



UNIVERSIDADE D
COIMBRA

Laura Carneiro Penetra Pires Angeja

AUTOMATION OF BEHAVIOURAL TESTS ON
ANIMAL MODELS OF MACHADO JOSEPH'S
DISEASE

Thesis submitted to the Faculty of Sciences and Technology of the
University of Coimbra for the degree of Master in Biomedical
Engineering with specialization in Biomedical Instrumentation,
supervised by Dr. Rui Nobre and Prof. Dr. Luís Pereira Almeida.

September 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Automation of Behavioural Tests on Animal Models of Machado Joseph's Disease

Laura Carneiro Penetra Pires Angeja

Thesis submitted to the Faculty of Sciences and Technology of the University of Coimbra
for the degree of Master in Biomedical Engineering with specialization in Biomedical
Instrumentation

Supervisors:

Dr. Rui Nobre (CNC-UC)

Prof. Dr. Luís Pereira Almeida (CNC-UC)

Coimbra, 2023

This work was developed in collaboration with:

Department

Group of Vectors and Gene Therapy, at the Center for Neuroscience and Cell Biology of the University of Coimbra under the supervision of Doctor Rui Jorge Nobre and Prof. Dr. Luís Pereira de Almeida.

Institutes

Center for Neuroscience and Cell Biology (CNC), Gene and Stem Cell Therapies for the Brain Group, University of Coimbra, 3004-504 Coimbra, Portugal.

Center for Innovative Biomedicine and Biotechnology (CIBB), Vectors, Gene and Cell Therapy Group, University of Coimbra, 3004-504 13 Coimbra, Portugal.

Esta cópia da tese é fornecida na condição de que quem a consulta reconhece que os direitos de autor são da pertença do autor da tese e que nenhuma citação ou informação obtida a partir dela pode ser publicada sem a referência apropriada.

This thesis copy has been provided on the condition that anyone who consults it understands and recognizes that its copyright belongs to its author and that no reference from the thesis or information derived from it may be published without proper acknowledgement.



Agradecimentos

Chegaram ao fim estes anos que eu tanto ansiava que durassem para sempre. Olhando para trás, tenho a confirmação de que o meu percurso académico esteve repleto de pessoas marcantes e sem as quais não teria sido possível.

Em primeiro lugar, quero agradecer aos meus orientadores. Ao professor Luís Almeida, por me ter dado esta oportunidade e por me ter recebido no seu grupo de investigação com muita simpatia. Ao Rui, por me ter mostrado este projeto e ter confiado em mim para levá-lo a cabo. Obrigada por ter estado sempre disponível para ajudar, para resolver problemas e por me incentivar em todos os momentos. À Sara, que mostrou toda a amabilidade e sem a ajuda da qual este projeto não teria tomado este rumo. Agradeço também ao grupo do professor Peça, em especial ao Pedro, pela disponibilidade. Não posso deixar de agradecer ao professor César, por me ter aconselhado em vários momentos.

Aos membros da Team R, por me terem acolhido desde o primeiro dia. À Carina, que foi a primeira a sair do seu caminho para me ajudar. Ao António e ao Gui, os meus amigos da "meta". E ao João que, estando no mesmo patamar que eu, sempre me conseguiu ajudar.

À minha casa nesta faculdade, o DF, e a todos os que dele fazem e fizeram parte, os que pertenceram à minha vida académica durante estes 5 anos, um obrigada muito especial. Não pensei terminar este percurso com tanta certeza de que a minha passagem por esta academia foi única e realmente inesquecível.

À Bia, uma pessoa que por alguma sorte entrou na minha vida este ano e que me acompanhou neste e noutros projetos. Obrigada pela maneira como nunca me deixaste sentir desamparada, me ouviste e me guiaste a resolver os meus problemas. Aprendi imenso e sem ti não teria alcançado tudo o que me propus. Deixo, por tudo, um obrigada incondicional.

À minha família de praxe, à minha madrinha Rita, às minhas afilhadas Abigail e Inês, obrigada por tantos jantares repletos de gargalhadas, de conselhos e parvoíces.

Aos meus amigos d'Os Cruzados, o João, a Mari, a Castela, o Tomé, a Rita, o Caramelo, o Luís, a Martini, o Costini, o Prata, o Mendonça, o Xico, a Mati e a Eva por viverem estes anos ao meu lado e por se terem tornado no melhor grupo de amigos que Coimbra já viu.

Às três farmacêuticas que conheci em Bologna e que desde aí não larguei, Ana Marçal, Sara e Matilde, obrigada. Levo a vossa amizade no coração. Para a Ana, um agradecimento

especial, por me ajudares a crescer e superar os meus medos, pelas conversas longas enquanto cozinhamos um "piatto di pasta".

Aos meus amigos do secundário, Princesas e Anões, obrigada por termos crescido juntos e por nos continuarmos a apoiar uns aos outros.

À Eva, a amiga que me acompanhou em tudo desde a primeira aula, por uma vida juntas em Bologna, até esta última etapa. Foi assim até aqui, juntas, não tenho dúvidas que será assim daqui para a frente.

Ao Diogo, um obrigada enorme por teres acreditado em mim, mesmo quando eu não acreditava, por me apoiares e por estares sempre pronto a ouvir todos os meus queixumes. Obrigada pela companhia, carinho e paciência em todos os momentos.

Por último e talvez o mais importante, agradeço à minha família. À minha avó Lete, que se vira ao contrário para me ajudar, ao meu avô João que tem sempre uma palavra de sabedoria para me ensinar e ao meu tio Daniel que me ajuda a escapar deste mundo e a entrar noutra, de música, filmes e imaginação. Um obrigada especial ao meu irmão, que me ensinou a trabalhar ao som de qualquer música. Tenho a certeza que esta skill virá a dar jeito. Um obrigada do tamanho do mundo ao meu pai, que revira todos os cantos e recantos do universo para me ajudar, e à minha mãe, aquela a quem eu recorro para contar os meus problemas, pois sei que ela sabe sempre o que fazer, e na qual confio incondicionalmente. Não há pais mais incríveis, vocês são a minha maior fonte de inspiração. Obrigada por tudo que fizeram e continuam a fazer por mim.

Funding

This work was funded by the European Regional Development Fund (ERDF), through the Centro 2020 Regional Operational Program; through the COMPETE 2020 – Operational Programme for Competitiveness and Internationalization, and Portuguese national funds via FCT under the projects: UIDB/04539/2020, UIDP/04539/2020, LA/P/0058/2020, SpreadSilencing (POCI-01-0145-FEDER029716), ViraVector (CENTRO-01-0145-FEDER-022095), BDforMJD (CENTRO-01-0145-FEDER-181240), ModelPolyQ2.0 (CENTRO-01-0145-FEDER-181258), and MJDEDIT (CENTRO-01-0145-FEDER-181266); ARDAT under the IMI2 JUgrant agreement no. 945473 supported by the European Union’s H2020 programme and EFPIA; by the American Portuguese Biomedical Research Fund (APBRF) and the Richard Chin and Lily Lock Machado-Joseph Disease Research.

Abstract

Machado Joseph disease (MJD) or Spinocerebellar Ataxia Type 3 (SCA3) is a neurodegenerative disorder characterised by an expansion of a CAG repeat in the coding region of the MJD1 gene. This abnormal expansion translates into a toxic gain of function of the ataxin-3 protein, which triggers multiple pathogenic mechanisms leading to neurodegeneration in several brain regions. The absence of available treatment for MJD encourages further investigation towards possible therapeutic approaches.

This study aimed to develop a semi-automated system and to expand the insights on behaviour assessment studies in video recordings of mouse models of MJD in pre-clinical research.

For that purpose, we exploited two behaviour tracking methods that presented two distinct approaches to our problem. Firstly, we constructed a low-cost setup specific to open field assessment tests. For analysis of the recorded tests, we used Bonsai RX to be able to track the centre of mass of the animal under study. For our second approach, we used an already existing tracking system, DeepLabCut, based on pose estimation algorithms that specialise in body part prediction. Thus, a significant part of this dissertation was directed at the identification of optimal parameter values for the tracking algorithm. Both approaches proved suitable to automatically track mice in an open field test setting and calculate our customised measurements.

In summary, this work contributes to MJD investigation through the implementation of two semi-automated tracking approaches for animal behaviour studies which mitigates researchers' excess work related to test data analysis. Moreover, these developed pipelines are not restricted to MJD related investigation and may be applied to other SCAs research.

Keywords: animal tracking, MJD YAC mouse model, Bonsai RX, convolutional neural networks, DeepLabCut, neuroscience, open field test, Machado Joseph disease, Spinocerebellar Ataxia Type 3

Resumo

A doença de Machado Joseph (DMJ) ou ataxia espinocerebelosa do tipo 3 (SCA3) é uma doença neurodegenerativa caracterizada por uma expansão anômala de uma repetição de trinucleotídeos CAG na região codificadora do gene MJD1. Esta expansão anormal traduz-se num ganho de função tóxica da proteína ataxina-3, que desencadeia múltiplos mecanismos patogénicos que levam à neurodegeneração em várias regiões do cérebro. A ausência de tratamento disponível para a DMJ incentiva a investigação de possíveis abordagens terapêuticas.

Este estudo teve como objetivo expandir os conhecimentos sobre estudos de avaliação do comportamento de modelos de murganho, em gravações de vídeos, assim como desenvolver um sistema semi-automatizado para a monitorização do comportamento em modelos de murganho da DMJ.

Para esse efeito, explorámos dois métodos de rastreio do comportamento que apresentam duas abordagens distintas ao nosso problema. Em primeiro lugar, construímos uma instalação de baixo custo específica para testes de avaliação em campo aberto (open field). Para a análise dos testes recolhidos, utilizámos o programa Bonsai RX para conseguir localizar o centro de massa do animal em estudo. Para a nossa segunda abordagem, utilizámos um sistema de detecção já existente, o DeepLabCut, baseado em algoritmos de estimação de pose, especializados na predição de partes do corpo. Assim, uma parte significativa desta dissertação foi direccionada para a identificação dos valores óptimos dos parâmetros do algoritmo de seguimento. Ambas as abordagens provaram ser adequadas para seguir automaticamente murganhos num ambiente de teste em campo aberto e calcular as nossas medições específicas.

Em resumo, este trabalho contribui para a investigação da DMJ através da implementação de duas abordagens de rastreio semi-automatizadas, para estudos de comportamento animal, que atenuam o excesso de trabalho dos investigadores relacionado com a análise de dados de testes. Além disso, estas metodologias desenvolvidas não se restringem à investigação relacionada com a DMJ e podem ser aplicadas a outras investigações relacionadas com outras doenças de ataxias espinocerebelares (SCA).

Palavras-chave: rastreamento de animais, modelo de murganho DMJ YAC, Bonsai RX, redes neuronais convolucionais, DeepLabCut, neurociência, teste de campo aberto, doença de Machado Joseph, ataxia espinocerebelosa do tipo 3

Contents

List of Figures	xvii
List of Tables	xix
List of Abbreviations	xxi
1 Introduction	1
1.1 Context and Motivation	1
1.2 Objectives	2
1.3 Document Structure	2
1.4 Scientific Dissemination	3
2 Machado-Joseph Disease	5
2.1 Brief Description and Historical Context	5
2.1.1 Disease: <i>MJD1</i> Gene and <i>Ataxin-3</i> Protein	6
2.1.2 Symptomatology	6
2.1.3 Neuropathology	7
2.2 Animal Models of Machado-Joseph Disease	7
2.2.1 Mouse Models	8
2.2.2 MJD YAC Mouse Model - Phenotypical Profile	10
3 Automation of Behaviour Analysis	13
3.1 Animal Behaviour Tracking	13
3.2 Behavioural Video Tracking Analysis	14
3.2.1 Experiment Control Systems	15
	xiii

3.2.1.1	Data Collection System - Bonsai	15
3.2.2	2D Pose Estimation and Tracking	15
3.2.2.1	Current Deep Learning Tools	16
4	Deep Learning Methods	19
4.1	Machine Learning	19
4.2	Deep Learning	19
4.2.1	Convolutional Neural Network	21
4.3	Transfer Learning	22
5	Methodology	25
5.1	Mice	25
5.2	Behavioural Test: Open Field Test	26
5.3	Dataset	26
5.4	Video Acquisition	26
5.4.1	Experimental Setup for Video Acquisition	26
5.4.2	Hardware Tools for Video Acquisition and Analysis	27
5.4.3	Software Tools for Video Acquisition	27
5.5	Preprocessing - Ffmpeg	27
5.6	General Settings	28
5.7	Development of Video Tracking Analysis System	29
5.7.1	Protocol Optimization for Bonsai Tracking	29
5.7.2	Protocol Optimization for DeepLabCut Tracking	30
5.7.2.1	Pipeline	30
5.7.2.2	DeepLabCut Parameters	31
5.7.2.3	Frame Labelling	35
5.7.2.4	Human Labelling Error	35
5.7.2.5	Validation	36

6	Results and Discussion	37
6.1	Bonsai Tracking	37
6.2	DeepLabCut Tracking	39
6.2.1	Parameter Selection	39
6.2.2	Workflow and Results	40
6.3	Leveraging Keypoint Tracking in DeepLabCut	42
6.3.1	Wild-type Animal	42
6.3.2	Homo YAC PBS Animal	43
6.3.3	Body Parts Keypoint Tracking	45
6.4	Final Discussion	47
7	Conclusion	49
	References	51
	Appendices	61
A	DeepLabCut Parameter Comparison	63
A.1	Batch Size	63
A.2	Global Scale	64
A.3	Data Augmentation - Cropped Images	65
A.4	Likelihood Threshold - <i>p-cutoff</i>	66
B	Supplementary Figures of Bonsai Analysis	69
C	Supplementary Figures of DeepLabCut Analysis	71
D	Supplementary Figures of DeepLabCut Keypoint Tracking Analysis	73

