knowledge representation and even abstract thinking (**Minsky et al., 1980**). As such, the capacity to deal with large amounts of data, as well as the increase in computational power, boosted the usage of AI to tackle biological issues (**Han and Liu, 2019**). Although GPCRs are still far from being fully documented, the existence of a large amount of GPCR related information is fueling the construction of AI models applied to these clinically relevant proteins (**Jabeen and Ranganathan, 2019**). This technology was used to help widen the research on several subtopics: GPCR-ligand interaction (i); feature engineering of both GPCRs and ligands (ii); molecule representation of both GPCRs and ligands (iii); GPCR identification (iv); ligand de novo modeling (v); drug repurposing (vi); drug characterization—Absorption, Distribution, Metabolism, Excretion and Toxicity (ADMET) (vii); protein structural stabilization (viii); signaling pathways construction and analysis (ix) and other subfields in which the use of AI is emerging more recently. This article reviews the main applications of Machine Learning (ML) methods, a subfield of AI, to GPCRs data to help uncover new insights about their structure and function.

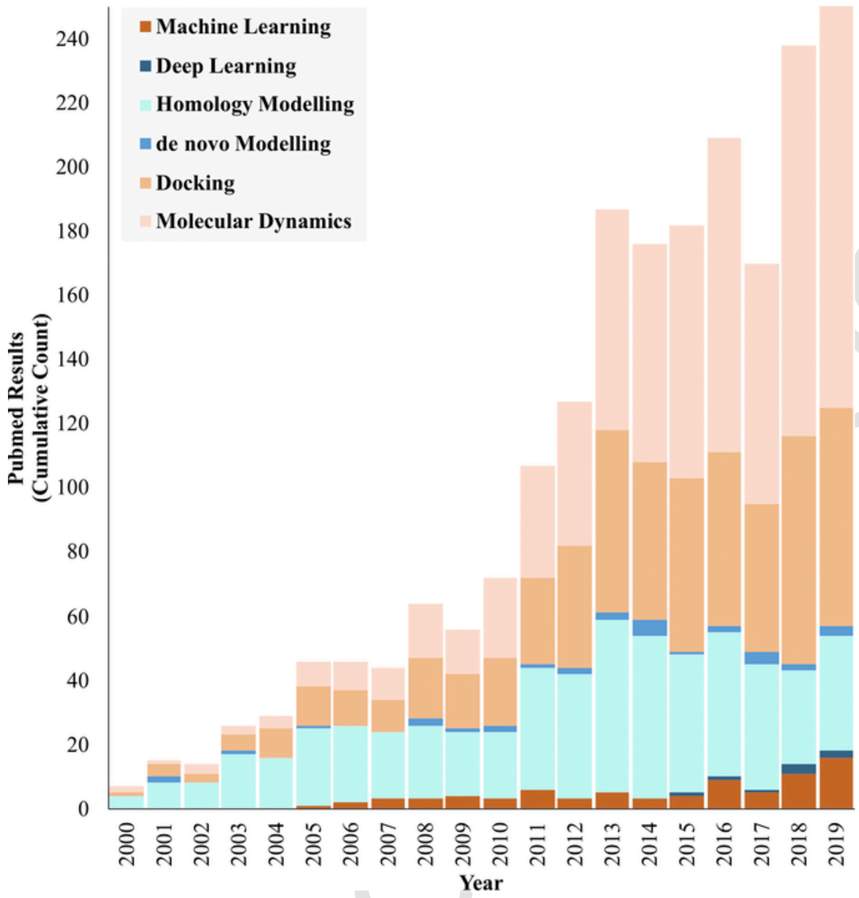


Fig. 1 The cumulative count of PubMed results by year using the following keywords: Machine Learning, Deep Learning, Homology Modeling, de novo Modeling, Docking and Molecular Dynamics, regarding GPCRs.

2 Machine learning

ML is a sub-area of AI that focuses on algorithms that can identify patterns in the input data and improve a given task without being explicitly programmed to do so (**Kubat, 2017**). To develop a more accurate ML model, the data must be expressed in the form of meaningful features that should represent variable samples according to the main objective. ML is a valuable tool for many computational biology fields since it potentiates data-analysis, Text Mining (TM), drug design and many more. The development of robust ML models requires that some general conditions are fulfilled to assure the quality of the output model. First, it is important to have a comprehensive and large dataset to create the circumstances for the algorithm to find patterns and successively build models that can recognize the relationship between the data and perform a specific task (**Baldi, 2012**). The application of ML methods to this complex data also demands a high computational power, even more, if the process must be time effective. ML methods can be split into supervised, unsupervised or Reinforcement Learning (RL). Furthermore, ML methods can be combined through ensembles, in which two or more of the best ML algorithms can be merged through a system of voting to form a unique predictor that can, ideally, outperform the individual predictors (**Zhang and Ma, 2012**).

2.1 Supervised learning

Supervised learning consists of training a machine with labeled data, i.e., the label is the target characteristic of interested to be predicted. Using this type of data, researchers guide the final model based on a pre-determined correlation between data and labels and so, the past experiences will be the stepstone to create a generalizable model. After this process, the user provides the model with unseen data, which will be label autonomously (**Cunningham et al., 2008**). If the model is well-suited to the problem, and the dataset is well constructed with meaningful and representative data, the labeling process shows high performance. These methods can be applied to data with discrete labels (classification) or continuous labels (regression) (**Rajoub, 2020**). Two prevalent issues that affect supervised learning algorithms are overfitting and underfitting. In overfitting, the prediction model excessively fits data, meaning it has difficulties in classifying new unseen samples, although it fits most of the training samples correctly. When underfitting is present, the model is incapable of correctly classifying the training data, thus suggesting that the features are inappropriate to fit the model and predict a class (**Badillo et al., 2020**). Some of the simplest supervised learning approaches include k-Nearest Neighbors (kNN), Linear Regression, Naïve Bayes (NB), or Decision Trees (DT). In contrast, more complex approaches include Random Forests (RF), Support Vector Machines (SVM), Extreme Randomized Trees (ERT), and Artificial Neural Networks (ANN) (**Uddin et al., 2019**).

2.2 Unsupervised learning

Contrarily to supervised learning, unsupervised learning is used to study datasets without labels. Users do not explicitly specify to the algorithm the data label, letting the algorithm build a relationship between features on the dataset (**Francis, 2014**). These algorithms are relevant if the user does not have full access to the labels, or if the user does not completely understand the nature of the data. For instance, datasets of huge size require dimensionality reduction. Features with low importance can be removed based on simple statistical concepts like variance; however, this can be too simplistic when the data is not fully understood. Techniques such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) are slightly more complex techniques for dimensionality reduction that can be used; however, they can also be too reductive depending on the problem (**Tenenbaum et al., 2000**). Autoencoders are one of the unsupervised algorithms that can be helpful in this situation. They are neural networks that encode all the features from a dataset in a chosen smaller dimension, and they do it autonomously and automatically. Another well-known application of unsupervised learning are clustering algorithms that compute the similarity between pairs of examples. As such, these algorithms are able to weigh the importance of each feature and reorganize the dataset in clusters of data (**Kulis et al., 2009**).

2.3 Reinforcement learning

Similarly, to unsupervised learning, RL does not require labeled data. However, in addition to the data, researchers should also provide a signal to guide the algorithm to an output that makes sense to the problem in hand (**Szepesvári, 2010**). When deploying RL algorithms there is an environment upon which an agent (the algorithm) should act. After this action is finished, the agent is signaled with a "reward", e.g., win or lose. Depending on this reward the algorithm will shift itself to be more likely to achieve the desired output (**Mousavi et al., 2018**).

3 Machine learning key algorithms

Throughout this section, we explore the most commonly applied AI approaches to GPCRs as well as list innovative ones not yet extensively applied in the field. A few of the algorithms comprise simple approaches; however, some of the most recent approaches deal with more complex algorithms, particularly ANNs.

3.1 Decision trees, random forests and extreme randomized trees

A DT algorithm works through nodes, branches, and leaves. When considering a particular dataset, all the features are measured against the class to determine the Gini impurity of the classification (the Gini impurity score accounts for the probabilities of each classification option given the feature under scope). The feature with the lowest Gini impurity score is defined as the root node. This node ramifies through branches to new nodes (internal nodes) on which the Gini impurity score will be reassessed, excluding the already used feature. The process is repeated until all the features are used so that their inclusion lowers the Gini impurity score. Thus, if a feature has a Gini impurity score higher than the value achieved without that feature, the DT will stop at that point in what is called a leaf. Leaves are the final stop of a DT, more objectively determined as the point that is connected to prior nodes but is not connected to subsequent nodes. The measurement of a Gini impurity score, for a new sample, at each of the leaves, will yield the final classification (**Kingsford and Salzberg, 2008**).

RF, on the other hand, encompasses multiple DTs, forming an ensemble in which the different attributes were tested in random combinations and the final decision is made taking into consideration the output of the individual DTs. ERT, a variation of RF, were effectively used in several biological problems although still not yet applied to GPCRs (**Basith et al., 2018b**; **Manavalan et al., 2019**; **Preto and Moreira, 2020**). This method has increased randomization, compared to RF, picking not only samples but also attributes at random. Furthermore, it chooses node cut-off points fully at random; this means that for continuous variables, which have to be split according to a threshold, instead of following a standard approach such as DT and RF (by calculating Gini impurity scores for the samples until the lowest Gini score is found), ERT uses random cut-off points, which can help eliminate sample dependent bias. These differences might make it suitable to handle Drug-Target Interaction (DTI) prediction, as this task exhibits many of the difficulties of other biological problems, such as structural characterization of proteins through Hot-spot identification (**Preto and Moreira, 2020**).

3.2 Hidden Markov models

When considering a Hidden Markov Model (HMM), there is a non-observable (hidden) variable for which the algorithm will try to solve based on known variables, typically chained sequentially. Thus, HMM is typically appropriate to predict the likelihood of time-dependent events. However, it can also be useful to derive inferences from non-timed yet sequentially chained data, as seen in its usage on biological sequence data, particularly the inclusion of evolutionary data from protein sequences alignments (**Bartoli et al., 2009**). Based on Bayesian probability, HMM considers, by default, both transition and emission probabilities. The transition probabilities are related to previous samples or states and influence the probability of the current sample prediction introducing a sequential bias. On the other hand, the emission probabilities associate individual events with their probability of occurrence without being influenced by previous events. The collective impact of these probabilities is calculated to estimate the event of the highest likelihood (**Jurafsky and Martin, 2009**).

3.3 Support vector machines

An SVM is an algorithm that differentiates samples according to their features. Depending on the features number (n), an n-dimensional space is generated, and samples are assessed in terms of proximity. The n-dimensional space is then split into k sections, with k being the number of classes that the algorithm is bound to determine. The k sections are split among space by n-dimensional support vectors that define the places that are going to be occupied by the samples (**Noble, 2006**). First, to define a support vector, an edge is defined as the object between two samples. Secondly, all the samples are evaluated according to this edge, with the possibility of some being misclassified. Next, soft margins are calculated as the distances from the samples used to define the edge and the edge itself. The space occupied between the soft margins (with the edge in between) is called a support vector. Samples outside the support vector should be correctly classified; however, there is a possibility of finding samples inside the soft margins. The hardest and most time and resource consuming task when training a SVM is cross-validation in order to find the best support vector so that the amount of misclassified samples is the lowest while maintaining the performance on new samples, and as such, achieving a generalizable algorithm (**Noble, 2006**).

3.4 Deep learning

Deep Learning (DL) refers to a class of algorithms that stem from (and are), ANNs which, in turn, were originally inspired by the biological structure of the human brain and how neurons communicate. ANNs can also be interpreted and referred to as Multilayer Perceptron's (MLP). MLPs, as the name indicates, stem from their counterpart, the perceptron. A perceptron is a single input variable that is related to an output variable through a function. As such, a simple linear regression can qualify as a perceptron (**Goodfellow et al., 2016**). A perceptron is usually graphically depicted as a single circle (input node or neuron) connected to another circle (output node or neuron) through a line (edge). This description is also applicable to a simple graph (not to be confused with Graph Neural Networks (GNNs), discussed subsequently) (**Alpaydin, 2014**).

Upon combining multiple perceptrons, such that a row of input nodes would be connected to the second row of nodes, we have our first MLP. If we add a third layer, we now refer to the first layer as the input layer, the second layer as a hidden layer and the third layer as the output layer. This is now an ANN, with the output layer providing a value that can, depending on the problem, be of a continuous or discrete nature. When we have more than one hidden layer, we can now refer to the network as a Deep Neural Network (DNN), although also still an ANN. The ability to arbitrarily add hidden layers can give ANNs the capacity to abstract information and achieve higher performance than other methods, particularly when using increasingly larger amounts of data. Furthermore, this also opens the gates to tackle problems of supervised, unsupervised and RL, among others (**Goodfellow et al., 2016**). A DL based model takes the input features, which should have the same size as the input layer and passes them along with the hidden layers by activating nodes (neurons), until it reaches a final vector of values, in the output layer. The activation process depends on the activation function (initially typically sigmoid, currently, more often, ReLU), weights and biases (usually randomly initialized), and the network's architecture. The final output values can then be assessed according to a cost function, against actual values. The algorithm can then be backpropagated to fine-tune the parameters (weights and biases are improved according to a learning rate) until the model has converged and the loss is no longer significantly decreasing (**Goodfellow et al., 2016**). One of the most significant disadvantages of DL is that it is quite demanding in terms of computational resources, especially when dealing with massive datasets. In the last years, DL has proven to effectively perform different computational tasks, mainly of categorical and regression nature (**Tavanaei et al., 2019**) (**Schmidhuber, 2015**).

The set of steps described opens a gateway to a whole new family of ML algorithms, nowadays referred to as DL, in which the parameters can be tuned, and the architecture can be twitched in order to have the best performing algorithm for each task (**Koutsoukas et al., 2017**). We elaborate some of these algorithms on the subsections below and provide the most significant examples applied to GPCRs.

3.4.1 Convolutional neural networks

Convolutional Neural Networks (CNN) are another subtype of ANNs that use a DL architecture. CNNs, unlike other ANNs, do not connect all the neurons in each layer. Instead, they activate subsets of neurons depending on a specific batch of input features and then overlap part of the information not to miss any data. When considering image processing, this would be the equivalent of analyzing subsets of subjacent pixels and finding patterns among and in them. Furthermore, CNN's usually have pool layers that reduce the noise and standardize the information (**Li et al., 2017**).

3.4.2 Autoencoders

Autoencoders are unsupervised DL algorithms inside the ANNs; however, they have some peculiar characteristics. There are three specialized groups of layers on the hidden layers: the encoder, the bottleneck, and the decoder (**Liou et al., 2014**). The encoder represents the hidden layers situated immediately downstream from the input. The number of layers in this structure varies; however, the number of neurons in each layer will gradually decrease in each layer until we reach the bottleneck. The bottleneck is situated downstream from the encoder and upstream from the decoder. It is usually the middle layer of the network and the layer with fewer neurons on the autoencoder. The decoder is the layer downstream from the bottleneck and upstream of the output layer. This structure is almost always the encoder's mirror, with fewer neurons close to the bottleneck and more neurons on layers close to the output layer (Fig. 2) (Hinton and Salakhutdinov, 2006). Data is fed to the input layer, and it flows through the encoder decreasing in dimension. When the flow gets to the bottleneck, it reaches the most compressed state. Typically, the bottleneck information represents the input data as it is that data encoded in a smaller state. The decoder segment transforms the bottleneck information on the original data, and so the autoencoder output is equal to the input data. This algorithm can also be applied to dimensionality reduction, denoising and inpainting images, among others (Boehmke and Brandon, 2019).

3.4.3 Graph neural networks

Before explaining what the GNN algorithm is and how it works, it is necessary to introduce the concept of a graph. A graph is a data structure composed of nodes and edges, being the last one responsible for establishing the relationship between nodes. This type of data structure presents a significant difference compared to the other ones, as it does not necessarily consider spatial features and enables the inclusion of data of different sources in a graph representation (the representation of molecules as graphs is one example of GNN usage in biological problems (Shui and Karypis, 2020)). GNN is a DL tool that allows processing, representing, and collecting information from graphs, an example of which are graph embedding techniques (Zhou et al., 2019). GNNs aims to discover the weight vector called embedding state (Scarselli et al., 2009; Zhou et al., 2019). Each node has a state in the graph, and both the node and the edges have features. The embedding state is then calculated in an iterative process through a local transition function dependent on the node, edges, state, and neighborhood features. After setting the embedding state, it is possible to determine the output using a local output function that considers this state and the nodes' features. Finally, when obtaining all the nodes' functions, these are stacked, generating a single global function, either for the embedding state or the output (Scarselli et al., 2009; Zhou et al., 2019).

3.5 Text-mining

TM involves a plethora of methods that take advantage of standard and existing ML techniques as well as other ones specifically developed to access, retrieve and process text information (Feldman and Sanger, 2007). The resulting text data can be used by other data mining and ML techniques. TM first employs data mining techniques to retrieve unstructured text information. The obtained unstructured data, the *corpus*, is then pre-processed using, tailored to the context, Natural Language Processing (NLP) methods. Despite being able to comprise a myriad of steps that increase the complexity of the text pre-processing, steps of data cleaning and normalization are usually included. Data cleaning aims to flatten the text formatting by making it even despite its origin. Text normalization can be achieved through the employment of, for example, tokenization, lemmatization or stemming of the words. Tokenization includes the segmentation of the text into smaller units as words or terms named tokens. While stemming is a simpler form of text normalization where, following a rule-based algorithm, a word is cut from its suffixes to its stem word; lemmatization is a more mature method where the word is converted to its lemma, the base form of the word taking into consideration lexical analysis and proximal dictionary-term. The stop word removal stage is usually used to discard common words to streamline further analysis (Wachsmuth, 2015) (Fig. 3).

The processed text is then used to extract meaningful features. Some of the most common approaches to do this are the n-grams, Bag of Words (BoW), and Term Frequency-Inverse Document Frequency (TF-IDF). N-grams is a method that analyses the text by splitting it into groups of n words. This model can then predict the next word's probability, given the n − 1 previous words (Niesler and Woodland, 1999). BoW essentially counts the number of occurrences of every word for a given document, generating a numeric vector that describes it (Zhao and Mao, 2018). Finally, the representation of words can be done employing word embeddings. Given a dictionary of words,

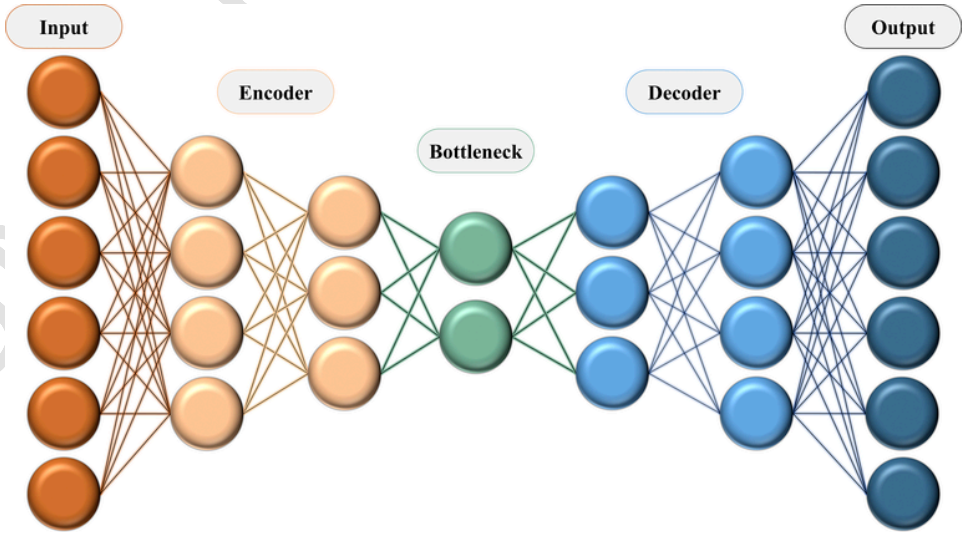


Fig. 2 Autoencoder graphical depiction detailing the three autoencoder-specific sets of layers (encoder, bottleneck and decoder).

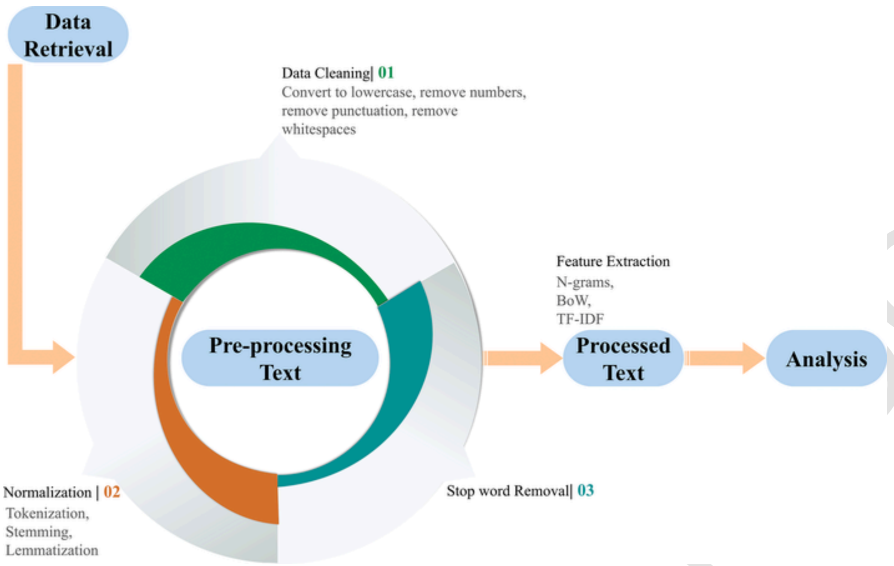


Fig. 3 Example of TM processing pipeline (*BoW*, Bag of words; *TF-IDF*, Term Frequency-Inverse Document Frequency).

these models will generate vectors to represent each word concerning its meaning and surrounding context. The closer the vectors, the more similar context the words have. Full-TM pipelines use word embedding to extract features that can be employed for biologic purposes (**Asgari and Mofrad, 2015**).

4 Computer aided drug design: Ligand design and discovery

In the next sections AI implementation to different sub-fields of GPCR world were listed in greater detail. For an easier analysis by the reader, **Table 1** summarizes their key data.

4.1 Chemical coverage

Drug discovery is a challenging task mainly due to the expensive and time-consuming Research & Development (R&D) pipeline. This process can follow two different directions: the development of new ones (**Talevi, 2018**) or more recently the scaffold repurposing of existing drugs (drug repositioning or drug reprofiling). The search for a compound that modulates a specific target involves screening chemical libraries to determine potential molecules to become drug candidates. This process requires the development and maintenance of large compound libraries, application or development of specific assays, and employment of High-Throughput Screening (HTS). HTS is a highly regarded technique used to identify lead compounds within a library through cell-based and biochemical assays (**Yadav and Tripathi, 2018**).

Computer-Aided Drug Design (CADD) uses accurate computational methods that are less expensive and time-consuming than experimental/bench approaches. Furthermore, CADD is particularly useful in identifying lead compounds among extensive chemical libraries (**Pinzi and Rastelli, 2019**). Sliwoski et al. defines three major CADD applications: (i) selection of sets of active compounds from chemical libraries, consequently submitted to experimental tests; (ii) optimization of lead compounds (affinity, metabolism, or pharmacokinetic properties) (**Sliwoski et al., 2014**); and (iii) design of novel compounds (**Sliwoski et al., 2014**). We consider vital to add to these a fourth application: (iv) understanding of DTIs in order to detect new targets for existing drugs. Conventionally, CADD approaches are broadly classified into Ligand-Based Drug Design (LBDD) and Structure-Based Drug Design (SBDD) (**Sliwoski et al., 2014**). LBDD uses known ligands and predicts the on and off-target interactions. It follows the premise that similar ligands display similar properties, thereby binding to similar proteins. So, through the chemical structure, the aim is to understand which functional groups (and why) are responsible for binding to the protein-target towards the development of novel analogs (**Sliwoski et al., 2014**; **Ezzat et al., 2019**). LBDD is extremely useful when the target's 3D structure is not available (**Sliwoski et al., 2014**; **Baig et al., 2016**). Quantitative Structure-Activity Relationship (QSAR) and ligand-based pharmacophore modeling are the most used LBDD methods. QSAR evaluates the relationship between a structure and its biological activity to predict the activity of analogues. Pharmacophore modeling depends on common properties between ligands with the same biological activity (**Baig et al., 2016**). LBDD models' performance is limited by the number of ligands used in the process, meaning that the lower the number of ligands used to build the model, the lower its performance (**Sliwoski et al., 2014**). Contrarily to LBDD, in SBDD, the targets' 3D structure is required, which is usually obtained through experimental approaches such as X-Ray Crystallography and Nuclear Magnetic Resonance (NMR) or, more recently, via Cryo-EM (**Baig et al., 2016**). When the targets' 3D structure is not available, it can be predicted by several in silico approaches. Homology modeling can

Table 1 Known applications of AI to GPCRs.

Task type	*Description*	*Year*	*Software name*[a]	*References*
Database	GPCR-ligand database	2008	GLIDA	Okuno (2008)
Database	GPCR ligand library (GLL) and GPCR decoy database (GDD)	2012	GLL/GDD	Gatica and Cavasotto (2012)
Database	Database generated from RF predictions of receptor-ligand pairings in conjunction with TM	2012	ReLiance	Iacucci et al. (2012)
Database	GPCR-ligand experimentally validated database	2015	GLASS	Chan et al. (2015)
Database	GPCR-related information database	2016	GPCRdb	Munk et al. (2016)
Database	Drug-abuse related GPCRs information repository	2019	DAKB-GPCRs	Chen et al. (2019)
Database	Data-independent acquisition mass-spectrometry and DL driven database for proteomic profiling	2020		Lou et al. (2020)
DTI detection	Using ML algorithms used to aid the detection of GPCR-ligand pairs in live cell microscopy	2020		Allikalt et al. (2020)
DTI predictor	Using SVM algorithm to predict GPCR-ligand interactions without using GPCR structural information	2008		Jacob et al. (2008)
DTI predictor	Using SVM and Bayesian methods algorithms to determine predict adenosine receptors antagonists	2010		Lee et al. (2010)
DTI predictor	SVM to predict 5-HT receptor agonist or antagonist drugs	2011		Zhu et al. (2011)
DTI predictor	DTI Predictor based on GPCR pseudo amino-acid composition (PseAAC) and drug fingerprint	2013	iGPCR-drug	Xiao et al. (2013)
DTI predictor	Supervised neural networks ANN using IFPs	2016		Vass et al. (2016)
DTI predictor	Using various ML methods to predict active/non-active ligands for GPCRs as well as ligand toxicity	2016		Mansouri and Judson (2016)
DTI predictor	DTI prediction using RF and evolutionary features	2016	TargetGDrug	Hu et al. (2016b)
DTI predictor	DTI predictor with Bayesian Ranking Prediction for Drug repurposing	2017	BRDTI	Peska et al. (2017)
DTI predictor	DTI prediction with SVM based on features calculated with discrete wavelet transform	2017	DAWN	Shen et al. (2017)
DTI predictor	Sequence-based approach with RVM to predict DTI networks	2017	PDTPS	Meng et al. (2017)
DTI predictor	Testing Several ML algorithms against used to predict dataset of VS GPCR inhibitors	2018		Raschka et al. (2018)
DTI predictor	ML algorithms used to unveil new 5-HTA2 receptor agonists. Using ML	2019		Diez-Alarcia et al. (2019)
DTI predictor	DTI prediction with protein sequences and drug fingerprints	2019		Li et al. (2019b)

Table 1 (Continued)

Task type	*Description*	*Year*	*Software name*[a]	*References*
DTI predictor	Using RF to predict DTI with evolutionary information and chemical structure	2019	LRF-DTIs	Shi et al. (2019)
DTI predictor	DTI prediction with wrapper feature selection and class balancing	2020		Redkar et al. (2020)
DTI predictor	GPCR-ligand prediction with hub and cycle feature	2020		Breer et al. (1985)
Feature representation	Protein-ligand fingerprint to mine chemogenomic space	2009		Weill and Rognan (2009)
Feature representation	Using n-grams as protein subspaces and SVM algorithms to identify class C GPCR motifs	2014		König et al. (2014)
Feature representation	Using SVM algorithm used to represent GPCR information and a RF approach to classify GPCRs and non-GPCRs	2016		Cai et al. (2003) and Liao et al. (2016)
Feature representation	Shape similarity profile between ligands and structural samples of GPCR-binding molecules from PDB	2016		Hu et al. (2016a)
Feature representation	DL and physicochemical properties to characterize class C GPCRs	2018		Cruz-Barbosa et al. (2018)
Feature representation	Text mining-based techniques to generate features and classify GPCR-ligand interactions	2020		Wang et al. (2020)
Functional selectivity	Using ML algorithms used to predict interactions between GPCR and PDZ domain proteins interactions	2009		Eo et al. (2009)
Functional selectivity	GPCR classification up to three levels (class A) making use of several ML algorithms	2009		Kumari et al. (2009)
Functional selectivity	GPCR classification with grey incidence degree	2011		Zia-Ur-Rehman and Khan (2011)
Functional selectivity	Classification of GPCRs using family specific motifs selected by the distinguishing power evaluation technique	2011	GPCRBind	Cobanoglu et al. (2011)
Functional selectivity	Ligand classification algorithm with adaptively boosting ensemble stumps	2011	LiCABEDS	Ma et al. (2011)
Functional selectivity	GPCR classification with an ensemble of nearest neighbor, SVM, grey incidence degree and probabilistic neural network	2012		Zia-Ur-Rehman and Khan (2012)
Functional selectivity	Automatic classification of GPCRs using a Genetic Ensemble	2012	GPCR-MPredictor	Naveed and Khan (2012)
Functional selectivity	ML algorithms used to identify chemical substructures and amino-acid properties associated with ligand binding	2013		Shiraishi et al. (2013)

Table 1 (Continued)

Task type	*Description*	*Year*	*Software name*[a]	*References*
Functional selectivity	Pseudo amino-acid composition and physicochemical properties to classify GPCRs	2013		Rehman et al. (2013)
Functional selectivity	Using an SVM ensemble to predict GPCR glycosylation sites	2013		Xie et al. (2013)
Functional selectivity	Hierarchical classification method based upon an SVM that is able to identify GPCR subtype levels	2013		Gao et al. (2013)
Functional selectivity	Using SVM algorithms used to classify GPCRs/non-GPCRs and GPCRs subfamilies	2015		Nie et al. (2015)
Functional selectivity	Using SVM algorithm to identify GPCR misclassifications	2015		König et al. (2015)
Functional selectivity	Gaussian Process Models for VS	2016		Bieler et al. (2016)
Functional selectivity	Using RF ensembles to identify misclassified GPCRs	2017		Shkurin and Vellido (2017)
Functional selectivity	GPCR classification using CNN and TM based techniques	2017		Li et al. (2017)
Functional selectivity	Using an SVM algorithm used to discriminate and characterize class C GPCRs	2018		König et al. (2018)
Functional selectivity	DL and RF to predict ligand bioactivity prediction using DL and RF	2018	WDL-RF	Wu et al. (2018)
Functional selectivity	MD data analysis with DNN	2019		Plante et al. (2019)
Functional selectivity	SVM to predict SVM-neuropeptide pair	2019		Shiraishi et al. (2019)
Functional selectivity	Prediction of allosteric modulators for metabotropic glutamate receptors with QSAR	2019		Butkiewicz et al. (2019)
Functional selectivity	Function prediction for GPCRs through TM and induction matrix	2019	TM-IMC	Wu et al. (2019)
Functional selectivity	Ligand discovery with using DL and RF	2020		Tsou et al. (2020)
Functional selectivity	Combination of ML with metadynamics to study GPCR-ligand kinetics	2020		Lamim Ribeiro et al. (2020)
Functional selectivity	A learning to rank algorithm to explore target-drug correlations	2020		Ru et al. (2020)
Model quality assessment	SVM to predict MP models' quality	2010	ProQM	Ray et al. (2010)
Network visualization	Application of an unsupervised clustering algorithm for detection of GPCR sequences	2017	MSC	Hu et al. (2017)
PPI predictor	GPCR-GPCR interaction predictor with SVM	2016	GGIP	Nemoto et al. (2016)

Table 1 (Continued)

Task type	*Description*	*Year*	*Software name*[a]	*References*
Structural modeling and enhancement	Using HMM algorithms to predict coiled-coil regions from MSA information	2009	CCHMM_PROF	**Bartoli et al. (2009)**
Structural modeling and enhancement	Predict Multiple ML classifiers to predict transmembrane inter-helix contacts. With multiple ML classifiers	2013	MemBrain	**Yang et al. (2013)**
Structural modeling and enhancement	Ligand de novo modeling with the assistance of ML methods	2014		**Reutlinger et al. (2014)**
Structural modeling and enhancement	Conditional random fields for transmembrane topology prediction	2016	dCRF-TM	**Wu et al. (2017)**
Structural modeling and enhancement	GPCR thermostabilizing point mutations with MD and ML using 5-HT2C receptor as case study	2018	CompoMug	**Popov et al. (2018)**
Structural modeling and enhancement	Protein modeling from sequences with using DL algorithms	2018	DeepFam	**Seokjun et al. (2018)**
Structural modeling and enhancement	GPCR mutants thermostability enhancement with four ML methods	2019		**Muk et al. (2019)**
Structural modeling and enhancement	Using ML algorithms used to aggregate MD trajectories information	2019		**Ferraro et al. (2020)**
Structural modeling and enhancement	Using ML algorithms used to distinguish active from inactive GPCRs	2020		**Bemister-Buffington et al. (2020)**

[a]Software name is provided when available in the original research article.

be used to develop a virtual model of the target with an unknown structure, using a known target structure (template) and considering the sequence similarities between them, which should be as high as possible to minimize the model's error. This approach assumes that similar structures hold similar functions and binding site conservation (**Baig et al., 2016**; **Muhammed and Aki-Yalcin, 2019**). The main methods used for SBDD are docking and Virtual Screening (VS). A docking simulation allows identifying promising drug candidates for a given target and studying structure-activity relationships (**Li et al., 2019a**; **Pinzi and Rastelli, 2019**). Docking simulations also allow the ranking of drug-target candidates based on rough estimates of binding affinity, where 3D predictions are made by using both targets' and ligands' 3D structure (**Li et al., 2019a**; **Sachdev and Gupta, 2019**). In turn, VS, which can be both structure-based or ligand-based, is a computational analysis of an extensive chemical library to identify lead compounds capable of interacting with the target of interest (**Baig et al., 2016**). Agreeable to these definitions, docking and VS are not mutually exclusive, in fact, it is common procedure to deploy VS upon large sets of docked structural models (**Kontoyianni, 2017**). To complement these methods, Molecular Dynamics (MD) simulations are another SBDD methodology often used to give further insights into the GPCRs coupling to a variety of ligands/drugs (**Jabeen and Ranganathan, 2019**).

Through ligand-based VS regarding the analysis of GPCR ligands, ML can be applied to distinguish active from non-active compounds and to reckon the functional groups' influence in their biological activity. For example, Raschka et al. used DT and RF in 3,12-diketo-4,6-petromyzonene-24-sulfate (DKPES) analogs binding to GPCR and revealed that the presence of sulfur atoms compromises ligands' activity in 35% while sulfate ester groups induced around 20% of ligands' activity (**Raschka et al., 2018**). Thus, the knowledge of the functional groups involved in the biological activity was used to evaluate a ligand's analog and infer whether it could or could not interact with the desired target (**Raschka et al., 2018**).

4.2 Ligand representation

Ligand representation is an essential step of CADD since it determines how much the information available can be maximized for the subsequent tasks. In ML-based tasks, ligand representation features are usually used to train and develop models, such as DTI prediction. Ligands can be represented through atomic or structural data as well as molecular descriptors (**Chandrasekaran et al., 2018**; **Grisoni et al., 2018**). These molecular descriptors are mathematical representations of molecules' features (**Chandrasekaran et al., 2018**), which quantify molecules' physical and chemical properties, either ligands or proteins (**Chandrasekaran et al., 2018**). In CADD, the main features used to study biologically active compounds are: (i) atom composition, (ii) molecular weight, (iii) functional groups constituting a drug, (iv) bonds connecting different functional groups, and (v) distances between different atoms or functional groups and the Polar and Non-Polar Surface Area (PSA and NPSA, respectively) (**Chandrasekaran et al., 2018**; **Grisoni et al., 2018**). From another perspective, molecular descriptors can be categorized in 1D, 2D, and 3D (**Chandrasekaran et al., 2018**). 1D descriptors are the simplest type and can be easily calculated using the chemical formula of the ligand. These descriptors consist, for instance, on the frequency of a given atom or functional group, its type, molecular weight and sum or average of atomic properties (e.g., atomic Van der Waals volumes) (**Chandrasekaran et al., 2018**; **Grisoni et al., 2018**). Yet, 1D information is narrow and can assume the same values for different molecules, meaning that is frequently not specific enough (**Grisoni et al., 2018**). As for 2D descriptors, they are calculated using a representation of the molecule in a plain, where the atoms are laid and connected by bonds, but without the third dimensional component of space (**Grisoni et al., 2018**). In this way, 2D descriptors define atoms' connections. Also, this type of representation allows the calculation of several topological indices which represent properties, like adjacency and connectivity, depending on the size, shape, symmetry, branching and cyclicity of the molecule overcoming some of the 1D descriptors disadvantages (**Grisoni et al., 2018**; **Chandrasekaran et al., 2018**). Lastly, 3D descriptors give information on the molecule's conformation, identifying and quantifying its interaction(s) (**Grisoni et al., 2018**). In addition, PSA and NPSA surface area, intramolecular hydrogen bonding and valence electron distribution are often calculated. To calculate these descriptors, quantum mechanics can be a significant addition since the molecules under scope are often relatively small and cannot be as accurately described with the more standard Newtonian physics (**Chandrasekaran et al., 2018**).

Molecular descriptors play a fundamental role in the progression of computational biology (**Grisoni et al., 2018**). They are the foundation of fingerprints, a tool that, indeed, connects experimental evidence to in silico methods, once it relates the information beyond a molecule with experimental data through mathematical algorithms in a fast and inexpensive way (**Grisoni et al., 2018**). Like molecular descriptors, fingerprints are used to implement ML in CADD (**Kearnes et al., 2016**). A fingerprint often is a binary sequence describing the chemical composition, structural features, and physical properties of a compound (**Kearnes et al., 2016**). They allow a comparison of different ligands turning the evaluation of molecules similarity into a more straightforward task (**Cereto-Massagué et al., 2015**). Fingerprints can also store 2D information, thus called 2D fingerprints, or 2D and 3D information, in which case they are most known as pharmacophore fingerprints (**Cereto-Massagué et al., 2015**). Cereto-Massagué et al. categorized and thoroughly described fingerprints according to the type of information and how it was stored as: substructure keys-based, topological or path-based, circular pharmacophore, hybrid and other types of molecular fingerprints (**Cereto-Massagué et al., 2015**).

A more recent approach uses graphs to exploit a new form of ligand representation, which can be very useful when considering larger ligands, as for example small peptides. Bandholtz et al. used a GNN as a genetic algorithms' fitness function to optimize a GPCR ligands' metabolic activity, preventing bioactivity loss. As input, the GNN receives the peptide ligand's linear structure and two biochemical properties, agonistic activity and metabolic stability. It then translates the peptides' structure into a graph where each node corresponds to an amino-acid and each edge to the connections between them. Furthermore, this network output allows for exploring possible virtual ligands without relying on information from a three-dimensional structure (**Bandholtz et al., 2012**).

4.3 Drug-target interactions

Countless ligands can interact with GPCRs, making it imperative to improve our understanding and characterization of GPCR-ligand structures (**Shiraishi et al., 2013**; **Bueschbell et al., 2019**). DTIs can be studied through GPCRome-ligand information to shed light on the binding process and accelerate drug discovery. Drug-target coupling depends on several factors, such as binding energy, electrostatic energy, intermolecular energy, the interaction energy of van der Waals or intermolecular forces (**Chakraborty et al., 2017**). Most intermolecular DTIs result from van der Waals forces, weaker than hydrogen bonds or hydrophobic interactions (**Van Oss et al., 1986**; **Chakraborty et al., 2017**). Although this superfamily has a low overall reduced sequence similarity among its members, excepting a transmembrane conservative region (**Sanders et al., 2011**), some structure-based approaches were developed and can predict GPCR binding motifs (**Shiraishi et al., 2013**).

Throughout the drug discovery process, compounds are modified to improve ligand properties such as bioactivity and selectivity (**Klabunde and Hessler, 2002**). GPCR-ligand structures and binding conformational information are helpful tools to achieve compound optimization. Ligand properties can be improved using ML methods to identify chemical and residue properties associated with protein-ligand coupling (**Shiraishi et al., 2013**). GPCR-specific descriptors and statistical scores were established to predict residues and chemical structures in GPCR-ligand interactions using different kernels comparison (**Shiraishi et al., 2013**). Reliable protein-ligand predictions seems to directly depend on positive and negative pairs for the predictor (**Weill and Rognan, 2009**). Noninteraction pairs are limited in public databases and pairs with unknown interaction can be randomly selected to overcome this limitation, although this is a less than desirable solution. To select more plausible noninteracting pairs, residue pairs with low co-occurrence scores can be chosen to improve model accuracy (**Shiraishi et al., 2013**). These approaches can be applied to GPCR ligand VS and active compounds modification, to predict GPCR-ligand binding.

GPCR-ligand can also be represented as Interaction FingerPrints (IFPs), one-dimension binary representations determining the occurrence of contacts between ligand and protein amino-acids (**Vass et al., 2016**). Each IFP encodes seven interaction types between pocket residues and ligand as 1 if a contact is present or 0 if it is absent (**Kooistra et al., 2016**). The seven interaction types encoded are hydrophobic contact, aromatic face-to-face, aromatic edge-to-face, H-bond donor-acceptor, H-bond acceptor-donor, ionic positive-negative, and ionic negative-positive (**Manning et al., 2002**). Protein-ligand fingerprints are a proper biochemical structure representation and suitable to work with substantial amounts of data (**Vass et al., 2016**). IFPs can ultimately be used to predict GPCR binding sites (**Deng et al., 2004**), process virtual ligand screening (**Da and Kireev, 2014**), predict GPCR structure modulation (**Kruse et al., 2013**), ligand functional activity prediction (**Cereto-Massagué et al., 2015**) and drug discovery for new targets binding sites (**Lavecchia, 2015**). By constructing an IFP dataset, it is possible to use algorithms such as ANNs to predict GPCR ligands' IFPs and accurate docking decoys

(**Chupakhin et al., 2013**; **Vass et al., 2016**). Furthermore, it is possible to unravel new target-ligand associations with algorithms such as PROFILER, designed for polypharmacology prediction, that applies a DT to choose ligand-based or structure-based approaches given the amount and the quality of available data (**Meslamani et al., 2013**).

4.4 Drug-target interaction predictors

Recently, eight sequence-based ML models were developed in order to predict DTIs, all trained with enzymes, ion channels, GPCR, and nuclear receptor datasets with an accuracy higher than 82% (**Xiao et al., 2015**; **Meng et al., 2017**; **Peska et al., 2017**; **Shen et al., 2017**; **Shi et al., 2019**; **Li et al., 2019b**; **Redkar et al., 2020**; **Zhan et al., 2020**). In this section, we explore in more detail some of these DTI prediction approaches that were applied to study GPCRs and summarize their performance (**Table 2**). Different classifiers were previously used such as Rotation Forest (**Li et al., 2019b**), Relevance Vector Machine (RVM) (**Meng et al., 2017**), SVM (**Shen et al., 2017**), and RF (**Redkar et al., 2020**) (**Shi et al., 2019**). These classifiers can make use of a large array of features for the target, Position-Specific Scoring Matrix (PSSM) (**Jones, 1999**; **Gautam et al., 2013**; **Meng et al., 2017**; **Li et al., 2019b**), pseudo-position PSSM (PsePSSM) (**Shi et al., 2019**) and BI-Gram Probabilities (BIGP) (**Sharma et al., 2013**), among others. For the drug features, substructure fingerprints (**Ojansivu and Heikkilä, 2008**; **Meng et al., 2017**; **Shen et al., 2017**; **Li et al., 2019b**) as well as other molecular fingerprinting representations (**Shi et al., 2019**), were used to attain good performing DTI predictors. In some cases, data processing tools such as Component Analysis (**Meng et al., 2017**), LASSO algorithm (**Ghosh and Chinnaiyan, 2005**; **Shi et al., 2019**), or wrapper feature selection (**Redkar et al., 2020**) shown to improve the overall model performance.

Table 2 GPCR DTI predictors performance metrics.

Name	*ACC (%)*	*SE (%)*	*SP (%)*	*AUC*	*PE (%)*	*MCC (%)*	*RC (%)*	*FM (%)*	*STR (%)*	*References*
BRDTI	N/A	N/A	N/A	0.96	N/A	N/A	N/A	N/A	N/A	**Peska et al. (2017)**
DAWN	89.0	88.8	89.1	0.95	N/A	N/A	N/A	N/A	N/A	**Shen et al. (2017)**
Diez-Alarcia et al., 2019	86.5	95.4	85.6	N/A	N/A	N/A	N/A	N/A	N/A	**Diez-Alarcia et al. (2019)**
iDrug-Target	90.3	97.6	86.7	N/A	N/A	80.7	N/A	N/A	N/A	**Xiao et al. (2015)**
iGPCR-drug	85.5	80.0	88.3	N/A	N/A	67.8	N/A	N/A	N/A	**Xiao et al. (2013)**
Jacob, L. et al., 2008	78.1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**Jacob et al. (2008)**
Lee, J.H. et al., 2010	97.9	93.3	98.8	1.00	N/A	92.0	N/A	N/A	N/A	**Lee et al. (2010)**
Li, Y. et al., 2019	82.2	81.3	N/A	0.87	82.8	70.6	N/A	N/A	N/A	**Li et al. (2019b)**
LRF-DTIs	95.7	95.3	96.1	0.99	N/A	N/A	N/A	N/A	N/A	**Shi et al. (2019)**
Mansouri, K. and Judson R.S., 2016	95.0	100.0	91.0	N/A	N/A	N/A	N/A	N/A	N/A	**Mansouri and Judson (2016)**
PDTPS	86.8	84.9	88.4	N/A	87.9	79.7	N/A	N/A	N/A	**Meng et al. (2017)**
Redkar, S. et al., 2020	90.8	N/A	92.4	N/A	90.1	81.5	88.9	89.3	N/A	**Redkar et al. (2020)**
TargetGDrug	80.8	83.1	79.6	N/A	N/A	60.0	N/A	N/A	81.3	**Hu et al. (2016b)**
Zhan X. et al., 2020	82.3	81.2	N/A	0.89	83.4	70.9	N/A	N/A	N/A	**Zhan et al. (2020)**
Zhu, X. et al., 2011	92.7	91.7	94.1	0.91	N/A	N/A	N/A	N/A	N/A	**Zhu et al. (2011)**

Abbreviations: *ACC*, accuracy; *SE*, sensitivity; *SP*, specificity; *AUC*, Area under Receiver Operating Characteristic Curve; *PE*, precision; *MCC*, Matthews's correlation coefficient; *RC*, Recall; *FM*, F-Measure; *STR*, Strength. *N/A*, Not Available.

Mainly, there are now an extensive array of features regarding simultaneously target and drugs and complementary tools that are driving GPCR-DTI prediction. The development of more standard approaches has also opened the gates to tackle the problem using more elaborate and complex ones. For instance, BRDTI, a DTI predictor using Bayesian Ranking Prediction matrix factorization with drug-centric repositioning was developed (**Peska et al., 2017**). Wang et al. described a model with stacked auto-encoders used to extract features from a GPCR dataset (**Wang et al., 2018**). Those features were then used to predict the DTIs with an RF algorithm that achieved near 87% accuracy. In another approach, target bias was included as a way to simulate possible target conformations (**Peska et al., 2017**). In this work, the GPCR dataset had a normalized Discounted Cumulative Gain (nDCG) of 92.9%, outperforming previous approaches (**Peska et al., 2017**). The authors described these performance metrics as evaluating a drug-centric approach better than the area under the ROC curve (**Table 2**). Moreover, the authors state that their approach could predict completely new DTIs (**Peska et al., 2017**).

Another issue to keep in mind regarding DTI prediction of GPCRs is the subproblem that arise when considering how different these proteins can be. Since it is challenging to identify ligands for specific GPCR subtypes, an approach using ML, docking and multiple scoring methods was proposed for this task using two Neurotransmitter Serotonin (5-HT) receptors (**Rataj et al., 2018**). In this particular case, Neighboring Substructures Fingerprints were applied to select compounds from a large database, creating a two-dimension fingerprint with substructural compound features. The compound selection was then conducted considering interactions with protein binding pockets (**Rataj et al., 2018**). The examples presented were considered mainly to display the wide variety of features, methods, classifiers and overall strategies that are on the rise in GPCR DTI prediction. To the best of our knowledge, LRF-DTIs (**Shi et al., 2019**) DTI predictor appears to have achieved the best performance (accuracy, specificity, and AUC) (**Table 2**).

4.5 Ligand effect prediction

GPCR-ligand binding site classification and dynamic molecular changes are crucial for a more precise ligand effect understanding. Plante et al., by deploying DL upon MD trajectories developed a method able to extract useful information to reveal distinct ligand characteristics and molecular factors and ultimately discriminate GPCR structure and function with high accuracy on the test set (> 98%) (**Plante et al., 2019**). On another example, a SVM prediction model was developed to estimate if a ligand is a GPCR's agonist or antagonist showing an accuracy of 86.5% (**Zhu et al., 2011**). Although the dataset was specific of 5-HT1A Receptor (5-HT1AR), this approach may be applied to other GPCR receptors. Another study compared several ML models with other ligand-based VS methods and showed that DNN and RF were able to enhance GPCR agonists prediction even with less compounds in the training set (**Tsou et al., 2020**). In an effort to generate a method able to extract more precise information than simply agonist/antagonism classification, Wu et al., firstly used weighted DL for molecular fingerprinting and then used a RF algorithm to assess ligand bioactivity (**Wu et al., 2018**). The model achieved an average root-mean square error of 1.33 and a correlation coefficient of 0.80 (**Wu et al., 2018**). The metabotropic glutamate receptor 1 is a GPCR target for neuropathic pain treatments with a druggable allosteric site that, when blocked, is unable to complete the receptor response. A VS approach to detect allosteric modulators, molecules that bind to a protein at a location different to the binding site (**May et al., 2007**), in metabotropic glutamate receptor 1 was conducted using structural models derived from homology modeling and MD. Compounds from several libraries were selected based on known positive and negative antagonists and classified with a NB model. Selected ligands were docked into the protein target binding site and binding modes were calculated and used to improve inhibitory activities (**Jang et al., 2016**).

Throughout this section we demonstrated how VS is often used in CADD approaches focusing in a single GPCR. However, only a few are based on ML methods, which opens new venues of research, such as the automatization of new target identification given a pool of drugs. This has been explored by Ru et al. that developed a model which incorporates a RF classifier to rank putative new drug-target pairs (**Ru et al., 2020**).

4.6 ADMET prediction

From the first tests to final approval, getting a single drug to the market takes a long time and involves many resources (**Sacks et al., 2018**). However, only a few drug candidates that reach clinical trials are approved for human use, representing a substantial waste of time and money (**Lysenko et al., 2018**). Most of the issues related to this enormous failure rate in drug development are associated with undesirable pharmacokinetics and toxicity. Therefore, it has been widely accepted that ADMET properties should considered in the early stages of drug discovery to increase drug development success, especially later on during the clinical phase (**Wu et al., 2020**). Moreover, post-marketing safety issues have led to several drug withdrawals and unexpected mortality and morbidity concerns boosting the need to apply ADMET prediction even after drug approval (**Basile et al., 2019**).

Computational approaches emerged as crucial tools to evaluate ADMET properties in a cheaper but still efficient way (**Basile et al., 2019**). Several ADMET-related databases, which incorporate pharmacokinetics and toxicity parameters can be used for shape and/or pharmacophore screening to obtain further information about bioactivity on similar models that match the input query compound (**Wu et al., 2020**). Furthermore, databases like ADME Database (**Shang et al., 2017**), SuperToxic (**Schmidt et al., 2008**), PKKB (**Cao et al., 2012**), and DSSTox (**Williams et al., 2017**) were reported as reliable and comprehensive sources for training and development of ML models for ADMET prediction, namely to predict drug metabolites and toxicity, which are ultimately responsible for drug efficacy and safety (**Basile et al., 2019**; **Litsa et al., 2020**).

Traditionally, several in silico ADMET approaches tend to establish a relationship between different molecular descriptors and ADME properties by applying statistical models or ML algorithms, which includes ANN, DT, kNN, and SVM (**Shen et al., 2010**). Among them, SVM is one of the most applied algorithms for building ADMET prediction models. One example of SVM algorithms application is the widely used SwissADME tool, a free web tool able to assess small molecules' pharmacokinetic profile, alongside their physicochemical properties and drug-likeness (**Daina et al., 2017**). Other methodologies were applied to study the ADMET profile of drug candidates, particularly its toxicity, by testing several drug features combined with target-based predictions and QSAR studies. QSAR models were mainly applied to assess several drug safety endpoints, such as Lethal Dose 50% (LD50), tissue-specific toxicity, and skin and eye irritation (**Patlewicz and Fitzpatrick, 2016**). PrOCTOR (**Gayvert et al., 2016**) and TargeTox (**Lysenko et al., 2018**) are two examples of freely target-based toxicity prediction tools based on QSAR models. On the one hand, PrOTOR combines the target-based features with the drugs' chemical features and drug-likeness properties to generate a RF-based classifier that differentiates FDA approved drugs from Failure To Thrive (FTT) ones, ultimately predicting compounds that are likely to fail (due to toxicity issues) during clinical trials. Besides predicting tolerable toxicity, this model gives insights into the chemical and target-based properties able to foster or prevent toxicity (**Gayvert et al., 2016**). On the other hand, TargeTox combines a network-based approach with a gradient boosting classifier to predict drug toxicity. This model considers the information about drug's on- and off-targets and off-targets, as well as functional impact and biological network data, to generate a protein networks' distance metric, since that neighboring biological molecules display similar functional roles. Therefore, authors assumed that toxicity effects can be confined to a specific network region (**Lysenko et al., 2018**). The creation of a protein network and the combination of several pharmacological and functional features make TargeTox a good ML classifier for toxicity prediction. Unlike other approaches, TargeTox can not only generate protein network data but also integrate pharmacological and functional features into a ML classifier able to predict toxicity. However, the limited information about all possible bound proteins could potentially compromise the effectiveness of TargeTox (**Lysenko et al., 2018**).

Specific ML-based models for metabolism prediction were also developed in recent years due to its recognized impact on the pharmacokinetics and pharmacodynamics of xenobiotics and their derivatives (**Litsa et al., 2020**). However, most of these models have a limited scope, coverage, and performance. In order to overcome this issue, a freely available software package, BioTransformer (**Djoumbou-Feunang et al., 2019**), which associates a ML with a knowledge-based approach, was developed allowing both metabolism prediction and compound identification. Another interesting tool, MetaTrans, consists of a learning-based technique to predict human metabolites of small molecules. A transfer learning approach was applied by first using chemical reactions' data to pre-train a transformer model, after which it was fined-tuned using a human metabolic reactions' dataset from freely available databases, which includes metabolism not only of xenobiotics but also of endogenous molecules and comprises all enzyme classes' spectrum (**Litsa et al., 2020**). An ensemble model combining the output of several fined-tuned models and considering different metabolites was then built. Authors showed that their method displayed an equivalent performance in comparison with other drugs metabolite prediction approaches, such as SyGMa (**Ridder and Wagener, 2008**), GLORYx and BioTransformer, considering the major enzyme families screened. Furthermore, it seems able to identify metabolites using fewer common enzymes (**Litsa et al., 2020**).

The ADMET prediction tools reported above can be applied in an undifferentiated way for almost all drug candidates, including GPCRs ligands. It is noteworthy to highlight that Mansouri et al. reported two QSAR model approaches involving 18 GPCR cell-free HTS assays (**Mansouri and Judson, 2016**). Several software and genetic algorithms were then employed to calculate and select the best molecular descriptors used for the development of ML models. This strategy included as a first step the development of classification models to distinguish active and inactive chemicals to ultimately rank them considering target Molecular Initiating Events (MIE) of Adverse Outcome Pathways (AOPs). Afterward, a regression model was generated to predict the potency of active chemicals was performed. In both tasks several model-fitting methods like SVM and kNN were applied (**Mansouri and Judson, 2016**).

5 GPCR characterization and selection

Although GPCR characterization and feature representation is necessary for CADD, it is also required for a number of other AI-based tasks involving GPCRs. In this section we explore some of the main features that can be used to represent target protein information.

5.1 Target features

The development of ML models requires features that can represent and accurately describe the full dataset. When considering GPCRs proteins, these features can be attained by different methods, depending on data availability. Structural data is less abundant than sequence-based data; however, their use to train ML models' can improve their performance.

Sequence-based features are extracted from the protein sequence, and comprise a wide array of information, such as amino-acid properties, whole-protein sequence features, and conservation information. When considering amino-acid properties for sequence-based feature extraction, information such as the known composition of the amino-acids (e.g., number of sulfur atoms, number of carbon atoms, presence of aromatic rings, etc.) can be used. Experimentally determined values (e.g., pKa values, secondary structure propensity and average accessible area) in particular those available at the Biological Magnetic Resonance Data Bank (**Ulrich et al., 2008**) are also commonly used. These features characterize each amino-acid of the protein individually or when using window-based features an overall environment of each amino-acid (**Krallinger et al., 2005**; **Preto and Moreira, 2020**). Whole-sequence protein features are descriptors common to all amino-acids of the system, but that complement the variability introduced by single amino-acid level analysis. Furthermore, these features can be particularly useful to characterize Protein-Protein Interactions (PPI) as they provide thorough characterizations of the protein chains (**Cao et al., 2013**; **Rehman et al., 2013**; **Xie et al., 2013**; **Xiao et al., 2015**; **Chen et al., 2018**). Features encompassing conservation information presume the calculation of a Multiple Sequence Alignment (MSA), which takes the target protein sequence as input and aligns it with other known protein sequences. Several tools were developed and fine-tuned for this purpose, such as Clustal Omega (**Sievers and Higgins, 2014**), Basic Local Alignment Search Tool (BLAST) and Psi-BLAST (**Database resources of the National Center for Biotechnology Information, 2018**). Upon these alignments, a PSSM can be calculated, and used to score every amino-acid position according to its conservation, depending on its accordance with the remaining aligned protein sequences. The conservation scores for each amino-acid are valuable features, as highly conserved residues tend to be more relevant in both protein structure and function. This information allows the PSSM to represent structural information and as such, albeit being sequence-based, methods, provide meaningful contributions to overall prediction models (**Marks et al., 2011**; **Preto et al., 2018**), and in particular to GPCR-ligand DTI predictors (**Hu et al., 2016b**; **Shi et al., 2019**; **Li et al., 2019b**). A more recent approach successfully uses representation learning to automatically extract the most significant characteristics of GPCR protein sequences and express them as features (**Cruz-Barbosa et al., 2018**). Differently, other approaches focus on minimizing the noise of less relevant features using methods such as wrapper feature selection (**Redkar et al., 2020**).

However, researchers should take into account that if structural data is available and easy to use, it is generally more reliable than sequence-based partly as it also comprises sequence-based information (**Cross, 2018**; **Jastrzębski et al., 2019**). Some approaches can take the raw atom coordinates and process them inside DL architectures, whereas others can add a prior step in which structural features are abstracted from the coordinates before they are subject to prediction tasks. The construction of feature vectors from contact matrices between the amino-acids and physicochemical distance matrices is one of the approaches that was already applied to GPCRs (**Eo et al., 2009**).

5.2 GPCR classification

Various GPCR classification systems were proposed so far, considering different criteria, such as structural and ligand binding (**Bockaert and Pin, 1999**), phylogenetics (**Fredriksson et al., 2003**), amino-acid composition (**Chou and Elrod, 2002**), ligand-specific features (**Okuno et al., 2006**), and proteins' physicochemical properties (**Davies et al., 2007**). Over the last years, ML was extensively used to aid GPCRs classification, to better understand these receptors and assist drug discovery, developing new and more selective drugs with fewer side effects (**Cobanoglu et al., 2011**).

Many of the methods used for GPCR classification are based on SVMs algorithms (**Zhu et al., 2011**; **Nie et al., 2015**; **Hu et al., 2016a**; **Shen et al., 2017**). However, other methods deploy different approaches. For example, Kumari et al. used the domain predictions from five different software as input to all classifiers present (DT J48, Bagging, Naïve Bayes (NB) and Bayes Net) in order to classify GPCRs (**Kumari et al., 2009**). In another work, Cobanoglu et al. developed the GPCRBind method, which classifies Class A GPCRs family through sequence-derived motifs that specify the different subfamilies by identifying the critical ligand interaction sites (**Cobanoglu et al., 2011**). This method makes use of TFI-DF for motif characterization and DT to select the motifs. More recently, CNN was used in conjunction with TM techniques to perform GPCR classification (Man **Li et al., 2017**). Alternatively to the previous methods, GPCR-MPredictor uses a genetic algorithm to construct an ensemble of classifiers (SVM, KNN, PNN, and J48) for GPCR classification into family, subfamily, sub-subfamily, and subtypes with over 80% accuracy for every level (**Naveed and Khan, 2012**). Additionally, RF was used to correctly classify GPCRs that had been previously misclassified (**Shkurin and Vellido, 2017**).

5.3 Importance of pathway analysis

GPCRs participate in many intracellular signaling pathways that are known to trigger several cellular and physiological consequences. Several studies have shown increasing evidence that drugs acting over the same GPCR have different physiological effects, since they modulate different intracellular signaling pathways (**Kenakin, 2019**). Therefore, the stimulation of a pathway by a certain drug has important implications (**Diez-Alarcia et al., 2019**). Indeed, there are multiple signaling pathways activated by a single receptor due to the multiplicity of G proteins known for a variety of receptors that provide one mechanism for this type of activation (**Yang et al., 2020**).

The key importance of pathway mapping is being explored in recent years. For example, a model developed by Diez-Alarcia et al., seems to determine the probability of a molecule to interact with a different GPCR considering pathway information. This method combines concepts from Perturbation Theory (PT) and ML. For a practical case, the authors focused on the prediction of pharmacological compounds with affinity for Serotonin 5-HT2A Receptor (5-HT2AR) (**Diez-Alarcia et al., 2019**). The results of the predictive and experimental experiences indicated that some drugs, formerly defined as selective 5-HT2AR agonist, antagonist, or inverse agonist, are not so specific for this receptor and could demonstrate intrinsic activity different to that previously stated. The conclusions for the 5-HT2AR displayed that this computational approach could help to design new antipsychotic drugs with better efficacy and tolerability profiles (**Diez-Alarcia et al., 2019**).

6 Other areas of AI application to GPCRs

While most AI applications were developed to facilitate in silico drug development and design in terms of HTS, there are also other illustrations in the GPCR field where AI was particularly useful.

6.1 Database construction

For instance, the DAKB-GPCRs database was explicitly developed to compile research of GPCRs involved in Drug Abuse (DA) (**Chen et al., 2019**). The chemogenomic knowledgebase contains information about DA-related protein targets (258 proteins in total, 86 of them are GPCRs), small molecules, and algorithms for data analysis and visualization (**Chen et al., 2019**). Since the structural resolution of DA-related GPCRs has stagnated (only 29 out of 52 published structures in the last 18 years are relevant (**Xiang et al., 2016**)) homology modeling, and MDs were very helpful to build accurate models. Within the DAKB-GPCRs database, the accuracy and diversity of such models' conformations were further optimized by pre-screening them against a training set of GPCR active and inactive ligands after

MDs, using tools such as HTDocking, TargetHunter, NGL, and ANN. As output, for example, a spider plot can be generated to visualize and analyze the data (**Chen et al., 2019**).

6.2 Structural modeling improvement

Another extremely challenging area in the GPCR field is attaining a particular 3D-structure resolution (either by X-ray crystallization or Cryo-EM). The flexible nature of GPCRs and their ability to quickly switch between different conformational states (apo-state, active agonist-bound, antagonist-bound, inactive) makes them difficult candidates for protein purification. Consequently, most of this protein family structures (87%) remain unsolved (**Muk et al., 2019**). The knowledge about a target's structure is, however, crucial for SBDD. The most profitable method for structure solvation is finding thermostabilizing mutants suitable for the crystallization process (**Tate and Schertler, 2009**; **Muk et al., 2019**). The principle of thermostabilization of GPCRs via point mutations has helped to solve more than 30 structures (**Popov et al., 2019**) by using either systematic alanine scanning (**Errey et al., 2015**), protein evolution (**Schütz et al., 2016**) or a combination of both. Therein lies the challenge (and the costs) for determining the suitable combination of mutations. To avoid intensive laboratory labor, computational applications have helped to accelerate the process. A study by Muk et al. made use of four different ML approaches (RF, cost-sensitive RF, adaptive boosting and gradient boosting) to improve the method of thermostabilized mutant GPCRs via systematic alanine scanning mutations (**Muk et al., 2019**). Their method combines sequence-, structure-, and dynamics-based molecular properties of GPCRs that recapitulate their stability to predict thermostable mutations ahead of experiments. Similar approaches were also developed, for example, the GPCRdb (**Munk et al., 2019**; **Popov et al., 2019**) construct design tool and the CompoMug (**Popov et al., 2018**), which used information about known thermostabilizing point mutations of any GPCR and applies it to the target of interest.

Besides the application of ML-methods to improve the process of structure determination of GPCRs, in silico 3D-modeling has become indispensable over the years and is continuously improving with the increasing numbers of solved X-ray crystal structures and computational techniques. Many algorithms were already employed to describe and predict the membrane-embedded alpha-helical polytopic nature of GPCRs (**Yang et al., 2013**). Mostly, the TransMembrane Helix domain topology (TMH), such as inter-TMH residue contacts, TMH-TMH interactions and residue-residue contact patterns (crucial for ab initio protein folding) have proven to be appropriate targets for ML predictions (**Yang et al., 2013**). Published methods propose predicting such information from the primary sequence by either Correlated Mutation Analysis (CMA) and/or ML-based methods (**Yang et al., 2013**). Some of these examples include TMHit (**Lo et al., 2009**), MemBrain (CMA + ML-based method (**Yang et al., 2013**)), PSIPRED (**Buchan and Jones, 2019**), MEMSAT3 (**Jones, 2007**), DeepMetaPSICOV (**Kandathil et al., 2019**) and many applications from the ZhangLab (https://zhanglab.ccmb.med.umich.edu/research/) such as GPCR-I-Tasser (**Zhang et al., 2015b**).

6.3 GPCR stabilization and modeling

As it was already highlighted, modeling MPs is still very challenging compared to soluble proteins (**Ray et al., 2010**; **Almeida et al., 2017**). Many standard sequence-based methods for model quality evaluation (not based upon ML) were initially developed for water-soluble proteins but can also be applied to MPs (**Forrest et al., 2006**; **Ray et al., 2010**). Indeed, methods such as ProQ (**Wallner and Elofsson, 2006**), Rosetta (**Rohl et al., 2004**), and many more were developed in the last years to evaluate such structure prediction models. ProQ can generate many models either by using multi-template alignments or a hybrid template generated from the several individual templates (**Wallner and Elofsson, 2006**), while Rosetta uses a different approach by sampling different regions of the conformational space (**Rohl et al., 2004**). For the best model's discrimination an ideal scoring function is the output of such methods, measuring the distance between the model and the native structure correlating between score and quality (**Ray et al., 2010**). Available scoring functions can be split into three categories: physics, knowledge and learning-based (**Ray et al., 2010**). Physics scoring functions describe the interaction between atoms as accurately as possible; typical examples are molecular mechanics force fields such as CHARMM (**Brooks et al., 1983**) or AMBER (**Weiner et al., 1984**). Knowledge-based scoring function (based upon the Boltzmann device) derives a probability distribution from native structures features (**Sippl, 1990**; **Lüthy et al., 1992**; **Samudrala and Moult, 1998**; **Zhang and Kim, 2000**; **Ray et al., 2010**). Lastly, learning-based functions such as ANNs or SVMs are trained to distinguish between correct and incorrect models based on structural features to predict the actual quality of a given model (**Martinez et al., 1991**; **Fain et al., 2002**; **Wallner and Elofsson, 2003**; **Pawlowski et al., 2008**; **Ray et al., 2010**). Nevertheless, for GPCRs, only some methods focus on the general problem of predicting the correctness of different parts of a structural model, rather than just evaluating the global model (**Wallner and Elofsson, 2006**). Moreover, insertion into the lipid bilayer environment is an additional peculiarity, which most predictors do not address. The developers of ProQRes and ProQM (**Ray et al., 2010**), tried to include in their scoring function all relevant parameters for membrane-specific properties such as topology and *Z*-coordinate prediction as well as the grade of conservation and sequence profile information.