List of Figures

2.1	Schematic representation of distribution of neuronal loss in the Central Nervous System (CNS) of Machado Joseph disease (MJD) patients.	7
3.1	Schematic Overview of Markerless Motion Capture.	16
4.1	Representation of a perceptron.	20
4.2	Representation of an Artificial Neural Network, with the input layer (yellow), the hidden layers (green) and output layer (red).	21
5.1	Experimental Setup and Scheme for Video Acquisition.	27
5.2	Adopted Pipeline for DeepLabCut Network Training and Video Analysis.	30
5.3	Schematic Representation of Mouse Labelled Points.	35
6.1	Workflow diagram for first strategy, Bonsai RX.	38
6.2	Open field test performance measured using Bonsai.	39
6.3	Workflow diagram for second strategy, DeepLabCut.	40
6.4	Open field test performance measured using DeepLabCut.	41
6.5	Point tracking trajectory of three tracked body parts of a wild-type mouse.	43
6.6	Point tracking trajectory of three tracked body parts of a Homo YAC PBS mouse.	44
6.7	Open field test performance of nose point across Wild-Type and Homo YAC PBS using DeepLabCut.	45
6.8	Open field test performance of body centre point across Wild-Type and Homo YAC PBS using DeepLabCut.	46
6.9	Open field test performance of tail centre point across Wild-Type and Homo YAC PBS using DeepLabCut.	46
A.1	Comparison of Train and Test error of Batch Size 1, 2 and 4.	64

A.2	Comparison of Train and Test error of Global Scale = 0.8 and Global Scale = 0.9.	65
A.3	Comparison of Train and Test error of Crop = TRUE and Crop = FALSE.	66
A.4	Comparison of Train and Test error with and without p -cutoff = 0.1.	67
A.5	Comparison of Train and Test error with and without p -cutoff = 0.8.	67
B.1	Comparison of distance travelled across gender using Bonsai.	69
B.2	Comparison of open field test performance across Wild-Type and Transgenic Homo YAC PBS using Bonsai.	70
C.1	Comparison of distance travelled across gender using DeepLabCut.	71
D.1	Total Distance detailed by a wild-type per body part.	74
D.2	Total Distance detailed by a Homo YAC PBS per body part.	75

List of Tables

2.1	Mouse Models of Machado Joseph Disease.	9
3.1	Overview of Popular Deep Learning Tools for Animal Motion Capture.	17
5.1	Number of animals divided by condition.	25
5.2	Human Labelling Average Error in pixels.	36
6.1	Selected Values for DeepLabCut Hyperparameter Comparison.	40
A.1	Standard Values for DeepLabCut Hyperparameter Comparison.	63
D.1	Values of the measurements for Wild-Type group.	76
D.2	Values of the measurements for Homo PBS YAC group.	76

List of Abbreviations

- ANN** Artificial Neural Network. xvii, 19, 20, 21
- ATXN3** Ataxin 3 protein. 8, 10
- CAG** Cytosine-Adenine-Guanine Trinucleotide Repeats. ix, 9, 10, 11
- CDS** Coding Sequence. 9, 10
- CMV** Cytomegalovirus. 10
- CNN** Convolutional Neural Network. 21, 22, 31, 32, 33, 34, 36, 37, 63, 64, 65, 66
- CNS** Central Nervous System. xvii, 7
- CNV** Copy Number Variation. 10
- DLC** DeepLabCut. 17, 31
- DRPLA** Dentatorubral–Pallidoluyisian Atrophy. 5
- GUI** Graphical Use Interface. 15, 17
- HD** Huntington’s disease. 5
- MAE** Mean Average Euclidean Error. 35, 36
- MJD** Machado Joseph disease. ix, xvii, 1, 2, 5, 6, 7, 8, 10, 11, 12, 13, 25, 37, 49, 50
- ML** Machine Learning. 19
- mut-ATXN3** Mutant Form of Ataxin 3 Protein. 9
- NES** Nuclear Export Signal. 9
- NII** Neuronal Intranuclear Inclusions. 9
- NLS** Nuclear Localization Signal. 9
- PBS** Phosphate-buffered saline. 25, 26, 42, 43
- polyQ** Polyglutamine Diseases. 5
- SCA** Spinocerebellar Ataxia. ix, 5, 49
- SCA3** Spinocerebellar Ataxia Type 3. ix, 1, 5
- SPMA** Spinal–Bulbar Muscular Atrophy. 5
- YAC** Yeast Artificial Chromosome. 2, 8, 9, 10, 11, 25, 26, 37, 42, 43, 44, 48

Introduction

1.1 Context and Motivation

Machado Joseph disease (MJD), also known as Spinocerebellar Ataxia Type 3 (SCA3), is a neurodegenerative hereditary disorder that affects 1.5 individuals in every 100 000 worldwide [Orphanet, 2022]. In Portugal, it presents a higher prevalence, particularly in the Azores, where the highest incidence per capita is reported [Bettencourt and Lima, 2011]. It is an extremely debilitating condition, characterized by progressive cerebellar dysfunction. This degeneration causes a spectrum of problems affecting gait, balance, speech, eye movements, among other manifestations [Taroni and DiDonato, 2004].

At present, there is no cure or effective treatment to prevent symptom progression. Extensive research efforts have been devoted into developing therapies for this disease. Part of research studies in this domain incorporate pre-clinical investigations, which entail the use of animal testing to analyse the effects of therapeutic strategies. Particularly, mouse models are commonly used due to their physiological similarities with humans. In cases where behaviour assessment is involved, the tests conducted are often recorded for posterior analysis. Analysing behaviour data is a laborious and time-consuming task often burdening researchers. Manual scoring, up to now a common approach, presents several additional drawbacks, including inter-rater variability, dependence on rater's experience and on human detectable behaviours. Despite the availability of various proprietary software to address these issues, they are often expensive, closed source and rarely disease specific, making their use somewhat limited.

Open source solutions emerged to mitigate this gap by permitting researchers to customise the software to meet to their specific requirements. Additionally, these systems grant researchers greater control over design and coordination of their experiments [Siegle et al., 2017, Lopes et al., 2015]. Subsequent advances in the field of deep learning and video monitoring technology led to the development of alternatives that generated meaningful outcomes, such as animal tracking, keypoint extraction and pose estimation, some even capable of outperforming closed source commercial systems [Sturman et al., 2020]. In fact, deep learning algorithms are the most powerful among pose estimation algorithms [Toshev and Szegedy, 2014, Insafutdinov et al., 2016]. Such tools originate new possibilities in the neuroscience rodent research. Current animal pose estimation packages provide tools for the researcher to generate and train tailored networks to

user-focused needs. This generally involves an initial frame labelling stage, a training/testing stage in which a neural network is optimized to predict poses from images it is fed, followed by performance evaluation and network refinement.

The extraction of novel behavioural measurements from pre-existing videos has the potential to add information to the already collected data. However, managing these open source packages is a complex process, especially for neuroscientists and researchers in this field who may lack programming skills. In fact, our primary motivation for this work was to make behavioural analyses easier and faster for researchers at our research centre.

1.2 Objectives

The main goal of this project was to assess the applicability of open source methods for analysing mouse behaviour in experimental studies of MJD. The various aims our project hoped to achieve are delineated below:

- Establish two working pipelines to analyse the open field behaviour test, from acquisition to result extraction and analysis, creating a more cost-effective experimentally oriented way to conduct mouse behaviour testing in MJD research.
- Extract meaningful behavioural data from the open field test, increasing the level of confidence in the results and broadening the test's utility.
- Create a functional end-to-end framework for behaviour testing with the open field assay to be used in the future in MJD studies at our investigation centre.

Furthermore, we intend to make a meaningful contribution to MJD related research by proposing an alternative approach to quantify behaviour function. We envision this thesis as a comprehensive guide, offering detailed information about the toolboxes used, parameters selected, calculated metrics and all the steps taken for the sake of clarity and reproducibility.

1.3 Document Structure

This document is structured as follows:

Chapter 2 provides a simple introduction to Machado Joseph disease, followed by an overview of the disease's transgenic animal models and a description of the mouse model utilised, the Yeast Artificial Chromosome (YAC) mouse model.

Chapter 3 describes the state of the art for behavioural video tracking analysis.

Chapter 4 introduces the theoretical background behind deep learning methods, the backbone of keypoint tracking and pose estimation algorithms described in the previous chapter.

Chapter 5 thoroughly describes the tools and the methodology used in the various steps of this study.

Chapter 6 presents the results obtained and discusses them through comparative and interpretive analysis.

Chapter 7 provides a conclusion and outlines potential directions for future work.

1.4 Scientific Dissemination

The work developed during this project resulted the following scientific contributions:

- A. Silva*, L. Angeja*, G.L. Gabriel*, C. Henriques, L.S. Gaspar, L. Pereira de Almeida, R.J. Nobre. "Motor performance in mouse models of Spinocerebellar ataxia type 3: a meta-analysis", 2023. (*) these authors contributed as co-first authors. [Silva* et al., 2023]. Submitted to "Journal of Translational Medicine" for publication.
- Repository with our established pipeline available in [Angeja, 2023].

Machado-Joseph Disease

This chapter aims to outline the main concepts required to develop this work. Firstly, Section 2.1 introduces some pivotal concepts related to the disease at study, MJD. Section 2.2 presents the mouse models generated thus far, taking special attention to the transgenic model selected to conduct this thesis research, in Subsection 2.2.2.

2.1 Brief Description and Historical Context

Machado Joseph disease (MJD) or Spinocerebellar Ataxia Type 3 (SCA3) is an incurable autosomal dominant ataxia that belongs to the group of Polyglutamine Diseases (polyQ), which includes Huntington’s disease (HD), Dentatorubral–Pallidoluysian Atrophy (DRPLA), Spinal–Bulbar Muscular Atrophy (SPMA) and other forms of Spinocerebellar Ataxia (SCA). These diseases are triggered by the expansion of an unstable CAG repeat located in the coding region of the respective causative gene, which is responsible for the clinical and neuropathological similarities observed [Schöls et al., 2004, Katsuno et al., 2008].

MJD is considered the most common subtype of dominant inherited spinocerebellar ataxias worldwide [Schöls et al., 2004]. This neurodegenerative disorder was named after two families of Azorean ancestry, who migrated to the USA as of the first descriptions of the disease. Since then, MJD has been identified in many families worldwide, both of Portuguese and non-Portuguese ancestry [Sakai et al., 1983, Livingstone and Sequeiros, 1984, Eto et al., 1990, Ogun et al., 2015].

The disease’s relative frequency is variable depending on the region, being highest in Brazil, Portugal (particularly in Azores islands), Singapore and China [Bettencourt and Lima, 2011, Lima and Coutinho, 1980].

The genetic cause of MJD is characterized by the pathological expansion of CAG trinucleotide repeats in the coding region of the causative gene, which is identified as *MJD1* gene [Kawaguchi et al., 1994]. As a result, the expansion of CAG repeats translates into an abnormally long polyQ tract in the causative gene conferring a toxic gain of function to the ataxin-3 protein [Ikeda et al., 1996].

2.1.1 Disease: *MJD1* Gene and *Ataxin-3* Protein

In 1993, the causative gene, MJD1, locus was assigned to the long arm of chromosome 14 (14q24.3-q32) [Takiyama et al., 1993]. The following year, it was discovered that this mutation-causing gene resulted from an anomalous expansion of trinucleotide CAG repeats in the coding region (exon 10) of the MJD1 gene [Kawaguchi et al., 1994].

Ataxin-3, a polyubiquitin-binding protein with a physiological role associated with ubiquitin-mediated proteolysis [Burnett et al., 2003] is encoded by the MJD1 gene. Mutant ataxin-3 protein exhibits a long polyglutamine tract at the C-terminus due to CAG expansion in the causative gene [Ikeda et al., 1996]. Contrary to healthy individuals exhibiting a CAG tract spanning from 10 to 51 repeats, MJD patients have a trinucleotide repeat expansion extending above a critical threshold of 55 repetitions (from 55 to 87) [Gu et al., 2004, Maciel et al., 2001]. Larger repeat expansions have been strongly correlated with earlier disease onset and increased severity in clinical manifestations [Schöls et al., 2004, Maciel et al., 1995].

2.1.2 Symptomatology

MJD is a multissystem neurodegenerative disorder predominantly affecting:

- the cerebellum, that plays an important role in motor control;
- the pyramidal and extrapyramidal tracts, responsible for coordinating conscious controlled face and body movement and regulating posture and involuntary motor functions control of muscles, respectively;
- the oculomotor system, responsible for maintaining control of eye movement and visual stability.

Prevalent clinical manifestations of MJD include ataxia, i.e. impaired coordination of voluntary movement that impacts gait, balance, speech and gaze [Taroni and DiDonato, 2004]. Progressively over time, patients exhibit nystagmus, dysarthria and dysphagia [Paulson, 2012]. Furthermore, external progressive ophthalmoplegia, dystonia, intention fasciculation-like movements of facial and lingual muscles, and bulging eyes, pertain to moderately infrequent manifestations of the disease [Bettencourt and Lima, 2011]. In advanced disease stages, patients are bound to a wheelchair and have aggravated symptoms.

The mean age of disease onset is at around 40 years although it may be widely variable, with extremes as low as 4 and as high as 70 years old. The variability is a reflection of differences caused by the size of the mutant CAG repeat, as aforementioned in Subsection 2.1.1, which also accounts for severity discrepancies of MJD phenotype [Bettencourt and Lima, 2011, Maciel et al., 1995]. Due to clinical heterogeneity, MJD was divided into four phenotypical subtypes with different ages of onset (types I, II, III or IV), with a fifth additional one (type V) proposed later on [Lysenko et al., 2010]. Among the subtypes, MJD type I has the earliest onset, 5 to 30 years of age. MJD type II is the most common, with an onset of around 36 years of age. The second most common is type III, with a later onset age of around 50 years old. MJD type IV

age of onset is described as 38 to 47 years, and it occurs in patients with fewer CAG repeats expansions. This classification might not apply to the totality of MJD patients, and in some cases a patient can evolve from one type to the other [Nóbrega and de Almeida, 2012]. Mean survival time following disease onset is 21 years, reviewed in [Bettencourt and Lima, 2011].

2.1.3 Neuropathology

MJD's typical neurodegenerative profile reveals neuronal loss across selective regions of the Central Nervous System, these include cerebellum (spinocerebellar pathways and dentate nucleus), spinal cord, substantia nigra, thalamus, striatum, brainstem (pons and medulla oblongata) and cranial nerves, as Figure 2.1 portrays [Durr et al., 1996, Scherzed et al., 2012]. Brains of MJD patients appear, in many cases, to be considerably reduced in weight in comparison to those of individuals without history of neuropathological or psychiatric diseases [Iwabuchi et al., 1999].