More than 200 receptors have known ligands, either peptides or small molecules, which enable them to perform their biological function, although, for almost 100 receptors, the ligand remains unknown (orphan GPCRs) (**Foster et al., 2019**). Despite the undeniable importance for orphan GPCRs, such ligands' discovery is transverse to all GPCRs mainly due to their potential as a therapeutic target. In silico approaches can be applied to predict the peptide-receptor combination correctly, and the prediction of endogenous peptides originated from proteolytic sites in proteins (**Foster et al., 2019**). Active endogenous peptides are usually the result of Post-Translational Modifications (PTMs) to inactive precursors, which often carry an arginine as the first amino-acid in their N-terminus (**Kliger, 2010**), are highly conserved, and located between cleavage sites (**Foster et al., 2019**). Shiraishi et al. successfully integrated computational and experimental approaches to neuropeptides and receptors interactions (**Shiraishi et al., 2019**). An SVM model was used to obtain Peptide Descriptors (PDs) from chemical and sequence data of ligand-receptor pairs. These PDs were double-optimized using Genetic Algorithm-based Feature Selection (GAFS) to improve the result independently of the species, making the neuropeptide-GPCR pairs prediction with their method suitable for any species. Of the 29 pairs predicted in silico, 12 were validated by experimental methods, 11 of which were specific for the species under study. The phylogenetic tree analysis also revealed previously unknown interactions between neuropeptides and GPCRs, paving the way for future research. Similarly, Foster et al. also used an integrative approach that benefited from a first stage of computational prediction tool for peptide ligands and receptors before experimental validation to complement the ligand-GPCR signaling system (**Foster et al., 2019**). From previously described peptide ligands, they concluded that 67% regulate cellular functions through interaction with GPCRs, identifying several relationships where, on average, each GPCR is regulated by 2.9 peptides with higher affinity and potency levels. For the prediction of peptide ligands from the human proteome, the authors used a RF classifier obtained from similar length and evolutionary conservation values of previously described peptides that match the precursor signal peptide and cleavage sites. After experimental studies, the authors reported a 17% increase in known interactions of the human peptidergic signaling network along with 74% of the peptides being validated as developing receptor-dependent responses (**Foster et al., 2019**).

6.4 GPCR dimerization

In general, GPCR dimerization's influence during the receptor's life cycle was already summarized in several publications (**Terrillon and Bouvier, 2004**; Barreto et al., 2020a, 2020b; **Preto et al., 2020**). It can be pinpointed in three stages: ontogeny (localization), membrane-specific actions (ligand-promoted regulation, pharmaceutical diversity, signal transduction), and internalization. Ontogeny ensures the correct folding and maturation of the receptors and, consequently, allows cell surface delivery (**Bulenger et al., 2005**; **Lopez-Gimenez et al., 2007**). In contrast, the GPCR oligomerization may also anticipate limited receptor maturation and cell surface delivery by causing ER-retention (**Janovick et al., 2007**; **Lopez-Gimenez et al., 2007**). For the internalization process dimer −/oligomerization can promote the co-internalization of both receptors after stimulation of only one. By choice, an oligomeric structure may also prevent agonist-induced internalization of the targeted receptor (**Terrillon and Bouvier, 2004**). Co-internalization can also be associated with cross-desensitization of the signaling activities (**Terrillon and Bouvier, 2004**). Although the maturation and internalization process of GPCRs is of potential interest, the physiological consequences remain to be determined. In a nutshell, the cause for heterodimerization events at these two stages are due to naturally occurring mutations, which are of pathophysiological relevance (**Lopez-Gimenez et al., 2007**).

The relevance of GPCR-oligomers has increased over the last few years as more disease-specific heteromers are being identified (**Gupta et al., 2010**; **Rozenfeld et al., 2011**; **Gomes et al., 2013**; Barreto et al., 2020a, 2020b). Hence, it is now widely accepted that highly dynamic GPCR networks exist and that the monomer's functions such as ligand binding affinity and signaling may be altered through oligomer formation (**Terrillon and Bouvier, 2004**; Barreto et al., 2020a, 2020b). This paradigm shift from basic signal transduction towards a more holistic and multifactorial view on GPCRs challenges rational drug design (**Terrillon and Bouvier, 2004**). Therefore, computational studies (including AI approaches) on GPCR oligomerization are necessary to understand the disease mechanism and support experimental studies to reveal novel pathways for treating GPCR-linked illnesses (**Nemoto et al., 2016**). At the plasma membrane, a GPCR-complex can either be a target for dynamic regulation of ligand-binding, promote or inhibit ligand binding cooperativity or potentiating, attenuating downstream signaling or even changing G protein selectivity (**Terrillon and Bouvier, 2004**). For these kinds of PPI, several ML-based methods and web servers for the prediction of their interfaces, such as WHISCY (**de Vries et al., 2006**) and ISIS (**Ofran and Rost, 2007**), are well-established and were reviewed by Barreto et al. (2020a, 2020b). However, not all were developed explicitly for GPCR dimer interface prediction and their modulation. Until today, there is not a method that covers the complexity of oligomeric systems, and as such, these innovative ML-based methods may provide strategic prediction tools (Barreto et al., 2020a, 2020b).

6.5 Combining molecular dynamics with artificial intelligence

The increasing number of GPCRs available structures, in particular with intracellular partners G-Proteins and Arrestins, and the technological advances in computational power allows for bigger systems and longer timescales (microseconds) MD simulations, which created a "big data" problem in their analysis (**Díaz et al., 2019**). The currently reported integration of MD simulations and ML algorithms shows promising results (**Almeida et al., 2017**; **Plante et al., 2019**). For example, Plante et al. presented an ML approach to analyze GPCR-ligand MD simulations (**Plante et al., 2019**) using 5-HT2A and D2 receptors as study cases. The atomic coordinates calculated throughout the simulations were converted into RGB code to form an image that was readable by a DNN-based pipeline. This novel approach successfully classified GPCR conformations by ligand class (full, partial, and inverse agonist), and allowed authors to identify the structural motifs that undergo conformational changes for each type of molecule studied (**Plante et al., 2019**).

The Marta Filizola group recently published another ML/MD protocol to better estimate kinetic properties of (un)binding of a ligand to GPCR. The rate of dissociation of a drug is an important predictor of its in vivo efficacy. However, the timescale of drug dissociation is around the minute, which would be computationally inefficient to simulate. Enhanced sampling methods, such as infrequent metadynamics (used in Filizola's group work) were proposed to reduce simulation time required to observe drug dissociation. Nevertheless, these methods require identifying a Reaction Coordinate (RC) capable of successfully describing the dissociation process, which is an incredibly challenging task in a complex system such as GPCR. Filizola's group reported a possible solution to this problem by using features extracted from a short, unbiased MD as an initial dataset that was fed into a pipeline that used state of the art ML methods for dimensionality reduction with Automatic Mutual Information Noise Omission (AMINO). Furthermore, it used Reweighted Autoencoded Variational Bayes for Enhanced sampling to help determine the optimal Reaction Coordinate for infrequent metadynamics (RAVE). This protocol successfully estimated two prototypical opioid receptor drugs' kinetic rates at a reduced computational cost while granting atomic resolution of transitional structures throughout the unbinding pathway (**Lamim Ribeiro et al., 2020**).

7 R&D companies

Between 1985 and 2005, 91% of FDA approved drugs were associated with a patent detained by the private sector (**Sampat and Lichtenberg, 2011**). However, almost half of these patents cited a public research paper or patent (**Sampat and Lichtenberg, 2011**), which means that the private sector is leveraging the generated knowledge from the public sector to patent more improved and adapted ideas to meet the market's needs. This somewhat symbiotic relationship denotes the importance of valuing both the public and the private research. Although the private sector clearly dominates the development, the research part is still largely driven by the academia. **Table 3** shows a group of companies that developed, adapted, or enhanced computational methods or pipelines, that can be used to better study GPCRs.

8 Concluding remarks

In the present chapter, we reviewed over 60 works that deploy AI-based methods to improve GPCR characterization/prediction of function and/or structure. Many of these works were associated with DTI prediction, which is not surprising since this task is still cumbersome and far from being resolved. However, their study remains very promising, as it would trigger a whole new landscape of drug design and development.

Indeed, ML application in GPCRs is still in its embryo form; however, the deployment of similar techniques to soluble proteins validates and boosts their usage to this clinically relevant class of MPs.

Funding

This work was funded by COMPETE 2020—Operational Program for Competitiveness and Internationalization and portuguese national funds via FCT—Fundação para a Ciência e a Tecnologia, under projects POCI-01-0145-FEDER-031356, PTDC/QUI-OUT/32243/2017and UIDB/04539/2020. António Preto, Carlos Barreto, Beatriz Buschbell, Nícia Rosário-Ferreira and Catarina Marques-Pereira were also supported by FCT through Ph.D. scholarships SFRH/BD/144966/2019, SFRH/BD/145457/2019, SFRH/BD/149709/2019, PD/BD/135179/2017 and 2020.07766.BD, respectively. Raquel Gouveia was also supported by FCT and DGES—Direção Geral do Ensino Superior, through fellowship for Curso de Verão in "Metodologias de Investigação Científica"—Módulo de I&D "Metodologias

Table 3 R&D companies, with significant reported advances.

Company name	*Description*	*URL*
Accutar Biotech	DNN for target-ligand docking	https://www.accutarbio.com/products/orbital/
BenevolentAI	AI to discover potential new drugs for chronic kidney disease and idiopathic pulmonary fibrosis	https://www.benevolent.com/what-we-do
Confo Therapeutics	Structure-based drug design	https://www.confotherapeutics.com/confo-technology-suite/
Exscientia Sumitomo Dainippon	Active learning on the discovery of bispecific compound that activates two GPCR receptors from different families	https://www.exscientia.ai/news-insights/exscientia-ltd-reaches-first-delivery-milestone-in-collaboration-with-sumitomo-dainippon-pharma-co-ltd
MedChemica	Drug combination evaluation through RF	https://www.medchemica.com/case-studies/target-id/
Menten.AI	Quantum computer protein-based drug design	https://menten.ai/about
NuMedii	Repurpose of FDA approved GPCR related drugs for small cell lung cancer	**Jahchan et al. (2013)**
PharmCADD	Prediction of 3D structures of proteins	http://pharmcadd.com/pharmulator-2/
ProteinQure	Design of protein-based therapeutics	https://www.proteinqure.com/
Schrödinger, Inc.	Physics-based computational platform applied to a variety of GPCRs	https://www.schrodinger.com/
Sosei Heptares	Drug discovery, GPCRs modeling and determination, novel GPCR agent design	https://soseiheptares.com/our-science/proprietary-research-platform.html
Vernalis Research	Fragment and structure-based drug discovery	https://www.vernalis.com/resources/#software
WuXi AppTec	DELopen (4.4 billion compounds), fragment-based drug discovery	http://rsd.wuxiapptec.com/discovery-services

Avançadas para o Estudo do Cérebro", under the program "Verão com Ciência". Ian Ferreira, Nádia Pereira and Ana Teresa Gaspar were also supported by FCT and DGES—Direção Geral do Ensino Superior, through fellowships for Curso de Verão in "Computational Biology", under the program "Verão com Ciência". Authors would also like to acknowledge ERNEST—European research Network on Signal Transduction, CA18133 and STRATAGEM—New diagnostic and therapeutic tools against multidrug-resistant tumors, CA17104.

References

Allikalt, A., et al., 2020. Quantitative analysis of fluorescent ligand binding to dopamine D3 receptors using live cell microscopy. FEBS Journal 1–19. https://doi.org/10.1111/febs.15519.

Almeida, J.G., et al., 2017. Membrane proteins structures: A review on computational modeling tools. Biochimica et Biophysica Acta - Biomembranes 1859 (10)https://doi.org/10.1016/j.bbamem.2017.07.008.

Alpaydin, E., 2014. Multilayer Perceptrons. In: Introduction to Machine Learning. MIT Press, pp. 267–316, Available at. https://ieeexplore.ieee.org/document/6917150.

Arakaki, A.K.S., Pan, W.-A., Trejo, J., 2018. GPCRs in cancer: Protease-activated receptors, endocytic adaptors and signaling. International Journal of Molecular Sciences 19 (7)https://doi.org/10.3390/ijms19071886.

Asgari, E., Mofrad, M.R.K., 2015. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One 10 (11), e0141287https://doi.org/10.1371/journal.pone.0141287.

Badillo, S., et al., 2020. An Introduction to machine learning. Clinical Pharmacology and Therapeutics 107 (4), 871–885. https://doi.org/10.1002/cpt.1796.

Baig, M.H., et al., 2016. Computer aided drug design: Success and limitations. Current Pharmaceutical Design 22 (5), 572–581. https://doi.org/10.2174/1381612822666151125000550.

Baldi, P., 2012. Autoencoders, unsupervised learning, and deep architectures. In: Guyon, I., et al. (Eds.). JMLR Workshop and Conference Proceedings (Proceedings of Machine Learning Research), Bellevue, Washington, USA. pp. 37–49, Available at:. http://proceedings.mlr.press/v27/baldi12a.html.

Ballesteros, J., et al., 1998. Functional microdomains in G-protein-coupled receptors. The conserved arginine-cage motif in the gonadotropin-releasing hormone receptor. The Journal of Biological Chemistry 273 (17), 10445–10453. https://doi.org/10.1074/jbc.273.17.10445.

Bandholtz, S., et al., 2012. Molecular evolution of a peptide GPCR ligand driven by artificial neural networks. PLoS One 7 (5), e36948https://doi.org/10.1371/journal.pone.0036948.

Barreto, C.A.V., et al., 2020. Prediction and targeting of GPCR oligomer interfaces. Progress in Molecular Biology and Translational Sciencehttps://doi.org/10.1016/bs.pmbts.2019.11.007.

Barreto, C.A.V., et al., 2020. Prediction and targeting of GPCR oligomer interfaces. In: Giraldo, J., Ciruela, F. (Eds.), Oligomerization in Health and Disease: From Enzymes to G Protein-Coupled Receptors, Academic Press, pp. 105–149. https://doi.org/10.1016/bs.pmbts.2019.11.007, ch. 4.

Bartoli, L., et al., 2009. CCHMM_PROF: A HMM-based coiled-coil predictor with evolutionary information. Bioinformatics 25 (21), 2757–2763. https://doi.org/10.1093/bioinformatics/btp539.

Basile, A.O., Yahi, A., Tatonetti, N.P., 2019. Artificial intelligence for drug toxicity and safety. Trends in Pharmacological Sciences 40 (9), 624–635. https://doi.org/10.1016/j.tips.2019.07.005.

Basith, S., Cui, M., et al., 2018. Exploring G protein-coupled receptors (GPCRs) ligand space via cheminformatics approaches: Impact on rational drug design. Frontiers in Pharmacology 9, 128. https://doi.org/10.3389/fphar.2018.00128.

Basith, S., Manavalan, B., et al., 2018. iGHBP: Computational identification of growth hormone binding proteins from sequences using extremely randomised tree. Computational and Structural Biotechnology Journal 16, 412–420. https://doi.org/10.1016/j.csbj.2018.10.007.

Bemister-Buffington, J., et al., 2020. Machine learning to identify flexibility signatures of class A GPCR inhibition. Biomolecules 10 (3)https://doi.org/10.3390/biom10030454.

Bieler, M., et al., 2016. Designing multi-target compound libraries with gaussian process models. Molecular Informatics 35 (5), 192–198. https://doi.org/10.1002/minf.201501012.

Bockaert, J., Pin, J.P., 1999. Molecular tinkering of G protein-coupled receptors: An evolutionary success. The EMBO Journal 18 (7), 1723–1729. https://doi.org/10.1093/emboj/18.7.1723.

Boehmke, B., Brandon, M.G., 2019. Hands-On Machine Learning With R, 1st edn Chapman and Hall/CRC.

Breer, H., Kleene, R., Hinz, G., 1985. Molecular forms and subunit structure of the acetylcholine receptor in the central nervous system of insects. The Journal of Neuroscience 5 (12), 3386–3392. https://doi.org/10.1523/JNEUROSCI.05-12-03386.1985.

Brooks, B.R., et al., 1983. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. Journal of Computational Chemistry 4 (2), 187–217. https://doi.org/10.1002/jcc.540040211.

Buchan, D.W.A., Jones, D.T., 2019. The PSIPRED protein analysis workbench: 20 years on. Nucleic Acids Research 47 (W1), W402–W407. https://doi.org/10.1093/nar/gkz297.

Bueschbell, B., et al., 2019. A complete assessment of dopamine receptor-ligand interactions through computational methods. Molecules 24 (7)https://doi.org/10.3390/molecules24071196.

Bulenger, S., Marullo, S., Bouvier, M., 2005. Emerging role of homo- and heterodimerization in G-protein-coupled receptor biosynthesis and maturation. Trends in Pharmacological Sciences 26 (3), 131–137. https://doi.org/10.1016/j.tips.2005.01.004.

Butkiewicz, M., et al., 2019. Identification of novel allosteric modulators of metabotropic glutamate receptor subtype 5 acting at site distinct from 2-methyl-6-(phenylethynyl)-pyridine binding. ACS Chemical Neuroscience 10 (8), 3427–3436. https://doi.org/10.1021/acschemneuro.8b00227.

Cai, C.Z., et al., 2003. SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Research 31 (13), 3692–3697. https://doi.org/10.1093/nar/gkg600.

Cao, D., et al., 2012. ADMET evaluation in drug discovery. 11. Pharmaco kinetics knowledge base (PKKB): A comprehensive database of pharmacokinetic and toxic properties for drugs. Journal of Chemical Information and Modeling 52 (5), 1132–1137. https://doi.org/10.1021/ci300112j.

Cao, D.-S., et al., 2013. PyDPI: Freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. Journal of Chemical Information and Modeling 53 (11), 3086–3096. https://doi.org/10.1021/ci400127q.

Cereto-Massagué, A., et al., 2015. Molecular fingerprint similarity search in virtual screening. Methods (San Diego, Calif.) 71, 58–63. https://doi.org/10.1016/j.ymeth.2014.08.005.

Chakraborty, C., et al., 2017. Micro-environmental signature of the interactions between druggable target protein, dipeptidyl peptidase-IV, and anti-diabetic drugs. Cell Journal 19 (1), 65–83. https://doi.org/10.22074/cellj.2016.4865.

Chan, W.K.B., et al., 2015. GLASS: a comprehensive database for experimentally validated GPCR-ligand associations. Bioinformatics 31 (18), 3035–3042. https://doi.org/10.1093/bioinformatics/btv302.

Chan, W., et al., 2018. GPCR-EXP: A Semi-Manually Curated Database for Experimentally-Solved and Predicted GPCR Structures. Available at. https://zhanglab.ccmb.med.umich.edu/GPCR-EXP/.

Chan, H.C.S., et al., 2019. New binding sites, new opportunities for GPCR drug discovery. Trends in Biochemical Sciences 44 (4), 312–330. https://doi.org/10.1016/j.tibs.2018.11.011.

Chandrasekaran, B., et al., 2018. Computer-Aided Prediction of Pharmacokinetic (ADMET) Properties. In: Dosage Form Design Parameters. Elsevier Inchttps://doi.org/10.1016/B978-0-12-814421-3.00021-X.

Chen, Z., et al., 2018. iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics 34 (14), 2499–2502. https://doi.org/10.1093/bioinformatics/bty140.

Chen, M., et al., 2019. DAKB-GPCRs: An integrated computational platform for drug abuse related GPCRs. Journal of Chemical Information and Modeling 59 (4), 1283–1289. https://doi.org/10.1021/acs.jcim.8b00623.

Chou, K.-C., Elrod, D.W., 2002. Bioinformatical analysis of G-protein-coupled receptors. Journal of Proteome Research 1 (5), 429–433. https://doi.org/10.1021/pr025527k.

Chun, L., Zhang, W., Liu, J., 2012. Structure and ligand recognition of class C GPCRs. Acta Pharmacologica Sinica 33 (3), 312–323. https://doi.org/10.1038/aps.2011.186.

Chung, S., Funakoshi, T., Civelli, O., 2008. Orphan GPCR research. British Journal of Pharmacology 153 (supplement 1), S339–S346. https://doi.org/10.1038/sj.bjp.0707606.

Chupakhin, V., et al., 2013. Predicting ligand binding modes from neural networks trained on protein-ligand interaction fingerprints. Journal of Chemical Information and Modeling 53 (4), 763–772. https://doi.org/10.1021/ci300200r.

Cobanoglu, M.C., Saygin, Y., Sezerman, U., 2011. Classification of GPCRs using family specific motifs. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8 (6), 1495–1508. https://doi.org/10.1109/TCBB.2010.101.

Coleman, J.L.J., Ngo, T., Smith, N.J., 2017. The G protein-coupled receptor N-terminus and receptor signalling: N-tering a new era. Cellular Signalling 33, 1–9. https://doi.org/10.1016/j.cellsig.2017.02.004.

Cross, J.B., 2018. Methods for virtual screening of GPCR targets: Approaches and challenges. Methods in Molecular Biology (Clifton, N.J.) 1705, 233–264. https://doi.org/10.1007/978-1-4939-7465-8_11.

Cruz-Barbosa, R., Ramos-Pérez, E.-G., Giraldo, J., 2018. Representation learning for class C G protein-coupled receptors classification. Molecules 23 (3)https://doi.org/10.3390/molecules23030690.

Cunningham, P., Cord, M., Delany, S.J., 2008. In: Cord, M., Cunningham, P. (Eds.), Supervised learning BT—Machine learning techniques for multimedia: Case studies on organization and retrieval. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 21–49. https://doi.org/10.1007/978-3-540-75171-7_2.

Da, C., Kireev, D., 2014. Structural protein-ligand interaction fingerprints (SPLIF) for structure-based virtual screening: Method and benchmark study. Journal of Chemical Information and Modeling 54 (9), 2555–2561. https://doi.org/10.1021/ci500319f.

Daina, A., Michielin, O., Zoete, V., 2017. SwissADME: A free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. Scientific Reports 7 (1), 42717. https://doi.org/10.1038/srep42717.

Davies, M.N., et al., 2007. On the hierarchical classification of G protein-coupled receptors. Bioinformatics (Oxford, England) 23 (23), 3113–3118. https://doi.org/10.1093/bioinformatics/btm506.

de Vries, S.J., van Dijk, A.D.J., Bonvin, A.M.J.J., 2006. WHISCY: what information does surface conservation yield? Application to data-driven docking. Proteins 63 (3), 479–489. https://doi.org/10.1002/prot.20842.

Deng, Z., Chuaqui, C., Singh, J., 2004. Structural interaction fingerprint (SIFt): A novel method for analyzing three-dimensional protein-ligand binding interactions. Journal of Medicinal Chemistry 47 (2), 337–344. https://doi.org/10.1021/jm030331x.

Díaz, , Dalton, J.A.R., Giraldo, J., 2019. Artificial intelligence: A novel approach for drug discovery. Trends in Pharmacological Sciences 40 (8), 550–551. https://doi.org/10.1016/j.tips.2019.06.005.

Diez-Alarcia, R., et al., 2019. Big data challenges targeting proteins in GPCR signaling pathways; combining PTML-ChEMBL models and [35S]GTPγS binding assays. ACS Chemical Neuroscience 10 (11), 4476–4491. https://doi.org/10.1021/acschemneuro.9b00302.

Djoumbou-Feunang, Y., et al., 2019. BioTransformer: A comprehensive computational tool for small molecule metabolism prediction and metabolite identification. Journal of Cheminformatics 11 (1), 2. https://doi.org/10.1186/s13321-018-0324-5.

Dror, R.O., et al., 2011. Pathway and mechanism of drug binding to G-protein-coupled receptors. Proceedings of the National Academy of Sciences of the United States of America 108 (32), 13118–13123. https://doi.org/10.1073/pnas.1104614108.

Eo, H.-S., et al., 2009. A machine learning based method for the prediction of G protein-coupled receptor-binding PDZ domain proteins. Molecules and Cells 27 (6), 629–634. https://doi.org/10.1007/s10059-009-0091-2.

Errey, J.C., et al., 2015. G protein-coupled receptors in drug discovery. In: Methods in Molecular Biology. Humana Press, New York, https://doi.org/10.1007/978-1-4939-2914-6_1.

Ezzat, A., et al., 2019. Computational prediction of drug-target interactions using chemogenomic approaches: An empirical survey. Briefings in Bioinformatics 20 (4), 1337–1357. https://doi.org/10.1093/bib/bby002.

Fain, B., Xia, Y., Levitt, M., 2002. Design of an optimal Chebyshev-expanded discrimination function for globular proteins. Protein Science 11 (8), 2010–2021. https://doi.org/10.1110/ps.0200702.

Fang, Y., Kenakin, T., Liu, C., 2015. Editorial: Orphan GPCRs as emerging drug targets. Frontiers in Pharmacology 6, 295. https://doi.org/10.3389/fphar.2015.00295.

Feldman, R., Sanger, J., 2007. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press.

Ferraro, M., et al., 2020. Multi-target dopamine D3 receptor modulators: Actionable knowledge for drug design from molecular dynamics and machine learning. European Journal of Medicinal Chemistry 188, 111975https://doi.org/10.1016/j.ejmech.2019.111975.

Flood, A.B., 1990. Peaks and pits of using large data bases to measure quality of care. International Journal of Technology Assessment in Health Care 6 (2), 253–262. https://doi.org/10.1017/s0266462300000775.

Forrest, L.R., Tang, C.L., Honig, B., 2006. On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins. Biophysical Journal 91 (2), 508–517. https://doi.org/10.1529/biophysj.106.082313.

Foster, S.R., et al., 2019. Discovery of human signaling systems: Pairing peptides to G protein-coupled receptors. Cell 179 (4), 895–908.e21. https://doi.org/10.1016/j.cell.2019.10.010.

Francis, L., 2014. Unsupervised learning. In: Frees, E.W., Meyers, G., Derrig, R.A. (Eds.), Predictive Modeling Applications in Actuarial Science: Volume 1: Predictive Modeling Techniques. International Series on Actuarial Science Cambridge University Press, Cambridge, pp. 280–312. https://doi.org/10.1017/CBO9781139342674.012.

Fredriksson, R., et al., 2003. The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints. Molecular Pharmacology 63 (6), 1256–1272. https://doi.org/10.1124/mol.63.6.1256.

Gao, Q.-B., Ye, X.-F., He, J., 2013. Classifying G-protein-coupled receptors to the finest subtype level. Biochemical and Biophysical Research Communications 439 (2), 303–308. https://doi.org/10.1016/j.bbrc.2013.08.023.

Gatica, E.A., Cavasotto, C.N., 2012. Ligand and decoy sets for docking to G protein-coupled receptors. Journal of Chemical Information and Modeling 52 (1), 1–6. https://doi.org/10.1021/ci200412p.

Gautam, A., et al., 2013. In silico approaches for designing highly effective cell penetrating peptides. Journal of Translational Medicine 11, 74. https://doi.org/10.1186/1479-5876-11-74.

Gayvert, K.M., Madhukar, N.S., Elemento, O., 2016. A data-driven approach to predicting successes and failures of clinical trials. Cell Chemical Biology 23 (10), 1294–1301. https://doi.org/10.1016/j.chembiol.2016.07.023.

Gether, U., 2000. Uncovering molecular mechanisms involved in activation of G protein-coupled receptors. Endocrine Reviews 21 (1), 90–113. https://doi.org/10.1210/edrv.21.1.0390.

Ghosh, D., Chinnaiyan, A.M., 2005. Classification and selection of biomarkers in genomic data using LASSO. Journal of Biomedicine and Biotechnology 2005, 427208. https://doi.org/10.1155/JBB.2005.147.

Gloriam, D.E., Fredriksson, R., Schiöth, H.B., 2007. The G protein-coupled receptor subset of the rat genome. BMC Genomics 8 (1), 338. https://doi.org/10.1186/1471-2164-8-338.

Gomes, I., et al., 2013. Disease-specific heteromerization of G-protein-coupled receptors that target drugs of abuse. Progress in Molecular Biology and Translational Science 117, 207–265. https://doi.org/10.1016/B978-0-12-386931-9.00009-X.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Grisoni, F., et al., 2018. Molecular descriptors for structure-activity applications: A hands-on approach. Methods in Molecular Biology 1800, 3–53. https://doi.org/10.1007/978-1-4939-7899-1_1.

Gupta, A., et al., 2010. Increased abundance of opioid receptor heteromers after chronic morphine administration. Science Signaling 3 (131), ra54https://doi.org/10.1126/scisignal.2000807.

Han, H., Liu, W., 2019. The coming era of artificial intelligence in biological data science. BMC Bioinformatics 712. https://doi.org/10.1186/s12859-019-3225-3.

Hauser, A.S., et al., 2017. Trends in GPCR drug discovery: New agents, targets and indications. Nature Reviews 16 (12), 829–842. https://doi.org/10.1038/nrd.2017.178.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507. https://doi.org/10.1126/science.1127647.

Hu, B., et al., 2016. Three-dimensional biologically relevant spectrum (BRS-3D): Shape similarity profile based on PDB ligands as molecular descriptors. Molecules (Basel, Switzerland) 21 (11), https://doi.org/10.3390/molecules21111554.

Hu, J., et al., 2016. GPCR-drug interactions prediction using random forest with drug-association-matrix-based post-processing procedure. Computational Biology and Chemistryhttps://doi.org/10.1016/j.compbiolchem.2015.11.007.

Hu, G.-M., Mai, T.-L., Chen, C.-M., 2017. Visualizing the GPCR network: Classification and evolution. Scientific Reports 7 (1), 15495https://doi.org/10.1038/s41598-017-15707-9.

Iacucci, E., et al., 2012. ReLiance: A machine learning and literature-based prioritization of receptor—Ligand pairings. Bioinformatics (Oxford, England) 28 (18), i569–i574. https://doi.org/10.1093/bioinformatics/bts391.

Jabeen, A., Ranganathan, S., 2019. Applications of machine learning in GPCR bioactive ligand discovery. Current Opinion in Structural Biology 55, 66–76. https://doi.org/10.1016/j.sbi.2019.03.022.

Jaber, M., et al., 1996. Dopamine receptors and brain function. Neuropharmacology 35 (11), 1503–1519. https://doi.org/10.1016/S0028-3908(96)00100-1.

Jacob, L., et al., 2008. Virtual screening of GPCRs: An in silico chemogenomics approach. BMC Bioinformatics 9, 1–16. https://doi.org/10.1186/1471-2105-9-363.

Jahchan, N.S., et al., 2013. A drug repositioning approach identifies tricyclic antidepressants as inhibitors of small cell lung cancer and other neuroendocrine tumors. Cancer Discovery 3 (12), 1364–1377. https://doi.org/10.1158/2159-8290.CD-13-0183.

Jang, J.W., et al., 2016. Novel scaffold identification of mGlu1 receptor negative allosteric modulators using a hierarchical virtual screening approach. Chemical Biology and Drug Design 87 (2), 239–256. https://doi.org/10.1111/cbdd.12654.

Janovick, J.A., et al., 2007. Refolding of misfolded mutant GPCR: Post-translational pharmacoperone action in vitro. Molecular and Cellular Endocrinology 272 (1–2), 77–85. https://doi.org/10.1016/j.mce.2007.04.012.

Jastrzębski, S., et al., 2019. Three-dimensional descriptors for aminergic GPCRs: Dependence on docking conformation and crystal structure. Molecular Diversity 23 (3), 603–613. https://doi.org/10.1007/s11030-018-9894-4.

Jones, D.T., 1999. Protein secondary structure prediction based on position-specific scoring matrices. Journal of Molecular Biology 292 (2), 195–202. https://doi.org/10.1006/jmbi.1999.3091.

Jones, D.T., 2007. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. Bioinformatics (Oxford, England) 23 (5), 538–544. https://doi.org/10.1093/bioinformatics/btl677.

Jurafsky, D., Martin, J.H., 2009. Speech and Language Processing, 2nd edn Prentice-Hall, Inc.

Kandathil, S.M., Greener, J.G., Jones, D.T., 2019. Prediction of interresidue contacts with DeepMetaPSICOV in CASP13. Proteins: Structure, Function, and Bioinformatics 87 (12), 1092–1099. https://doi.org/10.1002/prot.25779.

Karnik, S.S., et al., 2003. Activation of G-protein-coupled receptors: A common molecular mechanism. Trends in Endocrinology and Metabolism 14 (9), 431–437. https://doi.org/10.1016/j.tem.2003.09.007.

Katritch, V., Cherezov, V., Stevens, R.C., 2012. Diversity and modularity of G protein-coupled receptor structures. Trends in Pharmacological Sciences 33 (1), 17–27. https://doi.org/10.1016/j.tips.2011.09.003.

Katritch, V., Cherezov, V., Stevens, R.C., 2013. Structure-function of the G protein-coupled receptor superfamily. Annual Review of Pharmacology and Toxicology 53, 531–556. https://doi.org/10.1146/annurev-pharmtox-032112-135923.

Kearnes, S., et al., 2016. Molecular graph convolutions: Moving beyond fingerprints. Journal of Computer-Aided Molecular Design 30 (8), 595–608. https://doi.org/10.1007/s10822-016-9938-8.

Kenakin, T., 2019. Biased receptor signaling in drug discovery. Pharmacological Reviews 71 (2), 267–315. https://doi.org/10.1124/pr.118.016790.

Kingsford, C., Salzberg, S.L., 2008. What are decision trees?. Nature Biotechnology 26 (9), 1011–1013. https://doi.org/10.1038/nbt0908-1011.

Klabunde, T., Hessler, G., 2002. Drug design strategies for targeting G-protein-coupled receptors. ChemBioChem 3 (10), 928–944. https://doi.org/10.1002/1439-7633(20021004)3:10<928::AID-CBIC928>3.0.CO;2-5.

Kliger, Y., 2010. Computational approaches to therapeutic peptide discovery. Biopolymers 94 (6), 701–710. https://doi.org/10.1002/bip.21458.

König, C., et al., 2014. Reducing the n-gram feature space of class C GPCRs to subtype-discriminating patterns. Journal of Integrative Bioinformatics 11 (3), 254. https://doi.org/10.2390/biecoll-jib-2014-254.

König, C., et al., 2015. Label noise in subtype discrimination of class C G protein-coupled receptors: A systematic approach to the analysis of classification errors. BMC Bioinformatics 16, 314. https://doi.org/10.1186/s12859-015-0731-9.

König, C., et al., 2018. Systematic analysis of primary sequence domain segments for the discrimination between class C GPCR subtypes. Interdisciplinary Sciences, Computational Life Sciences 10 (1), 43–52. https://doi.org/10.1007/s12539-018-0286-3.

Kontoyianni, M., 2017. Docking and virtual screening in drug discovery. Methods in Molecular Biology 1647, 255–266. https://doi.org/10.1007/978-1-4939-7201-2_18.

Kooistra, A.J., et al., 2016. KLIFS: A structural kinase-ligand interaction database. Nucleic Acids Research 44 (D1), D365–D371. https://doi.org/10.1093/nar/gkv1082.

Koutsoukas, A., et al., 2017. Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. Journal of Cheminformatics 9 (1), 42. https://doi.org/10.1186/s13321-017-0226-y.

Krallinger, M., Padron, M., Valencia, A., 2005. A sentence sliding window approach to extract protein annotations from biomedical articles. BMC Bioinformatics 6 (supplement 1), S19. https://doi.org/10.1186/1471-2105-6-S1-S19.

Krishnan, A., et al., 2016. Classification, nomenclature, and structural aspects of adhesion GPCRs. Handbook of Experimental Pharmacology 234, 15–41. https://doi.org/10.1007/978-3-319-41523-9_2.

Krumm, B., Roth, B.L., 2020. A structural understanding of class B GPCR selectivity and activation revealed. Structure 28 (3), 277–279. https://doi.org/10.1016/j.str.2020.02.004.

Kruse, A.C., et al., 2013. Activation and allosteric modulation of a muscarinic acetylcholine receptor. Nature 504 (7478), 101–106. https://doi.org/10.1038/nature12735.