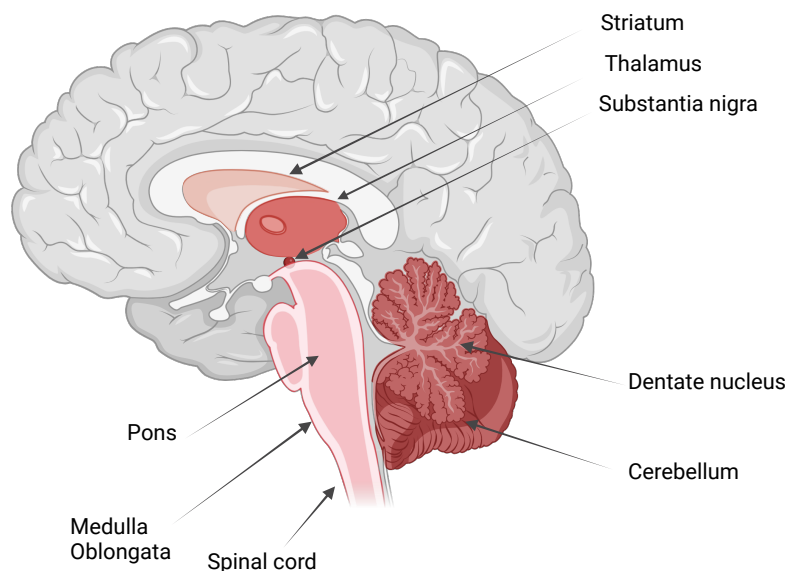


Figure 2.1: Schematic representation of distribution of neuronal loss in the CNS of MJD patients.

2.2 Animal Models of Machado-Joseph Disease

The development of cellular and animal disease models has proved to shed insight on the underlying mechanisms involved and helped in the testing of therapeutic strategies. Considering a disease such as MJD, using animal models presents the possibility of analysing pathogenic mechanisms and aspects with a late onset and during various disease stages, which is unable to

be conducted in human patients. Furthermore, animal models allow the assessment of possible treatments at a pre-clinical stage.

Despite the advantages of the existing models, each one presents its own drawbacks. It is an on-going challenge to create a model that simulates disease features the most faithfully. Organisms such as the fruit fly (*Drosophila melanogaster*) and the roundworm (*Caenorhabditis elegans*) have successfully been used to study several aspects of MJD [Schmidt and Schmidt, 2018]. However, the mouse (*Mus Musculus*) model is the most commonly used, due to their physiological similarities with humans [Ellenbroek and Youn, 2016].

2.2.1 Mouse Models

Transgenic animals expressing mutant human ataxin-3 comprise the majority of MJD mouse models. The transgene frequently consists of a complementary DNA (cDNA) encoding for a particular isoform and controlled by a specific promoter. These models differ in terms of CAG number, promoter selection, the number of integrated transgene copies, and the transgene itself, which may be the full-length or truncated Ataxin 3 protein (ATXN3) as discussed in [do Carmo Costa and Paulson, 2012].

Ikeda and collaborators developed the first mouse model for Machado Joseph disease in 1996, which has proved to imitate disease late stages while being cost and time effective [Ikeda et al., 1996]. Nonetheless, the model holds some limitations. Protein expression is restricted to a specific brain region, contrasting with the human condition, in which neuronal failure is observed across multiple brain regions. Furthermore, the model's high symptom severity and early onset might pose an obstacle to certain therapeutic approaches.

In an attempt to more accurately recreate the human disease features, Cemal and colleagues developed a novel mouse model, the YAC model. This model expresses the entire human ATXN3 coding sequence, exhibiting a moderate phenotype with tremors, wide gait, difficulties regarding body positioning and reduced weight gain. Ultimately, a model able to mimic and correlate numerous characteristics observed in human patients [Cemal et al., 2002].

As observable in Table 2.1, multiple alternative mouse models have been produced whilst exploiting different promoters and different transcript variants of ATXN3 expressions. In general, every model has contributed to advances in the understanding of MJD. Similar to what is observed in humans, the number of CAG repetitions is suggested to be correlated with the severity of the disease [Bichelmeier et al., 2007, Silva-Fernandes et al., 2010, Silva-Fernandes et al., 2014].

Truncated forms of the gene are more toxic, associated with earlier onset and more severe phenotype. This found toxicity, in addition to being related to the C-terminal truncated forms, is linked to an ATXN3 N-terminal fragment that promotes neurodegeneration, behavioural issues and early death [Hübener et al., 2011].

Aside from the transgenic mouse models of MJD, lentiviral-based mouse models have also been developed and have had an impact on disease pathogenesis study. Lentiviral-based mouse

models overexpress full-length human mut-ATXN3 allowing rapid testing of treatment strategies and quantitative analysis of neuropathological impairments. The fact that this model requires invasive procedures to produce a localized overexpression of the disease protein comprises its main drawback [Alves et al., 2008, Alves et al., 2010, Nóbrega et al., 2014, Simoes et al., 2012].

Table 2.1: Mouse Models of Machado Joseph Disease.

Transgene	Promoter/ Expression	Neuropathology	Motor Phenotype	Reference
Truncated and full length Human ATXN3 CDS with 79 CAGs	L7 promoter: expression in cerebellar Purkinje cells	Cerebellar atrophy	Ataxia Gait disturbances (4 weeks)	[Ikeda et al., 1996]
Full length human ATXN3 gene with 48 / 84 CAGs	Endogenous human promoter in YAC system – Ubiquitous expression	Intranuclear inclusions in Purkinje cells, pontine and dentate neurons; Neurodegeneration	Motor deficits (4 weeks)	[Cemal et al., 2002]
Full length ATXN3 CDS with 71 CAGs	Mouse prion promoter –Expression in several brain regions	NIIs in cerebellar nuclei and spinal cord. Abundant mutATXN3 fragments	Motor deficits, weight loss, premature death	[Goti et al., 2004, Gould et al., 2007]
Full length ATXN3 CDS with 70/ 148 CAGs with or without NLS and NES	Mouse prion promoter – Expression in several brain regions	Degeneration of Purkinje cells, reduced turnover of dopamine and serotonin.	Tremor, reduced activity, premature death.	[Bichelmeier et al., 2007]
Full length ATXN3 CDS with 79 CAGs	Mouse prion promoter - Expression in several brain regions	NIIs in pontine and dentate neurons, substantia nigra. Downregulated gene expression	Reduced motor coordination, ataxic gait (5-6 months) and weight loss	[Chou et al., 2008]
Truncated ATXN3 CDS with 69 CAGs	L7 promoter - expression in cerebellar Purkinje cells	Severe cerebellar atrophy with defects in Purkinje cells synaptic transmission	Severe ataxia	[Torashima et al., 2008, Oue et al., 2009]
Full length ATXN3 CDS with 77 CAGs under control of Tet-Off system	Hamster protein promoter - expression in several brain regions	NIIs in cerebellar cortex Neuronal dysfunction in the cerebellum	Reduced motor coordination, Gait ataxia (2 months)	[Boy et al., 2009]
Full length ATXN3 CDS with 148 CAGs	Rat huntingtin promoter (brain)	NIIs in pons and cerebellum, Degeneration in Purkinje	Hyperactivity, Motor deficits (1 year)	[Boy et al., 2010]

Full length ATXN3 CDS with 94 CAGs	CMV promoter - Ubiquitous expression	Neuronal atrophy and astrogliosis, no inclusions	Motor coordination impairments (4 months)	[Silva-Fernandes et al., 2010]
Truncated N-terminal ATXN3 CDS	Endogenous mouse ATXN3 promoter - expression in several brain regions	Neuronal cytoplasmic inclusions, impaired endoplasmic reticulummediated unfolded response	Premature death, tremor, claspings, gait ataxia, weight loss	[Hübener et al., 2011]
Full length ATXN3 CDS with 135 CAGs	CMV promoter - ubiquitous expression	Neuronal loss in pontine nuclei	Motor deficits (16 weeks), Weight reduction	[Silva-Fernandes et al., 2014]
Full length humanized ATXN3 CDS with 91 CAGs and small portion of human 3'UTR region	Endogenous mouse ATXN3 promoter – Ubiquitous expression	Inclusions in the cerebellum, cortex and hippocampus, neuroinflammation, cerebellar neurodegeneration	Mild motor coordination impairment (90 weeks)	[Switonski et al., 2015]
Full length mouse ATXN3 gene with 82 CAGs	Endogenous mouse ATXN3 promoter - ubiquitous expression	Abundant nuclear inclusions	No behavioural alterations	[Ramani et al., 2015]

2.2.2 MJD YAC Mouse Model - Phenotypical Profile

As described in Section 2.2.1, the YAC mouse model is recognized for its capacity to replicate certain characteristics witnessed in human MJD patients. Thus, it has emerged as the animal model that has incited our investigative interest within the context of the current study.

Cemal and colleagues developed several transgenic lines of YAC expressing the full length human mut-ATXN3 gene with expanded CAG repeats ranging in size from 48 to 84 repeat units. Five hemizygous transmitting lines of the transgenic founder mice were originated, namely MJD15.4, MJD72.1, MJD67.2, MJD22.1/84.1 and MJD84.2. Of the five transmitting lines, MJD84.2 mice exhibit the most evident functional and behavioural abnormalities.

The YAC mouse model can be designated as homozygous or hemizygous for MJD/SCA3, depending on their Copy Number Variation (CNV). Hemizygous mice carry two copies of the YAC-transgene and two endogenous genes. Homozygous mice, on the other hand, are obtained through breeding between hemizygous mice and thus have four copies of the transgene. Hence, crosses between MJD84.2 mice result in MJD84.2/84.2 homozygous mice, respecting the Mendelian inheritance ratios.

In order to phenotypically characterize their newly developed animal models, behavioural studies were conducted based on the SHIRPA protocol. In general, Cemal et al. observed that the mutated MJD1 gene in YAC mice was associated with lower body weight in comparison to

wild-type littermates [Cemal et al., 2002]. Other characteristics include limb claspings, reduced grip strength, tremor, abnormal gait, hypoactivity, an inability to correct geotaxis, and abnormal toe pinch responses. These may vary from mild to prominent phenotypic expression depending on copy number, which is consistent with the cerebellar neurodegeneration observed in MJD patients.

In fact, hemizygous MJD84.2 mice present abnormalities in gait and pelvic elevation from 4-6 weeks of age, also showing limb claspings from 24 weeks in forelimbs [Shakkottai et al., 2011, Cemal et al., 2002]. Signs of weakening in the limbs and/or lack of coordination were noticeable from 56 weeks. Cemal and colleagues further demonstrated that homozygous MJD84.2/84.2 mice portray more frequent limb claspings in both hind and forelimbs than hemizygous mice, noticeable by 16 weeks of age. They also observed these mice depict muscle weakness and inability to correct geotaxis from 12 weeks on, showing also excessive grooming. When compared to homozygous mice, all the hemizygous YAC transgenic lines exhibit less pronounced gait abnormalities and lower weight deficit [Cemal et al., 2002]. Moreover, in comparison to each other, hemizygous lines with larger repeat lengths have more prominent behavioural and functional abnormalities. Hence, mice with a higher gene dosage and expanded CAG repeats have earlier disease onset and exhibit more severe symptoms [Cemal et al., 2002]. This observation confirmed the impact of repeat expansions and dosage on disease severity in human patients. Since initially characterized by Cemal and colleagues, several studies have corroborated that MJD84.2 mice, both hemizygous and homozygous, show reduced weight gain compared to wild-type littermates [Shakkottai et al., 2011, do Carmo Costa et al., 2013]. Other findings further report hemizygous MJD84.2 mice display a mild gait abnormalities phenotype and homozygous MJD84.2 mice portray an early decrease in locomotor and exploratory activity on the open field test [do Carmo Costa et al., 2013],

We conducted a meta analysis review of mouse models of MJD which confirmed the YAC mouse model exhibits progressive motor dysfunction. Our investigation was focused on motor performance analysis, evaluated using the accelerating rotarod and footprint tests. In the accelerated rotarod test, lower latency to fall is an indicator of motor dysfunction. Whereas in the footprint test, stride length is evaluated aiming to depict gait abnormalities which are associated with MJD.

Our observations indicated that 10 week-old and 18-week-old YAC mice did not exhibit significant distinctions from wild-type mice in terms of accelerating rotarod performance and footprint stride length, respectively. However, it is noteworthy that 18-week-old YAC mice demonstrated a diminished latency to fall from the accelerating rotarod, while mice aged 23 to 24 weeks displayed a reduction in stride length during footprint analysis [Silva* et al., 2023].

These results strongly suggest that phenotypic alterations in the YAC MJD84.2 model have a representative phenotype from approximately 18 weeks on. Furthermore, we denoted that motor abnormalities in YAC mice might not manifest consistently in individuals younger than 10 weeks old. Our meta analysis review description of the MJD84.2 line corroborates the later onset of this model.

Henceforth, the model developed by Cemal and his colleagues has become a valuable tool for the investigation of therapeutic strategies through facilitating a comprehensive assessment of the genotypical and phenotypical characteristics of Machado Joseph disease.

Automation of Behaviour Analysis

In this chapter, we introduce a brief explanation of behaviour and its importance for neuroscience research, particularly in MJD. We present an overview of the current tools available for behaviour testing and analysis automation.

3.1 Animal Behaviour Tracking

Behaviour is the aggregate of the responses made by the organism in any given situation. It is how animals act, whether they are being constrained or moving freely in an environment. Neuroscience strives to understand brain function, its success is not merely in discovering the underlying mechanisms of the brain, but in connecting them to brain function - behaviour.

Effective quantification of animal behaviour is of extreme relevance to the study of neurodegenerative diseases, such as MJD. The relationship between cerebellar neurodegeneration and motor impairments is widely known and is object of intensive investigation.

It is possible to separate behaviour study into two distinct approaches. On the one hand, observation of behaviour in natural environments, with the purpose of understanding innate behaviour of individuals and common rules of interaction of the animals among each other. On the other hand, behaviour can be studied in a more controlled environment which is the case of laboratory settings. This second approach aims to reduce variability through settings' control to induce a particular outcome or to provoke a particular behaviour response.

In laboratory settings, to quantify the change in animal behaviour we must register their interactions and reactions to the environment. Technology for acquiring records of these occurrences (videography, photography, audiography, and others) made this registration possible. Taking advantage of advances in technology, researchers began to use enhanced tools that provided information beyond the reach of the naked human eye, such as infrared illumination and high-speed video. A more common practice in animal experiments, due to their complexity, consisted in marking each individual with sensors, some examples are GPS, accelerometers or RFID tags that transmit information over extended periods of time [Gomez-Marin et al., 2014, Peleh et al., 2019].

Our work will primarily focus on markerless approaches for behaviour tracking due to their advantages such as minimal preparation or extensive equipment required and minimal

interference with the subjects.

3.2 Behavioural Video Tracking Analysis

Videography is extremely popular among researchers for behavioural analysis as it allows for postliminary analysis. Recorded files entail the examination and extraction of meaningful information to study and draw conclusions from the animal's behaviour. In the past, researchers would often conduct data collection manually, which involved labour-intensive processes such as watching video playback, annotating animal movement times and locations, visually recognizing behavioural patterns, and manually scoring the collected data for events of interest. This approach is not only time-consuming but also prone to human error and subjectivity. Manual scoring and analysing parameters might vary across laboratories and even suffer changes when conducted by the same expert researcher at different points in time. Furthermore, such analysis requires expert knowledge about the species, disease, and experimental design at study. These methods regularly result in laboratories examining only a small subset of the sets of videos recorded during the experiment and storing substantial amounts of unused data. The described challenges associated with low reproducibility of research and its manual costs have had an impact on the field, with a vast number of scientists resorting to more easy-to-use measures in their experiments [Luxem et al., 2023]. These inherent constraints limit the potential of investigations to improve our comprehension of the intricate relationship between brain and behaviour.

In response to the aforementioned issues of heavy amounts of data for manual analysis and user bias, many advancements in the behaviour analysis field have arisen with several degrees of complexity and adaptability to laboratory settings.

Some scientific experiments require processing data of heterogeneous nature, such as video cameras, microphones, and electrodes that must simultaneously record data. Various tools to aid in data acquisition are available nowadays: open-source data collection systems such as Bonsai RX [Lopes et al., 2015], custom-written programs using well known libraries such as OpenCV [Bradski, 2000] and video recording software provided by video capture boards (eg. loopbio Motif). Subsequently, video recordings can be subjected to analysis using a range of tools.

Commercial products for automated annotation and tracking behaviour appeared (eg. Ethovision by Noldus, ANY-Maze by Stoelting), followed by open-source software tools (eg. JAABA, VSAMBR, Optimouse) both aiming to address the new era of behaviour analysis. Opposed to commercial software, open source software holds the advantage of making customisable modifications to the tool's algorithm according to the research or experiment specific purpose.

The available open source tools vary in the type of analysis conducted, either concentrating in 'centre of mass' tracking, in pose estimation or in extracting features for behaviour detection and subsequent quantification. This greatly influences the algorithm used, which might be deep learning based or non-neural network machine learning based. In this work, we will focus mainly on pose estimation algorithms based on deep learning. These concepts will be further explained in Chapter 4.

3.2.1 Experiment Control Systems

Taking a step back from post-experiment analysis, many experiments require tight control of laboratory settings in order to induce the desired behavioural outcome from the subject being tested. To attend to the needs of neuroscience behavioural research, hardware and software experiment control systems, both commercial and open-source technologies, have been developed to this extent. For defining complex state dependent tasks (where the current state determines how the task responds to events) and for simplifying the construction of complex experiment setups, open source systems (pyControl [Akam et al., 2022], Bpod [Sanworks LLC, nd]) have been made available. Different approaches such as Bonsai address experimental control in a very different form, as it functions through processing various data types in parallel.

3.2.1.1 Data Collection System - Bonsai

Bonsai RX was developed to solve the problem of monitoring several parallel data streams for one experiment in an efficient way, including video tracking of behaviour [Lopes et al., 2015]. It is a visual programming language, developed in a way that a researcher with little to no experience in programming is able to use it. This tool can be adapted to several different experimental scenarios that the researcher can build and combine with the Bonsai framework.

Through a Graphical Use Interface (GUI) the Bonsai framework uses observable sequences to represent asynchronous streams of data. An observable sequence is a data stream of elements that follow one another, such as a sequence of frames in a video recording. The sequences of data are manipulated and analysed as they arrive, through the graphic dataflow representation. Each node in the dataflow represents an observable sequence. Nodes can be sources of observable data or combinators, handling raw data streams or multiple sequences of data, respectively. Based on the manipulation of their inputs, combinators can be grouped into transformers, sinks or operator types. Bonsai incorporates visualizers to aid the debugging and inspection of data elements in the workflow. Essentially, this represents a graphical and adaptable approach for working with observable sequences in behaviour tracking [Lopes et al., 2015].

3.2.2 2D Pose Estimation and Tracking

Pose tracking techniques aim to derive positional data from the body parts of a moving animal, i.e. the algorithms can be understood as a function that decodes the frames from recordings to coordinates of various body parts [Mathis et al., 2020]. The body parts are represented as a list of keypoints, and these normally have a semantic meaning attached, e.g. nose, left ear, tail base, or tail tip (Figure 3.1). The algorithms might also group the keypoints, called part grouping, for the case of multiple animals, so that these can be extracted simultaneously. The coordinates of each body part are estimated through finding the pixel with the highest probability [Luxem et al., 2023, Isik and Unal, 2023, Mathis et al., 2020].

Deep learning algorithms, although others have been introduced, have proved to be the most powerful for the task of pose estimation [Moeslund et al., 2006, Poppe, 2007]. One can describe pose estimation as a system of an encoder that extracts relevant features from an image

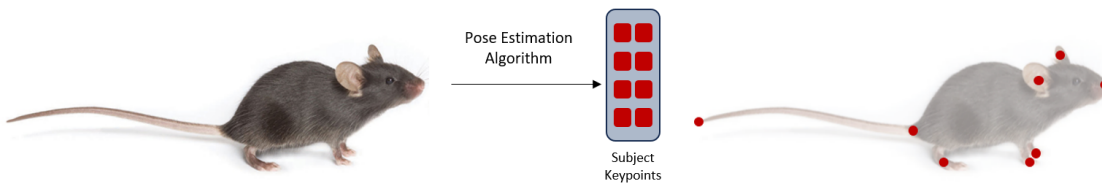


Figure 3.1: Schematic Overview of Markerless Motion Capture.

The pixel representation of an image or a sequence of images (left) is converted into a list of keypoints (right). Semantic information associated to the object and keypoints are linked to the predictions.

or frame, which a decoder will then use to predict the points of interest (body parts) together with the coordinates in said frame. Tools based on deep neural networks optimized for pose estimation tasks name the encoder the backbone and the decoder as the output head.

Machine learning systems consist of a dataset, model, loss function and an optimization algorithm [Goodfellow et al., 2016]. The dataset assigns a particular pose (output) from a particular image (input), defining the desired input-output relationship, that is then used to train the model. Parameters of the model or weights are iteratively updated by the optimizer in order to bring the loss function to its minimum. Hence, the loss function is the measure of quality of the predicted poses, when compared to the ground truth data (manual annotations of body parts).

3.2.2.1 Current Deep Learning Tools

The overall procedure for pose estimation in computer vision research follows a similar strategy. However, researcher-oriented toolboxes differ from one another by providing code to run inference and train on an architecture, for one’s specific datasets. Nonetheless, these two crucial characteristics alone are insufficient to generate suitable neural network toolboxes for a neuroscience laboratory or researcher. Current packages explore different state-of-the-art architectures whilst providing a comprehensive pipeline, accommodating functionalities for labelling a customised dataset, generating the train and test fractions, data augmentation and loaders, neural architectures, performance evaluation code, running video inference and post-processing tools for easy interpretation of obtained outputs. Here, we highlight five state-of-the-art open source toolboxes that are mice specific or species agnostic and concentrate their efforts on usability as well as achieving near human accuracy in pose estimation, as summarized in Table 3.1.

DeepLabCut

Through exploiting a state-of-the-art neural network method for multi-human pose estimation, Mathis and his collaborators generated a pose estimation toolbox targeting animal body parts detection in laboratory settings [Mathis et al., 2018]. To develop DeepLabCut, the features detectors of DeeperCut, a pose estimation algorithm with distinguished performance, were

Table 3.1: Overview of Popular Deep Learning Tools for Animal Motion Capture. Adapted from [Mathis et al., 2020].

	Any Species	Multi-Animal	Training Code	Full GUI	Example Data	PT-NNs	Release Date
DeepLabCut	yes	yes	yes	yes	yes	many	2018
LEAP	yes	no	no	no	no	no	2018
DeepPoseKit	yes	no	yes	partial	yes	no	2019
SLEAP	yes	yes	yes	no	no	yes	2022
B-KinD	no	yes	no	no	no	self supervised	2022

Here, we indicate whether each tool may be used to build customized networks or if it only offers tools for specific animals. Only when pre-trained neural networks (PT-NNs) that go beyond humans are accessible, did we highlight them. We include the code’s release date. Note that SLEAP has replaced and deprecated LEAP.

used [Pishchulin et al., 2016, Insafutdinov et al., 2016]. This was made feasible through transfer learning, as the feature detectors were built upon extremely deep neural networks pretrained on ImageNet, a large-scale dataset widely used for benchmarking object recognition algorithms [Deng et al., 2009].

DeepLabCut (DLC) is a deep convolutional network that combines the strengths of pre-trained ResNets and deconvolutional layers, two components of object recognition and semantic segmentation algorithms. The default network architecture comprises modified ResNets, originally trained on the ImageNet dataset [Deng et al., 2009]. DeepLabCut leverages deconvolutional layers to up-sample visual information in order to generate spatial probability densities, rather than delivering the classification layer output of the ResNet. Spatial probability densities indicate the likelihood of body parts in a particular location. For it to be fine-tuned to a specific task, the network’s weights are adjusted through iterative training on labelled data. During training, the weights are readjusted so that annotated keypoints (body parts or other objects present in the frame) are allocated high probabilities in detriment of unannotated keypoints. Through this method, the network is able to acquire feature detectors specific to labelled keypoints [Mathis et al., 2018].

Later, Mathis and his colleagues introduced structural changes to DeeperCut’s feature detector, which had not been done in the original DLC implementation [Mathis et al., 2021]. Different types of backbone networks were added, MobileNets and EfficientNets, however for the scope of this thesis only the original ResNet backbone was leveraged.

Social LEAP and LEAP

The deep learning-based framework SLEAP (Social SLEAP) is inspired by the earlier method LEAP (LEAP Estimates Animal Pose), which focused on single animal pose estimation [Pereira et al., 2019, Pereira et al., 2022]. The previous LEAP method consisted of a fully convolutional neural network, that consisted of 15-layers of repeated convolutions and pooling. SLEAP is able to employ several fully convolutional neural network configurations, the default architecture type is a version UNet. Described as a simple encoder-decoder framework, it uses repeated convolutional and pooling layers that produce a set of probability distributions for the location of each body part in an image. To estimate the confidence maps of the keypoints’ locations during training, the model makes use of up-sampling or deconvolutional layers. Addi-

tionally, SLEAP allows the use of fixed network architectures (ResNet, MobileNets and others) as the backbone of the encoder of the model, connecting intermediate-layer activations with the decoder portion, thus enabling transfer learning.

DeepPoseKit

DeepPoseKit’s model is a novel implementation of fully convolutional DenseNet or FC-DenseNet [Jégou et al., 2017] inspired by stacked hourglass network [Newell et al., 2016], named Stacked DenseNet. While also using the encoder-decoder design through convolutional layers downsampling the input, this model adds intermediate supervision at the output of each network. Intermediate supervision involves additional auxiliary loss functions at intermediate layers of the network, aiming for these layers to learn more distinctive features to improve the overall performance of the network [Graving et al., 2019].

B-KinD

Other approaches such as **B**ehavioural **K**eypoint **D**iscovery (B-KinD) [Sun et al., 2022] utilise self-supervised learning to decrease the burden of labelling frames, inherent to all deep learning methods. This method also uses an encoder-decoder setup. However, instead of focusing on image reconstruction, it investigates a reconstruction target around spatiotemporal differences. In behavioural recordings, the background is usually static, and the objects of interest are the only ones moving, e.g. a mouse moving around in an open field arena. Taking advantage of the objects of interest providing the only source of movement, Sun et al. infer location and movement through the spatiotemporal differences [Sun et al., 2022]. However, occlusions in video recordings might pose a challenge for this method and similar.

Deep Learning Methods

4.1 Machine Learning

The field of Machine Learning (ML) is a subfield of Artificial Intelligence. This method utilizes mathematical models of data to facilitate computer learning without explicit instruction [IBM, 2020]. It involves the development of statistical models and algorithms that enable computers to increase their performance related to a specific task through past experience. In simpler terms, the computer learns and improves in a way that mimics human reasoning.

Machine learning can be categorised based on learning nature in supervised learning, unsupervised learning and reinforcement learning [Azure, nd].

- In **Supervised Learning**, the models are designed to learn through examples. Models are trained with labelled data, i.e. each input corresponding to a paired output. During training, the model adjusts its weights until an appropriate mapping function is achieved. The model is then used to classify data or predict outcomes. Neural networks, random forest and support vector machine (SVM) are among some of the methods used in supervised learning.
- In **Unsupervised Learning**, the algorithm aims to discover patterns or similarities in the input data without a human previous labelling or categorising them. The model has the ability to categorise data according to its features, without a ‘correct’ answer for comparison. It is used to association and clustering type problems. Some methods include k-means clustering and k-nearest neighbour algorithms.
- **Reinforcement learning** algorithms learn through trial and error. The model is reinforced with a system of rewards and punishments in response to the model’s performance. The model aims to maximize its rewards.

4.2 Deep Learning

Deep learning is a ML branch based on Artificial Neural Networks (ANN), which are algorithms inspired by the structure of the brain’s neurons. Neural networks have existed for decades, its recent resurgence is attributed to the era of Big Data, an era of exponential growth of information available worldwide, accompanied by advancements in hardware and software that

meet the requirements of these algorithms, large amounts of data and lots of computational power. Unlike traditional machine learning methods where feature extraction must be explicitly programmed, deep learning algorithms can learn to extract features related to a particular task.

Tasks such as object detection, where manual feature extraction is challenging, have been made easier by the automatic feature extraction process. As a result, deep learning has become the state-of-the-art in several tasks, including pose estimation which falls under object detection, with exceptional results that are equivalent to or might occasionally outperform human performance [He et al., 2015, Mathis et al., 2018]. We must first comprehend the concept of artificial neural networks in order to fully comprehend what deep learning is.

An **Artificial Neural Network (ANN)** is a model developed to mimic the human brain. Analogous to synapses propagating signals in the brain, ANNs are composed of units resembling neurons, and connections resembling synapses. The fundamental building block of every neural network is the Perceptron, a single neuron or unit [Rosenblatt, 1958]. A perceptron has internal parameters: a set of weights, applied to its inputs and a bias, which is a constant value. The perceptron takes M amount of inputs and linearly combines them with the respective weights, adds the bias, and finally applies an activation layer to the weighted sum. Weights are employed to give certain inputs greater importance, and the bias shifts the activation function. Weights are used to give more importance to certain inputs for the final decision, and the bias value shifts the activation function. The activation function introduces non-linearity to the decision boundary and approaches the result to complex functions [Amini, 2023]. The scheme and calculation are depicted in Figure 4.1 and Equation 4.1, respectively.