Kubat, M., 2017. An Introduction to Machine Learning, 2nd edn Springer, Cham, https://doi.org/10.1007/978-3-319-63913-0.

Kulis, B., et al., 2009. Semi-supervised graph clustering: A kernel approach. Machine Learning 74 (1), 1–22. https://doi.org/10.1007/s10994-008-5084-4.

Kumari, T., Pant, B., Pardasani, K.R., 2009. A model for the evaluation of domain based classification of GPCR. Bioinformation 4 (4), 138–142.

Lagerström, M.C., Schiöth, H.B., 2008. Structural diversity of G protein-coupled receptors and significance for drug discovery. Nature Reviews. Drug Discovery 7 (4), 339–357. https://doi.org/10.1038/nrd2518.

Lamim Ribeiro, J.M., Provasi, D., Filizola, M., 2020. A combination of machine learning and infrequent metadynamics to efficiently predict kinetic rates, transition states, and molecular determinants of drug dissociation from G protein-coupled receptors. The Journal of Chemical Physics 153 (12), 124105. https://doi.org/10.1063/5.0019100.

Lane, J.R., et al., 2017. A kinetic view of GPCR allostery and biased agonism. Nature Chemical Biology 13 (9), 929–937. https://doi.org/10.1038/nchembio.2431.

Lavecchia, A., 2015. Machine-learning approaches in drug discovery: Methods and applications. Drug Discovery Today 20 (3), 318–331. https://doi.org/10.1016/j.drudis.2014.10.012.

Lee, J.H., Lee, S., Choi, S., 2010. In silico classification of adenosine receptor antagonists using Laplacian-modified naïve Bayesian, support vector machine, and recursive partitioning. Journal of Molecular Graphics and Modelling 28 (8), 883–890. https://doi.org/10.1016/j.jmgm.2010.03.008.

Li, M., Ling, C., Gao, J., 2017. An efficient CNN-based classification on G-protein coupled receptors using TF-IDF and N-gram. In: 2017 IEEE Symposium on Computers and Communications (ISCC). pp. 924–931. https://doi.org/10.1109/ISCC.2017.8024644.

Li, L., et al., 2019. Predicting protein-ligand interactions based on bow-pharmacological space and Bayesian additive regression trees. Scientific Reports 9 (1), 1–12. https://doi.org/10.1038/s41598-019-43125-6.

Li, Y., et al., 2019. Drug-target interaction prediction based on drug fingerprint information and protein sequence. Molecules 24 (16), https://doi.org/10.3390/molecules24162999.

Liang, Y.-L., et al., 2017. Phase-plate cryo-EM structure of a class B GPCR-G-protein complex. Nature 546 (7656), 118–123. https://doi.org/10.1038/nature22327.

Liang, Y.-L., et al., 2020. Toward a structural understanding of class B GPCR peptide binding and activation. Molecular Cell 77 (3), 656–668.e5. https://doi.org/10.1016/j.molcel.2020.01.012.

Liao, Z., Ju, Y., Zou, Q., 2016. Prediction of G protein-coupled receptors with SVM-prot features and random forest. Scientifica 2016, 8309253https://doi.org/10.1155/2016/8309253.

Lindner, D., et al., 2009. Functional role of the extracellular N-terminal domain of neuropeptide Y subfamily receptors in membrane integration and agonist-stimulated internalization. Cellular Signalling 21 (1), 61–68. https://doi.org/10.1016/j.cellsig.2008.09.007.

Liou, C.-Y., et al., 2014. Autoencoder for words. Neurocomputing 139, 84–96. https://doi.org/10.1016/j.neucom.2013.09.055.

Litsa, E.E., Das, P., Kavraki, L.E., 2020. Prediction of drug metabolites using neural machine translation. Chemical Science https://doi.org/10.1039/D0SC02639E.

Lo, A., et al., 2009. Predicting helix–helix interactions from residue contacts in membrane proteins. Bioinformatics 25 (8), 996–1003. https://doi.org/10.1093/bioinformatics/btp114.

Lopez-Gimenez, J.F., et al., 2007. The α1b-adrenoceptor exists as a higher-order oligomer: Effective oligomerization is required for receptor maturation, surface delivery, and function. Molecular Pharmacology. American Society for Pharmacology and Experimental Therapeutics 71 (4), 1015–1029. https://doi.org/10.1124/mol.106.033035.

Lou, R., et al., 2020. Hybrid spectral library combining DIA-MS data and a targeted virtual library substantially deepens the proteome coverage. Science 23 (3), 100903. https://doi.org/10.1016/j.isci.2020.100903.

Lu, M., Wu, B., 2016. Structural studies of G protein-coupled receptors. IUBMB Life 68 (11), 894–903. https://doi.org/10.1002/iub.1578.

Lüthy, R., Bowie, J.U., Eisenberg, D., 1992. Assessment of protein models with three-dimensional profiles. Nature 356 (6364), 83–85. https://doi.org/10.1038/356083a0.

Lysenko, A., et al., 2018. An integrative machine learning approach for prediction of toxicity-related drug safety. Life Science Alliance 1 (6), e201800098https://doi.org/10.26508/lsa.201800098.

Ma, C., Wang, L., Xie, X.Q., 2011. Ligand classifier of adaptively boosting ensemble decision stumps (LiCABEDS) and its application on modeling ligand functionality for 5HT-subtype GPCR families. Journal of Chemical Information and Modeling 51 (3), 521–531. https://doi.org/10.1021/ci100399j.

Ma, S., et al., 2020. Molecular basis for hormone recognition and activation of corticotropin-releasing factor receptors. Molecular Cell 77 (3), 669–680.e4. https://doi.org/10.1016/j.molcel.2020.01.013.

Manavalan, B., et al., 2019. AtbPpred: A robust sequence-based prediction of anti-tubercular peptides using extremely randomized trees. Computational and Structural Biotechnology Journal 17, 972–981. https://doi.org/10.1016/j.csbj.2019.06.024.

Manglik, A., Kruse, A.C., 2017. Structural basis for G protein-coupled receptor activation. Biochemistry 56 (42), 5628–5634. https://doi.org/10.1021/acs.biochem.7b00747.

Manning, G., et al., 2002. The protein kinase complement of the human genome. Science 298 (5600), 1912–1934. https://doi.org/10.1126/science.1075762.

Mansouri, K., Judson, R.S., 2016. In silico study of in vitro GPCR assays by QSAR modeling. Methods in Molecular Biology 1425, 361–381. https://doi.org/10.1007/978-1-4939-3609-0_16.

Marks, D.S., et al., 2011. Protein 3D structure computed from evolutionary sequence variation. PLoS One 6 (12), e28766. https://doi.org/10.1371/journal.pone.0028766.

Martinez, F.J., Couser, J.I., Celli, B.R., 1991. Respiratory response to arm elevation in patients with chronic airflow obstruction. The American Review of Respiratory Disease 143 (3), 476–480. https://doi.org/10.1164/ajrccm/143.3.476.

May, L.T., et al., 2007. Allosteric modulation of G protein-coupled receptors. Annual Review of Pharmacology and Toxicology 47, 1–51. https://doi.org/10.1146/annurev.pharmtox.47.120505.105159.

Meng, F.R., et al., 2017. Prediction of drug–target interaction networks from the integration of protein sequences and drug chemical structures. Molecules 22 (7)https://doi.org/10.3390/molecules22071119.

Meslamani, J., et al., 2013. Computational profiling of bioactive compounds using a target-dependent composite workflow. Journal of Chemical Information and Modeling 53 (9), 2322–2333. https://doi.org/10.1021/ci400303n.

Minsky, M., Hillis, D., Rudisch, G., 1980. Artificial intelligence. The New England Journal of Medicine 1482.

Moreira, I.S., 2014. Structural features of the G-protein/GPCR interactions. Biochimica et Biophysica Acta 1840 (1), 16–33. https://doi.org/10.1016/j.bbagen.2013.08.027.

Mousavi, S.S., Schukat, M., Howley, E., 2018. Deep reinforcement learning. In: Bi, Y., Kapoor, S., Bhatia, R. (Eds.). An Overview BT—Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016. Springer International Publishing, Cham, pp. 426–440.

Muhammed, M.T., Aki-Yalcin, E., 2019. Homology modeling in drug discovery: Overview, current applications, and future perspectives. Chemical Biology & Drug Design 93 (1), 12–20. https://doi.org/10.1111/cbdd.13388.

Muk, S., et al., 2019. Machine learning for prioritization of thermostabilizing mutations for G-protein coupled receptors. Biophysical Journal 117 (11), 2228–2239. https://doi.org/10.1016/j.bpj.2019.10.023.

Munk, C., et al., 2016. GPCRdb: The G protein-coupled receptor database—An introduction. British Journal of Pharmacology https://doi.org/10.1111/bph.13509.

Munk, C., et al., 2019. An online resource for GPCR structure determination and analysis. Nature Methods 16 (2), 151–162. https://doi.org/10.1038/s41592-018-0302-x.

Naveed, M., Khan, A.U., 2012. GPCR-MPredictor: Multi-level prediction of G protein-coupled receptors using genetic ensemble. Amino Acids 42 (5), 1809–1823. https://doi.org/10.1007/s00726-011-0902-6.

NCBI Resource Coordinators, 2018. Database resources of the National Center for Biotechnology Information. Nucleic Acids Research 46 (D1), D8–D13. https://doi.org/10.1093/nar/gkx1095.

Nemoto, W., et al., 2016. GGIP: Structure and sequence-based GPCR-GPCR interaction pair predictor. Proteins 84 (9), 1224–1233. https://doi.org/10.1002/prot.25071.

Neumann, J.-M., et al., 2008. Class-B GPCR activation: Is ligand helix-capping the key?. Trends in Biochemical Sciences 33 (7), 314–319. https://doi.org/10.1016/j.tibs.2008.05.001.

Nie, G., et al., 2015. A novel fractal approach for predicting G-protein-coupled receptors and their subfamilies with support vector machines. Bio-medical Materials and Engineering 26 (supplement 1), S1829–S1836. https://doi.org/10.3233/BME-151485.

Niesler, T.R., Woodland, P.C., 1999. Variable-length categoryn-gram language models. Computer Speech & Language 13 (1), 99–124. https://doi.org/10.1006/csla.1998.0115.

Noble, W.S., 2006. What is a support vector machine?. Nature Biotechnology 24 (12), 1565–1567. https://doi.org/10.1038/nbt1206-1565.

Ofran, Y., Rost, B., 2007. ISIS: Interaction sites identified from sequence. Bioinformatics 23 (2), e13–e16. https://doi.org/10.1093/bioinformatics/btl303.

Ojansivu, V., Heikkilä, J., 2008. Blur insensitive texture classification using local phase quantization. In: Elmoataz, A., et al. (Eds.), Image and Signal Processing. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 236–243.

Okuno, Y., 2008. In silico drug discovery based on the integration of bioinformatics and chemoinformatics. Yakugaku Zasshi 128 (11), 1645–1651. https://doi.org/10.1248/yakushi.128.1645.

Okuno, Y., et al., 2006. GLIDA: GPCR-ligand database for chemical genomic drug discovery. Nucleic Acids Research 34 (Database issue), D673–D677. https://doi.org/10.1093/nar/gkj028.

Pándy-Szekeres, G., et al., 2018. GPCRdb in 2018: Adding GPCR structure models and ligands. Nucleic Acids Research 46 (D1), D440–D446. https://doi.org/10.1093/nar/gkx1109.

Patlewicz, G., Fitzpatrick, J.M., 2016. Current and future perspectives on the development, evaluation, and application of in silico approaches for predicting toxicity. Chemical Research in Toxicology 29 (4), 438–451. https://doi.org/10.1021/acs.chemrestox.5b00388.

Pawlowski, M., et al., 2008. MetaMQAP: A meta-server for the quality assessment of protein models. BMC Bioinformatics 9 (1), 403. https://doi.org/10.1186/1471-2105-9-403.

Peska, L., Buza, K., Koller, J., 2017. Drug-target interaction prediction: A Bayesian ranking approach. Computer Methods and Programs in Biomedicine 152, 15–21. https://doi.org/10.1016/j.cmpb.2017.09.003.

Pinzi, L., Rastelli, G., 2019. Molecular docking: Shifting paradigms in drug discovery. International Journal of Molecular Sciences 20 (18)https://doi.org/10.3390/ijms20184331.

Plante, A., et al., 2019. A machine learning approach for the discovery of ligand-specific functional mechanisms of GPCRs. Molecules 24 (11)https://doi.org/10.3390/molecules24112097.

Popov, P., et al., 2018. Computational design of thermostabilizing point mutations for G protein-coupled receptors. eLife 7, https://doi.org/10.7554/eLife.34729.

Popov, P., Kozlovskii, I., Katritch, V., 2019. Computational design for thermostabilization of GPCRs. Current Opinion in Structural Biology 55, 25–33. https://doi.org/10.1016/j.sbi.2019.02.010.

Preininger, A.M., Meiler, J., Hamm, H.E., 2013. Conformational flexibility and structural dynamics in GPCR-mediated G protein activation: A perspective. Journal of Molecular Biology 425 (13), 2288–2298. https://doi.org/10.1016/j.jmb.2013.04.011.

Preto, A.J., Moreira, I.S., 2020. Spotone: Hot spots on protein complexes with extremely randomized trees via sequence-only features. International Journal of Molecular Sciences 21 (19)https://doi.org/10.3390/ijms21197281.

Preto, A.J., et al., 2018. Computational tools for the structural characterization of proteins and their complexes from sequence-evolutionary data. In: Encyclopedia of Analytical Chemistry. American Cancer Society, pp. 1–19. https://doi.org/10.1002/9780470027318.a9615.

Preto, A.J., et al., 2020. Understanding the binding specificity of G-protein coupled receptors toward G-proteins and arrestins: Application to the dopamine receptor family. Journal of Chemical Information and Modeling 60 (8), 3969–3984. https://doi.org/10.1021/acs.jcim.0c00371.

Prioleau, C., et al., 2002. Conserved helix 7 tyrosine acts as a multistate conformational switch in the 5HT2C receptor. Identification of a novel "locked-on" phenotype and double revertant mutations. The Journal of Biological Chemistry 277 (39), 36577–36584. https://doi.org/10.1074/jbc.M206223200.

Rajoub, B., 2020. Ch. 3: Supervised and unsupervised learning. In: Zgallai, W., et al. (Eds.), Developments in Biomedical Engineering and Bioelectronics. Academic Press, pp. 51–89. https://doi.org/10.1016/B978-0-12-818946-7.00003-2.

Raschka, S., et al., 2018. Automated inference of chemical discriminants of biological activity. Methods in Molecular Biology 1762, 307–338. https://doi.org/10.1007/978-1-4939-7756-7_16.

Rataj, K., et al., 2018. Fingerprint-based machine learning approach to identify potent and selective 5-HT2BR ligands. Molecules 23 (5), 1–15. https://doi.org/10.3390/molecules23051137.

Ray, A., Lindahl, E., Wallner, B., 2010. Model quality assessment for membrane proteins. Bioinformatics (Oxford, England) 26 (24), 3067–3074. https://doi.org/10.1093/bioinformatics/btq581.

Redkar, S., et al., 2020. A machine learning approach for drug-target interaction prediction using wrapper feature selection and class balancing. Molecular Informatics 39 (5)https://doi.org/10.1002/minf.201900062.

Rehman, Z.-U., et al., 2013. Predicting G-protein-coupled receptors families using different physiochemical properties and pseudo amino acid composition. Methods in Enzymology 522, 61–79. https://doi.org/10.1016/B978-0-12-407865-9.00004-2.

Reutlinger, M., et al., 2014. Multi-objective molecular de novo design by adaptive fragment prioritization. Angewandte Chemie 53 (16), 4244–4248. https://doi.org/10.1002/anie.201310864.

Ribeiro, J.M.L., Filizola, M., 2019. Insights from molecular dynamics simulations of a number of G-protein coupled receptor targets for the treatment of pain and opioid use disorders. Frontiers in Molecular Neuroscience 12, 1–13. https://doi.org/10.3389/fnmol.2019.00207.

Ridder, L., Wagener, M., 2008. SyGMa: Combining expert knowledge and empirical scoring in the prediction of metabolites. ChemMedChem 3 (5), 821–832. https://doi.org/10.1002/cmdc.200700312.

Rohl, C.A., et al., 2004. Protein structure prediction using Rosetta. Methods in Enzymology 383, 66–93. https://doi.org/10.1016/S0076-6879(04)83004-0.

Rosenbaum, D.M., Rasmussen, S.G.F., Kobilka, B.K., 2009. The structure and function of G-protein-coupled receptors. Nature 459 (7245), 356–363. https://doi.org/10.1038/nature08144.

Rozenfeld, R., et al., 2011. AT1R-CB$_1$R heteromerization reveals a new mechanism for the pathogenic properties of angiotensin II. The EMBO Journal 30 (12), 2350–2363. https://doi.org/10.1038/emboj.2011.139.

Ru, X., et al., 2020. Exploration of the correlation between GPCRs and drugs based on a learning to rank algorithm. Computers in Biology and Medicine 119, 103660https://doi.org/10.1016/j.compbiomed.2020.103660.

Sachdev, K., Gupta, M.K., 2019. A comprehensive review of feature based methods for drug target interaction prediction. Journal of Biomedical Informatics 93, 103159. https://doi.org/10.1016/j.jbi.2019.103159.

Sacks, D., et al., 2018. Multisociety consensus quality improvement revised consensus statement for endovascular therapy of acute ischemic stroke. International Journal of Stroke 13 (6), 612–632. https://doi.org/10.1177/1747493018778713.

Sampat, B.N., Lichtenberg, F.R., 2011. What are the respective roles of the public and private sectors in pharmaceutical innovation?. Health Affairs 30 (2), 332–339. https://doi.org/10.1377/hlthaff.2009.0917.

Samudrala, R., Moult, J., 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. Journal of Molecular Biology 275 (5), 895–916. https://doi.org/10.1006/jmbi.1997.1479.

Sanders, M.P.A., et al., 2011. ss-TEA: Entropy based identification of receptor specific ligand binding residues from a multiple sequence alignment of class A GPCRs. BMC Bioinformatics 12, 332. https://doi.org/10.1186/1471-2105-12-332.

Sangmin, S., et al., 2018. Prediction of GPCR-ligand binding using machine learning algorithms. Computational and Mathematical Methods in Medicine 2018, 6565241https://doi.org/10.1155/2018/6565241.

Scarselli, F., et al., 2009. The graph neural network model. IEEE Transactions on Neural Networks 20 (1), 61–80. https://doi.org/10.1109/TNN.2008.2005605.

Schiöth, H.B., Fredriksson, R., 2005. The GRAFS classification system of G-protein coupled receptors in comparative perspective. General and Comparative Endocrinology 142 (1–2), 94–101. https://doi.org/10.1016/j.ygcen.2004.12.018.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Schmidt, U., et al., 2008. SuperToxic: a comprehensive database of toxic compounds. Nucleic Acids Research 37 (supplement 1), D295–D299. https://doi.org/10.1093/nar/gkn850.

Schütz, M., et al., 2016. Directed evolution of G protein-coupled receptors in yeast for higher functional production in eukaryotic expression hosts. Scientific Reports 6 (1), 21508. https://doi.org/10.1038/srep21508.

Seokjun, S., et al., 2018. DeepFam: Deep learning based alignment-free method for protein family modeling and prediction. Bioinformatics 34 (13), i254–i262. https://doi.org/10.1093/bioinformatics/bty275.

Shang, J., et al., 2017. Comparative analyses of structural features and scaffold diversity for purchasable compound libraries. Journal of Cheminformatics 9 (1), 25. https://doi.org/10.1186/s13321-017-0212-4.

Sharma, A., et al., 2013. A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. Journal of Theoretical Biology 320, 41–46. https://doi.org/10.1016/j.jtbi.2012.12.008.

Shen, J., et al., 2010. Estimation of ADME properties with substructure pattern recognition. Journal of Chemical Information and Modeling 50 (6), 1034–1041. https://doi.org/10.1021/ci100104j.

Shen, C., et al., 2017. An ameliorated prediction of drug–target interactions based on multi-scale discretewavelet transform and network features. International Journal of Molecular Sciences 18 (8)https://doi.org/10.3390/ijms18081781.

Shi, H., et al., 2019. Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure. Genomics 111 (6), 1839–1852. https://doi.org/10.1016/j.ygeno.2018.12.007.

Shiraishi, A., et al., 2013. Chemical genomics approach for GPCR-ligand interaction prediction and extraction of ligand binding determinants. Journal of Chemical Information and Modeling 53 (6), 1253–1262. https://doi.org/10.1021/ci300515z.

Shiraishi, A., et al., 2019. Repertoires of G protein-coupled receptors for Ciona-specific neuropeptides. Proceedings of the National Academy of Sciences of the United States of America 116 (16), 7847–7856. https://doi.org/10.1073/pnas.1816640116.

Shkurin, A., Vellido, A., 2017. Using random forests for assistance in the curation of G-protein coupled receptor databases. Biomedical Engineering 16 (supplement 1), 75. https://doi.org/10.1186/s12938-017-0357-4.

Shui, Z., Karypis, G., 2020. Heterogeneous Molecular Graph Neural Networks for Predicting Molecule Properties.

Sievers, F., Higgins, D.G., 2014. Clustal Omega, accurate alignment of very large numbers of sequences. Methods in Molecular Biology 1079, 105–116. https://doi.org/10.1007/978-1-62703-646-7_6.

Sippl, M.J., 1990. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. Journal of Molecular Biology 213 (4), 859–883. https://doi.org/10.1016/s0022-2836(05)80269-4.

Sliwoski, G., et al., 2014. Computational methods in drug discovery. Pharmacological Reviews 66 (1), 334–395. https://doi.org/10.1124/pr.112.007336.

Szepesvári, C., 2010. Algorithms for reinforcement learning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning. 4, Morgan & Claypool Publishers, pp. 1–103. https://doi.org/10.2200/S00268ED1V01Y201005AIM009.

Talevi, A., 2018. Computer-aided drug design: An overview. In: Gore, M., Jagtap, U.B. (Eds.), Computational Drug Discovery and Design. Springer New York, New York, NY, pp. 1–19. https://doi.org/10.1007/978-1-4939-7756-7_1.

Tate, C.G., Schertler, G.F.X., 2009. Engineering G protein-coupled receptors to facilitate their structure determination. Current Opinion in Structural Biology 19 (4), 386–395. https://doi.org/10.1016/j.sbi.2009.07.004.

Tavanaei, A., et al., 2019. Deep learning in spiking neural networks. Neural Networks 111, 47–63. https://doi.org/10.1016/j.neunet.2018.12.002.

Tenenbaum, J.B., de Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500), 2319–2323. https://doi.org/10.1126/science.290.5500.2319.

Terrillon, S., Bouvier, M., 2004. Roles of G-protein-coupled receptor dimerization. EMBO Reports 5 (1), 30–34. https://doi.org/10.1038/sj.embor.7400052.

Tsou, L.K., et al., 2020. Comparative study between deep learning and QSAR classifications for TNBC inhibitors and novel GPCR agonist discovery. Scientific Reports 10 (1), 16771. https://doi.org/10.1038/s41598-020-73681-1.

Uddin, S., et al., 2019. Comparing different supervised machine learning algorithms for disease prediction. BMC Medical Informatics and Decision Making 19 (1), 281. https://doi.org/10.1186/s12911-019-1004-8.

Ulrich, E.L., et al., 2008. BioMagResBank. Nucleic Acids Research 36 (Database Issue), D402–D408. https://doi.org/10.1093/nar/gkm957.

Unal, H., Karnik, S.S., 2012. Domain coupling in GPCRs: The engine for induced conformational changes. Trends in Pharmacological Sciences 33 (2), 79–88. https://doi.org/10.1016/j.tips.2011.09.007.

Usman, S., et al., 2020. The current status of anti GPCR drugs against different cancers. Journal of Pharmaceutical Analysis https://doi.org/10.1016/j.jpha.2020.01.001.

Van Oss, C.J., Good, R.J., Chaudhury, M.K., 1986. The role of van der Waals forces and hydrogen bonds in “hydrophobic interactions” between biopolymers and low energy surfaces. Journal of Colloid and Interface Science 111 (2), 378–390. https://doi.org/10.1016/0021-9797(86)90041-X.

Vass, M., et al., 2016. Molecular interaction fingerprint approaches for GPCR drug discovery. Current Opinion in Pharmacology 30, 59–68. https://doi.org/10.1016/j.coph.2016.07.007.

Velgy, N., Hedger, G., Biggin, P.C., 2018. GPCRs: What can we learn from molecular dynamics simulations?. Methods in Molecular Biology 1705, 133–158. https://doi.org/10.1007/978-1-4939-7465-8_6.

Venkatakrishnan, A.J., et al., 2013. Molecular signatures of G-protein-coupled receptors. Nature 494 (7436), 185–194. https://doi.org/10.1038/nature11896.

Visiers, I., Ballesteros, J.A., Weinstein, H., 2002. Three-dimensional representations of G protein-coupled receptor structures and mechanisms. In: G Protein Pathways Part A: Ribonucleases. Academic Press, pp. 329–371. https://doi.org/10.1016/S0076-6879(02)43145-X.

Wachsmuth, H., 2015. Text Analysis Pipelines. Springer International Publishinghttps://doi.org/10.1007/978-3-319-25741-9.

Wallner, B., Elofsson, A., 2003. Can correct protein models be identified?. Protein Science 12 (5), 1073–1086. https://doi.org/10.1110/ps.0236803.

Wallner, B., Elofsson, A., 2006. Identification of correct regions in protein models using structural, alignment, and consensus information. Protein Science 15 (4), 900–913. https://doi.org/10.1110/ps.051799606.

Wang, L., et al., 2018. A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network. Journal of Computational Biology 25 (3), 361–373. https://doi.org/10.1089/cmb.2017.0135.

Wang, P., et al., 2020. Identifying GPCR-drug interaction based on wordbook learning from sequences. BMC Bioinformatics 21 (1), 150. https://doi.org/10.1186/s12859-020-3488-8.

Weill, N., Rognan, D., 2009. Development and validation of a novel protein-ligand fingerprint to mine chemogenomic space: Application to G protein-coupled receptors and their ligands. Journal of Chemical Information and Modeling 49 (4), 1049–1062. https://doi.org/10.1021/ci800447g.

Weiner, S.J., et al., 1984. A new force field for molecular mechanical simulation of nucleic acids and proteins. Journal of the American Chemical Society 106 (3), 765–784. https://doi.org/10.1021/ja00315a051.

Williams, A.J., et al., 2017. The CompTox Chemistry Dashboard: A community data resource for environmental chemistry. Journal of Cheminformatics 9 (1), 61. https://doi.org/10.1186/s13321-017-0247-6.

Wootten, D., et al., 2018. Mechanisms of signalling and biased agonism in G protein-coupled receptors. Nature Reviews. Molecular Cell Biology 19 (10), 638–653. https://doi.org/10.1038/s41580-018-0049-3.

Wright, S.C., et al., 2019. A conserved molecular switch in Class F receptors regulates receptor activation and pathway selection. Nature Communications 10 (1), 667. https://doi.org/10.1038/s41467-019-08630-2.

Wu, H., et al., 2017. Deep conditional random field approach to transmembrane topology prediction and application to GPCR three-dimensional structure modeling. IEEE/ACM Transactions on Computational Biology and Bioinformatics 14 (5), 1106–1114. https://doi.org/10.1109/TCBB.2016.2602872.

Wu, J., et al., 2018. WDL-RF: Predicting bioactivities of ligand molecules acting with G protein-coupled receptors by combining weighted deep learning and random forest. Bioinformatics 34 (13), 2271–2282. https://doi.org/10.1093/bioinformatics/bty070.

Wu, J., et al., 2019. Function prediction for G protein-coupled receptors through text mining and induction matrix completion. ACS 4 (2), 3045–3054. https://doi.org/10.1021/acsomega.8b02454.

Wu, F., et al., 2020. Computational approaches in preclinical studies on drug discovery and development. Frontiers in Chemistry 8, 726. https://doi.org/10.3389/fchem.2020.00726.

Xiang, J., et al., 2016. Successful strategies to determine high-resolution structures of GPCRs. Trends in Pharmacological Sciences 37 (12), 1055–1069. https://doi.org/10.1016/j.tips.2016.09.009.

Xiao, X., et al., 2013. iGPCR-drug: A web server for predicting interaction between GPCRs and drugs in cellular networking. PLoS One 8 (8)https://doi.org/10.1371/journal.pone.0072234.

Xiao, X., et al., 2015. iDrug-target: Predicting the interactions between drug compounds and target proteins in cellular networking via benchmark dataset optimization approach. Journal of Biomolecular Structure & Dynamics 33 (10), 2221–2233. https://doi.org/10.1080/07391102.2014.998710.

Xie, H.-L., Fu, L., Nie, X.-D., 2013. Using ensemble SVM to identify human GPCRs N-linked glycosylation sites based on the general form of Chou’s PseAAC. Protein Engineering, Design & Selection 26 (11), 735–742. https://doi.org/10.1093/protein/gzt042.

Yadav, B.S., Tripathi, V., 2018. Recent advances in the system biology-based target identification and drug discovery. Current Topics in Medicinal Chemistry 18 (20), 1737–1744. https://doi.org/10.2174/1568026618666181025112344.

Yang, J., et al., 2013. High-accuracy prediction of transmembrane inter-helix contacts and application to GPCR 3D structure modeling. Bioinformatics (Oxford, England) 29 (20), 2579–2587. https://doi.org/10.1093/bioinformatics/btt440.

Yang, L.-K., Hou, Z.-S., Tao, Y.-X., 2020. Biased signaling in naturally occurring mutations of G protein-coupled receptors associated with diverse human diseases. Biochimica et Biophysica Acta, Molecular Basis of Disease 1867 (1), 165973. https://doi.org/10.1016/j.bbadis.2020.165973.

Zalewska, M., Siara, M., Sajewicz, W., 2014. G protein-coupled receptors: Abnormalities in signal transmission, disease states and pharmacotherapy. Acta Poloniae Pharmaceutica 71 (2), 229–243.

Zhan, X., et al., 2020. Ensemble learning prediction of drug-target interactions using GIST descriptor extracted from PSSM-based evolutionary information. BioMed Research International 2020, 4516250https://doi.org/10.1155/2020/4516250.

Zhang, C., Kim, S.H., 2000. Environment-dependent residue contact energies for proteins. Proceedings of the National Academy of Sciences of the United States of America 97 (6), 2550–2555. https://doi.org/10.1073/pnas.040573597.

Zhang, C., Ma, Y., 2012. In: Zhang, C., Ma, Y. (Eds.), Ensemble Machine Learning. Springer, Boston, https://doi.org/10.1007/978-1-4419-9326-7.

Zhang, H., et al., 2015. Structural basis for ligand recognition and functional selectivity at angiotensin receptor. The Journal of Biological Chemistry 290 (49), 29127–29139. https://doi.org/10.1074/jbc.M115.689000.

Zhang, J., et al., 2015. GPCR-I-TASSER: A hybrid approach to G protein-coupled receptor structure modeling and the application to the human genome. Structure 23 (8), 1538–1549. https://doi.org/10.1016/j.str.2015.06.007.

Zhang, X., Dong, S., Xu, F., 2018. Structural and druggability landscape of frizzled G protein-coupled receptors. Trends in Biochemical Sciences 43 (12), 1033–1046. https://doi.org/10.1016/j.tibs.2018.09.002.

Zhang, Y., et al., 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. Scientific Data 6 (1), 52. https://doi.org/10.1038/s41597-019-0055-0.

Zhao, R., Mao, K., 2018. Fuzzy bag-of-words model for document representation. IEEE Transactions on Fuzzy Systems 26 (2), 794–804. https://doi.org/10.1109/TFUZZ.2017.2690222.

Zhou J et al. (2019) Graph Neural Networks: A Review of Methods and Applications.

Zhu, X.L., et al., 2011. Classification of 5-HT1A receptor agonists and antagonists using GA-SVM method. Acta Pharmacologica Sinica 32 (11), 1424–1430. https://doi.org/10.1038/aps.2011.112.

Zia-Ur-Rehman, Khan, A., 2011. Prediction of GPCRs with pseudo amino acid composition: employing composite features and grey incidence degree based classification. Protein and Peptide Letters 18 (9), 872–878. https://doi.org/10.2174/092986611796011491.

Zia-Ur-Rehman, Khan, A., 2012. Identifying GPCRs and their types with Chou's pseudo amino acid composition: An approach from multi-scale energy representation and position specific scoring matrix. Protein and Peptide Letters 19 (8), 890–903. https://doi.org/10.2174/092986612801619589.

Zou, Y., Ewalt, J., Ng, H.-L., 2019. Recent insights from molecular dynamics simulations for G protein-coupled receptor drug discovery. International Journal of Molecular Sciences 20 (17)https://doi.org/10.3390/ijms20174237.

Glossary

Artificial intelligence An interdisciplinary wide-ranging branch of computer science concerned with building smart machines capable of performing assignments that typically require human task-accomplishing skills.

Computer-aided drug design Comprises a wide range of theoretical and computational methods to reduce the time and resource bottlenecks involved in drug design and discovery.

Deep learning A subset of Machine Learning (ML) that makes use of Artificial Neural Network (ANN). These networks mimic their biological counterpart (brain neurological networks) and have been explored through diverse architectures and problems.

Drug-target interaction The physical and chemical interaction that occurs between a small ligand (drug) and a protein target. Ideally, its modulation can lead to pharmaceutical solutions to biomedical problems.

G protein-coupled receptors A superfamily of MPs that mediate a vast array of biological processes constituting the target of around 35% of all pharmaceutical drugs in the market.

Machine learning The usage of statistical and logical tools, in conjunction with computers, to deploy AI methods and optimize task-solving processes without explicitly programming the computer to do so.

Orphan receptors GPCRs without known endogenous ligand or physiological function. The finding of endogenous ligands and physiological function is called deorphanization, since it allows the inclusion of said GPCR on the appropriate family.

3.2.2. DrugTax: package for drug taxonomy identification and explainable feature extraction

Preto *et al. Journal of Cheminformatics (2022) 14:73*
https://doi.org/10.1186/s13321-022-00649-w

Journal of Cheminformatics

SOFTWARE Open Access

DrugTax: package for drug taxonomy identification and explainable feature extraction

A. J. Preto[1,2], Paulo C. Correia[3] and Irina S. Moreira[1,3,4*]

Abstract

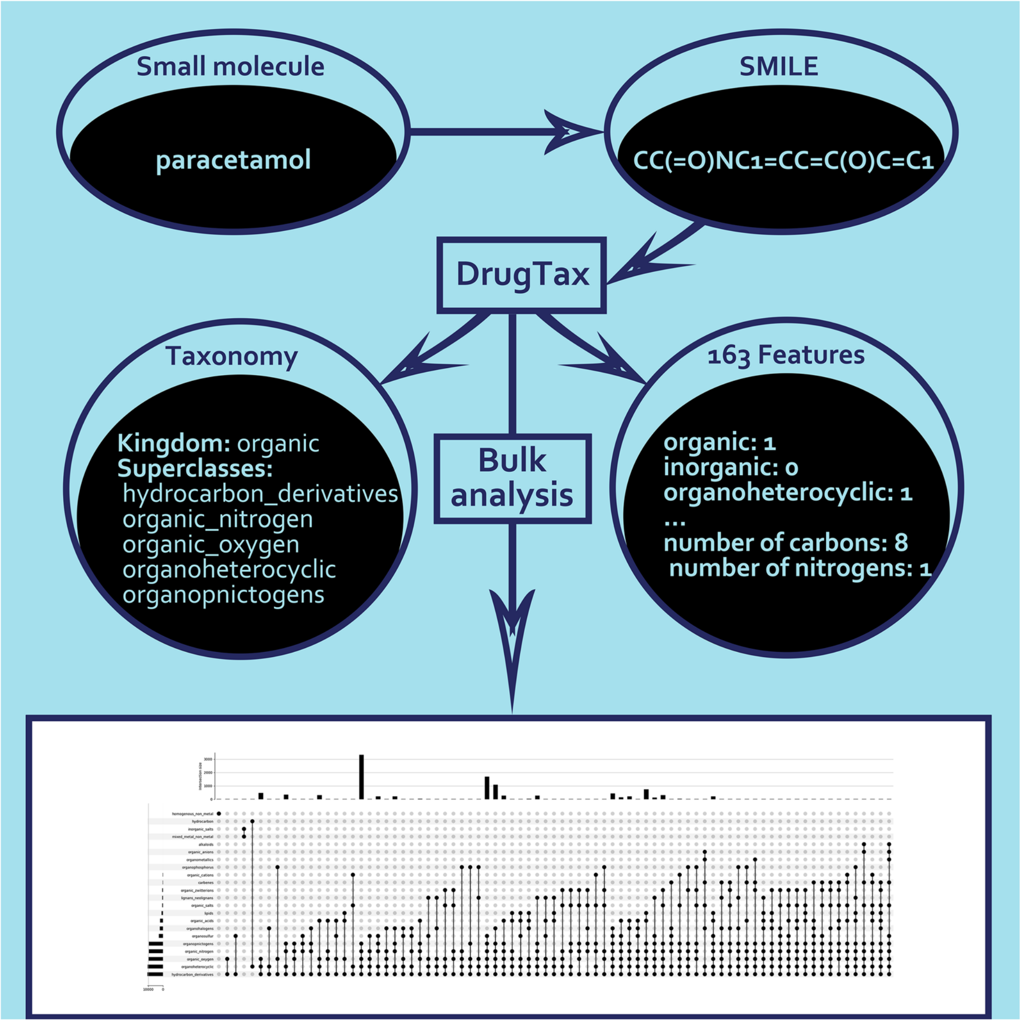
DrugTax is an easy-to-use Python package for small molecule detailed characterization. It extends a previously explored chemical taxonomy making it ready-to-use in any Artificial Intelligence approach. DrugTax leverages small molecule representations as input in one of their most accessible and simple forms (SMILES) and allows the simultaneously extraction of taxonomy information and key features for big data algorithm deployment. In addition, it delivers a set of tools for bulk analysis and visualization that can also be used for chemical space representation and molecule similarity assessment. DrugTax is a valuable tool for chemoinformatic processing and can be easily integrated in drug discovery pipelines. DrugTax can be effortlessly installed via PyPI (https://pypi.org/project/DrugTax/) or GitHub (https://github.com/MoreiraLAB/DrugTax).

Keywords: DrugTax, Small molecules, Machine learning, Explainability, Python

*Correspondence: irina.moreira@cnc.uc.pt

[3] Department of Life Sciences, University of Coimbra, Calçada Martim de Freitas, 3000-456 Coimbra, Portugal
Full list of author information is available at the end of the article

Graphical Abstract



Introduction

PubChem [1] registers over 111 million compounds and 278 million substances (August 2022). According to Drugbank [2] there are 2725 approved drugs, among 11,937 possible drugs. ChEMBL [3] reports over 2.2 million compounds and 14,000 drugs. The abundance of drugs or drug-like compounds is evidently overwhelming, which is often problematic, when considering automatized approaches.

The surge of Artificial Intelligence (AI) and its subfield Machine Learning (ML) to tackle problems involving drugs or, overall, small ligands has been significant in the last few years [4]. For this purpose, it is advantageous to be able to provide a deeper understanding of the drugs' characteristics while also being able to numerically describe them [5]. Feature extraction is a focus when considering ML-based approaches, as it is a crucial and necessary step for any algorithms to be able to distinguish between the different patterns within the data. Under the scope of drug discovery, several packages have been developed to this end. Open Babel [6] is a broad example, providing a set of chemical tools to describe and manipulate drugs and other small molecules. More recently, packages such as Mordred [7] or ChemmineR [8] have also been developed. Alternatively, a different type of approaches can also be used for ML processing, such as the ones based on graph [9, 10] and voxel-based [11] drug representations. The chemical characterization of small molecules is a cornerstone for further understanding and essential for bulk data approaches, and as such we explored the usage of this type of knowledge for data grouping and feature extraction, some of the characterizations stemming from the root biochemical definitions [12].

Our new developed Python package, DrugTax, follows the definitions made available by ChemOnt and Classyfire [13]. The Classyfire protocol [13] is very useful for small molecule taxonomy classification as it performs a levelled classification in 11 different levels (Kingdom, SuperClass, Class, SubClass, etc.), yielding over 4800 different categories. We also explored the chemical ontology (ChemOnt), developed by the same authors, which allows the classification of the small molecules solely by rule-based steps. However, these protocols still presented some shortcomings: (i) the API, although properly documented, is faulty in bulk submissions; (ii) although both the browser and the API are available, the ChemOnt code for small molecule taxonomic classification is not accessible, limiting the users to using the authors' API; and finally, (iii) while the same-level categories are not necessarily mutually exclusive, Classyfire [13] yields a single classification for each compound. This means that molecules belonging to more than one superclass, are overlooked, leading to major oversights of information when considering multiple molecules' comparison. These shortcomings are particularly relevant if the research's main aim is to group small ligands according to their characteristics.

DrugTax solves that problem by allowing the user to install and inspect the code that generates the small molecules classes in an easy-to-use package. DrugTax provides the prior classification between the two possible kingdoms, organic and inorganic, and, respectively, their 26 and 5 superclasses. These superclasses are returned in the form of a list, thus allowing overlapping superclasses. Subsequently, DrugTax displays UpSet plots [14], which are ideal for identifying and inspecting large volumes of intersecting sets to provide the user an approach to further tailor the groupings to their needs. Finally, DrugTax provides an option to use features derived from the taxonomic analysis up until superclasses. This innovation can be promptly used for ML purposes or simply small molecule data visualization.

Methods and implementation

DrugTax is centered around a Python object class that takes as input a Simplified Molecular Input Line Entry System (SMILES) [15] and computes several necessary steps for the upcoming kingdom and superclass assignment. If a SMILES representation is not provided, DrugTax will default to download its isomeric form from a provided name. All Code Snippets (C.S.) can be found in Additional file 1. Figure 1 illustrates molecules belonging to the 31 superclasses that will be listed next. Organic molecules are highlighted in green, while inorganic molecules are shown in red.

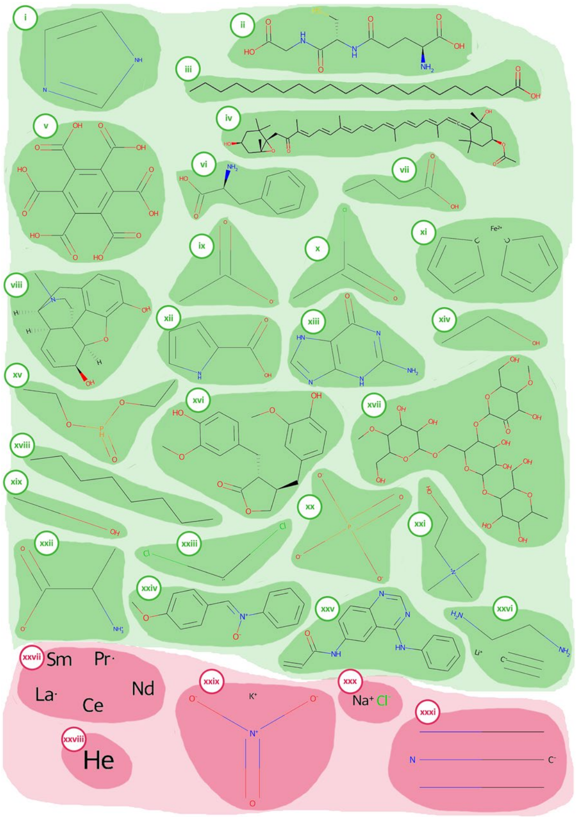


Fig. 1 Graphical representation of each of the 31 superclasses. Organic molecules are highlighted in green, while inorganic molecules are shown in red. The molecules depicted are: organoheterocyclic-imidazole (**i**); organosulphur-glutathione (**ii**); lipid molecule-behenic acid (fatty acid) (**iii**); allene-fucoxanthin (**iv**); benzenoid-benzene hexacarboxylic acid (**v**); phenylpropanoid-phenylalanine (**vi**); organic acid-butyric acid (**vii**); alkaloid-morphine (**viii**); organic salt-acetate (**ix**); organohalogen-acetyl chloride (**x**); organometallic-ferrocene (**xi**); organic nitrogen-pyrrole-2-carboxylate (**xii**); nucleotide-guanine (**xiii**); organic oxygen-ethanol (**xiv**); organophosphorus-diethyl phosphonate (**xv**); lignans and neolignans-matairesinol (**xvi**); organic polymer-starch (**xvii**); hydrocarbon-octane (**xviii**); hydrocarbon derivative-ethanol (**xix**); organic anion-phosphate (**xx**); organic cation-choline (**xxi**); organic zwitterion-ammonium propionate (**xxii**); carbene-dichlorocarbene (**xxiii**); organic 1,3-dipolar-nitrone molecule (**xxiv**); organopnictogen-N-(4-phenylamino-quinazolin-6-yl)-acrylamide (**xxv**); acetylide-lithium acetylide (**xxvi**); homogenous metal - cerium with mixed metals (**xxvii**); homogenous non-metal-noble gas helium (**xxviii**); mixed metal/non-metal-potassium nitrate (**xxix**); inorganic salt-sodium chloride (**xxx**); miscellaneous inorganic-cyanide (**xxxi**)

DrugTax class, helper functions and variables

Prior to starting the calculations, a few variables (C.S.1—Halogens, metals and group-15/nitrogen atoms lists) helper functions were constructed (C.S.2—To retrieve only ordered atom sequence and C.S.3—To allow atom rings identification). Furthermore, two functions were made available for upcoming feature extraction: one allows for the count of characters on SMILES (C.S.4), while the other initializes an empty dictionary of superclass feature data (C.S.5). Finally, the DrugTax class object itself is initialized with the computation of several useful characteristics (C.S.6 – DrugTax class object initialization).

Kingdoms: organic and inorganic

The general rule to assess whether a compound is organic, or inorganic depends on the existence of at least one carbon atom, in which case it is categorized as an organic compound. There are a few exceptions. For example, some compounds, although containing carbon atoms, are nonetheless, considered inorganic, e.g., isocyanide/cyanide, thiophosgene, carbon diselenide, carbon monosulphide, carbon disulphide, carbon subsulphide, carbon monoxide, carbon suboxide and dicarbon monoxide. The code accessible in C.S.7 allows the discrimination between the two possible kingdoms. Subsequently the matching superclasses will be called, in accordance with C.S.6.

Organic compounds

As previously mentioned, an in accordance with ClassyFire [13], DrugTax considers 26 possible superclasses for organic compounds, listed below and for which the code to compute them from the basic SMILES is displayed in Additional file 1.

Organoheterocyclic

According to the Nomenclature of Organic Compounds "*Organic heterocyclic systems contain one or more foreign elements such as oxygen, sulphur, or nitrogen in addition to carbon*" [16]. As such, we considered organoheterocyclic compounds those which contain a ring with least one carbon atom and one non-carbon atom (C.S.8). The organoheterocyclic superclass is illustrated with an imidazole molecule in Fig. 1-i.

Organosulphur

According to Arya et al. [17], "*Organosulphur compounds are organic molecules that contain sulphur and are associated with the pungent odors*" [17], and as such, we identified organosulfur compounds as those with at least one carbon–sulphur bond (C.S.9). The organosulphur superclass is depicted with a glutathione in Fig. 1-ii.

Lipids

According to the definition by Jones [18], "*Lipids may be classified as a mixed group of substances with the common characteristics of solubility in organic solvents*". This group of biological molecules can be further split into simple lipids (i), such as fats—neutral esters of glycerol with satured and unsaturated acids; compound lipids (ii) consist of a fatty acid, an alcohol and at least one group containing atoms such as phosphorus or nitrogen; derived lipids (iii) are fatty acids that stem from simple or compound lipids by means of hydrolysis.

As seen above, the chemical definition of lipids is quite broad. Within DrugTax implementation, we narrowed it down to fatty acids and their derivatives, as well as substances related biosynthetically or functionally to these compounds. This corresponds to the occurrence of carboxyl group as well as a carbon chain at least four carbons long, regardless of chain saturation (C.S.10). These criteria were driven by literature assessment, in agreement with Aslan and Aslan, 2017 definition [19]. Behenic acid (fatty acid) is shown in Fig. 1-iii.

Allenes

"*Allenes are organic compounds in which one carbon atom has double bonds with each of its two adjacent carbon centres*" in accordance with IUPAC Gold Book allenes entry [20]. The definition includes both the hydrocarbon molecules and their derivatives obtained by substitution (C.S.11). The allenes superclass is depicted with a fucoxanthin in Fig. 1-iv.

Benzenoids

According to Gutman and Babić [21], benzenoids are aromatic compounds containing one or more benzene rings, formed solely by carbon atoms. The code for benzenoid superclass attribution can be consulted at C.S.8. Benzene hexacarboxylic acid, an example, is representated in Fig. 1-v.

Phenylpropanoids and polyketides

According to Zhang and Stephanopoulos [22], "The phenylpropanoids are a family of organic compounds with an aromatic ring and a three-carbon propene tail and are synthesized by plants from the amino acids phenylalanine and tyrosine" [23]. Regarding polyketides, Korman et al. says: "Polyketides are a large class of structurally diverse, acetate derived natural products that exhibit a wide range of bioactivities." [24]. As such, phenylpropanoids and polyketides are organic compounds that are synthesized either from the amino acid phenylalanine (phenylpropanoids) or the decarboxylative condensation of malonyl-CoA (polyketides). Phenylpropanoids are aromatic compounds based on the phenylpropane skeleton. Polyketides usually consists of alternating carbonyl and methylene groups (beta-polyketones), biogenetically derived from repeated condensation of acetyl coenzyme A (via malonyl coenzyme A) (C.S.12). The phenylpropanoids and polyketides superclass is depicted with a phenylalanine in Fig. 1-vi.

Organic acids and derivatives

According to Richter et al. [25] "Organic acids are weak acids with pK_a values that range widely from as low as 3 (carboxylic) to as high as 9 (phenolic)". Furthermore, according to Papagianni 2011, "Organic acids contain one or more carboxylic acid groups, which may be covalently linked in groups such as amides, esters, and peptides." Although we are aware that there are different definitions, some of which consider organic acids without a carboxyl group [26], we considered organic acids those with carboxyl groups (C.S.13). The organic acids superclass is depicted using butyric acid as an example in Fig. 1-vii.

Alkaloids

According to Kurek, "Alkaloids are a huge group of naturally occurring organic compounds which contain nitrogen atom or atoms (amino or amido in some cases) in their structures. These nitrogen atoms cause alkalinity of these compounds" [27]. DrugTax classifies small molecules as alkaloid it exists nitrogen atom(s) and they have a negative net charge (C.S.14). The alkaloid superclass is depicted with a morphine molecule in Fig. 1-viii.

Organic salts

Organic compounds consist of an assembly of cations and anions, of which one must be organic. According to Seçken, Nilgün, "Organic salts, however, are compounds that are formed from at least one anion and one cation. Their anions are organic acid based" [28] (C.S.15). Acetate molecule was used to exemplify this superclass in Fig. 1-ix.

Organohalogen compounds

According to Roberts and Caserio. "*The general term of "organohalogen" refers to compounds with covalent carbon-halogen bonds*" [29]. As such, by listing the halogen atoms in C.S.1, using the code below it is possible to identify organohalogens (C.S.16). The organohalogen compounds superclass is depicted with an acetyl chloride in Fig. 1-x.

Organometallic compounds

According to Abbot et al. the existence of at least on metal–carbon allows the classification into Organometallic compounds [30]. Given this definition, DrugTax identifies organometallic compounds using the same code as for organohalogens (C.S.16) but accessing the metals list instead (C.S.1). The organometallic compounds superclass is depicted with ferrocene in Fig. 1-xi.

Organic nitrogen compounds

According to Moreno and Peinado, "Nitrogen compounds can be classified as mineral or organic. (…) Organic compounds, in contrast, are carbon and hydrogen compounds that contain a nitrogen atom" [31]. In the context of DrugTax, organic nitrogen compounds are simply organic compounds that contain nitrogen atoms. As such, we identify nitrogen atoms upon kingdom attribution completion (C.S. 17). Pyrrole-2-carboxylate, an example of this superclass, can be found in Fig. 1-xii.

Nucleosides and nucleotides

According to Sparkman et al. "*Nucleosides consist of a purine or a pyrimidine base and a ribose or a deoxyribose sugar connected*" [32]. Nucleotides, on the other hand, are defined by Joseph, A. as "*A nucleotide is a subunit of DNA or RNA that consists of a nitrogenous base (A, G, T, or C in DNA; A, G, U, or C in RNA), a phosphate molecule, and a sugar molecule (deoxyribose in DNA, and ribose in RNA)*" [33]. Considering these definitions, nucleotides are simply nucleosides with phosphate groups. As such, to identify nucleosides and nucleotides is necessary to encounter any combination of cytosine, adenine, guanine, thymine, uracil with either ribose or deoxyribose (C.S.18). The nucleosides and nucleotides superclass are represented with guanine in Fig. 1-xiii.

Organic oxygen compounds

As shown by Lee and Meyer [34], the quantification of oxygen in organic compounds can be detrimental in characterizing said compounds. DrugTax also identifies whether the input drug has oxygens or not (C.S.17). The organic oxygen compounds superclass is illustrated with ethanol Fig. 1-xiv.

Organophosphorus compounds

According to Müller "*Organophosphorus compounds with phosphorus–carbon multiple bonds provide a rich and fascinating coordination chemistry*" [35]. By identifying phosphorus in an organic compound (C.S.17), we can recognize organophosphorus compounds. The organophosphorus compounds superclass is depicted with diethyl phosphonate Fig. 1-xv.

Lignan and neolignans

Sang and Zhu states: "*Lignans form a group of phenolic compounds with a backbone of two phenylpropanoid*

(C6C3) units" [36]. According to this definition, DrugTax identifies lignans and neolignans according to the occurrence of either p-propyphenol or phenylpropane (C.S.19). The lignans and neolignans superclass is shown with matairesinol Fig. 1-xvi.

Organic polymers

Yadav and Sinha states that organic polymers are long, chained macromolecules composed of many repeating monomer units" [37]. As such, DrugTax identifies repeating patterns in the molecules of the organic kingdom to identify organic polymers (C.S.20). The organic polymers superclass is depicted with starch Fig. 1-xvii.

Hydrocarbons

According to Enerijiofi "*Hydrocarbons are a group of chemical organic compounds composed of carbon and hydrogen*" [38]. In this case, if the input molecule has not atoms besides carbon and hydrogen, DrugTax will classify the molecule as a hydrocarbon (C.S.21). The hydrocarbons superclass is depicted with octane Fig. 1-xviii.

Hydrocarbon derivatives

Extending from the definition of Enerijiofi, hydrocarbon derivatives are organic compounds derived from hydrocarbon in which there are atoms different from carbon and hydrogen. DrugTax uses the same function (C.S.21) to identify both hydrocarbons and hydrocarbon derivatives. The hydrocarbon derivatives superclass is portrayed with ethanol Fig. 1-xix.

Organic anions

According to Sekine et al.:"*Organic anions are chemically heterogeneous substances possessing a carbon backbone and a net negative charge*" [39]. As such, DrugTax accounts identifies as organic cations the organic molecules with a negative net charge (C.S.22). The organic anions superclass is showed with phosphate Fig. 1-xx.

Organic cations

In contrast with Sekine et al.'s definition of organic anions, organic cations carry a net positive charge. As such, the same process can be applied (C.S.22), this time considering an overall positive net charge. The organic cations superclass is shown with choline Fig. 1-xxi.

Organic zwitterions

According to Hadjesfandiari and Parambath: "*Zwitterions contain both positive- and negative-charged groups, with an overall neutral charge*" [40]. Considering this definition, DrugTax leverages the same approach of the previous two superclasses (C.S.22), for organic cations and anions. However, in this case, it is important to highlight that zwitterions are not merely organic compounds without a charge. They must have an equal number of negative and positive charges. The organic zwitterions superclass is depicted with ammonium propionate in Fig. 1-xxii.

Carbenes

Savin states: "*A carbene is a neutral divalent carbon species containing two electrons that are not shared with other atoms*" [41]. As such, DrugTax identifies carbenes as organic molecules with unpaired electrons at a carbon atom (C.S.23). The carbenes superclass is depicted by dichlorocarbene in Fig. 1-xxiii.

Organic 1,3-dipolar compounds

The IUPAC Compendium of Chemical Terminology defines dipolar compounds as "*Electrically neutral molecules carrying a positive and a negative charge in one of their major canonical descriptions*" [42]. Further along, it extends the definition to 1,3-dipolar compounds as "*those in which a significant canonical resonance form can be represented by a separation of charge over three atoms*" [42]. According to this definition, DrugTax identifies organic 1,3-dipolar compounds if they simultaneously possess positive and negative charges. However, the net charge should be neutral, and the compound must have one atom separating the atoms with the opposing charges (C.S.24). Nitrone molecule was chosen as an example, and it is depicted in Fig. 1-xxiv.

Organopnictogen compounds

IUPAC defines pnictogens as an atom belonging to group 15 of the periodic table, which include nitrogen, phosphorus, arsenic, antimony and bismuth [43]. To identify organopnictogens, DrugTax leverages the list of the group 15 atoms (C.S.1) and checks whether there are any bounds between these atoms and carbons (C.S.25). The organopnictogen superclass is depicted with N-(4-phenylamino-quinazolin-6-yl)-acrylamide in Fig. 1-xxv.

Acetylides

According to the IUPAC Compendium of Chemical Terminology, acetylides obey the following principles: "Compounds arising by replacement of one or both hydrogen atoms of acetylene (ethyne) by a metal or other cationic group. E.g., NaC≡CH monosodium acetylide. By extension, analogous compounds derived from terminal acetylenes, RC≡CH" [44]. By using the list of metal atoms (C.S.1), DrugTax identifies acetylides as organic compounds with a triple covalent bond between two carbon atoms, with at least one of them, bounded to a metal atom (C.S.26). Lithium acetylide is portrayed as an example of this superclass in Fig. 1-xxvi.

Inorganic

As previously mentioned, and in accordance with ClassyFire [13], DrugTax considers five possible superclasses for inorganic compounds, listed in the next subsections. As these definitions are overall quite straightforward and elementary, we will present equally simple definitions.

Homogenous metal compounds

Homogenous metal compounds are inorganic compounds that contain only metal atoms. These atoms, however, are not necessarily all atoms of the same metal. The list of metals was retrieved from C.S.1. The code to identify homogenous metal compounds can be found at C.S.27. The homogenous metal superclass is illustrated as cerium with mixed metals Fig. 1-xxvii.

Homogenous non-metal compounds

Homogenous non-metal compounds are inorganic compounds that contain only non-metal atoms. The list of metals was retrieved from C.S.1. The code to identify homogenous non-metal compounds can be found at C.S.28. As an example, gas helium is shown in Fig. 1-xxviii.

Mixed metal/non-metal compounds

Mixed metal/non-metal compounds are inorganic compounds that can contain simultaneously metal and non-metal atoms. The list of metals was retrieved from C.S.1. The code to identify homogenous non-metal compounds can be found at C.S.29. Potassium nitrate is depicted as an example in Fig. 1-xxix.

Inorganic salts

The superclass of inorganic salts consists of inorganic compound with one or more charges, either negative or positive ones. The code to identify inorganic salts can be found at C.S.30. The inorganic salts superclass is depicted with sodium chloride in Fig. 1-xxx.

Miscellaneous inorganic compounds

The identification of miscellaneous inorganic compounds is dependent on the previous four inorganic superclasses. If a given compound does not fit any of these superclasses, it is considered a miscellaneous inorganic compound. Cyanide (Fig. 1-xxxi) was chosen to illustrate this superclass.

DrugTax bulk analysis and plotting tools

One of the main purposes of this work was to allow bulk analysis of chemical properties of drugs to enable proper, tailored, and comprehensive categorization of small ligands. With that in mind, DrugTax has an additional tool for bulk ligand analysis, which makes use of kingdom and superclass attribution to perform categorization of small molecules. These categories account for multiple superclasses, in the cases in which this is possible. Firstly, it was added a short functionality to fetch the isomeric SMILES from the drug name, by using pubchempy (C.S.31). Then, using C.S. 1–30, the different superclasses for each ligand are listed (C.S.32).

By retrieving summary data from the input list of SMILES, DrugTax uses individual small ligand information to generate a fast characterization tool of small molecule datasets. Furthermore, by making use of UpSetPlot [14], DrugTax can depict many intersecting sets (in the form of small ligand superclasses), which is often limited by more conventional forms of visualization. The plots are generated from the summary information previously retrieved and can be tuned to avoid close to empty superclass aggregations (C.S.33).

Results and case study

To exemplify the usage of DrugTax, we developed a short approach that assembles a dataset focused on drugs associated with a variety of known viruses. Firstly, we performed a query using PUG-REST (Power User Interface–Representational State Transfer) [45], a web interface of PubChem [1] that allows the programmatic access of information of chemical compounds present in the database. The requests to the server are made through URLs (Uniform Resource Locators). To comply with PUG-REST's request volume limit, 100 compounds are fetched at a time, while the total amount of compounds to be analyzed must be specified by the user. This parameter ultimately affects the size of the resulting dataset. The compounds are scraped by the iterating over the list of CIDs (Compound ID).

Another parameter that must be specified by the user are the keywords related to the dataset one wants to create. These keywords must be present in the more relevant bioassays titles, in this case, the keywords were chosen after looking at the most frequently appearing terms in the titles of *Journal of Virology* [46] studies (accessed on the 29th of July 2022). The chosen keywords affect the size, diversity, and quality of the dataset, and so a good selection is key. It is also to note that these keywords are case sensitive and can also be present inside a word. The used keywords were: DENV, HIV, H1N1, virus, viral, Viral, SARS, Virus, HCV, influenza, Influenza, HSV, HHV, EBOV, MERS. This query was performed over 700.000 compounds.

To build a dataset relevant in the settings of both a biological problem and ML implementation, it was relevant to narrow the compounds according to their activity. As such, we selected only compounds that were featured in biological activity studies. To fulfill these criteria, we explored the information related to bioassays, regarding our compounds, in PubChem [1]. Bioassays are analytical methods to calculate the potency of chemical compounds in biological beings, making them a good source of experimentally proven data that can be accessed easily through PUG-REST [45]. We retrieved the corresponding bioassays for each compound.

Regarding the bioassays that were relevant for DrugTax's purpose a selection took place, respecting the following conditions:

- Exclusive to the compound: The study must have the compound as the only studied chemical (an activity value is presented).
- Related to the input keywords: The study title must have at least one of the keywords introduced by the user.
- Conclusive: The result of the bioassay must be either "Active" or "Inactive", any other results like "Unspecified" or "Inconclusive" were excluded.
- Target protein: There must be an ID of a protein target.

After performing this selection, our dataset was reduced to 10.567 unique compounds, targeting 367 unique proteins. However, several bioassays can involve the same protein-compound pair, and therefore were subsequently removed. As the activity values can vary, a pair was only considered as active if more than 50% of the studies indicate so, the same applies to the inactive, but if it is exactly 50% the pair was taken as inconclusive and removed. This analysis was performed by replacing the activity values by numbers (1 for active and 0 for inactive). As such, we simultaneously consider the positively reported interactions (active) and their counterpart (inactive). The surge of ML-based approaches further stressed out the need to report both positive and negative results, giving rise to new research terms like Structure Inactive Relationships (SIR), which complements the more standard Structure Activity Relationships (SAR) approaches [47]. After performing this final step of pre-processing, the dataset still tallied a total of 10.556 unique compounds and 367 unique proteins.

Finally, it was necessary to retrieve these compounds in a usable format, for which we considered SMILES. A request was conducted PUG-REST [45] returning the isomeric SMILES string of the compound using the CID. Achieving a list of 10.556 SMILES representing unique virus-related compounds, these were tested using our new developed package—DrugTax. Running the DrugTax class on the compounds, their object representation, including superclass categorization and DrugTax features did not exceed 10 s, on a common portable laptop (16 Gb RAM and 11th Gen Intel Core i7-11370H, 3.30 GHz CPU). After retrieving the computed data on table format, we proceeded with the bulk analysis and plotting devices of DrugTax, yielding the UpSetPlot [14] in Fig. 2. As expected, most of the compounds belong to the organic kingdom, although a few exceptions were observed in the form of inorganic salts and/or mixed metal/non-metal inorganic compounds. The most recurring superclass was hydrocarbon derivatives, with few hydrocarbons present (organic molecules containing only carbon and hydrogen). The most populated aggregation of superclasses were organic molecules that fit the superclasses: hydrocarbon derivatives, organoheterocyclic, organic oxygen, organic nitrogen and organopnictogens.

Applications

DrugTax was developed to simplify molecule characterization. In particular, we deliver a comprehensible molecule categorization as well as clear and humanly

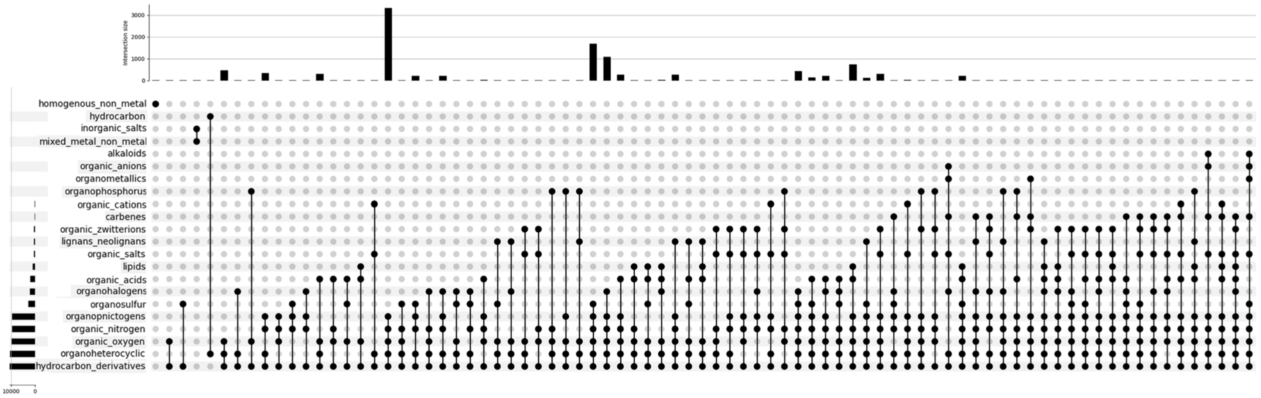


Fig. 2 UpSetPlot displaying the bulk analysis of 10.567 unique compounds related to virus research

interpretable features, which yields a set of simple and fundamental level applications. For example, DrugTax package could be applied to generate similarity searches, chemical space visualization, clustering, taxonomy-property relationships, among others. The results could then be combined with different easy-to-implement visualization tools. For instance, for similarity search, a hierarchical clustering plot could capture the stratified difference between the various molecules. Likewise, for chemical space visualization, by using DrugTax features and projecting the feature vectors into two dimensions with Principal Component Analysis (PCA) or the more recent Uniform Manifold Approximation and Projection (UMAP), users could then produce different scatterplots colored by taxonomic kingdom or superclass.

Due to its easy deployment and installation, DrugTax is a tool whose potential can unfold extensively.

Conclusions

DrugTax exhibits very fast performance with an easy-to-use interface available on PyPI (https://pypi.org/project/DrugTax/) and GitHub (https://github.com/MoreiraLAB/DrugTax). It extends on the work of Classyfire [13] with novel features oriented towards data science, ML and AI solutions. Its heavily focused on interpretable pharmacological data and features, key for the scientific community, as well as the Pharma sector. DrugTax offers flexible solutions in an intuitive setting that explores the possibilities of SMILES representations for ML and AI solutions on a data-centric setting.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-022-00649-w.

Additional file 1. Code Snippets regarding DrugTax.

Acknowledgements

Authors would like to acknowledge STRATAGEM—New diagnostic and therapeutic tools against multidrug-resistant tumors, CA17104.

Author contributions

AJP-conceptualization; methodology; software; validation; formal analysis; investigation; resources; writing—review and editing; visualization. PCC-methodology; software; writing—study case. ISM-writing—review and editing; supervision; project administration; funding acquisition. All authors read and approved the final manuscript.

Funding

This work was supported by the European Regional Development Fund through the COMPETE 2020—Operational Programme for Competitiveness and Internationalization and Portuguese national funds via Fundação para a Ciência e a Tecnologia (FCT) [LA/P/0058/2020, UIDB/04539/2020, UIDP/04539/2020, and DSAIPA/DS/0118/2020]. FCT also supported A.J.P. with a PhD scholarship [SFRH/BD/144966/2019]. Funding for open access charge: Fundação para a Ciência e a Tecnologia [DSAIPA/DS/0118/2020].

Availability of data and materials

DrugTax if of simple installation and usage in any computer that carries Python 3.6.x, with very few dependencies. Most of its extended dependencies emerge when using the bulk analysis and plotting options. Having been deposited in PyPI (https://pypi.org/project/DrugTax/), DrugTax is available through pip installation (C.S.34). Alternatively, DrugTax can be cloned from GitHub (https://github.com/MoreiraLAB/DrugTax).
Project name: DrugTax.
Project home page: https://pypi.org/project/DrugTax/
Project source code: https://github.com/MoreiraLAB/DrugTax
Operating system(s): Platform independent.
Programming language: Python.
Other requirements: Python 3.6.x or higher.
License: GNU GPL.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Center for Neuroscience and Cell Biology, University of Coimbra, 3004-504 Coimbra, Portugal. [2]PhD Programme in Experimental Biology and Biomedicine, Institute for Interdisciplinary Research (IIIUC), University of Coimbra, Casa Costa Alemão, 3030-789 Coimbra, Portugal. [3]Department of Life Sciences, University of Coimbra, Calçada Martim de Freitas, 3000-456 Coimbra, Portugal. [4]CIBB - Center for Innovative Biomedicine and Biotechnology, University of Coimbra, 3004-504 Coimbra, Portugal.