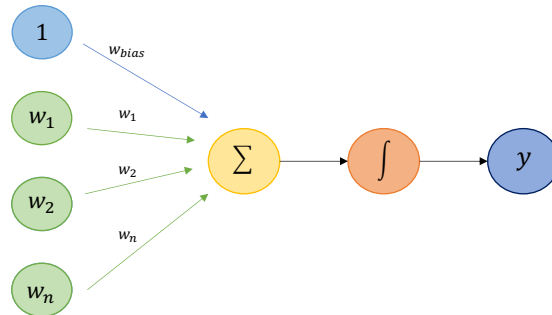


Figure 4.1: Representation of a perceptron.

$$y = h\left(\sum_{i=0}^n x_i * w_i + b\right) \quad (4.1)$$

Perceptrons can be organised together and added in layers where, within the same layer, all the units are connected to all the inputs. If we add several layers of units with connections to each other, we create a multi-layer neural network capable of solving complex non-linearly separable problems. Layers between the input and output are designated hidden layers, and the

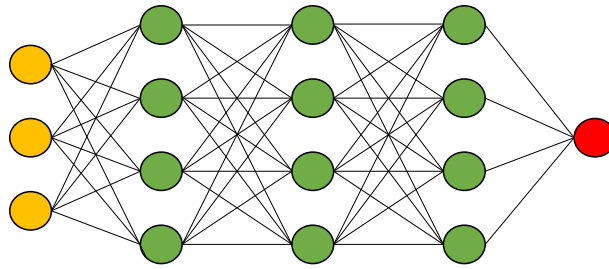


Figure 4.2: Representation of an Artificial Neural Network, with the input layer (yellow), the hidden layers (green) and output layer (red).

number of hidden layers of a network defines its depth (Figure 4.2). The term deep learning refers to the usage of deep artificial neural networks, which in turn are named after their large number of hidden layers.

During training, a network’s performance is improved through backpropagation, a process defined by an optimization algorithm steers the loss function to its minimum. The loss is minimized using a gradient descent algorithm and propagated by moving backward through the layers and their units. The gradient of the units is calculated, and their weights are updated.

There are several types of neural network, such as Recurrent Neural Networks, Feedforward Neural Networks and Convolutional Neural Network (CNN)s. The latter are primarily employed for tasks involving images.

4.2.1 Convolutional Neural Network

Convolutional Neural Networks, also known as ConvNets or CNNs, are a deep learning architecture used for multidimensional data, such as image recognition and classification problems. A CNN is able learn the features of an image and assign different importance to various elements within the image, generating weights used for the final classification. They are able to extract features through convolution and pooling operations [LeCun et al., 2015, Wu et al., 2020].

A typical CNN architecture is structured with three types of layers: convolutional, pooling and fully connected layers.

The first layer, the **convolutional layer**, is composed of filters, or kernels, that ”slide” over the input image and apply a convolution operation to generate a feature map. The kernels may vary in size, but are usually smaller than the input image.

The **pooling layer** is applied in order to reduce the input image dimensionality. Similar to convolutional operations, in pooling operations a filter is used to sweep the input image. However, unlike convolutional layer’s filters, these do not contain weights. Instead, the filter calculates either the maximum value (max pooling) or the average value (average pooling) from the image portion in the receptive field, i.e. the region covered by the filter. Despite some

information being lost in this operation, pooling layers are crucial because they help the CNN reduce the complexity and increase training efficiency.

Multiple layers of convolutional and pooling layers can be repeated and mixed in various combinations along the network. As the data passes through the layers, it becomes increasingly complex which allows the detection of more complicated features. The final layer of a CNN is a **fully connected layer**, where a unit is connected to all the units of the previous layer. The features extracted in these past layers are used as inputs to make decisions such as classification.

Several CNN architectures have been developed and gained popularity in the scientific community for image classification, including VGG [Simonyan and Zisserman, 2014], AlexNet [Krizhevsky et al., 2012], ResNet [He et al., 2016] and MobileNet [Howard et al., 2017]. In fact, these models have been trained on very large generic datasets, such as ImageNet [Deng et al., 2009]. A common approach in deep computer consists of using these pre-trained CNNs by reusing their weights for the task at hand, through transfer learning techniques. In fact, pose estimation algorithms in neuroscience typically use pre-trained CNNs, such as MobileNetV2s [Sandler et al., 2018], ResNets [He et al., 2016], DenseNets [Huang et al., 2017] or EfficientNets [Tan and Le, 2019].

ResNet

ResNet, short for Residual Neural Network, is part of a family of deep CNNs developed to address a common problem in very deep networks, the issue of vanishing gradients. The underlying idea of ResNets consists of incorporating residual blocks that allow for the propagation of gradients through the network. Instead of learning the whole mapping, the network can learn residual functions that represent the difference between the convolutional layers' input and output [He et al., 2016]. There are different ResNet models composed of multiple residual blocks, such as ResNet50, ResNet101 and ResNet152. For instance, ResNet50 has fifty layers using these blocks.

4.3 Transfer Learning

Transfer learning is a machine learning technique where an already trained model is repurposed for a novel task. Its basic concept relies on the idea that the knowledge a model has learned may be exploited in order to solve a new but related problem. This capability proves useful in scenarios where data availability for training a model from scratch is limited, as well as when we are confronted with a new task closely aligned with the original one where the pre-trained model can be reused to new task with minimal modifications.

Transfer learning strives to transfer as much knowledge as possible from the prior task on which the model was trained to the new task at hand. Depending on the problem and the data, this knowledge might take numerous forms. For example, the animal pose estimation toolbox DeepLabCut (Mathis et al., 2018) modified DeeperCut [Insafutdinov et al., 2016], which was built with a ResNet backbone network but modified stride the by dilated convolutions [Chen

et al., 2017] to retain a higher spatial resolution.

When compared to training a model from the ground up, transfer learning normally diminishes convergence time [Arac et al., 2019, He et al., 2019, Mathis et al., 2021] and increases robustness [Mathis et al., 2021].

Methodology

The methodology adopted to conduct this project is outlined in this chapter. Firstly, we provide details of the dataset, including the animals used (Sections 5.1 and 5.3). In Section 5.2, we briefly describe the open field behaviour assay. Section 5.4 provides a description of the experimental setup for dataset acquisition, the hardware and software used for this work. The following Section 5.5, goes through the preprocessing necessary in all our recordings. The subsequent Section 5.6 delineates the overall experimental settings employed in our analyses. Lastly, Section 5.7 outlines our developed analysis systems, firstly describing the Bonsai RX pipeline, followed by the training of the network model using the DeepLabCut toolbox.

5.1 Mice

Twelve MJD84.2 homozygous YAC transgenic mice [Cemal et al., 2002] and 4 wild-type littermates were used to perform open field test, at 36 weeks of age (9 months), as demonstrated in Table 5.1. Out of the twelve YAC MJD mice, half of them were treated with Phosphate-buffered saline (PBS), while the remaining half underwent prior gene therapy treatment to assess therapeutic efficacy. We decided to employ the term "MJD Homo YAC mouse" or "Homo YAC mouse" when referring to the MJD84.2 homozygous YAC mice in this experiment.

Table 5.1: Number of animals divided by condition.

Mouse Group	n
WT	4
Homo YAC PBS	6
Homo YAC Treated	6
Total	16

All animals were housed in a temperature-controlled room maintained on a 12 h light/12 h dark cycle during the study. Food and water were provided ad libitum. The experiments involving mice were previously approved by the Responsible Organization for the Animals Welfare of the Faculty of Medicine and Centre for Neuroscience and Cell Biology of the University of Coimbra, (ORBEA and FMUC/CNC, Coimbra, Portugal), and were carried out in accordance with the European Community directive (86/609/ EEC) for the care and use of laboratory animals. All efforts were made to minimize suffering.

The wild-type mice acted as a healthy control group due to their unaltered genotype or phenotype. Additionally, a separate control group consisted of YAC transgenic mice, which were treated with Phosphate-buffered saline (PBS). Members of this control group received a placebo treatment (PBS), which helped us assess the specific impact of the administrated therapy on the experimental group of YAC transgenic mice.

5.2 Behavioural Test: Open Field Test

The mice performed a one time open field behavioural test at 36 weeks old, before they were euthanized. It is one of the most basic tests to assess locomotor and exploratory activity which consists of observing and documenting the animal's movements in an open arena.

When placed in the centre of an open space, a mouse tends to head towards the walls on the edge and then navigate its way around the arena, maintaining proximity to the wall. As time progresses and the mouse familiarizes with the new surroundings, anxiety reduces, leading the mouse to venture into the central parts of the arena before returning to the edges. This characteristic behaviour is the foundation for investigating mice's activity in the open field. These experiments usually involve recording the observations so they can later be examined for a number of activities, including mouse trajectory, time spent near the edges or the central parts, time spent exploring or resting, as well as registering measures, e.g. distanced travelled and velocity.

5.3 Dataset

The first step of this work was to collect images for video analysis and inference. The dataset is composed of one video recording of open field test experiments per animal with a duration of approximately 30 minutes. The videos were conducted at a single time point, where mice were recorded during their active period, in an adapted dark room illuminated with red light.

The same resolution, encoding and sharpness were used, however we captured some of the videos with different frame rates. They were recorded in RGB with a resolution of 480x640 pixels.

5.4 Video Acquisition

5.4.1 Experimental Setup for Video Acquisition

Experiments were performed utilising a simple experimental setup that was assembled for the purpose, as presented in Figure 5.1. Reutilising a damaged and inoperable Panlab 50 x 50 cm open field arena, we covered the box's floor with white vinyl paper to create a more noticeable contrast between the animal's dark fur and the white background. A C920 PRO HD Webcam connected to a Windows computer was placed approximately 60cm above the arena.

The webcam was secured by a tripod stand, adjustable in height.

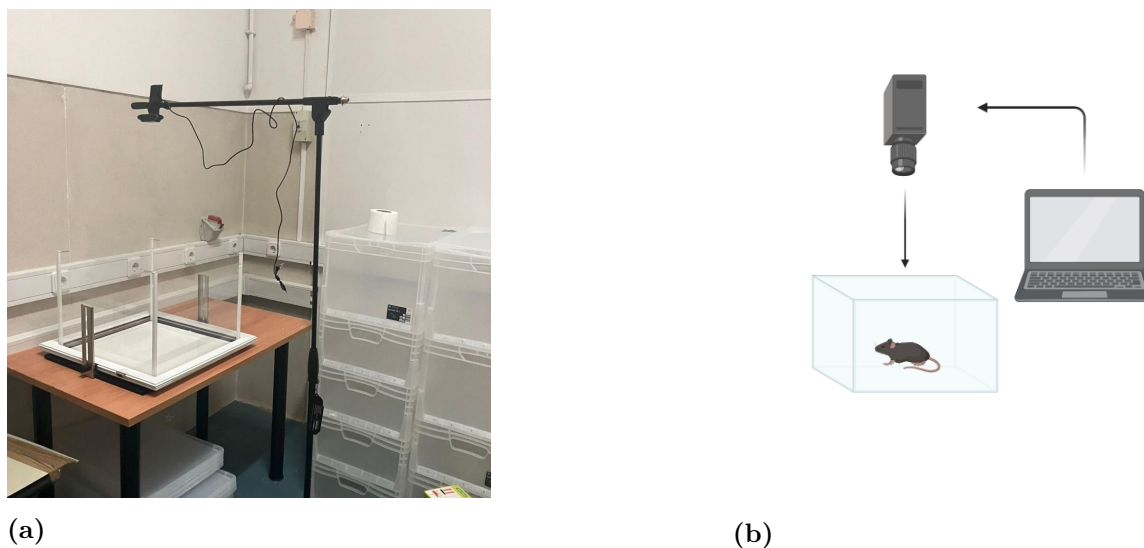


Figure 5.1: (a) Setup for video acquisition of open field test and (b) Schematic of behaviour testing acquisition.

5.4.2 Hardware Tools for Video Acquisition and Analysis

Video acquisition and analysis was performed in an ASUS TUF Dash F15 computer with 8GB of RAM, installed with Windows 11 Pro. The computer's processor is an 11th generation Intel(R) Core (TM) i7-11370H @ 3.30GHz 3.30 GHz. We also integrated the NVIDIA card drivers, CUDA and TensorFlow. Due to its many dependencies that might have conflicted with several versions of software installed for several non-related needs, we created a Conda (Anaconda) virtual environment specific for DeepLabCut toolbox installation.

5.4.3 Software Tools for Video Acquisition

Recordings were conducted using the modular programming framework Bonsai RX, with a simple custom script that allowed us to initiate the recording through keyboard pressing and save the videos in compressed AVI file format.

After connecting the webcam to the computer, the node *Camera Capture* starts generating a sequence of images from the camera. Applying a sequence of nodes, we were able to start and enable the recording by pressing a specified computer key. Using the *Video Writer* node, we could select a specific frame rate for our data collection, which would then be saved into a compressed AVI file in the desired folder.

5.5 Preprocessing - Ffmpeg

FFmpeg is an open source software project designed to handle video and audio processing. Through FFmpeg, we are able to encode, decode, transcode, mux, demux, filter, stream and play most video and audio file types [FFmpeg, nd].

We used FFmpeg to convert the videos from AVI format to MP4 to ensure no there was quality compromise. The AVI to MP4 conversion was necessary due to a problem that emerged when we first started to run the AVI format videos in the Bonsai framework and persisted when running the video analysis using DLC functionalities. The problem consisted in the videos stopping to run at seemingly random times, making it impossible to run the videos until the end as these were somehow corrupted. After all the files were converted to MP4 format, we no longer experienced any issues with the video analysis.

After a quick installation, we chose the specifications for the conversion in such a way that the converted video suffered from minimal quality loss.

For the video files, we used the following command:

```
ffmpeg -i input.avi -c:v libx264 -preset slow -crf 19 -r 30 output.mp4
```

-c:v libx264: sets the video codec, we decided to keep the codec in the conversion

-preset slow: specifies the encoding preset as “slow”. The trade-off between encoding speed and output file size is determined by the presets. Slower presets provide better compression efficiency.

-crf 19: is the constant rate factor that controls the output video quality. In this case is set to 19, normally ranges from 18 to 52. Lower values produce higher quality outputs but large file sizes.

-r 30: sets the desired frame rate for the output video. This value might be influenced by other factors such as the codecs or the input video properties. In our work, this value was specified to maintain the frame rate of the video files, which resulted in all the videos keeping their frame rates.