Received: 8 August 2022 Accepted: 3 October 2022
Published online: 27 October 2022

References

1. Kim S, Chen J, Cheng T et al (2021) PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res 49(D1):D1388–D1395. https://doi.org/10.1093/NAR/GKAA971
2. Wishart DS, Feunang YD, Guo AC et al (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res 46(D1):1074–1082. https://doi.org/10.1093/NAR/GKX1037
3. Gaulton A, Hersey A, Nowotka ML et al (2017) The ChEMBL database in 2017. Nucleic Acids Res 45(D1):D945–D954. https://doi.org/10.1093/NAR/GKW1074
4. Paul D, Sanap G, Shenoy S, Kalyane D, Kalia K, Tekade RK (2021) Artificial intelligence in drug discovery and development. Drug Discov Today 26(1):80. https://doi.org/10.1016/J.DRUDIS.2020.10.010
5. Vamathevan J, Clark D, Czodrowski P et al (2019) Applications of machine learning in drug discovery and development. Nat Rev Drug Discov 18(6):463. https://doi.org/10.1038/S41573-019-0024-5
6. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open Babel: an open chemical toolbox. J Cheminform. https://doi.org/10.1186/1758-2946-3-33
7. Moriwaki H, Tian YS, Kawashita N, Takagi T (2018) Mordred: a molecular descriptor calculator. J Cheminform 10(1):1–14. https://doi.org/10.1186/S13321-018-0258-Y/FIGURES/6
8. Cao Y, Charisi A, Cheng LC, Jiang T, Girke T (2008) ChemmineR: a compound mining framework for R. Bioinformatics 24(15):1733–1734. https://doi.org/10.1093/BIOINFORMATICS/BTN307
9. Li J, Cai D, He X (2017) Learning graph-level representation for drug discovery. arXiv. https://arxiv.org/abs/1709.03741
10. Kearnes S, McCloskey K, Berndl M, Pande V, Riley P (2016) Molecular graph convolutions: moving beyond fingerprints. J Comput Aided Mol Des 30(8):595–608. https://doi.org/10.1007/s10822-016-9938-8
11. Skalic M, Varela-Rial A, Jiménez J, Martínez-Rosell G, de Fabritiis G (2019) LigVoxel: inpainting binding pockets using 3D-convolutional neural networks. Bioinformatics 35(2):243–250. https://doi.org/10.1093/BIOINFORMATICS/BTY583
12. Nelson DL, Cox M (2013) Lehninger principles of biochemistry, 6th edn. W.H. Freeman and Company, New York

13. Djoumbou Feunang Y, Eisner R, Knox C et al (2016) ClassyFire: automated chemical classification with a comprehensive, computable taxonomy. J Cheminform 8(1):1–20. https://doi.org/10.1186/S13321-016-0174-Y
14. Lex A, Gehlenborg N, Strobelt H, Vuillemot R, Pfister H (2014) UpSet: visualization of intersecting sets. IEEE Trans Vis Comput Graph 20(12):1983–1992. https://doi.org/10.1109/TVCG.2014.2346248
15. Weininger D (1988) SMILES, a chemical language and information system: 1: introduction to methodology and encoding rules. J Chem Inf Comput Sci 28(1):31–36. https://doi.org/10.1021/CI00057A005/ASSET/CI00057A005.FP.PNG_V03
16. Fletcher JH, Dermer OC, Fox RB (1974) Heterocyclic systems-nomenclature of organic compounds. In: Fletcher JH, Dermer OC, Fox RB (eds) Advances in Chemistry, vol 126. Acs Publications, Washington, pp 49–64. https://doi.org/10.1021/BA-1974-0126.CH006
17. Arya R, Saldanha SN (2018) Dietary phytochemicals, epigenetics, and colon cancer chemoprevention. Epigenetics Cancer Prev. https://doi.org/10.1016/B978-0-12-812494-9.00010-X
18. Jones ML (2008) Lipids. In: Jones ML (ed) Theory and practice of histological techniques. Elsevier, Amsterdam, pp 187–215. https://doi.org/10.1016/B978-0-443-10279-0.50019-1
19. Aslan I, Aslan M (2017) Plasma polyunsaturated fatty acids after weight loss surgery. Metab Pathophysiol Bariatr Surg. https://doi.org/10.1016/B978-0-12-804011-9.00058-3
20. McNaught AD, Wilkinson A (2019) IUPAC. Compendium of chemical terminology, 2nd edn. Blackwell Scientific Publications, Oxford
21. Gutman I, Babić D (1991) Characterization of all-benzenoid hydrocarbons. J Mol Struct Theochem 251:367–373. https://doi.org/10.1016/0166-1280(91)85159-5
22. Zhang H, Stephanopoulos G (2016) Co-culture engineering for microbial biosynthesis of 3-amino-benzoic acid in Escherichia coli. Biotech Method 11(7):981–987. https://doi.org/10.1002/biot.201600013
23. Kawaguchi H, Ogino C, Kondo A (2017) Microbial conversion of biomass into bio-based polymers. Bioresour Technol 245:1664–1673. https://doi.org/10.1016/J.BIORTECH.2017.06.135
24. Korman TP, Ames B, Tsai SC (2010) Structural enzymology of polyketide synthase: the structure-sequence-function correlation. In: Mander L, Liu HW (eds) Comprehensive natural products II: chemistry and biology, vol 1. Elsevier, Amsterdam, pp 305–345. https://doi.org/10.1016/B978-008045382-8.00020-4
25. de Richter B, Oh NH, Fimmen R, Jackson J (2007) The Rhizosphere and soil formation. In: Cardon ZG, Whitbeck JL (eds) The Rhizosphere. Elsevier, Amsterdam, pp 179–200. https://doi.org/10.1016/B978-012088775-0/50010-0
26. Perez GV, Perez AL (2000) Organic acids without a carboxylic acid functional group. J Chem Educ 77(7):910–915. https://doi.org/10.1021/ED077P910
27. Kurek J (2019) Introductory chapter: alkaloids —their importance in nature and for human life. In: Kurek J (ed) Alkaloids-their importance in nature and human life. Intechopen, London. https://doi.org/10.5772/INTECHOPEN.85400
28. Seçken N (2010) Identifying student's misconceptions about SALT. Proc Soc Behav Sci. https://doi.org/10.1016/j.sbspro.2010.03.004
29. Roberts JD, Caserio MC (2022) Chapter 29. Polymers. Basic principles of organic chemistry. pp 1419–1459. http://resolver.caltech.edu/CaltechBOOK:1977.001%5Cn; http://authors.library.caltech.edu/25034/30/BPOCchapter29.pdf. Accessed 30 Jun 2022
30. Abbott JKC, Dougan BA, Xue ZL (2011) Synthesis of organometallic compounds. Mod Inorg Synth Chem. https://doi.org/10.1016/B978-0-444-53599-3.10013-7
31. Moreno J, Peinado R (2012) Enological chemistry. Academic Press, Cambridge
32. Sparkman OD, Penton ZE, Kitson FG (2011) Nucleosides (TMS derivatives). In: Sparkman OD (ed) Gas chromatography and mass spectrometry: a practical guide. Elsevier, Amsterdam, pp 369–371. https://doi.org/10.1016/B978-0-12-373628-4.00027-7
33. Joseph A (2017) The role of oceans in the origin of life and in biological evolution. In: Joseph A (ed) Investigating seafloors and oceans. Elsevier, Amsterdam, pp 209–256. https://doi.org/10.1016/B978-0-12-809357-3.00004-7
34. Lee TS, Robert M (1955) A new method for the determination of oxygen in organic compounds. Anal Chim Acta 13:340–349. https://doi.org/10.1016/S0003-2670(00)87954-4
35. Müller C (2019) Copper(I) complexes of low-coordinate phosphorus(III) compounds. In: Müller C (ed) Copper(I) chemistry of phosphines, functionalized phosphines and phosphorus heterocycles. Elsevier, Amsterdam, pp 1–19. https://doi.org/10.1016/B978-0-12-815052-8.00001-4
36. Sang S, Zhu Y (2014) Bioactive phytochemicals in wheat bran for colon cancer prevention. In: Sang S (ed) Wheat and rice in disease prevention and health. Elsevier, Amsterdam, pp 121–129. https://doi.org/10.1016/B978-0-12-401716-0.00010-6
37. Yadav A, Sinha N (2021) Organic polymers for drinking water purification. In: Yadav A (ed) Reference module in materials science and materials engineering. Elsevier, Amsterdam. https://doi.org/10.1016/B978-0-12-820352-1.00140-1
38. Enerijiofi KE (2020) Bioremediation of environmental contaminants: a sustainable alternative to environmental management. In: Enerijiofi KE (ed) Bioremediation for environmental sustainability: toxicity, mechanisms of contaminants degradation, detoxification and challenges. Elsevier, Amsterdam, pp 461–480. https://doi.org/10.1016/B978-0-12-820524-2.00019-5
39. Sekine T, Cha SH, Endou H (2000) The multispecific organic anion transporter (OAT) family. Pflügers Arch 440(3):337–350. https://doi.org/10.1007/S004240000297
40. Hadjesfandiari N, Parambath A (2018) Stealth coatings for nanoparticles: polyethylene glycol alternatives. In: Hadjesfandiari N (ed) Engineering of biomaterials for drug delivery systems: beyond polyethylene glycol. Elsevier, Amsterdam, pp 345–361. https://doi.org/10.1016/B978-0-08-101750-0.00013-1
41. Savin KA (2014) Reactions involving acids and other electrophiles. In: Savin KA (ed) Writing reaction mechanisms in organic chemistry. Elsevier, Amsterdam, pp 161–235. https://doi.org/10.1016/B978-0-12-411475-3.00004-X
42. McNaught AD, Wilkinson A (2008) Dipolar compounds. The IUPAC compendium of chemical terminology. Blackwell Scientific Publications, Oxford. https://doi.org/10.1351/GOLDBOOK.D01753
43. Connelly NG, Damhus T, Hartshorn RM, Alan T (2022) Hutton. Nomenclature of inorganic compounds. IUPAC recommendations 2005. P 377. http://old.iupac.org/publications/books/rbook/Red_Book_2005.pdf. Accessed 12 Sept 2022
44. McNaught AD, Wilkinson A (2008) Acetylides. The IUPAC compendium of chemical terminology. Blackwell Scientific Publications, Oxford
45. Kim S, Thiessen PA, Cheng T, Yu B, Bolton EE (2018) An update on PUG-REST: RESTful interface for programmatic access to PubChem. Nucleic Acids Res 46(W1):W563–W570. https://doi.org/10.1093/NAR/GKY294
46. American Society for Microbiology. Journal of Virology. ASM Journals
47. López-López E, Fernández-de Gortari E, Medina-Franco JL (2022) Yes SIR! On the structure-inactivity relationships in drug discovery. Drug Discov Today 27(8):2353–2362. https://doi.org/10.1016/J.DRUDIS.2022.05.005

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

3.3. Drug synergy prediction of cancer cell lines

3.3. SYNPRED: prediction of drug combination effects in cancer using different synergy metrics and ensemble learning

GigaScience, 2022, 11, 1–15
DOI: 10.1093/gigascience/giac087
TECH NOTE

SYNPRED: prediction of drug combination effects in cancer using different synergy metrics and ensemble learning

António J. Preto[1,2], Pedro Matos-Filipe[1], Joana Mourão[4] and Irina S. Moreira[3,4,*]

[1]Center for Neuroscience and Cell Biology, University of Coimbra, 3004-504 Coimbra, Portugal
[2]PhD Programme in Experimental Biology and Biomedicine, Institute for Interdisciplinary Research (IIIUC), University of Coimbra, Casa Costa Alemão, 3030-789 Coimbra, Portugal
[3]Department of Life Sciences, University of Coimbra, Calçada Martim de Freitas, 3000-456 Coimbra, Portugal
[4]CNC—Center for Neuroscience and Cell Biology, CIBB—Center for Innovative Biomedicine and Biotechnology, 3004-504 Coimbra, Portugal
*Correspondence address. Irina S. Moreira. E-mail: irina.moreira@cnc.uc.pt

Abstract

Background: In cancer research, high-throughput screening technologies produce large amounts of multiomics data from different populations and cell types. However, analysis of such data encounters difficulties due to disease heterogeneity, further exacerbated by human biological complexity and genomic variability. The specific profile of cancer as a disease (or, more realistically, a set of diseases) urges the development of approaches that maximize the effect while minimizing the dosage of drugs. Now is the time to redefine the approach to drug discovery, bringing an artificial intelligence (AI)–powered informational view that integrates the relevant scientific fields and explores new territories.

Results: Here, we show SYNPRED, an interdisciplinary approach that leverages specifically designed ensembles of AI algorithms, as well as links omics and biophysical traits to predict anticancer drug synergy. It uses 5 reference models (Bliss, Highest Single Agent, Loewe, Zero Interaction Potency, and Combination Sensitivity Score), which, coupled with AI algorithms, allowed us to attain the ones with the best predictive performance and pinpoint the most appropriate reference model for synergy prediction, often overlooked in similar studies. By using an independent test set, SYNPRED exhibits state-of-the-art performance metrics either in the classification (accuracy, 0.85; precision, 0.91; recall, 0.90; area under the receiver operating characteristic, 0.80; and F1-score, 0.91) or in the regression models, mainly when using the Combination Sensitivity Score synergy reference model (root mean square error, 11.07; mean squared error, 122.61; Pearson, 0.86; mean absolute error, 7.43; Spearman, 0.87). Moreover, data interpretability was achieved by deploying the most current and robust feature importance approaches. A simple web-based application was constructed, allowing easy access by nonexpert researchers.

Conclusions: The performance of SYNPRED rivals that of the existing methods that tackle the same problem, yielding unbiased results trained with one of the most comprehensive datasets available (NCI ALMANAC). The leveraging of different reference models allowed deeper insights into which of them can be more appropriately used for synergy prediction. The Combination Sensitivity Score clearly stood out with improved performance among the full scope of surveyed approaches and synergy reference models. Furthermore, SYNPRED takes a particular focus on data interpretability, which has been in the spotlight lately when using the most advanced AI techniques.

Keywords: ensemble learning, interpretability, omics, biophysics, drug synergy, cancer

Background

Cancer, a heterogeneous group of diseases, is one of the leading causes of mortality and the most significant barrier to increasing life expectancy worldwide. The International Agency for Research on Cancer estimates that, by 2040, approximately 30.2 million new cancer cases and 16.3 million deaths will be reported, mainly due to the population's growth and aging [1]. One of the significant contributors to this disease's global burden is the development of therapy resistance and, consequently, tumor relapse. Drug resistance in cancer is a multifactorial problem driven by the tumor microenvironment and genetic and nongenetic/epigenetic mechanisms that, along with cell plasticity, contribute to tumor heterogeneity [2]. In clinical settings, this problem is minimized with a combination of drugs administered together or in sequence (i.e., polytherapy). Targeting multiple components of different or interconnected cancer pathways is an efficient strategy to block vital biological processes [3, 4].

Drug combinations with a synergistic effect (i.e., when the total therapeutic effect of both drugs is greater than the expected additive monotherapy effect) [5] were successfully developed and applied in the treatment of different types of tumors, such as human epidermal growth factor receptor 2–positive breast cancer [6], chronic myeloid leukemia [7], prostate cancer [8], or BRAF-mutated tumors [9]. Nevertheless, this simultaneous administration can also result in a reduced therapeutic effect and possible toxicity (designated antagonism) or in the same beneficial effect when compared with the expected additive monotherapy effect (additivity) [5]. The experimental identification of successful synergistically effective combinations is a well-known time-consuming and expensive task. Therefore, there is still a signifi-

Received: December 18, 2021. **Revised:** June 14, 2022. **Accepted:** August 18, 2022

cant need for efficient and user-friendly computational methods, available in easy to use interfaces, to complement and speed up the traditional approaches by predicting the best synergistic drug combinations [10, 11].

In the past years, the development and improvement of high-throughput technologies and computational tools boosted the use of large volumes of multiomics data (e.g., genomic, transcriptomic, proteomic) essential to dissect and uncover the complex molecular signatures of cancer. Machine learning (ML) algorithms have attracted particular attention for their ability to learn new associations and extract valuable insights from this type of data. A few ML models based on extreme gradient boosting, random forest, elastic nets, support vector machine, and naive Bayes were already developed to predict the best combination of anticancer drugs by the integration of omics data with chemoinformatic properties of drugs or network information of their targets [12–15]. Likewise, deep learning (DL) implemented via deep neural networks (DNNs) was particularly useful in dealing with the high multidimensionality of omics data in supervised and unsupervised contexts. DL classification and regression models such as AuDNNsynergy [16], DeepDDS [17], DeepSynergy [18], DeepSignalingSynergy [19], Matchmakers [20], TranSynergy [21], or the work by Xia and colleagues [22] were recently developed for drug combination prediction. Nearly all the surveyed works developed drug synergy prediction models based upon a single reference model, which is in most cases the Loewe reference model [14, 16–18, 20, 21]. Currently, there is a wide scope of well-studied available reference models, including the Bliss independence [23], highest single agent (HSA) [24], Loewe additivity [25, 26], and zero interaction potency (ZIP) [27]. Furthermore, recently Malyutina et al. [15] developed the Combination Sensitivity Score (CSS), which measures drug combination synergy using their IC_{50}. As such, this led us to the question of whether the development of a novel prediction approach should be based solely upon a single reference model. Besides, most of the available web interfaces such as DECREASE [28] or DrugComb [29] require for synergy prediction the upload of a full or partial mandatory dose–response matrix (experimentally determined), which hinders its systematic use by the scientific community and handicaps its usefulness.

To overcome the current problems found in the field, we developed SYNPRED (SYNergy PREDiction), a collection of *in silico* ensemble classification and regression models that considers several synergy references models: Bliss, Loewe, HSA, ZIP, and CSS. It was developed by integrating multiomics features of cell lines and phenotypic and biophysical data, particularly physicochemical and structural features of drugs. SYNPRED displays a good predictive performance and inherently addresses the issue at a broader and more profound angle than the existing approaches, which generally focus on either classification or a single regression task and typically use a single synergy reference model. We made available the stand-alone deployment at https://github.com/MoreiraLAB/synpred, which allows the user the opportunity to undergo bulk prediction with SYNPRED. Additionally, for the first time, a user-friendly web-based application was assembled and made freely available online at http://www.moreiralab.com/resources/synpred/ to predict drug combinations, requiring only the upload of the 2 drugs' simplified molecular-input line-entry system (SMILEs) to be tested. This interactive platform will allow users with different backgrounds, from scientists to clinicians, to test, reproduce, and validate our models and data. The workflow used for the development of SYNPRED is depicted in Fig. 1.

Data and Methods

Experimental drug combination phenotypic data

Drug combination phenotypic data were acquired via bulk-download from the largest-to-date dataset from National Cancer Institute—A Large Matrix of Anti-Neoplastic Agent Combinations (NCI ALMANAC) through https://wiki.nci.nih.gov/display/NCIDTPdata/NCI-ALMANAC [30]. To this date, the dataset includes phenotypic data of tested cancer cell lines (growth percentage) of 105 unique drugs approved by the US Food and Drug Administration (FDA). These drugs were tested in combination against 61 cell lines from 9 cancer types currently included in the NCI [31, 32], comprising a total of 311,466 drug pair/cell line combinations. Drug sensitivity assays included in NCI ALMANAC were performed at the NCI's Frederick National Laboratory for Cancer Research, the Stanford Research Institute, and the University of Pittsburgh. Briefly, for each assay, cells were cultivated for 48 hours in a 3 × 3 or a 5 × 3 concentration matrix (different concentration values for each drug in combination) and the endpoint determined by Sulforhodamine B or CellTiter-Glo [30]. From these records, the authors retrieved the cell growth percentage at each drug concentration point, which corresponds to the percentage of growth of the cell lines in the presence of each combination, yielding a final viability assessment.

Combination scores and class definition

The phenotypic data from high-throughput drug combination screens were retrieved from DrugComb [29]. DrugComb extends its synergy metrics calculations from "SynergyFinder" [33], which leverages the percentage of cell growth included in the dataset to assess the degree of combination for each pair of drug concentrations by using several synergy reference models. As such, only the most well-studied synergy reference models described in the literature were included as they were the only ones that met the criteria of characterizing the effects of a drug pair on a cell line with a final single synergy score. This approach narrowed down our options to the 4 most well-known synergy reference models: Bliss independence (Equation 1) [23], Loewe additivity (Equation 2) [25, 26], HSA (Equation 3) [24], and ZIP (Equation 4) [27]. In addition to the mentioned synergy reference models, we also used the CSS metric [15], a higher sensitivity score [29].

$$yBliss \;=\; y1 + y2 - y1y2 \tag{1}$$

Bliss independence model: yBliss is the Bliss response, y1 is the drug 1 response, and y2 is the drug 2 response.

$$yLoewe \;=\; \frac{Emin + Emax\left(\frac{x1+x2}{m}\right)^{\lambda}}{1+\left(\frac{x1+x2}{m}\right)^{\lambda}} \tag{2}$$

Loewe additivity model: yLoewe is the Loewe response, Emin is the minimum drug response, Emax is the maximum drug response, *m* is the dose that produces a midpoint effect between Emin and Emax, λ is the shape parameter indicating the slope of the curve, x1 is the drug 1 dose, and x2 is the drug 2 dose.

$$yHSA \;=\; \max\,(y1, y2) \tag{3}$$

HSA model: yHSA is the HSA response, y1 is the drug 1 response, and y2 is the drug 2 response.

$$yZIP \;=\; \frac{\left(\frac{x1}{m1}\right)^{\lambda 1}}{1+\left(\frac{x1}{m1}\right)^{\lambda 1}} + \frac{\left(\frac{x2}{m2}\right)^{\lambda 2}}{1+\left(\frac{x2}{m2}\right)^{\lambda 2}} - \left(\frac{\left(\frac{x1}{m1}\right)^{\lambda 1}}{1+\left(\frac{x1}{m1}\right)^{\lambda 1}} * \frac{\left(\frac{x2}{m2}\right)^{\lambda 2}}{1+\left(\frac{x2}{m2}\right)^{\lambda 2}}\right) \tag{4}$$

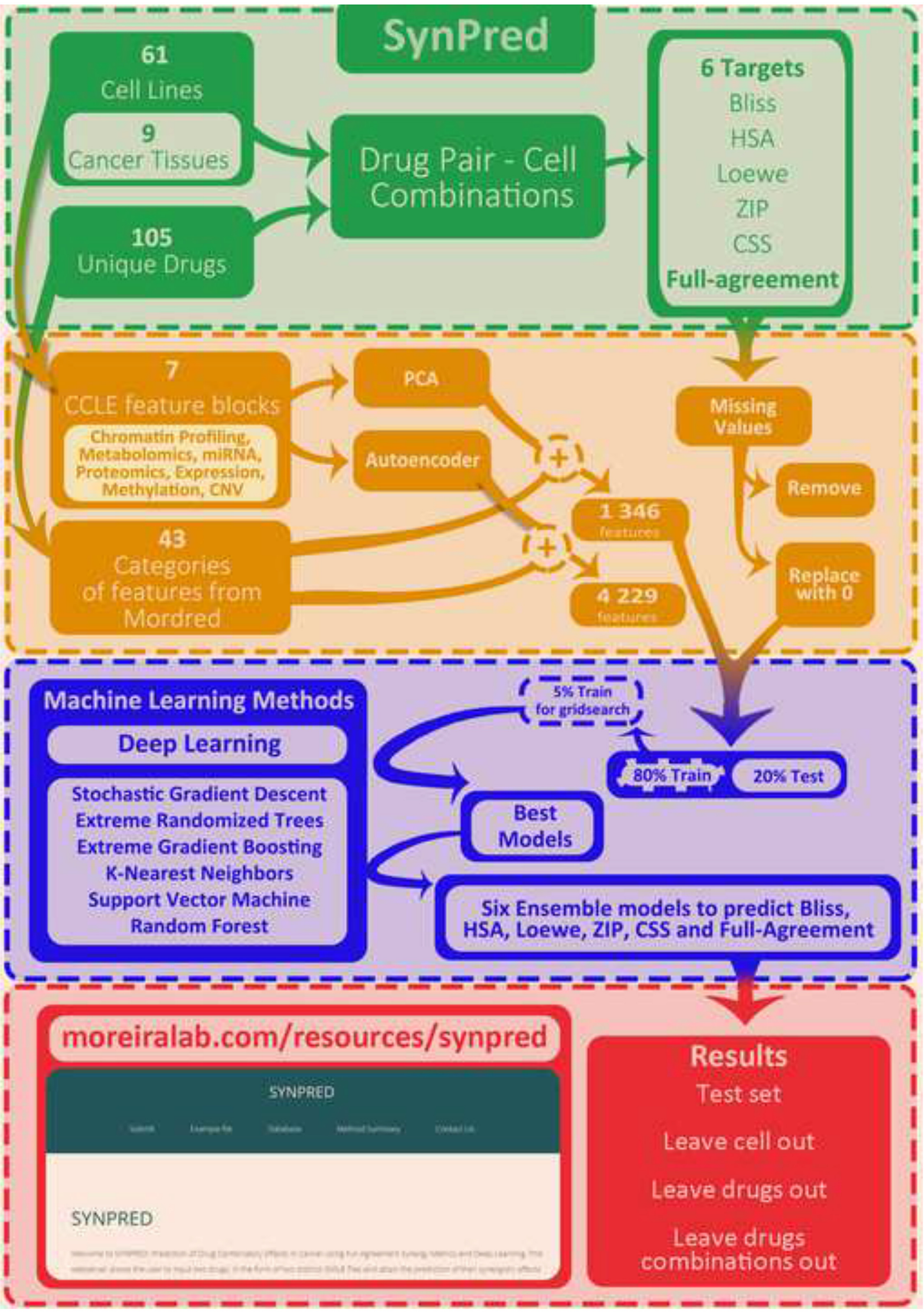


Figure 1: SYNPRED workflow summary. Green: Dataset construction. The National Cancer Institute—A Large Matrix of Anti-Neoplastic Agent Combinations database (phenotypic data) and the Cancer Cell Line Encyclopedia (CCLE) (multiomics data) were used for this purpose. Four reference models (Bliss, HSA, Loewe, ZIP) in addition to the CSS were used to quantify the combination degree and retrieve a full agreement between all metrics. Orange: Feature extraction and data preprocessing. Included normalization and dimensionality reduction using autoencoder or principal component analysis (PCA). Blue: Grid search and prediction model development using a training set. Red: Model evaluation using different classification and regression metrics in an independent test set and 3 different scenarios: (i) leave cell out dataset, (ii) leave drugs out dataset, and (iii) leave drug combinations out dataset.

ZIP model: yZIP is the ZIP response, $x1$ is the drug 1 dose, $x2$ is the drug 2 dose, $m1$ is the dose that produces a midpoint effect for drug 1, $m2$ is the dose that produces a midpoint effect for drug 2, $\lambda 1$ is the shape parameter indicating the slope of the curve for drug 1, and $\lambda 2$ is the shape parameter indicating the slope of the curve for drug 2.

Having computed Bliss, HSA, Loewe, ZIP, and CSS, a binary classifier was first developed to identify the type of combinatory effect present in each drug pair–cell line sample, where the values above the threshold (0, as defined for each metric by SynergyFinder [33]) (https://synergyfinder.fimm.fi/synergy/synfin_docs/) were defined as synergistic, and the remaining ones were classified as nonsynergistic. The dataset used for classification training considered full-agreement combination assessment (i.e., we only kept the instances on which combination classification was the same across the 4 previous reference predictors). For the dataset used, this process yielded 29,779 synergistic samples and 9,029 nonsynergistic samples. For the regression model deployment, we used the values attained directly from DrugComb to each synergy reference model (Bliss, HSA, Loewe, ZIP) as well as CSS. Most synergy reference model values were in similar scales (Loewe = [−116.63, 86.69], ZIP = [−36.08, 66.66], HSA = [−81.75, 64.29], Bliss = [−77.07, 78.65]) (Fig. 2, Figs. W1–W6 of the SYNPRED webserver). CSS stood in the interval [−54.05, 99.84], albeit with larger interquartile distances than the synergy reference models.

Drug molecular descriptors

Each drug included in NCI ALMANAC was analyzed to extract its physicochemical and structural features. A SMILE representation of the drugs was acquired from PubChem [34]. SMILEs were then used to mine molecular descriptors using the Python package "Mordred" (Version 1.1.2) [35]. In total, an array of 1,613 numeric features of 43 different categories was retrieved, making a 2-dimensional molecular description of the drugs. Feature arrays comprising nonnumerical attributes or displaying zero variance were deleted. This preprocessing left 586 features describing each drug included in the NCI ALMANAC, distributed across 28 categories (Table 1). The resulting features were subjected to normalization by removing the mean and scaling to unit variance with scikit-learn's StandardScaler [36].

Omics data of cancer cell lines

Omics data (expression, copy number variation, methylation, global chromatin profiling, metabolomics, microRNA, proteomic profiling) describing the cancer cell lines were acquired via bulk download from the Cancer Cell Line Encyclopedia (CCLE) (https://sites.broadinstitute.org/ccle/) [37]. The number of cell lines included in the CCLE varies depending on the type of omics data available at the time. Correspondence of cell line IDs between the NCI ALMANAC and CCLE was performed according to data available at the Swiss Institute of Bioinformatics Cellosaurus website [38]. According to the affected tissue, annotations acquired through Cellossaurus split the CCLE cell lines into 21 different cancer types. In agreement with the original publications [37, 39], expression data were obtained through RNA sequencing and processed to obtain level expression in transcripts per million by the expectation-maximization algorithm (file: CCLE_RNAseq_rsem_genes_tpm_20180929.txt.gz). Copy number variation (CNV) data were acquired from the Affymetrix SNP6.0 Arrays (file: CCLE_copynumber_byGene_2013–12-03.txt.gz). Copy numbers were normalized by the most similar HapMap normal samples [40]. Segmentation of normalized $\log_2$ (CN/2) ratios was achieved using the circular binary segmentation algorithm [37, 41]. Methylation data were derived by quantifying CpG islands using Reduced Representation Bisulfite Sequencing (file: CCLE_RRBS_tss_CpG_clusters_20181022.txt.gz). Global chromatin profiling was attained using multiple reaction monitoring for 42 combinations of histone marks (file: CCLE_GlobalChromatinProfiling_20181130.csv). Metabolomics data were acquired in parallel with global chromatin profiling by reporting the abundance measures of 225 metabolites (file: CCLE_metabolomics_20190502.csv). MicroRNA associated with cancer dependencies was correlated, regarding 734 microRNAs, with the Achilles gene dependency dataset. Protein profiling was measured with Reverse Phase Protein Arrays for 213 antibodies (file: CCLE_RPPA_20181003.csv) [39].

Dimensionality reduction of omics data

Data were normalized by removing the mean and scaling to unit variance with scikit-learn's StandardScaler [36]. Due to the omics data's high complexity, we performed dimensionality reduction to minimize the noise introduced in the dataset by highlighting the essential features. The datasets already described were used to build and train a multilayer perceptron (MLP) autoencoder, an unsupervised artificial neural network (ANN) with a typical "hourglass" architecture, which is often used to perform dimensionality reduction in vast and high-dimensional datasets such as the ones observed with omics data [42–44]. This type of MLPs usually consists of 3 parts: an encoder that abstracts the input into hidden variables (i.e., a latent-space representation), a bottleneck layer that holds the smallest hidden layer (HL) (for purposes of dimensionality reduction, this is the layer that defines the size of the reduced dataset), and a decoder that reconstructs the original input data from the hidden data [45, 46]. Seven autoencoders, one for each of the CCLE feature blocks, were developed by using Keras with a TensorFlow for graphics processing unit (GPU) (Version 2.3.1) backend [47]. Each of the autoencoders comprised 7 layers, of which 5 were HLs. The input and output layers follow the number of available features in all cell lines, as displayed in Table 2. The number of nodes within the bottleneck layer of each of the 7 autoencoders (used for extraction of the encoded features) corresponds to the autoencoder's final number of features. The 2 HLs in each of the encoder and decoder sections vary in size according to the number of samples and features available (Supplementary Table S1). In this stage, all models used Adam [48] as an optimizer function with a learning rate of 0.001. Rectified linear unit (ReLU) activation function was used in all layers. Mean square error (MSE) was used as a loss function. The models were trained for 1,000, 250, or 100 epochs, depending on the dataset size (Supplementary Table S2). After training, each autoencoder's bottleneck layer was used to perform dimensionality reduction of the omics data according to Table 2.

Principal component analysis (PCA), a commonly used method for dimensionality reduction [49], was also applied in the same datasets as the autoencoder, for which 25 principal components (PCs) were defined. It means that by using PCA, each dataset was transformed to yield only 25 features, totaling 175 features to describe each unique cell line. As shown in Table 2, each feature block from CCLE had its variance explained in a range from 0.89 to 0.99. Since the 7 blocks were used simultaneously for each sample, each cell line is thoroughly described by the components extracted with the PCA. Missing values (in both autoencoder and PCA) were processed by either dropping the sample entirely or replacing the missing values with zero.

Model evaluation and performance metrics

After data acquisition and preprocessing, we gathered all datasets, and to evaluate the results in the most unbiased manner possible, we randomly isolated 3 datasets considering different scenarios:

i) Leave cell out dataset: 3 randomly chosen cell lines belonging to different tissue types (regression dataset: 13,810 combinations; classification dataset: 1,396 synergistic and 429 nonsynergistic samples after processing the 13,810 combinations for full agreement) (for the tissue type classification, see Fig. W1 of the SYNPRED webserver).

ii) Leave drugs out dataset: 5 drugs with the majority belonging to different hierarchical clusters (regression dataset: 25,993 combinations; classification dataset: 2,934 synergistic and

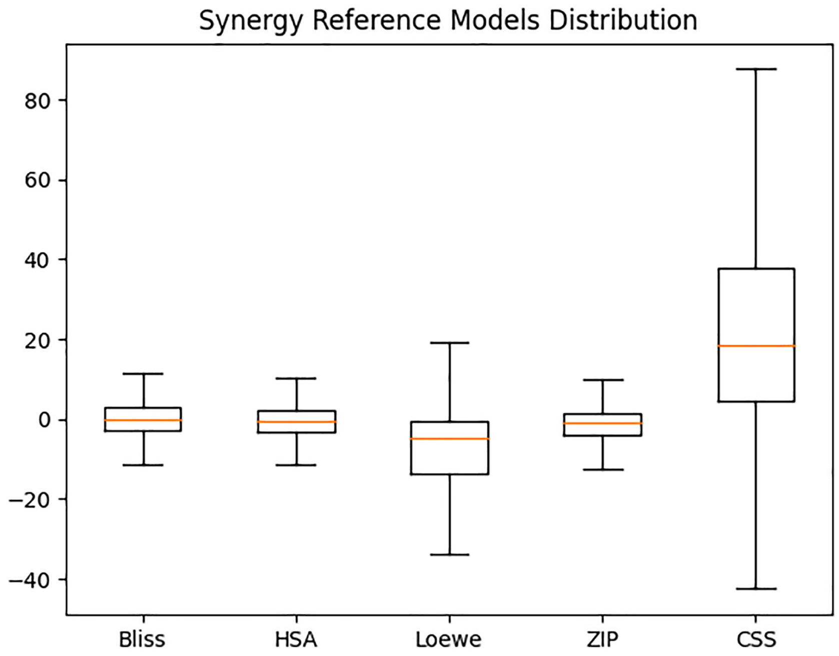


Figure 2: Box plot representing the distribution of synergy scores (y-axis) with respect to the 5 reference models: Bliss, HSA, Loewe, ZIP, and CSS (x-axis). The black boxes represent the difference between the upper 75% and the lower 25% quartiles (interquartile range); the horizontal orange line is the median; the whiskers are the lower and upper values that are not outliers or extremes (not represented as some of these values are off range).