5.6 General Settings

In both Bonsai and DeepLabCut, we had to set several parameters to be used throughout the experiments, we will call these types of parameters, the experimental settings. These settings influence the metrics calculation and need to be adjusted according to each video properties and experimental setup. Furthermore, these parameters can easily be adjusted to extrapolate the customised code for different types of videos and experimental tests. They are as described below:

- Video Frame Rate - This is a setting that depends on the video file properties and must be adjusted accordingly.
- Conversion Ratio - During the assessment of velocity, distance traveled, and resting time of mice in the open field test, the calculations were performed using pixel-based measurements due to the camera’s inherent pixel-based image capturing capability. Consequently, it became necessary to convert the outcomes of these metrics into our preferred measurement systems for meaningful assessment and interpretation. Hence, a conversion ratio of 0.11

was employed, taking into consideration the width of the arena in centimeters and its width in pixels.

- **Movement Threshold** - For the scope of our problem, we had to define a value for determining whether the mouse was moving. The value set for this movement threshold was 2 cm/s, which for example translates in 0.2 cm per frame, considering a frame rate of 10 frames per second. To determine the equivalent value in pixels, we further need to divide this value by the calculated Conversion Ratio.
- **Time Bins and their duration** - In the open field test and other behaviour assays, it is common practice to separate and categorize them in stages, through dividing the experiments into specific time intervals, often referred to as time bins, with the purpose of grouping certain events or observe patterns. In the open field test, the testing session is divided into intervals of equal duration. As such, it is possible to analyse and compare the mouse's behaviour within each interval. Also, this interval division allows for the examination of changes in activity levels and temporal patterns. Hence, we define if there will be time bin analysis or if the parameters will be evaluated for the whole video and if time bins are chosen, their duration. We decided to, if time bins were enabled, split the test into three time bins of ten minutes each, as the testing sessions were of approximately 30 minutes.

5.7 Development of Video Tracking Analysis System

5.7.1 Protocol Optimization for Bonsai Tracking

Our first approach consisted in analysing all the videos through Bonsai RX 2.7.2 [Lopes et al., 2015]. We leveraged an already existing script, kindly provided by a fellow researcher, Pedro Costa (Peça Lab, CNC-UC), with minimal changes adapted to our problem's needs. With the Bonsai framework, we were tracking the one moving object in the camera's field of view. To create the data processing timeline, Bonsai's node based framework was used, connecting different nodes to define the flow of the data, and the changes applied at each step.

For this purpose, the videos were processed through a sequence of nodes, including pre-processing nodes (such as applying a gray scale filter). Then a node was applied to extract the binary region properties (in our case, the mouse body) and a subsequent node was used to extract the centroid properties that were then used to calculate the desired parameters. Python Transform nodes are used to transform the elements that arrive sequentially, hence, in our work, were used to calculate the distance travelled, average instant speed and time spent still (or resting time).

The resulting output from the nodes was then aggregated, also utilising Python Transform nodes, to be then passed through a Text Writer node, in order to be organized and stored into a text file containing the information calculated in the sequence.

5.7.2 Protocol Optimization for DeepLabCut Tracking

Prior to choosing this tool for analysis, we followed the extensive protocol and pipeline made available by the authors in the tool’s GitHub repository and website [Nath* et al., 2019]. This step allowed us to familiarize with the tool and rapidly attest its utility for our purpose. The following step in our approach was to construct a pipeline adequate for our domain. The subsequent sections provide a more thorough explanation of the labelling and DeepLabCut parameters.

In order to fine-tune the CNN’s parameters, we selected 6 videos of 6 individuals from the dataset, 2 of each experiment group. The remaining videos were used for analysis. No image pre-processing was applied to the videos or images, as DLC is able to receive these without pre-treatment. Lastly, DLC output analysis was performed using custom Python scripts.

5.7.2.1 Pipeline

The implemented DLC pipeline navigates the processes shown in Figure 5.2, beginning with the video frames being extracted from the input videos and manually labelling selected frames as source dataset. Then, several DeepLabCut functions were exploited to train and analyse the neural network, to generate the output from a video. The final output data consisted of the location of tracked body parts in each image (x and y coordinates) and the degree of confidence of the trained network’s identification. We were also able to extract a labelled video where the featured body parts are graphically represented.

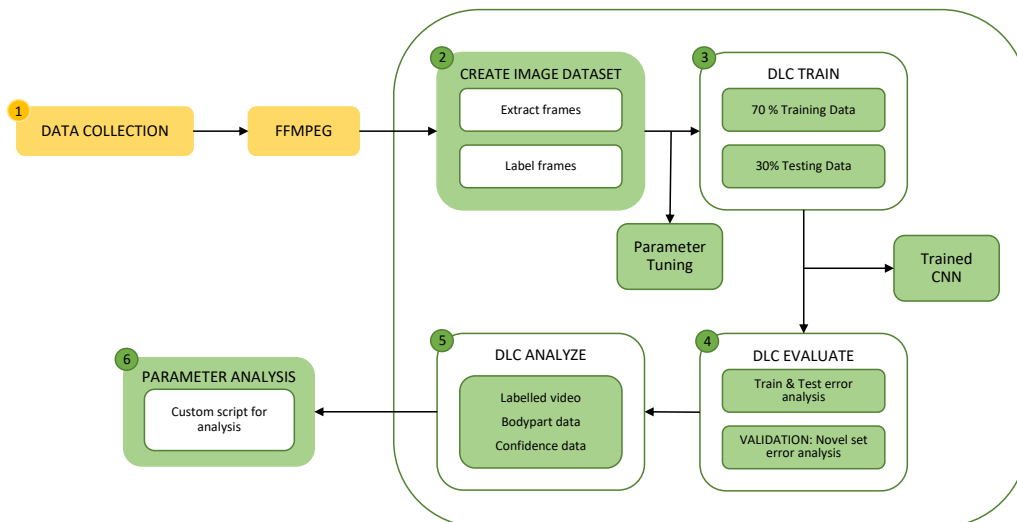


Figure 5.2: Adopted Pipeline for DeepLabCut Network Training and Video Analysis.

The first stage in the entire process, was to select a portion of the video files and extract video frames to create the training dataset. We decided to select six videos, which represent six

individuals, two from each group, i.e. two transgenic animals, two healthy control animals and two treated transgenic animals. The frames were extracted with k-means clustering function of the toolbox. Through k-means clustering method, n frames are assigned to K clusters. Frames from different clusters are then selected, assuring that the frames represent different postures. Hence, our training dataset was composed of 260 images, with 60 images from one of the video files and 40 from each of the remaining five.

The next step consisted of manually labelling the frames. This task was carried out using the Python Napari plugin package for keypoint annotation with DeepLabCut. This stage also entailed the visual verification of the labelling output, with the appropriate DLC function.

The third step was creating the training dataset that was then fed into DLC’s training function. For the totality of the trained models, the images in the dataset were randomly divided into train and test datasets, with sizes of 70% and 30%, respectively. Six training datasets were created, in which both training and testing images were shuffled to introduce further randomness and diversity in the learning process. After network training, the output was presented in a form of so called “snapshots” of the iterations’ checkpoints, containing the weights of the trained CNN.

We then performed the network’s evaluation, through examining the train and test error, with and without p-cutoff. The train and test errors were generated for every saved snapshot from the previous step. This is measure was helpful to draw conclusions on the CNNs error variation throughout the training process.

The video analysis and creation of an annotated video constituted the fifth step. Here, we chose the snapshot with better results, i.e., with smaller train and test errors, and used the weighted CNN from that snapshot to analyse an entire video. A file with the x , y locations and probability of each tracked point was created. With this file, we were able to build an annotated video enabling visual analysis.

The final assignment of the pipeline involved the validation of the labelling accuracy of the CNN, comparing unseen labelled images and the DLC predictions. In this process, we were validating the final output produced in the previous step.

We were able to configure the parameters of the described pipeline through the configuration files made available upon the project creation, using the DLC toolbox. The sections bellow provide more information on the set of parameters and their corresponding values.

5.7.2.2 DeepLabCut Parameters

In this Section, we go through the fundamental parameters for DLC configuration. Naturally, certain factors have greater influence on the outcomes than others. Finding a good set of values for these parameters that performs best in the task at hand is the challenge of most CNNs. We chose to test different values for only a small number of parameters in the configuration files, designated parameters of interest in [Nath* et al., 2019]. In addition to the recommended parameters of interest, we decided to further test the value for batch size, due to

the substantial impact of this parameter in CNNs.

Network Architecture

The available network configuration allows us to select between alternative configurations. The initial weights used for training belong to the variant of ResNet50 model from DeepLab-Cut, previously trained on the popular dataset for image recognition, ImageNet. The pre-trained ResNet50 network was selected as it has proven to be sufficient for studies alike [Lang et al., 2020], with deeper ResNet networks, with 101 and 152 layers, showing merely subtle decreases in the training and testing errors [Mathis et al., 2020]. Available MobileNets have been advised when a GPU is unavailable or a GPU with little memory. EfficientNets require careful handling, such as parameter tuning, and the Resnets have been benchmarked for these types of studies [Mathis et al., 2021]. The parameter *init_weights* (a file containing the specific network weights and layout) must be set. We decided to use the pre-trained ResNet50 supported by the aforementioned literature findings.

Batch Size

The batch size is a hyperparameter that specifies the number of samples that must be analysed before the internal model parameters are updated. The default batch size is 1, Stochastic Gradient Descent. As demonstrated in [Mathis and Warren, 2018], batch processing should improve inference speed. Hence, we decided to test batch sizes of 1, 2, 4, 8 and 16, in order to benchmark the conclusion made by Mathis and his colleagues. However due to hardware constraints, we were unable to train networks with batch sizes beyond a size of 4. As a result and after training the viable batch sizes, we chose a batch size of 2 (demonstrated in Appendix Section A.1).

Global Scale

Before being processed by the CNN, every image in the dataset is re-scaled by a scaling factor, *global_scale*. The values assigned to the parameter *global_scale* range between 0 and 1, with 0.8 as the default value. Since the image is being re-scaled, its resolution is being re-defined, and higher resolution images are slower to analyse. This parameter aims to keep the CNN fast while still maintaining a reasonably high image resolution, so that the performance of the network is uncompromised. As such, we tried to maximize this value to obtain the best results possible. Considering we are using standard definition videos (640x480 pixels), we tested the values of 0.8 and of 0.9 and subsequently analysed the result, as depicted in Appendix Section A.2.

Data Augmentation

Data augmentation is used to, as the name implies, augmenting the dataset when obtaining more data is not possible. With this process, we are artificially expanding the training set by varying the images' scale, through cropping or mirroring them. Several methods for data

augmentation are able to be implemented in the DeepLabCut pipeline.

The default augmentator is `imgaug`, which we chose to keep because it has previously shown good results. `imgaug` randomly varies the scale of images, and also randomly mirrors and crops them. The resolution of the images fed to the CNN is defined by the global scale parameter above. However, on each iteration, the actual scale used on the images is changed according to parameters `scale_jitter_lo` and `scale_jitter_up`. The training function scales the images by the factor `global_scale` randomly modified by a factor between the values of the two parameters. The default values for these parameters are 0.5 and 1.25 respectively.

To configure the mirroring behaviour, we have parameter `mirror`, which is a boolean parameter that allows DeepLabCut to invert (or not) the symmetry of images in the vertical axes. In our case, the focus of the tracking is a mouse, that itself is symmetric around the vertical axis. Hence, in theory, we should set this parameter to `TRUE`. However, we need to remember that the mouse is moving during the video in all directions. So, we already have a high level of symmetry of the mouse included in the training sets.

We also identified left-side and right-side features (ears, hips and side body) in our videos. For this reason, we wanted to avoid interference in the results that may be introduced by this process. We then set this parameter to `FALSE`.

The boolean parameter `crop` defines the application of a cropping mechanism applied during training, and the associated `crop_ratio` parameter defines the size ratio of cropped images. We are able to define the boundaries of the cropping in the configuration file with the parameters `leftwidth`, `rightwidth`, `bottomheight`, `topheight`. The parameter `minsize` determines the minimum size of the cropped image, excluding from training all smaller images. The images are randomly cropped within these boundaries.

The `max_input_size` parameter ensures that images larger than this value are not used for training, which prevents training from crashing with out of memory exception for very large images.

In our trials, thoroughly explained in Appendix Section A.3, we tested the network with parameter `cropping` for both `TRUE` and `FALSE` values. We determined that it did not have a beneficial effect on the results, and therefore set this parameter to `FALSE`.

Iterations

The parameter `display_iters` is used to exhibit the training iterations and respective loss, throughout the training process. It is a helpful tool to assess whether the training is progressing without errors and at a normal rate. At first, we set this parameter to 100, allowing us to inspect the training iterations. It was then changed to 1000 for the training process of all the videos, since we could trust all was running smoothly.

The network's weights are stored every N iteration, with the N value being set in the `save_iters` parameter. The parameter sets the periodicity with which the snapshots (containing

the weights) are stored in a separate file. This allows us to resume training from each saved snapshot. Unfortunately, these snapshots are of considerable size (> 80 MB) meaning saving too many might pose a drawback. We altered this parameter for different training sessions. Ultimately, as we decided to train every CNN for around 200 000 iterations, we set this value to 50 000.

Associated with this parameter, is the *max_snapshots_to_keep*, existing to ensure that not all the saved snapshots are kept, to tackle the possible lack of space problem. Although not the case for the totality of the cases, this parameter was set to 4, establishing that only the last 4 snapshots would be stored, independent of the iterations value when the training was stopped.

Hence, for the most of our cases, when running 200 000 iterations we had 4 snapshots for evaluation and comparison analysis, in order to select the best one.

Distance Threshold

The feature detectors consider all locations that fall within this pixel distance threshold as positive training samples. Setting this value too high, would cause many errors to be fed to the detector phase. Similarly, setting a too low value to the parameter, would mean too few images being accepted. Based on the discussion presented in [Mathis et al., 2018], we decided to subscribe to the default value of 17 pixels.

Likelihood Threshold - *p-cutoff*

DeepLabCut’s output data consists of the coordinates estimated by the network for the location of the labelled body parts accompanied by its confidence of that estimate. This value is denominated the likelihood of a point corresponding to that assigned body part, ranging from 0 to 1.

During training, the likelihood value is compared to the *p-cutoff* parameter. P-cutoff determines the threshold of which the likelihood will be considered, aiding in distinguishing uncertain body parts from likely body parts. As a default, *p-cutoff* is set to 0.1, which results in all labels with a value lower than 0.1 being discarded as good predictions (displayed as circles in a visual output), whilst labels with a likelihood equal to or higher than 0.1 being taken into consideration (visually displayed as dots). Assigning a *p-cutoff* to the network leads to a CNN rejecting the uncertain body parts when forming a labelled video.

Within the scientific community, there seems to have yet been established an acceptable error value. Typically, a commonly adopted criterion is an error rate of 5% or 0.05, which translates to a likelihood of 0.95 in our specific context. However, the default value set by DeepLabCut is 0.1, a drastically distant value from 0.95 for this criterion. Given the absence of consensus and substantial disparities observed among distinct cases, it became imperative to compare results obtained using different likelihood values. As our problem does not pose as an excessively challenging problem for the network and we still intend to obtain reliable results, we decided to test *p-cutoff* value of 0.8. In comparison to the default value (0.1), we found that

the network performed slightly better, as demonstrated in Appendix Section A.4.

5.7.2.3 Frame Labelling

As a result of transfer learning, this tool requires little labelled data. Taking advantage of that, we decided to work with a small dataset of 260 images. Using the Napari plugin for DeepLabCut, we annotated eleven body parts across all images: nose, left and right ear, left side, right side and centre of the body, left and right hip, and tail base, centre and tip, as depicted in Figure 5.3. The choice for these particular body parts was made in light of the behaviour test we decided to address, the open field test, and the surrounding conditions. We are filming the arena from above which entails impeded visibility of the mouse’s paws, a helpful measure of gait posture. Therefore, through labelling the animal’s hips and body’s sides and centre, we intend to extract similar or equally significant measurements for gait analysis.

When occluded in the image, body parts were not annotated but were instead left missing, e.g. when the animal was grooming and facing down, the nose keypoint was left unannotated. Missing labels were reported to be nearly insignificant in corrupting the accuracy of keypoint prediction of the network [Mathis et al., 2020].

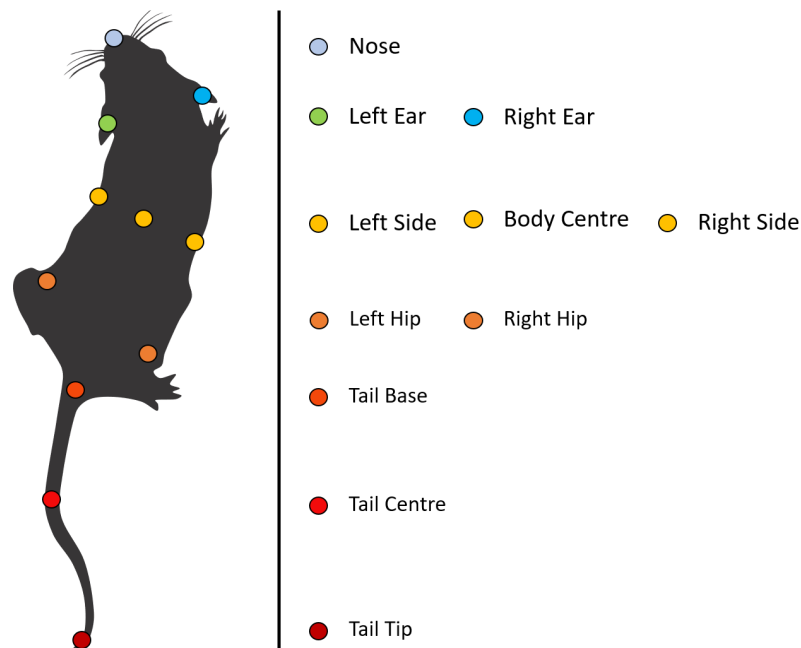


Figure 5.3: Schematic Representation of Mouse Labelled Points.

5.7.2.4 Human Labelling Error

We also investigated the impact of human labelling on introducing errors in the process. To assess this, we labelled the same set of frames twice. Subsequently, we calculated the Euclidean distance for each pair of points. This enabled us to calculate the mean of the Euclidean distances, which we can refer to as the Mean Average Euclidean Error. We chose to execute this error evaluation as this is same evaluation method the DeepLabCut toolbox utilises. This analysis

helped us estimate the amount of error that human variability may introduce into the network training procedure.

This was accomplished by having a colleague label the same set of frames. For this evaluation, we decided to label twice 120 frames from 3 separate videos. Table 5.2 displays the computed values for the x and y axis differences and the mean average Euclidean error. This establishes a lower boundary for what a CNN can be expected to achieve in order to match human-level accuracy (4.49 pixels).

Table 5.2: Human Labelling Average Error in pixels.

	pixels
ΔX	2.94
ΔY	2.79
MAE	4.49

The Mean Average Euclidean Error (MAE) of our selected snapshot, further explained in Appendix A, had a training error with *p-cutoff* of 1.65 pixels and a testing error of 2.94 pixels. These error values provide confidence in the predictions of our trained model, as we believe it has achieved a performance comparable to human standards.

5.7.2.5 Validation

In order to validate our selected parameters' values we had to take several factors into consideration. Firstly, to evaluate our trained network and before using it in a whole unlabelled video, we decided to perform an evaluation feeding a novel set of labelled yet unseen images to the trained model. In such a way, we were able to investigate how the network would behave before unseen data, while still having the ability to inspect the individual frame and obtaining the euclidean error for this dataset.

The output of each analysed video is the x and y coordinates of each feature accompanied by the likelihood of the label, for each frame. Thus, we can look at the likelihood of the network's predictions as a validation method. We can analyse the average likelihood of each feature and consider that all values above our training likelihood of 0.8 are a success. If a feature's average likelihood has a value lower than that, we can remove that feature from analysis. The average likelihood of all the features within the video can also be calculated, enabling the extrapolation of the videos' global average likelihood. We used this method in order to confirm our results were extracted with good likelihood values, instilling confidence to proceed with these results for further investigation.

Results and Discussion

This dissertation was conducted with the main objective of developing a functional open source pipeline. Our intention was for it to be used as a new strategy for mouse behaviour analysis that takes part in experimental studies of MJD. Two open source approaches were tested and evaluated for their performance as well as compared for their benefits regarding our particular usage, Bonsai RX and DeepLabCut.

This chapter presents all the results obtained within the scope of this thesis, accompanied by their analysis and discussion. On Section 6.1, we examine the results of using our Bonsai RX based workflow for the desired measurements' extraction. In Section 6.2, we then present the results obtained through our DeepLabCut based pipeline, including our parameter selection during CNN training and fine tuning. Using a CNN allowed us to perform a more in-depth analysis on our recorded animals, as presented in Section 6.3. Finally, we analyse our outcomes and discuss them as a way to provide a global evaluation of this entire methodology.

6.1 Bonsai Tracking

As part of our research, we conducted an analysis of the open field test using the Bonsai system. The primary aim of this analysis was to assess the suitability of this method for our research objectives. In doing so, we extracted three critical measures related to locomotor and exploratory activity: total traveled distance, mean velocity and resting time.

A comprehensive assessment of all recorded open field experiments was conducted. The videos were batch analysed via the custom Bonsai script, enabling the extraction of valuable insights from mice's locomotor and exploratory behaviour, as illustrated in Figure 6.1a. Consequently, TXT files were generated, containing the metrics of interest measured during the recorded test. As explained in further detail in Chapter 5, the videos were processed through a sequence of nodes, displayed in Figure 6.1c, and could be monitored through each node's *Visualizer*, as illustrated in Figure 6.1b.

As previously stated in the Methods chapter, in this manuscript, we will use the term "MJD Homo YAC mouse" or solely "Homo YAC" to describe the homozygous YAC MJD84.2 line that was used.

In Figure 6.2 we present our outcomes. Wild-type mice travelled longer distances when

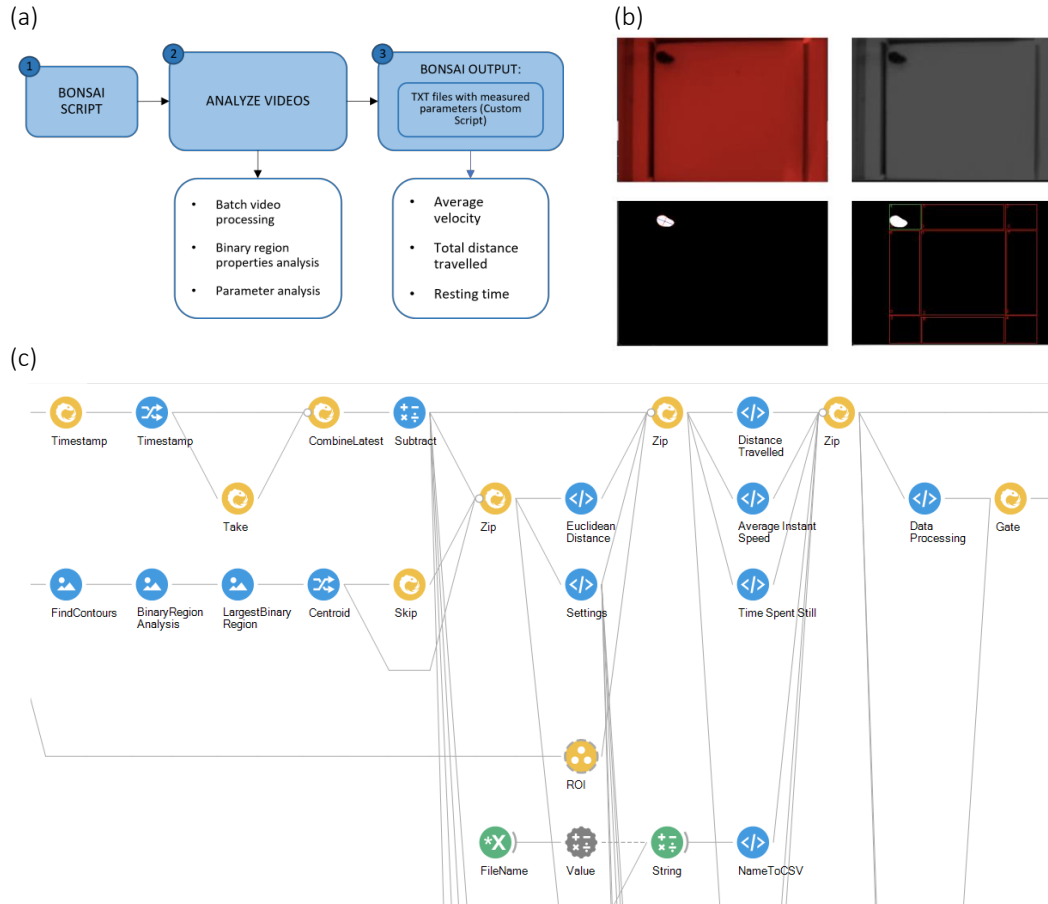


Figure 6.1: Workflow diagram for first strategy, Bonsai RX. (a) Scheme of data analysis using Bonsai, (b) Different stages of video processing on mice identification and arena segmentation in the open field video and (c) Snippet of Bonsai node-like script visualization.

compared to both Homo YAC PBS and Homo YAC Treated mice, respectively. In fact, a statistically significant difference was detected in the distance travelled between WT and Homo YAC PBS groups (11646.7 ± 3315.6 cm vs. 5015.1 ± 1157.9 cm). The Homo YAC Treated group travelled a mean of 6820.4 ± 4001.7 cm. Regarding the average speed of the mice throughout the open field session, no pronounced variations were observed amongst the three mice groups. Their average velocities were 6.87 ± 0.73 , 5.91 ± 0.64 , 6.32 ± 0.83 cm/s for wild-type, Homo YAC PBS mice and Homo YAC Treated mice, respectively. Although the average resting time was smaller in wild-type mice than in Homo YAC PBS and Homo YAC Treated groups (6.04 ± 3.02 vs 12.82 ± 2.05 vs 12.70 ± 5.17 min), this difference was not statistically significant.

These results show that wild-types travelled longer distances, the Homo YAC PBS mice performed worse, expectedly due to the recognized gait impairments, and Homo YAC Treated mice had a slightly, but not significant, better performance, which we may attribute to a positive impact of the treatment.

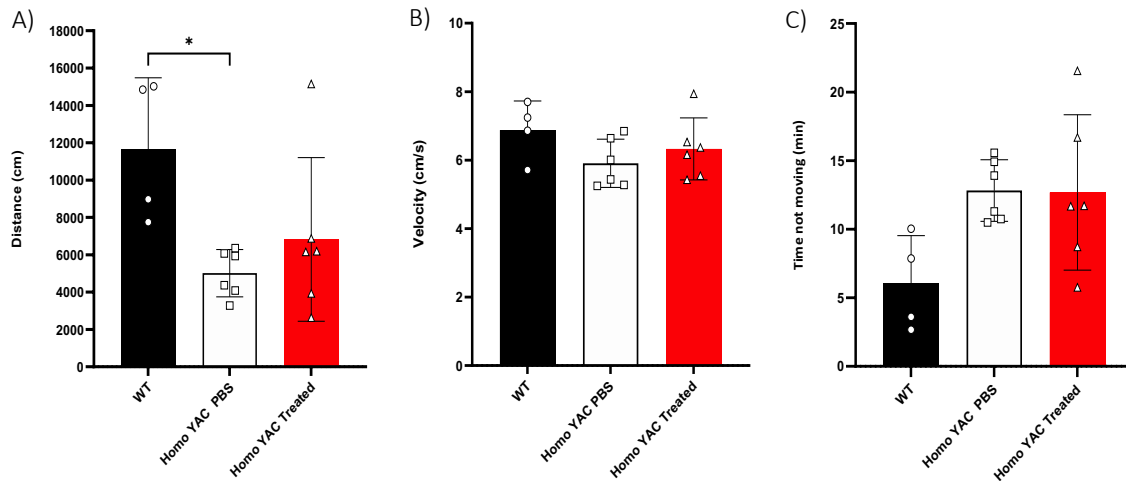


Figure 6.2: Open field test performance measured using Bonsai. Comparison of phenotypic features in wild-type, Homo YAC PBS and Homo YAC Treated groups. Locomotor activity was analysed by measuring: (a) total distance travelled (cm), (b) average velocity (cm/s) and (c) resting time (min). The data are presented as the mean \pm SD. Statistical analysis was performed using the ANOVA test, $*p < 0.05$.