Table 1: Number of features according to the molecular descriptor category of Mordred. Features are categorized as Energetic (E), Pharmacological (P), Structural (S), or Miscellaneous (M—in case of evaluating characteristics of multiple fields).

Number of features per descriptor category					
E	Acidity/Basicity	2	S	Information Content	36
P	ADME	3	S	Molecular Complexity	1
S	Aromatics	2	P	Molecular Operating Environment	51
S	Atom Count	16	S	Molecule Graph	5
S	Atom-Bond Connectivity	2	S	Path Count	21
M	Autocorrelation	180	E	Polarizability	2
S	Bond Count	9	S	Ring Count	66
E	Atomic Orbitals	10	S	Rotatable Bonds	1
S	Chirality	38	S	Topological Charges	21
S	Constitutional	14	S	Topological Index	7
E	Energy State	68	S	Topological Polar Surface Area	2
S	Fragment Complexity	1	S	Walk Counts	21
S	Framework	1	S	Weight	2
S	Hydrogen Bonds	2	M	Wildman–Crippen	2

Table 2: Number of features pertaining to the omics data and the corresponding amount for both the autoencoder and the principal component analysis (PCA) processing

Omics data	Number of available cell lines	Number of available features	Number of features after autoencoder	Number of features after PCA	Explained variance (PCA)
Expression	1,019	57,820	1,156	25	0.89
Copy number variation	1,043	23,316	466		0.91
Methylation	843	56,146	1,122		0.92
Global chromatin profiling	897	42	21		0.99
Metabolomics	928	225	112		0.99
MicroRNA	954	734	73		0.95
Proteomics	899	214	107		0.93

622 nonsynergistic samples after processing the 25,993 combinations for full agreement) (for drug hierarchical clustering, see Fig. W8 of the SYNPRED webserver).

iii)Leave drug combinations out dataset: 5 drug combinations (regression dataset: 360 combinations; classification dataset: 74 synergistic and 6 nonsynergistic samples after processing the 360 combinations for full agreement).

After extracting the datasets for validation, we split the remaining data into training and test sets on an 80/20 ratio (Supplementary Table S3). As such, the training dataset was composed of 195,996 combinations to be used for regression tasks that, upon full agreement processing, yielded 20,291 synergistic and 6,419 nonsynergistic samples for classification tasks. The test set was composed of 48,999 combinations to be used for regression tasks, which, upon full agreement processing, yielded 5,084 synergistic and 1,553 nonsynergistic samples for classification tasks. The described data splitting was performed before any model training, thus ensuring all the prediction models' performance evaluation is deployed on the same data. The binary classification models were evaluated through accuracy (acc), precision (prec), recall (rec), area under the receiver operating characteristic (AUROC), and F1-score as previously described [50]. The regression models were evaluated through the root mean square error (RMSE), mean squared error (MSE), mean absolute error (MAE) [51], Pearson and Spearman correlation coefficients [52].

Development of ML models

Neural networks with Keras

The classification and regression neural networks were fully developed using Keras with a TensorFlow (Version 2.3.1) backend [47]. Weights were updated using the Adam optimizer [48] and a learning rate of 0.0001 along 125 epochs with binary cross-entropy (classification) and MSE (regression) as the loss functions. All the HLs were connected through ReLU activation, while the output layer was subject to sigmoid (classification) or linear activation (regression). As an initial approach, we performed a grid search for parameter optimization using 5% of the training set, fully detailed in the "Parameter optimization" section. The best-performing parameters were further selected, and used to train the models with the complete train dataset.

ML algorithms with scikit-learn

The datasets presented in this work were also trained with the most commonly used algorithms for synergy prediction tasks, namely, random forest (RF) [53], extreme randomized trees (ETC) [50, 54], support vector machines (SVMs) [55], stochastic gradient descent (SGD) [56], k-nearest neighbors (kNNs) [57], and extreme gradient boosting (XGBoost) [58]. The RF, ETC, SVM, SGD, and kNN models were built using the Python package "SciKit Learn" (Version 0.22.1) [36]. The XGBoost model was built using its dedicated package for Python (available at the Python Package Index as "xgboost") [58]. These 6 algorithms were also subject to grid search for parameter optimization using 5% of the training set as described in the "Parameter optimization" section, with the best ones used to train the models with the full dataset.

Parameter optimization

To properly perform parameter optimization in all the algorithms described, a grid search was performed using in-house scripts for Keras DL models and scikit-learn's GridSearchCV with 3-fold cross-validation (for ML algorithms with scikit-learn). We used 5% of the training set [59], a value in agreement with subset usage for parameter optimization [60], since using the full training dataset would exponentially increase an already long task. For each of the Keras classification and regression DL models, we performed grid search with 192 runs with parameters covering the 4 available dimensionality reduction datasets (PCA, PCA_drop, autoencoder, autoencoder_drop), 12 different network architectures, and 4 different dropout rates (0.00, 0.25, 0.50, 0.75) (Supplementary Table S4). In the case of each of the 6 classification and regression ML models trained with scikit-learn, we used 820 runs, including different parameters and dataset combinations (Supplementary Table S5). Finally, for the 6 possible targets (full agreement, Bliss, HSA, Loewe, ZIP, and CSS), we trained each of the 6 ML models with the best corresponding performing parameters. We then assessed the best-performing architectures and dropout rates for the DL-based models. For each of the possible evaluation metrics, we then trained the best-performing parameters, which can lead to a different number of DL-based models depending on the synergy reference model used due to parameter overlap.

Ensemble algorithms

After selecting the previous best-performing models, we replaced the outliers with the average of the remaining prediction values. For some tasks, a few of the individual predictors had notably bad performance (mostly SGD and kNN). As such, we considered outliers the synergy prediction values above or below 10 times the average of the remaining prediction values; this was necessary to allow the ensemble neural networks to converge. These prediction values were used to constitute a new feature representation of the samples that could undergo ensemble model training. The ensemble models were first subjected to a new grid search for parameter optimization (Supplementary Table S6), taking the target probability of the selected algorithms as features, ultimately developing a neural network that worked as an ensemble method. This neural network had a learning rate of 0.0001, trained for 3 epochs, and used the Adam optimizer [48] and binary cross-entropy and MSE for classification and regression, respectively, as the loss functions. All the HLs were connected through ReLU activation, while the output layer was subject to sigmoid or linear activation for classification and regression, respectively. The best-performing ensemble models were trained with the prediction-based feature space.

Feature contribution

To understand what were the top contributors for accurate predictions, we assessed their predictive power. For that, we needed first to break down the process of assessing feature contribution into 2 stages due to the dimensionality reduction of cell lines. First, since the best-performing dimensionality reduction approach was the PCA, we considered the explained variance by each of the features concerning the respective PC. This information was then extracted as an attribute from the PCA object using scikit-learn [36]. Second, we used the eli5 package [61], with Python deployment, to assess the final feature weight by deploying permutation importance [53], a method that allows iterative exclusion of each of the features, to assess its contribution to the predictive model. The permutation importance was deployed on the test set because it would not be possible to assess the feature contribution under unbiased conditions if the training set had been used. However, it is worth noting that this evaluation occurs after all model training; hence, it does not influence the test results.

Benchmark

Benchmarking synergy prediction protocols is a very complicated process. As reviewed by Zagidullin et al. [29], the datasets available completely differ in the amount of information used, with DrugComb [29] assembling the most important ones (ALMANAC [30], ONEIL [62], FORCINA [63], CLOUD [64]). As shown by Kumar and Dogra [65], most authors used NCI ALMANAC data to train and the Loewe additivity synergy reference model [14, 16–18, 20, 21]. Furthermore, comparison to the available methodologies implies that authors adapt the published proposed DL architectures as these are not easily applied or not available in GitHub or similar platforms (e.g., pruning the data due to unavailability of a certain data modality, or changing the loss function to turn a model into a regressor).

As such, we followed a multistep approach to benchmark our pipeline:

i) Comparison of DL architectures and simpler ML algorithms (RF, ETC, SVM, SGD, kNN, and XGBoost models) with ensemble approaches in 4 different test scenarios.

ii) DeepSynergy [18] architecture implementation and comparison using our independent test set and validation sets as this is one of the most common approaches. As described in the original study, we retrained a model using 2 hidden layers, the first with 8,192 and the second with 4,096 neurons. Furthermore, 2 dropout layers were added, the first with a 0.2 rate and the second with a 0.5 rate. The activation function used between the hidden layers was a hyperbolic tangent, and on the output layer, linear activation was used. This DeepSynergy implementation was trained over 250 epochs with a learning rate of 0.00001 and an Adam optimizer.

iii)Comparison with published methods for synergy calculations using both regression (12 models) and classification (13 models) approaches as reviewed by Kumar and Dogra [65].

iv)Comparison of our regression approaches to algorithms for which the training dataset was clearly available to make sure the comparison would be as fair as possible. As such, we compared to the Matchmakers' algorithm [20] using the adapted DrugCombo (retrieved from Matchmakers' [20] GitHub) and NCI ALMANAC complete datasets, which, in turn, enables us also to compare with DeepSynergy [18] and TreeCombo [12] as these were also evaluated by the authors [20]. Upon the data considered, we performed our own feature extraction, as described in the SynPred pipeline. Thus, the comparison is now possible between the full methods, of which the feature extraction is a part, enabling us to compare with the values reported by the authors.

Web-based application interface implementation

The SYNPRED prediction models were implemented in a web-based application at http://www.moreiralab.com/resources/synpred/. The website's plots and front-end were constructed with plotly [66] and Flask [67], both freely available Python packages, on a framework that uses an in-house adaptation of Javascript, CSS, and HTML scripts. All the back-end hosting was mediated with Flask [67].

Results and Discussion

Measuring feature importance for model development

To understand the importance of each group of included features for the final model performance and to attain a more interpretable model, we analyzed each of the individual models with permutation importance. We perceived that more complex models, particularly DL-based models with different architectures, tend to make more extensive use of the omics-based features to over 70% of the total feature contribution (Figs. W9–W12 of the SYNPRED webserver). Contrarily, simpler models, such as kNN and SGD, made almost exclusive use of the drug features (above 90%) (Figs. W16 and W18 of the SYNPRED webserver). Other non-DL-based models made variable (between 20% and 80%) usage of the omics features (Figs. W13–W15 and W17 of the SYNPRED webserver). This observation highlights the importance of DL models to take full advantage of omics data for capturing the complexity of each cancer profile, thus improving drug pair–cell line combinations predictions. The advantages of using these algorithms when dealing with multidimensional omics data, particularly the great flexibility of DL architectures, were also previously emphasized [68].

We then looked for a possible biological relevance of the top 5 genes in each group of the most critical multiomics features to understand if genes contributing more to the prediction models were also implicated in tumorigenesis. Of the 15 ranked genes from expression, methylation, and CNV variations, all of them are used as prognostic cancer markers or have a role in tumor progression and treatment (Table 3). These data suggest that our models, especially DNNs, are likely to capture the most relevant information for each group of multiomics features for synergistic drug combinations. The remaining ranked genes organized by each ML model's best-contributing features are presented in interactive Sankey diagrams on the website landing page (Figs. W9–W18).

Tuning and choosing the best ML parameters

An appropriate choice of the best model parameters should always be performed, as ML performance and training time are deeply affected by them. With that in mind, we used a grid search approach to test a comprehensive array of parameters and dataset combinations, including parameters for several ML methods, a comprehensive set of DL configurations, and preprocessing setups, as described above. Regarding the preprocessing datasets, autoencoder datasets performed worse in the training sets and slightly worse for the test set. These results led us to discard them as there was no benefit to the increased training time caused by the significantly higher dimensionality. We proceed with the dataset in which PCA was used for dimensionality reduction and replacing the missing values with 0, as these approaches performed better for most grid search runs [80, 81] (Supplementary Table S3).

SYNPRED models for drug combination prediction

After selecting the best parameters for both DL with Keras and ML with scikit-learn, we trained models with the full training set according to the parameters in the best grid search performing metrics. The best individual models were used to attain each sample prediction to make the final ensemble for the 5-synergy reference model plus the full agreement. The final models were then evaluated in the test set and 3 different scenarios: leave cell out, leave drugs out, and leave drug combinations out, by attaining different classification (Supplementary Table S7) or regression (Supplementary Tables S8–S12) evaluation metrics.

Classification model performance. Prior to ensemble development, the best independent performing model was XGBoost with the following parameters: alpha = 0.25, max_depth = 6, n_estimators = 100. After ensemble, our final full-agreement SYNPRED comprised 4 DL-based and 6 ML-based models, attained with a DL architecture with 3 hidden layers of size 100 and a dropout rate of 0.60. When applied in an independent test set, our ensemble model displayed better performance (accuracy = 0.85, precision = 0.91, recall = 0.90, AUROC = 0.80, and F1-score = 0.90) than any other classic ML or DL models, including reference ones such as SVM, RF, or XGBoost frequently used for synergy prediction classification tasks (Table 4, Supplementary Table S7) [13, 14, 82]. In the 3 independent scenarios, the full-agreement ensemble SYNPRED achieved higher precision values by returning the most relevant results than any other of the individual models. However, we saw a significant drop in the leave cells, drugs, and drug combinations out datasets.

Regression model performance. Concerning the 5 regression tasks (Table 5), CSS (Supplementary Table S12) stands out—in either the metrics or the datasets considered—while the remaining 4 (ZIP, HSA, Bliss, and Loewe) (Supplementary Tables S8–S11) followed closely behind. Although in agreement with the presented data, this is unexpected considering the literature on the subject, which mainly uses Loewe. Indeed, historically, Loewe has been systematically chosen as the target regression reference model [14, 16–18, 20, 21]. For most cases in which this happens, there is no comparison with the remaining reference models. The few available comparative studies are mainly done outside the synergy prediction spectrum and somewhat under the scope of analyzing provided drug combination dose–response matrix data [33, 83]. By deploy-

Table 3: Permutation importance of the top 5 proteins associated with expression, methylation, and CNV features as well as their associated biological relevance

Type of feature	Protein name	Protein description	Biological relevance[a]
Expression	TMSB4X	Thymosin beta-4 X-linked	Prognostic marker in renal cancer (unfavorable)
	MTCO2	Mitochondrially encoded cytochrome c oxidase II	Prognostic marker in liver cancer (favorable) and pancreatic cancer (favorable)
	MT-RNR2	Mitochondrially encoded 16S rRNA	Associated with survival outcomes in patients with cancer [69]
	MT-CO3	Mitochondrially encoded cytochrome c oxidase III	Prognostic marker in pancreatic cancer (favorable) and liver cancer (favorable)
	COX6C	Cytochrome c oxidase subunit 6C	Associated with breast cancer, thyroid tumors, uterine cancer, prostate cancer, and esophageal cancer [70], although not reported as prognostic
Methylation	C11ORF52	Chromosome 11 open reading frame 52	Associated with lung cancer [71], although not reported as prognostic
	NPY1R	Neuropeptide Y receptor Y1	Prognostic marker in breast cancer (favorable)
	TMBIM6	Transmembrane BAX inhibitor motif containing 6	Prognostic marker in renal cancer (favorable), head and neck cancer (unfavorable), and breast cancer (unfavorable)
	C2CD4D	C2 calcium-dependent domain containing 4D	C2CD4D-AS1 overexpression contributes to the malignant phenotype of lung adenocarcinoma cells [72], although not reported as prognostic
	EDNRB	Endothelin receptor type B	Prognostic marker in renal cancer (favorable)
CNV	UTY	Ubiquitously transcribed tetratricopeptide repeat containing, Y-linked	Associated with cutaneous melanoma, bladder urothelial carcinoma, B-cell lymphoma, small cell lung cancer, oligodendroglioma, chondroblastic osteosarcoma, and cutaneous melanoma [73, 74], although not reported as prognostic
	MACROD2	Mono-ADP ribosylhydrolase 2	Associated with growth of intestinal tumors [75], although not reported as prognostic
	WWOX	WW domain containing oxidoreductase	Prognostic marker in renal cancer (favorable) and breast cancer (unfavorable)
	DAZ2	Deleted in azoospermia 2	Associated with oligozoospermia [76], which is, in turn, highly associated with testicular cancer [77], although not reported as prognostic
	KANK1	KN motif and ankyrin repeat domains 1	Upregulating Kank1 gene inhibits human gastric and lung cancer progress [78, 79], although not reported as prognostic

[a] The protein description and biological importance were retrieved from the Human Proteins Atlas (https://www.proteinatlas.org/) and the Human Gene Database (https://www.genecards.org/). When this information was not listed in these databases, we presented the study that supports the biological relevance. Favorable and unfavorable are related to gene/protein contribution for cancer progression.

Table 4: Best results obtained for the classification ensemble model

Subset used for evaluation[a]	Accuracy	Precision	Recall	AUROC	F1 score
Test	0.85	0.91	0.90	0.80	0.90
Leave cells out	0.37	0.89	0.13	0.55	0.22
Leave drugs out	0.33	0.86	0.13	0.53	0.22
Leave drug combinations out	0.24	1.00	0.21	0.61	0.35

[a] The final model had a dropout rate of [0.4] and an architecture of [10, 10, 10].

Table 5: Best results obtained for the regression ensemble models, considering the test dataset

Synergy reference model	RMSE	MSE	Pearson	MAE	Spearman
CSS	11.07	122.61	0.86	7.43	0.87
Loewe	10.58	111.92	0.71	6.49	0.68
Bliss	4.35	18.92	0.71	3.07	0.59
HSA	4.09	16.70	0.73	2.86	0.64
ZIP	3.86	14.87	0.70	2.74	0.66

ing an unbiased data-driven selection of the model, SYNPRED empirically assesses how realistically viable is the representation of 5 of the most common synergy reference models against a real biological dataset. Zagidullin et al. [29] have already pointed to the value of such agglomerative approaches.

The results of our best final ensemble regression model (CSS) outperformed all the individual predictors when evaluated in the test dataset and leave drug combinations out scenario, one of the most challenging ones (Table 5). Regarding correlation metrics and comparing to the literature standards [84], CSS achieved strong Pearson values (0.86 on the test and 0.74 on the leave cells out dataset). Concerning scale-depending performance metrics, the CSS had 11.07 and 13.63 RMSE on the test and leave cells out datasets, respectively. Considering that CSS values range within [−54.05, 99.84], our predictor was able to determine CSS synergy values with low error (Fig. 3). A similar pattern was exhibited by the Loewe ensemble predictor (Fig. 4).

Benchmark

We benchmarked our pipeline following a multistep approach as described in the Methods section:

i) Comparison of the best-performing individual DL and ML algorithms with the ensemble approaches for each prediction task—Supplementary Tables S7 to S12

ii) DeepSynergy [18] architecture implementation and comparison using our independent test set and validation sets—Supplementary Table S13

iii)Comparison with published methods for synergy calculations as reviewed by Kumar and Dogra [65]—Supplementary Table S14

iv)Comparison of our regression approaches to Matchmakers' algorithm [20], DeepSynergy [18], and TreeCombo [12]—Supplementary Tables S15 and S16

Regarding (i), ensemble/aggregation of algorithms consistently outperforms or stands very close to the best individual predictors. XGBoost and extreme randomized trees were typically the second-best predictors. These results showcase how SynPred leverages previous information on algorithms such as TreeCombo [12] (which uses an individual XGBoost algorithm) or DeepSynergy [18], which is, in essence, the literature parent of several of the neural networks with conic architecture we used. In fact, in (ii) (Supplementary Table S13), it can be seen that the DeepSynergy [18] implementation on SynPred's pipeline behaves similarly to other DNN approaches in SynPred. These are good performers but unable to beat the ensemble algorithms.

When comparing the reported performance for algorithms in their own settings (iii), as reviewed by Kumar and Dogra [65], once again we need to take into account a very broad array of circumstances, such as algorithms, datasets (filtered or postprocessed), and synergy reference models (Supplementary Table S14). The high possible combination of factors that leads to the final methods' performance is huge, and therefore this comparison has to be conducted with a limited few.

For instance, SynPred's highest performer predictor is the CSS predictor. However, it is impossible to justly compare our results to predictors that only focus on the Loewe synergy reference model. However, when considering the most recurring synergy reference model (Loewe), although SynPred shows lower Pearson and Spearman correlations, it also presents much lower errors (RMSE and MSE) compared to the best remaining algorithms. All these results highlight the need to consider different synergy reference models, which although not used before, were already suggested to be a valuable approach [29].

Finally (iv), we conducted closer comparisons (although still not optimal) with performances presented in Supplementary Table S15 and Supplementary Table S16. Regarding Supplementary Table S15, SynPred was run against Matchmakers' [20] processing of DrugComb [29]. Upon doing this, both CSS and Loewe predictors from SynPred stood very close to the performance of Matchmakers [20], which is remarkable since this was the dataset used by the authors [20] to train the model. When inspecting Supplementary Table S16, in which the predictors were deployed upon NCI ALMANAC [85] (the dataset used in this study), SynPred stands out in all the synergy reference models with Pearson and Spearman correlation performance increments between 30.51% and 42.37%, as well as between 36.36% and 56.36%, respectively. Although MSE metrics are particularly hard to compare between datasets and methods, significant improvements were also observed.

Web-based application description

The classification and regression models for predicting the type of combinatory effect in drug pair–cell line samples are available as a web-based application at http://www.moreiralab.com/resources/synpred/. All the 11 described single models are deployed on user submission, as well as the ensemble approach. The user needs to submit 2 drugs as input in the ∗.smile format and selects from a drop-down menu, the primary body site corresponding to the tested cancer cell lines. The drugs are then subject to feature extraction by Mordred and a standard preprocessing (feature elimination and normalization) as thoroughly described in the Methods section. The output, displayed in a downloadable heatmap, is the drug combination prediction effect for each of the individual cell lines calculated with the ensemble classification and regression models and using 5 synergy reference models (ZIP, HSA, Bliss, Loewe, CSS) plus the full-agreement metric. Furthermore, the final tally of synergistic queries predicted by all models based on the prediction values is also displayed in the last column ("Synergy Votes"). This additional option facilitates the visualization of the type of combinatory effect between the 2 drugs and aims at strengthening the value of the prediction due to the lack of consensus between the different synergy reference models. The results are returned to the provided e-mail and displayed on the submission webpage (as shown in Fig. 5). Additionally, users can assess, explore, and visualize through different plots as well as export a summary of the synergy scores (calculated using ZIP, Bliss, HSA, Loewe, and CSS synergy reference models) by cell line used to develop the original dataset of SYNPRED. To our knowledge, this is the first webserver that can predict new drug synergy combinations without the need of uploading a partial or full dose–response matrix. This feature is an advantage compared with other models implemented in webservers that need these types of data for drug combination response prediction [28, 29].

Conclusions

Synergistic anticancer drug combinations are a powerful tool to help tackle cancer drug resistance since they can simultaneously target multiple key molecules or pathways. The rational design of combination therapies is warranted to improve the efficacy, although this is a well-known time-consuming and expensive task. In recent years, ML algorithms' applicability for drug repurposing or novel drug design has been essential to demonstrate the importance of *in silico* methodologies to help overcome this problem. Some classification [13, 14, 17] and regression [16, 18–22] models using ML and omics data for predicting drug synergy combinations were already developed. However, the fittingness of the previously developed algorithms is sometimes hindered by using a single reference model (e.g., Bliss, Loewe, HSA, ZIP, or CSS) or by the difficulty in applying these models to new unseen data, since these are not straightforward to implement and require advanced bioinformatics skills. Our study leads to an innovative approach by highlighting the importance of choosing an appropriate synergy reference model, and explores how this choice influences the final predictor performance. Given the different sensitivity ob-

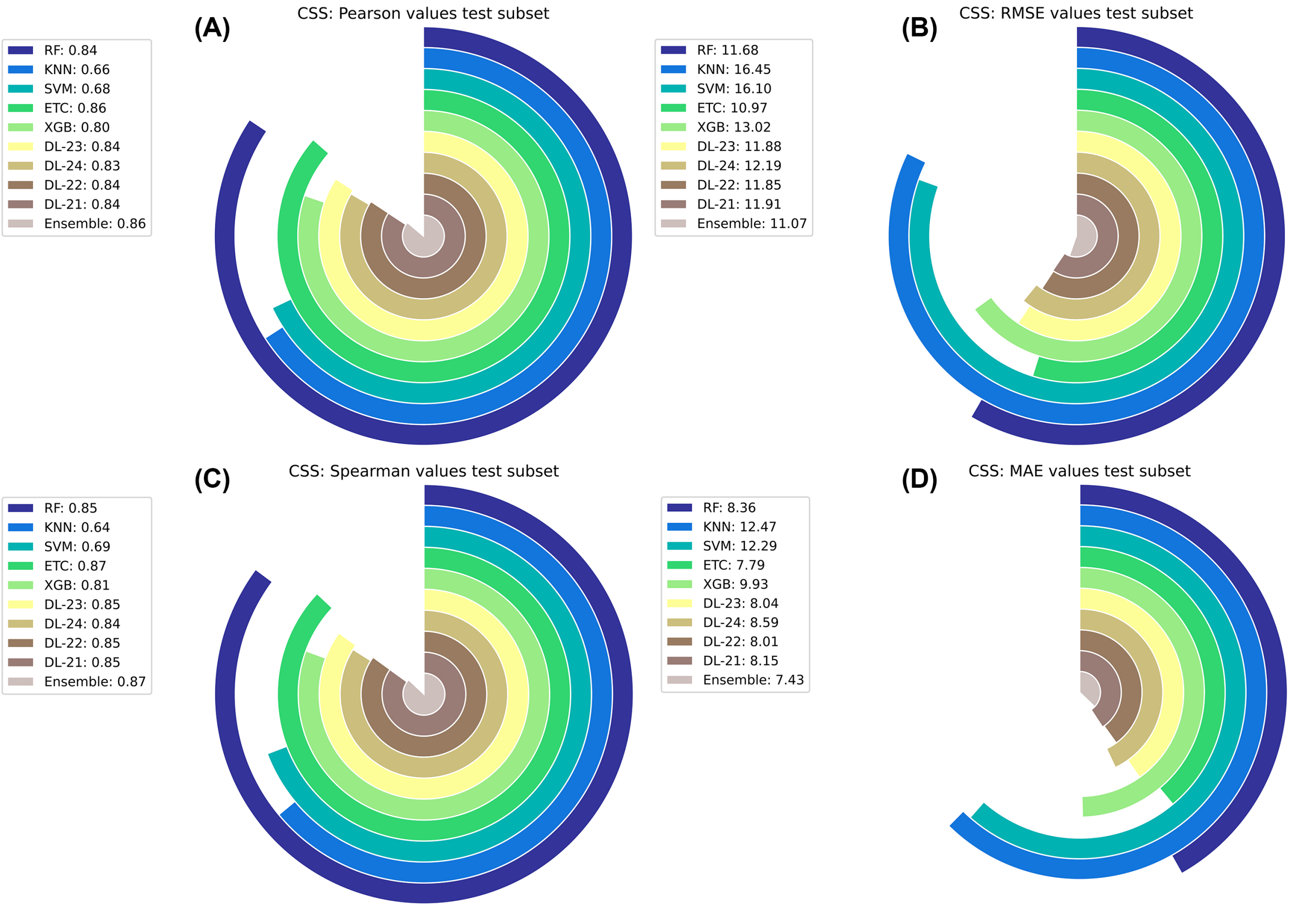


Figure 3: Circular bar plot representing the model's evaluation metrics for the CSS synergy reference model. (A) Model performance Pearson values evaluated in the test dataset. (B) Model performance RMSE values evaluated in the test dataset. (C) Model performance Spearman values evaluated in the test dataset. (D) Model performance MAE values evaluated in test dataset.

served between these reference models in evaluating the degree of combination, a more comprehensive and rigorous approach that leverages all metrics to predict drug synergy is an asset.

This study introduced a new synergy prediction model, SYNPRED, that combines comprehensive multiomics data of cancer cell lines with physicochemical and structural features of drugs. This work is one of the first that takes 5 different synergy reference models (Bliss, HSA, Loewe, ZIP, and CSS) and uses one of the most comprehensive and balanced databases regarding the synergistic–nonsynergistic distribution, the NCI ALMANAC. Our top-ranked classification and regression models, an ensemble developed with the best machine learning models, achieved state-of-the-art performance to predict synergistic drug combinations in an independent dataset. The best-performing prediction model in SYNPRED is, undoubtedly, CSS (RMSE, 11.07; MSE, 122.61; Pearson, 0.86; MAE, 7.43; Spearman, 0.87). However, we advise the users to considers the aggregate of results, albeit with a higher focus on CSS. We included a "Voting classifier" output that tallies the results of the 6 predictors to aid the user's interpretation of the results. If more than 5 predictors yield a positive result, the submission sample is likely to be synergistic, while if it is only 1 or lower, it is likely to be nonsynergistic. Besides, we provide the complete workflow for a standalone deployment in our GitHub coupled with a freely available and easy-to-use webserver (http://www.moreiralab.com/resources/synpred/) that requires only 2 drugs' SMILEs as inputs, thus alleviating the need for uploading a conventional and laborious dose–response matrix. SYNPRED can be a valuable tool to the scientific and medical community for drug repurposing or *in silico* discovery of new anticancer drug combinations.

Additionally, given the importance of multiomics data in cell line classification and therapy response, we combined all the available multiomics features in the CCLE database to explore their contribution to model development. The knowledge mined from this analysis demonstrates the capacity of different ML models to deal with multiomics data, with DL algorithms being much more able to learn and leverage this complex type of features. We found that the most ranked proteins in each of the most contributing multiomics features are important cancer biomarkers or have a role in tumorigenesis, demonstrating DNN models' capacity to capture their significance and use this information for the final model development. In the future, we expect to include protein–protein interactions data and network analysis to improve the model performance, aiming to identify drug combinations with potential new targets across different cell lines.

Availability of Supporting Source Code and Requirements

Project name: SYNPRED
Project homepage: https://github.com/MoreiraLAB/synpred
Operating system(s): Linux, Mac OS X, Windows

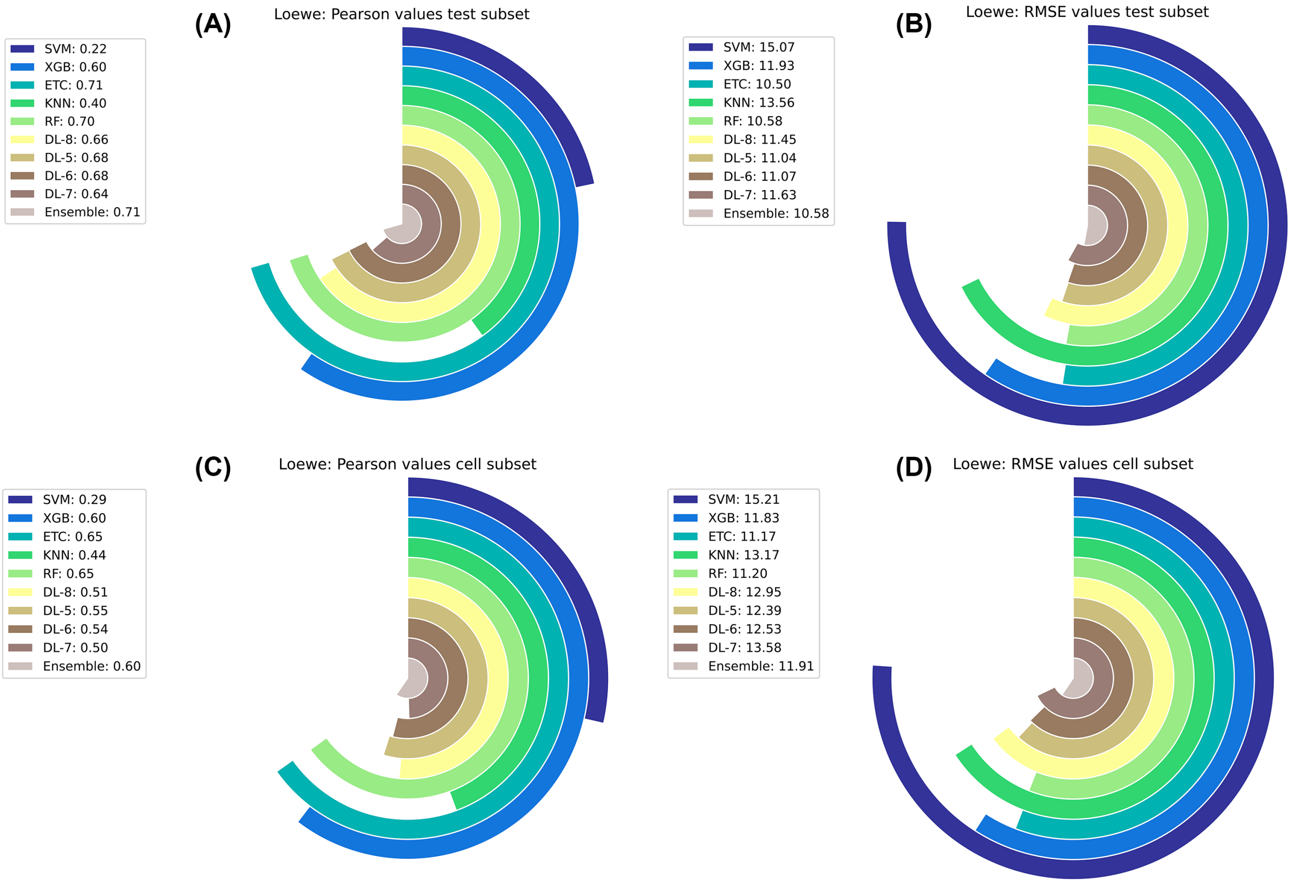
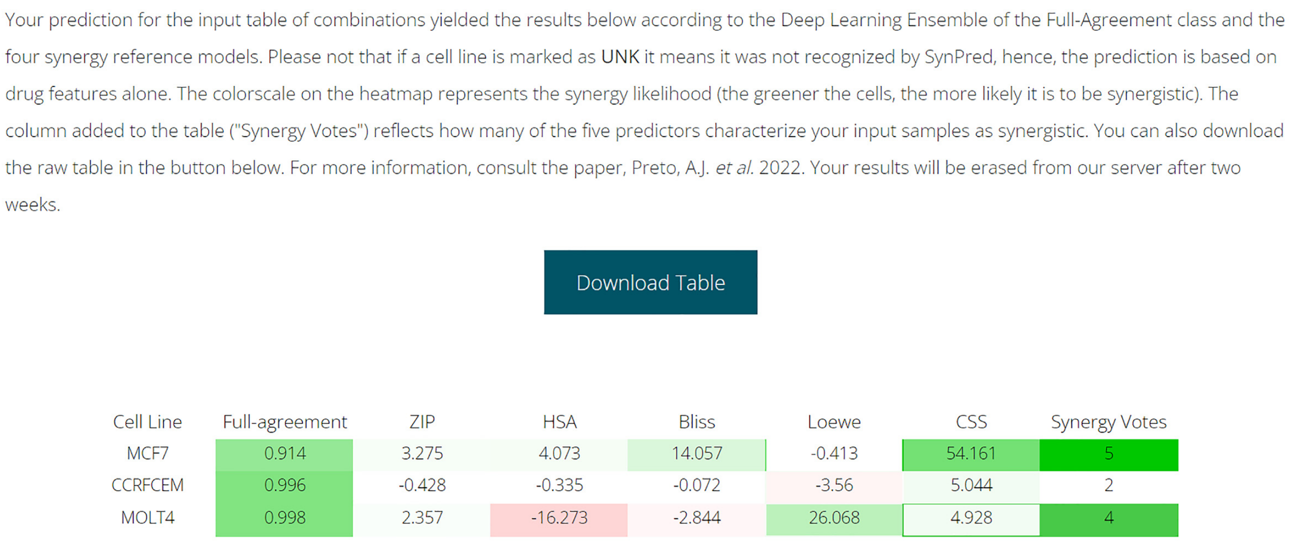


Figure 4: Circular bar plot representing the model's evaluation metrics for the Loewe synergy reference model. (A) Model performance Pearson values evaluated in the test dataset. (B) Model performance RMSE values evaluated in the test dataset. (C) Model performance Pearson values evaluated in the leave cells out dataset. (D) Model performance RMSE values evaluated in the leave cells out dataset.



Your prediction for the input table of combinations yielded the results below according to the Deep Learning Ensemble of the Full-Agreement class and the four synergy reference models. Please not that if a cell line is marked as UNK it means it was not recognized by SynPred, hence, the prediction is based on drug features alone. The colorscale on the heatmap represents the synergy likelihood (the greener the cells, the more likely it is to be synergistic). The column added to the table ("Synergy Votes") reflects how many of the five predictors characterize your input samples as synergistic. You can also download the raw table in the button below. For more information, consult the paper, Preto, A.J. *et al.* 2022. Your results will be erased from our server after two weeks.

Cell Line	Full-agreement	ZIP	HSA	Bliss	Loewe	CSS	Synergy Votes
MCF7	0.914	3.275	4.073	14.057	-0.413	54.161	5
CCRFCEM	0.996	-0.428	-0.335	-0.072	-3.56	5.044	2
MOLT4	0.998	2.357	-16.273	-2.844	26.068	4.928	4

Figure 5: Example of the SYNPRED output prediction. Green colored cells represent a synergistic prediction, while red colored cells represent the nonsynergistic ones.

Programming language: Python and R
Other requirements: Python 3.8.2 or higher, R 3.6.3 or higher
License: GPL-3.0
Biotools: Synpred
RRID: SCR_022693

Data Availability

SYNPRED is a free, open-source, web-based application available at http://www.moreiralab.com/resources/synpred/ without any login or registration requirements. The source code of the web-based application implementation is deposited in the GitHub repository (https://github.com/MoreiraLAB/synpred) to allow the stand-alone use of the application and further integration and comparison with other models. The code is fully developed in Python and R languages; hence, it can be deployed fully without charge. The multiomics data included in this study are available at the corresponding references mentioned in the main text. Supporting data and an archival copy of the code are also available via the GigaScience database GigaDB [89].

Additional Files

Supplementary Table S1. Conditions for dimensionality reduction with autoencoders. Hidden and bottleneck layers definition according to the number of features.
Supplementary Table S2. Conditions for dimensionality reduction with autoencoders. Number of epochs of the autoencoder training according to either the number of samples or number of features.
Supplementary Table S3. Final datasets to be subjected to training.
Supplementary Table S4. Grid search combination parameters using 5% on the training set with deep learning algorithms.
Supplementary Table S5. Grid search combination parameters using 5% on the training set with non–deep learning algorithms.
Supplementary Table S6. Grid search combination parameters of the ensemble neural network.
Supplementary Table S7. Final metrics of the classification models evaluated in an independent test set and 3 different scenarios (leave cell out, leave drugs out, and leave drug combinations out) using full-agreement synergy values.
Supplementary Table S8. Final metrics of the regression models evaluated in an independent test set and 3 different scenarios (leave cell out, leave drugs out, and leave drug combinations out) using the Bliss synergy reference model.
Supplementary Table S9. Final metrics of the regression models evaluated in an independent test set and 3 different scenarios (leave cell out, leave drugs out, and leave drug combinations out) using the HSA synergy reference model.
Supplementary Table S10. Final metrics of the regression models evaluated in an independent test set and 3 different scenarios (leave cell out, leave drugs out, and leave drug combinations out) using the Loewe synergy reference model.
Supplementary Table S11. Final metrics of the regression models evaluated in an independent test set and 3 different scenarios (leave cell out, leave drugs out, and leave drug combinations out) using the ZIP synergy reference model.
Supplementary Table S12. Final metrics of the regression models evaluated in an independent test set and 3 different scenarios (leave cell out, leave drugs out, and leave drug combinations out) using the CSS synergy reference model.
Supplementary Table S13. DeepSynergy [86] reimplementation on the dataset that yielded the best results for SynPred (with PCA preprocessing and missing values replacement with 0), against the synergy reference model the original work targeted—Loewe.
Supplementary Table S14. Comparison of final metrics of the classification and regression models of SynPred to the methods reviewed by Kumar and Dogra [65].
Supplementary Table S15. Comparison of the performance of SynPred and other recent algorithms, according to their respective reporting metrics upon deployment in DrugCombo [87].
Supplementary Table S16. Comparison of the performance of SynPred and other recent algorithms, according to their respective reporting metrics upon deployment in NCI ALMANAC [88].

Abbreviations

ACC: accuracy; AI: artificial intelligence; ANN: artificial neural network; AUROC: area under the receiver operating curve; CCLE: Cancer Cell Line Encyclopaedia; CNV: copy number variation; DL: deep learning; DNN: deep neural network; ENS: ensemble; ETC: extreme randomized trees; F1: F1-score; GPU: graphics processing unit; HL: hidden layer; HSA: highest single agent; kNN: k-nearest neighbor; MAE: mean absolute error; miRNA: microRNA; ML: machine learning; MLP: multilayer perceptron; MSE: mean square error; PC: principal component; PCA: principal component analysis; PREC: precision; REC: recall; ReLU: rectified linear unit; RF: random forest; RMSE: root mean square deviation; SGD: stochastic gradient descent; SMILE: simplified molecular-input line-entry System; SVM: support vector machine; SYNPRED: SYNergy PREDiction; XGBoost: extreme gradient boosting; ZIP: zero interaction potency.

Funding

This work was supported by the European Regional Development Fund through the COMPETE 2020–Operational Programme for Competitiveness and Internationalisation and Portuguese national funds via Fundação para a Ciência e a Tecnologia (FCT) (LA/P/0058/2020, UIDB/04539/2020, UIDP/04539/2020, POCI-01–0145-FEDER-031356, and DSAIPA/DS/0118/2020). FCT also supported A.J.P. with a PhD scholarship (SFRH/BD/144966/2019). J.M. is supported by a Individual Junior Postdoctoral Research Contract from FCT Scientific Employment Stimulus (2021.03416.CEECIND). Funding for open-access charge: Fundação para a Ciência e a Tecnologia (DSAIPA/DS/0118/2020).

Competing Interests

The authors declare that they have no competing interests.

Authors’ contributions

A.J.P., methodology; software; validation; formal analysis; investigation; resources; writing—review & editing; visualization. P.M.-F., methodology; software; investigation; resources; data curation; writing—original draft preparation. J.M., conceptualization; methodology; formal analysis; data curation; writing—original draft preparation; writing—review & editing; supervision; project administration. I.S.M., conceptualization; writing—review & editing; visualization; supervision; project administration; funding acquisition.

Acknowledgments

The authors acknowledge STRATAGEM—New diagnostic and therapeutic tools against multidrug-resistant tumors, CA17104.

References

1. IARC - Internation Agency for Research on Cancer, Estimated number of deaths and new cases tools from 2020 to 2040. GLOBOCAN - Cancer Tomorrow via Global Cancer Observatory. 2022.

2. Vasan, N, Baselga, J, Hyman, DM. A view on drug resistance in cancer. *Nature* 2019;**575**(7782):299–309.
3. Chatterjee, N, Bivona, TG. Polytherapy and targeted cancer drug resistance. *Trends Cancer* 2019;**5**(3):170–82.
4. Piochi, LF, AT, Gaspar,Rosário-Ferreira, N, *et al.* Single-omics to interactomics: how can ligand-induced perturbations modulate single-cell phenotypes? 2022 *Advances in Protein Chemistry and Structural Biology*. Rossen Donev.Academic Press.Swansea.
5. Roell, KR, Reif, DM, Motsinger-Reif, AA. An introduction to terminology and methodology of chemical synergy—perspectives from across disciplines. *Front Pharmacol* 2017;**8**(158).https://doi.org/10.3389/fphar.2017.00158
6. Brandão, M, Pondé, NF, Poggio, F, *et al.* Combination therapies for the treatment of HER2-positive breast cancer: current and future prospects. *Expert Rev Anticancer Ther* 2018;**18**(7):629–49.
7. Westerweel, PE, Te Boekhorst, PAW, Levin, M-D, *et al.* New approaches and treatment combinations for the management of chronic myeloid leukemia. *Front Oncol* 2019;**9**(665).https://doi.org/10.3389/fonc.2019.00665
8. Xu, J, Qiu, Y. Current opinion and mechanistic interpretation of combination therapy for castration-resistant prostate cancer. *Asian J Androl* 2019;**21**(3):270.
9. Ribas, A, Lawrence, D, Atkinson, V, *et al.* Combined BRAF and MEK inhibition with PD-1 blockade immunotherapy in BRAF-mutant melanoma. *Nat Med* 2019;**25**(6):936–40.
10. Wang, Z, Deisboeck, TS. Dynamic targeting in cancer treatment. *Front Physiol* 2019;**10**(96).https://doi.org/10.3389/fphys.2019.00096
11. Wang, Z, Li, H, Guan, Y. Machine learning for cancer drug combination. *Clin Pharmacol Ther* 2020;**107**(4):749–52.
12. Janizek, JD, Celik, S, Lee, S-I. Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine. 2018;*bioRxiv* 10.1101/331769v1.
13. Li, H, Li, T, Quang, D, *et al.* Network propagation predicts drug synergy in cancers. *Cancer Res* 2018;**78**(18):5446–57.
14. Celebi, R, Bear Don't Walk, O, Movva, R, *et al.* In-silico prediction of synergistic anti-cancer drug combinations using multi-omics data. *Sci Rep* 2019;**9**(1):8949.
15. Malyutina, A, Majumder, MM, Wang, W, *et al.* Drug combination sensitivity scoring facilitates the discovery of synergistic and efficacious drug combinations in cancer. *PLoS Comput Biol* 2019;**15**(5):e1006752.
16. Zhang, T, Zhang, L, Payne, PRO, *et al.* Synergistic drug combination prediction by integrating multiomics data in deep learning models. In: J Markowitz, editor. *Translational Bioinformatics for Therapeutic Development* 2021; New York, NY: Springer US;
17. Wang, J, Liu, X, Shen, S, *et al.* DeepDDS: deep graph neural network with attention mechanism to predict synergistic drug combinations. *Briefings Bioinf* 2022;**23**(1).
18. Preuer, K, Lewis, RPI, Hochreiter, S, *et al.* Predicting anti-cancer drug synergy with deep learning. *Bioinformatics* 2018;**34**(9):1538–46.
19. Zhang, H, Feng, J, Zeng, A, *et al.* Predicting tumor cell response to synergistic drug combinations using a novel simplified deep learning model. *AMIA Annu Symp Proc* 2021. **2020**:1364–1372.
20. Kuru, HI, Tastan, O, Cicek, AE. Matchmakers: a deep learning framework for drug synergy prediction. *IEEE/ACM Trans Comput Biol Bioinf* 2021;**19**(4):2334–44.
21. Liu, Q,Xie L. TranSynergy: mechanism-driven interpretable deep neural network for the synergistic prediction and pathway deconvolution of drug combinations. *PLoS Comput Biol* 2021;**17**(2):e1008653.
22. Xia, F, Shukla, M, Brettin, T, *et al.* Predicting tumor cell line response to drug pairs with deep learning. *BMC Bioinf* 2018;**19**(S18):71–79.
23. Bliss, CI. The toxicity of poisons applied jointly. *Ann Appl Biol* 1939;**26**(3):585–615.
24. Foucquier, J, Guedj, M. Analysis of drug combinations: current methodological landscape. *Pharmacol Res Perspectives*. 2015;**3**(3):e00149.
25. Loewe, S, Muischnek, H. Über Kombinationswirkungen. *Arch Exp Pathol Pharmakol* 1926;**114**(5–6):313–26.
26. Chou, T-C. Drug combination studies and their synergy quantification using the Chou-Talalay method. *Cancer Res* 2010;**70**(2):440–6.
27. Yadav, B, Wennerberg, K, Aittokallio, T, *et al.* Searching for drug synergy in complex dose-response landscapes using an interaction potency model. *Comput Structural Biotechnol J* 2015;**13**:504–13.
28. Ianevski, A, Giri, AK, Gautam, P, *et al.* Prediction of drug combination effects with a minimal set of experiments. *Nat Machine Intelligence* 2019;**1**(12):568–77.
29. Zagidullin, B, Aldahdooh, J, Zheng, S, *et al.* DrugComb: an integrative cancer drug combination data portal. *Nucleic Acids Res* 2019;**47**(W1):W43–51.
30. Holbeck, SL, Camalier, R, Crowell, JA, *et al.* The National Cancer Institute ALMANAC: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity. *Cancer Res* 2017;**77**(13):3564–76.
31. Shoemaker, RH. The NCI60 human tumour cell line anticancer drug screen. *Nat Rev Cancer* 2006;**6**(10):813–23.
32. National Cancer Institute. DCTD tumor repository—a catalog of in vitro cell lines, transplantable animal and human tumors and yeast. 2020. National Cancer Institute at Frederick,Maryland.
33. Zheng, S, Wang, W, Aldahdooh, J, *et al.* SynergyFinder Plus: towards a better interpretation and annotation of drug combination screening datasets. *Genomics, Proteomics & Bioinformatics* 2022;**S1672-0229**(22):00008–0.
34. Kim, S, Chen, J, Cheng, T, *et al.* PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res* 2019;**47**(D1):D1102–9.
35. Moriwaki, H, Tian, Y-S, Kawashita, N, *et al.* Mordred: a molecular descriptor calculator. *J Cheminformatics* 2018;**10**(1):1–14.
36. Pedregosa, F, Varoquaux, G, Gramfort, A, *et al.* Scikit-learn: machine learning in Python. *J Machine Learn Res* 2011;**12**: 2825–2830.
37. Barretina, J, Caponigro, G, Stransky, N, *et al.* The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* 2012;**483**(7391):603–7.
38. Bairoch, A. The Cellosaurus, a cell-line knowledge resource. *J Biomol Techniques* 2018;**29**(2):25–38.
39. Ghandi, M, Huang, FW, Jané-Valbuena, J, *et al.* Next-generation characterization of the Cancer Cell Line Encyclopedia. *Nature* 2019;**569**(7757):503–8.
40. The International HapMap Consortium. The International HapMap Project. *Nature* 2003;**426**(6968):789–96.
41. Venkatraman, ES, Olshen, AB. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics* 2007;**23**(6):657–63.
42. Chaudhary, K, Poirion, OB, Lu, L, *et al.* Deep learning–based multi-omics integration robustly predicts survival in liver cancer. *Clin Cancer Res* 2018;**24**(6):1248–59.
43. Zhang, L, Lv, C, Jin, Y, *et al.* Deep learning-based multi-omics data integration reveals two prognostic subtypes in high-risk neuroblastoma. *Front Genet* 2018;**9**(477). https://doi.org/10.3389/fgene.2018.00477

44. Simidjievski, N, Bodnar, C, Tariq, I, *et al.* Variational autoencoders for cancer data Integration: design principles and computational practice. *Front Genet* 2019;**10**(1205). https://doi.org/10.3389/fgene.2019.01205
45. Hinton, GE, Salakhutdinov, RR. Reducing the dimensionality of data with neural networks. *Science* 2006;**313**(5786):504–7.
46. Wang, Y, Yao, H, Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* 2016;**184**:232–42.
47. Abadi, M, Agarwal, A, Barham, P, *et al.* TensorFlow: large-scale machine learning on heterogeneous distributed dystems. arXiv:1603.04467. 2015.
48. Kingma, DP, Adam, BJ. A method for stochastic optimization. *arXiv:14126980 [cs]*. 2017.
49. Meng, C, Zeleznik, OA, Thallinger, GG, *et al.* Dimension reduction techniques for the integrative analysis of multi-omics data. *Briefings Bioinf* 2016;**17**(4):628–41.
50. Preto, AJ, Moreira, IS. SPOTONE: hot spots on protein complexes with extremely randomized trees via sequence-only features. *Int J Mol Sci* 2020;**21**(19):7281.
51. Botchkarev, A. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary J Information Knowledge Management* 2019;**14**:45–76.
52. de Winter, JCF, Gosling, SD, Potter, J. Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: a tutorial using simulations and empirical data. *Psychol Methods* 2016;**21**(3):273–90.
53. Breiman, L. Random forests. *Machine Learn* 2001;**45**(1):5–322001.
54. Geurts, P, Ernst, D, Wehenkel, L. Extremely randomized trees. *Machine Learn* 2006;**63**(1):3–422006.
55. Fan, RE, Chang, KW, Hsieh, CJ, *et al.* LIBLINEAR: a library for large linear classification. *J Machine Learn Res* **9**:1871–42008.
56. Zadrozny, B, Elkan, C. Transforming classifier scores into accurate multiclass probability estimates. Association for Computing Machinery. Edmonton, Alberta. 2002:694–699. *KDD '02: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
57. Altman, NS. An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* **46**:175–851992.
58. Chen, T, Guestrin, C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. arXiv:1603.02754,2016.
59. DeCastro-García, N, Castañeda, ÁLM, García, DE, *et al.* Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm. *Complex* 2019;**2019**:1–16.
60. Swersky, K, Snoek, J, Adams, RP. Multi-task Bayesian optimization. *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems*. Lake Tahoe Nevada. C.J. Burges and L. Bottou and M. Welling and Z. Ghahramani and K.Q. Weinberger. **2**:2004–122013.
61. Korobov, M, Lopuhin, K. ELI5 - Debug machine learning classifiers and explain their predictions: https://pypi.org/project/eli5/.2016.Python Software Foundation.
62. O'Neil, J, Benita, Y, Feldman, I, *et al.* An unbiased oncology compound screen to identify novel combination strategies. *Mol Cancer Ther* 2016;**15**(6):1155–62.
63. Forcina, GC, Conlon, M, Wells, A, *et al.* Systematic quantification of population cell death kinetics in mammalian cells. *Cell Syst* 2017;**4**(6):600–610.e6.
64. Licciardello, MP, Ringler, A, Markt, P, *et al.* A combinatorial screen of the CLOUD uncovers a synergy targeting the androgen receptor. *Nat Chem Biol* 2017;**13**(7):771–8.
65. Kumar, V, Dogra, N. A comprehensive review on deep synergistic drug prediction techniques for cancer. *Arch Comput Meth Eng* 2022;**29**(3):1443–61.
66. Plotly Technologies Inc. Collaborative Data Science. Montréal, QC: Plotly Technologies Inc. 2015.
67. Grinberg, M. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc 2018.Sebastopol, California.
68. Grapov, D, Fahrmann, J, Wanichthanarak, K, *et al.* Rise of deep learning for genomic, proteomic, and metabolomic data integration in precision medicine. *OMICS* 2018;**22**(10):630–6.
69. Lin, Y-H, Lim, S-N, Chen, C-Y, *et al.* Functional role of mitochondrial DNA in cancer progression. *Int J Mol Sci* 2022;**23**(3):1659.
70. Tian, B-X, Sun, W, Wang, S-H, *et al.* Differential expression and clinical significance of COX6C in human diseases. *Am J Transl Res* **13**:1–102021.
71. Wu, C-J, Cai, T, Rikova, K, *et al.* A predictive phosphorylation signature of lung cancer. *PLoS One* 2009;**4**(11):e7994.
72. Wang, B, Cai, Y, Li, X, *et al.* ETV4 mediated lncRNA C2CD4D-AS1 overexpression contributes to the malignant phenotype of lung adenocarcinoma cells via miR-3681-3p/NEK2 axis. *Cell Cycle* 2021;**20**(24):2607–18.
73. Wang, L, Shilatifard, A. UTX mutations in human cancer. *Cancer Cell* 2019;**35**(2):168–76.
74. Gozdecka, M, Meduri, E, Mazan, M, *et al.* UTX-mediated enhancer and chromatin remodeling suppresses myeloid leukemogenesis through noncatalytic inverse regulation of ETS and GATA programs. *Nat Genet* 2018;**50**(6):883–94.
75. Sakthianandeswaren, A, Parsons, MJ, Mouradov, D, *et al.* MACROD2 haploinsufficiency impairs catalytic activity of PARP1 and promotes chromosome instability and growth of intestinal tumors. *Cancer Discov* 2018;**8**(8):988–1005.
76. Fernandes, S, Huellen, K, Goncalves, J, *et al.* High frequency of DAZ1/DAZ2 gene deletions in patients with severe oligozoospermia. *Mol Hum Reprod* 2002;**8**(3):286–98.
77. Hanson, HA, Anderson, RE, Aston, KI, *et al.* Subfertility increases risk of testicular cancer: evidence from population-based semen samples. *Fertil Steril* 2016;**105**(2):322–328.e1.
78. Chen, T, Wang, K, Tong, X. In vivo and in vitro inhibition of human gastric cancer progress by upregulating Kank1 gene. *Oncol Rep* 2017;**38**(3):1663–9.
79. Gu, Y, Zhang, M. Upregulation of the Kank1 gene inhibits human lung cancer progression in vitro and in vivo. *Oncol Rep*. 2018;**40**:1243–50.
80. Pedersen, A, Mikkelsen, E, Cronin-Fenton, D, *et al.* Missing data and multiple imputation in clinical epidemiological research. *Clin Epidemiol* 2017;**9**:157–66.
81. Zhang, Z. Missing data imputation: focusing on single imputation. *Ann Transl Med*. 2016;**4**(1):9.
82. Gilvary, C, Dry, JR, Elemento, O. Multi-task learning predicts drug combination synergy in cells and in the clinic: 2019. https://doi.org/10.1101/576017
83. Di Veroli, GY, Fornari, C, Wang, D, *et al.* Combenefit: an interactive platform for the analysis and visualization of drug combinations. *Bioinformatics* 2016;**32**(18):2866–8.
84. Akoglu, H. User's guide to correlation coefficients. *Turkish J Emerg Med* 2018;**18**(3):91–3.
85. Sidorov, P, Naulaerts, S, Ariey-Bonnet, J, *et al.* Predicting synergism of cancer drug combinations using NCI ALMANAC data. *Front Chem* 2019;**7:509**. https://doi.org/10.1101/576017

86. Preuer, K, Lewis, RPI, Hochreiter, S, *et al.* DeepSynergy: predicting anti-cancer drug synergy with deep learning. *Bioinformatics* 2018;**34**(9):1538–46.
87. Zagidullin, B, Aldahdooh, J, Zheng, S, *et al.* DrugComb: an integrative cancer drug combination data portal. *Nucleic Acids Res* 2019;**47**(W1):W43–51.
88. Holbeck, SL, Camalier, R, Crowell, JA *et al.* The National Cancer Institute ALMANAC: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity. *Cancer Res* 2017;**77**(13): 3564–76.
89. Preto, AJ, Matos-Filipe, P, Mourão, J, *et al.* Supporting data for "SYNPRED: prediction of drug combination effects in cancer using different synergy metrics and ensemble learning." *GigaScience Database*. 2022. https://doi.10.20944/preprints202104.0395.v1

Chapter 4: Conclusions

The work in this thesis branched out in four main sets of results, unified around the goals and application of drug design and development. Two of these sets of results were focused on protein targets and aimed to improve the knowledge and tools available to characterize them all the way from amino acid to MP oligomer levels. The third set of results aimed at characterising drugs on an explainable setting while keeping open the option for bulk processing. Finally, the last set of results pertained tackling of a broad-encompassing subject directed centred on drug combination synergy prediction in cancer cell lines.

HS prediction from protein sequence-only data was conducted using state-of-the-art ML algorithms and delivered excellent results, opening the gates for in-depth protein characterization in low-information settings.

- Despite a low amount of available data samples, it was possible to build a HS/NS predictor using only protein-sequence data.
- Building a DL predictor with the available data yielded promising preliminary results.
- The robustness of SPOTONE emerged from a thorough treatment and splitting of the dataset, the usage of ERT and the exclusion of whole protein sequence features, leaving only residue-specific sequence-based features.
- Outstanding performance of the method: AUROC (0.83), accuracy (0.82), recall (0.82), precision (0.91) and f1-score (0.85).
- Its competitiveness performance against the best structural based predictors is complemented with the high versatility of using only sequence-based features prediction, which allows a much wider application in a variety of biological problems.

Membrane protein dimer characterization was thoroughly performed, providing information coherent with literature findings and new global insights on the subject. The results in this thesis also unveiled a hub for curated and newly generated MP data.

- MPs have a higher content of hydrophobic and aromatic residues, which contribute to the accuracy of ML models developed for predicting protein–protein binding sites.
- Glycine, Alanine and Serine (GAS) residues are significantly enriched at the MPs core and non-interfacial surface locations, in comparison to interfacial surface. These small residues are the strong driving force for membrane folding and favour thermodynamic interactions with the lipid bilayer. In contrast, charged residues are typically excluded from the MPs interface.
- Evolutionary conservation was shown to be lower for the surface, followed by the interface and then the protein core. The core and the interface of MPs are the most conserved regions, granting necessary structural stability at specific PPIs.

- Furthermore, b-factor values are lower for interfacial residues compared to non-interfacial surface ones, indicating that residues participating in PPIs are usually less motile.
- Aromatic residues are much more prone to establish close intermolecular atomic contacts at short distance than other residues. Coupled with and non-polar residues, they show a high number of hydrophobic contacts. This is, partly, due to certain types of interactions exclusive, or preferred, to them. Phenylalanine and tyrosine establish π–π stacking, T-stacking and cation–π interactions in different dimers. Cation–π interactions are also particularly relevant for arginine.
- MENSAdb is the first comprehensive resource dedicated explicitly to exposing the evolutionary and physicochemical features of dimeric MP structures.

Drug Characterization unfolded under a data-centric approach, while privileging feature interpretability and biological meaning.

- Taxonomic characterization of chemical compounds is delivered in the form of novel features oriented towards data science, ML and AI solutions. There is a heavy focus on interpretable pharmacological data and features, key for the scientific community, as well as the Pharma sector.
- Bulk small ligand analysis is provided, which makes use of kingdom and superclass attribution to perform small molecules' categorization. These categories account for multiple superclasses, in the cases in which this is possible.
- DrugTax retrieves summary data from an input list of drugs and uses individual small ligand information to generate a fast characterization tool of small molecules. Furthermore, by making use of up-to-date visualization tools, it can depict many intersecting sets (in the form of small ligand superclasses), which is often limited by more conventional forms of visualization.
- The test case yielded a list of 10.556 unique virus-associated compounds. Through bulk analysis, most of the compounds belonged to the organic kingdom. The most recurring superclass was hydrocarbon derivatives, with few hydrocarbons present. The most populated aggregation of superclasses were organic molecules that fit the combination of superclasses: hydrocarbon derivatives, organoheterocyclic, organic oxygen, organic nitrogen and organopnictogens.
- DrugTax simplifies molecule characterization and presents comprehensible molecule categorization as well as clear and interpretable features, which yields a set of simple and fundamental level applications. For example, it could be applied to generate similarity searches, chemical space visualization, clustering, taxonomy-property relationships, among others.
- Due to its easy deployment and installation, DrugTax is a tool whose potential can unfold extensively. It also exhibits very fast performance with an easy-to-use interface available on PyPI and GitHub.

Drug Combination synergy prediction in cancer lines was tackled on an in depth-protocol that traverses through omics data, feature interpretability, the most recent ML approaches and a thorough - literature-aware – protocol, delivering a final, usable, and useful prediction tool.

- This study introduced a new synergy prediction model, SYNPRED, that combines comprehensive multi-omics data of cancer cell lines with physicochemical and structural features of drugs. This work is the first that takes five different synergy reference models (Bliss, HSA, Loewe, ZIP, and CSS) and uses one of the most comprehensive and balanced databases regarding the synergistic/non-synergistic distribution, the NCI-ALMANAC.

- When comparing the reported performance for algorithms in their settings it is necessary to consider a very broad array of circumstances, such as prediction models, datasets, and synergy reference models. The high possible combination of factors that leads to the final methods' performance is huge, and therefore comparisons must be careful and avoid narrow-mindedness.

- Six final predictors were built, one for classification (full – agreement between the five different synergy reference models) and five for regression tasks. Furthermore, the final tally of synergistic queries predicted by all models based on the prediction values is also provided in the form of "Synergy Votes". This facilitates the visualization of the type of combinatory effect between the two drugs and aims at strengthening the value of the prediction due to the lack of consensus between the different synergy reference models.

- The top-ranked classification and regression models, ensembles developed with the best ML models, achieved state-of-the-art performance to predict synergistic drug combinations in an independent dataset. The best-performing prediction model in SYNPRED is, undoubtedly, CSS (RMSE, 11.07; MSE, 122.61; Pearson, 0.86; MAE, 7.43; Spearman, 0.87). However, it is advisable to consider the aggregate of results, albeit with a higher focus on CSS.

- For the classification task, the final ensemble full-agreement SYNPRED comprised 4 DL-based and 6 ML-based models. When applied in an independent test set, the ensemble model displayed better performance (accuracy = 0.85, precision = 0.91, recall = 0.90, AUROC = 0.80, and F1-score = 0.90) than any other classic ML or DL models, with the best individual performing model being XGB. However, there was a significant drop in the leave cells, drugs, and drug combinations out datasets.

- When comparing performance with predictors deployed upon NCI ALMANAC (the dataset used in this study), SYNPRED stands out in all the synergy reference models with Pearson and Spearman correlation performance increments between 30.51% and 42.37%, as well as between 36.36% and 56.36%, respectively. Although MSE metrics are particularly hard to compare between datasets and methods, significant improvements were also observed.

- More complex models, particularly DL-based models with different architectures, tend to make more extensive use of the omics-based features to over 70% of the total feature contribution. Contrarily, simpler models made almost exclusive use of the drug features (above 90%). Other non-DL-based models made variable (between 20% and 80%) usage of the omics features. This observation highlights the importance of DL models to take full advantage of omics data for capturing the complexity of each cancer profile, thus improving drug pair–cell line combinations predictions.

- Of the 15 ranked genes from expression, methylation, and Copy Number Variation (CNV), all of them are used as prognostic cancer markers or have a role in tumour progression and treatment. These data suggest that our models, especially DNNs, are likely to capture the most relevant information for each group of multi-omics features for synergistic drug combinations.

- To our knowledge, this is the first web server that can predict new drug synergy combinations without the need of uploading a partial or full dose–response matrix. This feature is an advantage compared with other models implemented in webservers that need these types of data for drug combination response prediction.

Despite the already large but still increasing investment in drug design and development, there are still many venues for growth, many of which inevitably through trial and error. During my thesis I focused on some of the cornerstones of drug design and development, making heavy use of the available ML tools to address both the targets and the drugs, as well as a specific problem of interest. In short, this resulted in the following contributions:

- Increased protein characterization at multiple levels.

- Deepened small molecule insights.

- A concrete solution for the drug combination synergy in cancer cell lines problem.

It is also worth mentioning that all the conducted research was adequately made available for replication and generated multiple tools that will allow its continued contribution to the field.

Appendices

Permissions

ELSEVIER LICENSE TERMS AND CONDITIONS

Oct 06, 2022

This Agreement between Mr. António Preto ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

The publisher has provided special terms related to this request that can be found at the end of the Publisher's Terms and Conditions.

License Number	5403090795603
License date	Oct 06, 2022
Licensed Content Publisher	Elsevier
Licensed Content Publication	Elsevier Books
Licensed Content Title	Comprehensive Pharmacology
Licensed Content Author	A.J. Preto,C Marques-Pereira,Salete J. Baptista,B. Bueschbell,Carlos A.V. Barreto,A.T. Gaspar,I. Pinheiro,N. Pereira,M. Pires,D. Ramalhão,D. Silvério,N. Rosário-Ferreira,R. Melo,J. Mourão,I.S. Moreira
Licensed Content Date	Jan 1, 2022
Licensed Content Pages	28
Start Page	135
End Page	162
Type of Use	reuse in a thesis/dissertation
I am an academic or government institution with a full-text subscription to this journal and the audience of the material consists of students and/or employees of this institute?	No
Portion	full chapter
Circulation	1
Format	electronic
Are you the author of this Elsevier chapter?	Yes
How many pages did you author in this Elsevier book?	28
Will you be translating?	No
Title	Deep-Learning application to in silico Drug Design
Institution name	University of Coimbra
Expected presentation date	Feb 2023
Order reference number	1
Attachment	B4.pdf
Requestor Location	Mr. António Preto Rua São João nº 39 Repeses Viseu, Viseu 3500-727 Portugal Attn: University of Coimbra
Publisher Tax ID	GB 494 6272 12

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Oct 06, 2022

This Agreement between Mr. António Preto ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5401470658349
License date	Oct 03, 2022
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Structural Characterization of Membrane Protein Dimers
Licensed Content Author	António J. Preto, Pedro Matos-Filipe, Panagiotis I. Koukos et al
Licensed Content Date	Jan 1, 2019
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes
Title	Deep-Learning application to in silico Drug Design
Institution name	University of Coimbra
Expected presentation date	Feb 2023
Order reference number	3
Requestor Location	Mr. António Preto Rua São João nº 39 Repeses Viseu, Viseu 3500-727 Portugal Attn: University of Coimbra
Total	**0.00 EUR**
Terms and Conditions	

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. Grant of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Oct 06, 2022

This Agreement between Mr. António Preto ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5401470764414
License date	Oct 03, 2022
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Predicting Hot Spots Using a Deep Neural Network Approach
Licensed Content Author	António J. Preto, Pedro Matos-Filipe, José G. de Almeida et al
Licensed Content Date	Jan 1, 2021
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes
Title	Deep-Learning application to in silico Drug Design
Institution name	University of Coimbra
Expected presentation date	Feb 2023
Order reference number	4
Requestor Location	Mr. António Preto Rua São João nº 39 Repeses Viseu, Viseu 3500-727 Portugal Attn: University of Coimbra
Total	**0.00 EUR**
Terms and Conditions	

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. Grant of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.