Categorising the animals by gender and conducting gender-specific analysis revealed a discrepancy. Male subjects exhibited greater mobility compared to their female counterparts (WT - males: 14933 ± 84 cm vs. females: 8360 ± 613 cm, Homo YAC PBS - males: 5504 ± 1007 cm vs. females: 4525 ± 1080 cm and Homo YAC Treated - males: 9404 ± 4069 cm vs. females: 4236 ± 1453 cm) as detailed in Appendix Figure B.1. Additionally, a focused statistical analysis aimed at comparing only the wild-type and Homo YAC PBS mice revealed significant differences in distance travelled and resting time within our experimental dataset. We applied the Unpaired t-test for this comparison, yielding the results presented in Appendix Figure B.2.

6.2 DeepLabCut Tracking

6.2.1 Parameter Selection

After conducting comprehensive testing and analysis, we decided on the most suitable hyperparameters for training our CNN, which led to the best results for our task. In Appendix A, we provide the reasoning for the values used from this point onward. The chosen parameters, outlined in Table 6.1, were as follows: batch size of 2, *p-cutoff* of 0.8, cropping parameter set to FALSE, and global scale set to 0.8. These specific values showed superior performance, offering a good balance between efficiency and accuracy in our image processing tasks. Our evaluations proved the efficacy of this parameter configuration, which we integrated into our CNN model for our research.

Table 6.1: Selected Values for DeepLabCut Hyperparameter Comparison.

Hyperparameters	Batch Size	P-Cutoff	Crop	Global Scale
Values	2	0.8	False	0.8

6.2.2 Workflow and Results

While the Bonsai framework had previously conveyed satisfactory results, we opted to also explore the DeepLabCut tool hoping for a more comprehensive and precise tracking of mice during open field tests. Given this tool was specifically designed for estimating animal posture and tracking specific points, we were confident this pipeline would yield a promising choice for our research requirements.

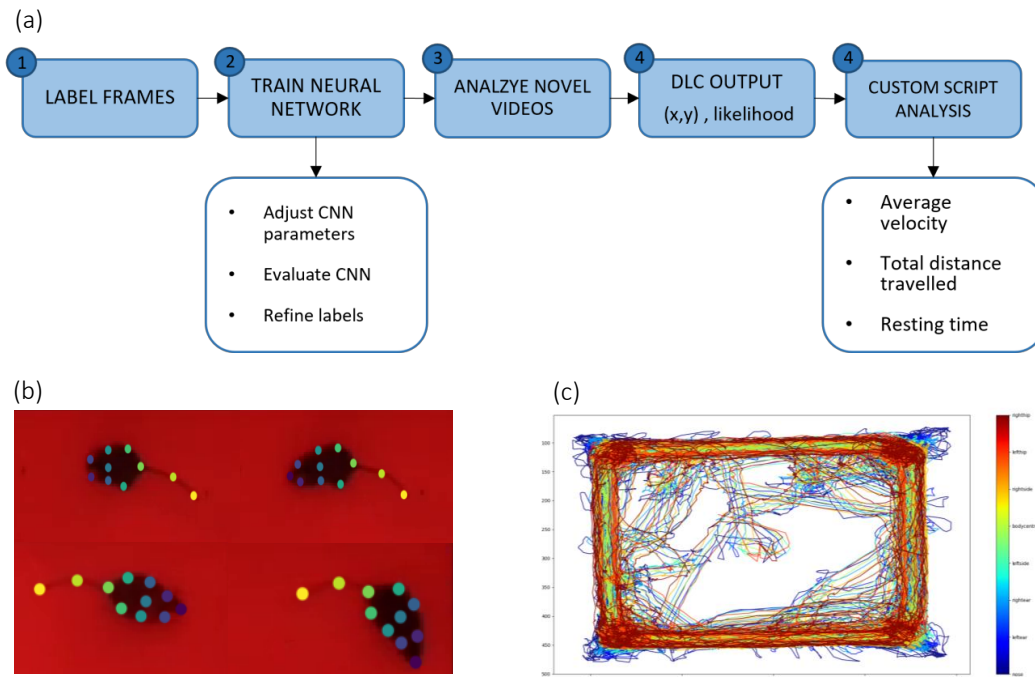


Figure 6.3: Workflow diagram for second strategy, DeepLabCut. (a) Scheme of data analysis using DeepLabCut, (b) DeepLabCut’s point estimation predictions and (c) Trajectory plot of all keypoints a YAC female mouse during an open field test.

A concise summary of our extensively optimized pipeline, which encapsulates the key steps we undertook, is presented in Figure 6.3a. Our initial steps involved the extraction of video frames from the input videos, and manually labelling the points of interest to create our training dataset. Subsequently, DeepLabCut’s functions were applied to train and validate the neural network with the carefully selected parameters. Once the network was effectively trained, it was applied to all our video dataset analysis, which culminated in an output, for each video, of body part coordinates and their respective likelihood. A more detailed explanation of these processes

is provided in Section 5.7.2.

In addition to the output file containing the coordinates and likelihood of every frame for each video, we also generated additional outputs, as a labelled video which enabled us to visualize the tracked keypoints and trajectory plots (Figures 6.3b and 6.3c). To assess the animal’s performance, we employed custom scripts (available in our repository [Angeja, 2023]) to compute identical metrics as those derived from the Bonsai experiment. These encompass the animal’s total distance travelled, average velocity and resting time throughout the test period.

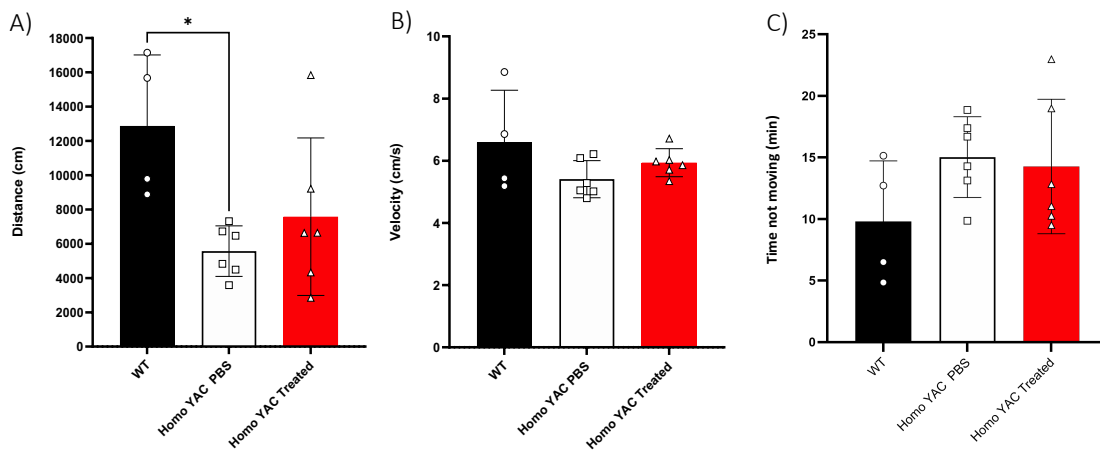


Figure 6.4: Open field test performance measured using DeepLabCut of bodycentre tracked point. Comparison of phenotypic features in wild-type, Homo YAC PBS and Homo YAC Treated groups. Locomotor activity was analysed by measuring: (a) total distance travelled in cm, (b) average velocity in cm/s and (c) resting time in min. The data are presented as the mean \pm SD. Statistical analysis was performed using the ANOVA test, * $p < 0.05$.

Analogous to the results obtained through Bonsai analysis, Figure 6.4 presents three graphical representations illustrating the values derived from DeepLabCut’s outputs and their subsequent processing using our script. Bonsai exclusively extracts the mouse’s centre of mass. Hence, for a more accurate comparison between the two approaches, we decided to use the body centre keypoint predictions to formulate these graphs.

Wild-type mice exhibited significantly greater travel distances than both transgenic groups. Specifically, the Homo YAC PBS group covered an average distance of 5570.1 ± 1342.5 cm, while wild-type mice traveled a 12872.6 ± 3591.4 cm and Homo YAC Treated mice covered 7581.8 ± 4195 cm. There were no substantial variations in average velocity across the three groups (wild-type: 6.58 ± 1.45 cm/s, Homo YAC PBS: 5.41 ± 0.55 cm/s, Homo YAC Treated: 5.94 ± 0.41 cm/s). It is worth noting that wild-type mice had a shorter average resting time (9.80 ± 4.25 min) compared to the Homo YAC PBS (15.03 ± 2.98 min) and Homo YAC Treated mice (14.26 ± 4.98 min), although this difference did not reach statistical significance.

When we segmented the groups based on gender and conducted separate analyses for males and females, a notable gender-based disparity became evident. Males exhibited overall higher

levels of locomotion in comparison to females, as detailed in Appendix Figure C.1.

Furthermore, when we exclusively compared the wild-type and Homo YAC PBS mice, we identified disparities between these two groups concerning movement distance within our experimental dataset. Unpaired t-test statistical test was used for this analysis, leading to the outcomes depicted in Figure 6.8, in the following Section.

6.3 Leveraging Keypoint Tracking in DeepLabCut

In order to fully capitalize on the coordinates' output generated by DeepLabCut, we decided to undertake a comprehensive examination of the body parts individually. In a first approach, we proceeded to analyse two mice, a wild-type and a Homo YAC PBS mouse, hoping to identify clear differences in body part tracking. We then repeated our measurement analysis for the complete test duration for the relevant body parts for wild-type and Homo YAC PBS mice groups.

Despite having labelled and tracked a total of eleven body parts with our DeepLabCut pipeline, here we focus on three (nose, body centre and tail centre) that we found most interesting for our purpose. The nose was selected due to mice's exploratory nature. As we well know, mice often use their noses to explore and be aware of their surroundings. Hence, we decided to further inspect this body part, expecting to see a slightly bigger travel distance for it, at least in the wild-type animal who should be curious and exploratory. In order to have a central measure for the mouse's trajectory and behaviour, we decided to select the body centre point. Finally, the tail centre was chosen in an effort to investigate whether the tail would behave differently from the rest of the body, namely from the body centre point. Some conditions might provoke shaking in some animals, which might be reflected in the tail. As such, we decided to investigate if the tail might be reflective of the condition present in the YAC mouse model. Tail centre was selected in detriment of tail tip due the difficulty in tail tip tracking. Tails are inherently difficult to track due to being very slim and therefore susceptible to being confused with the background. Our behaviour open field test was conducted during mice's active cycle, hence it was filmed in a dark room using a red light. Adding to the camera's not optimal resolution, this difficulty is enhanced because of harsh lighting. These constraints led us to prefer using the tail centre point rather than the tail tip.

We have decided to employ this form of representation to illustrate the behaviour of two mice, a wild-type and a Homo YAC PBS injected. In doing so, we have separated the data by body part and divided it into 10-minute intervals. This approach was adopted as we consider it yields potential value in investigations concerning a comprehensive examination of their exploratory behaviour throughout the test.

6.3.1 Wild-type Animal

Represented in the trajectory plots of Figure 6.5 is a wild-type animal that spent 30 minutes in an open field arena. As it is typical for wild-type mice, it presents a heightened exploratory

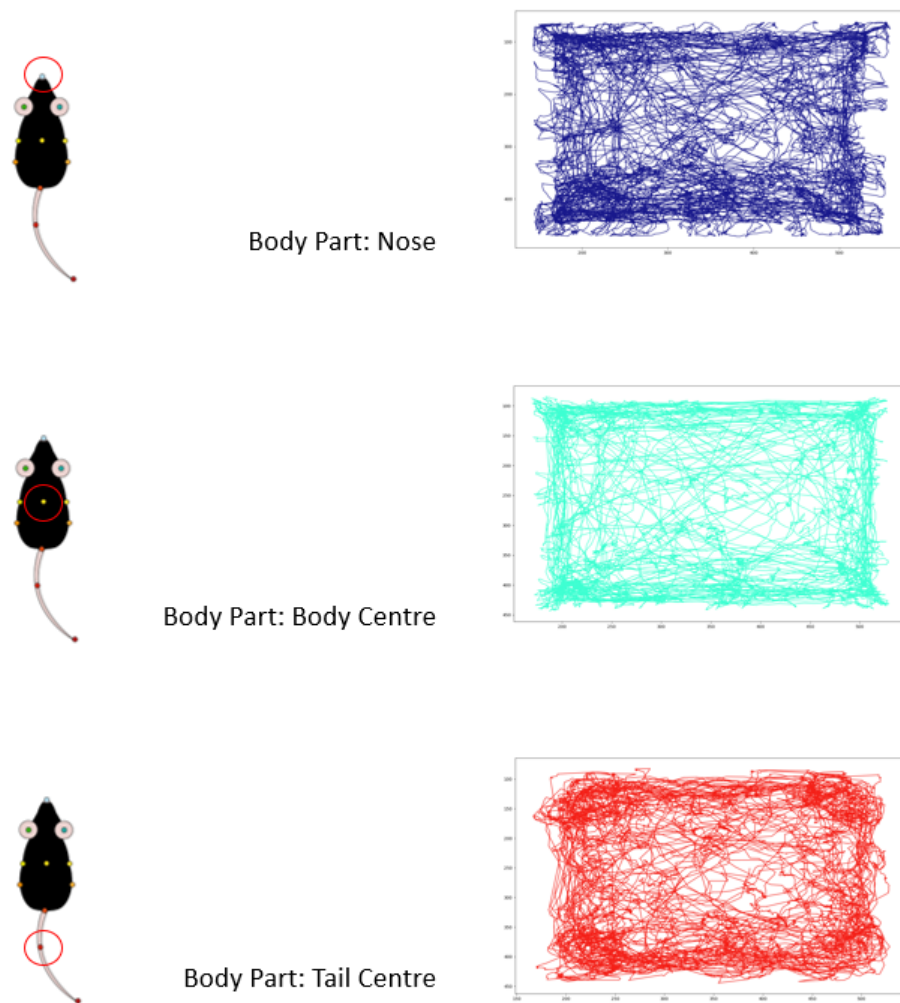


Figure 6.5: Point tracking trajectory of three tracked body parts of a wild-type mouse. Trajectories of nose, body centre, and tail centre, from top to bottom, respectively.

activity, exploring and occupying nearly every space of the arena for the duration of the test. The plots in Appendix Figure D.1 represent the average speed, resting time and distance travelled divided per body part in the open field test for a wild-type mouse for the time bins duration. Dividing the behaviour test into three time bins of 10 minutes each gives us the opportunity to analyse how each body part behaved throughout the test duration.

6.3.2 Homo YAC PBS Animal

We repeated the procedure with the DeepLabCut output file of a YAC female mouse treated with PBS. The results for the trajectory plots are as shown in Figure 6.6. This particular animal was very affected in its movements due to the progression of its condition. It exhibits little movement in the centre of the arena, which might be caused by limb impairments as well as

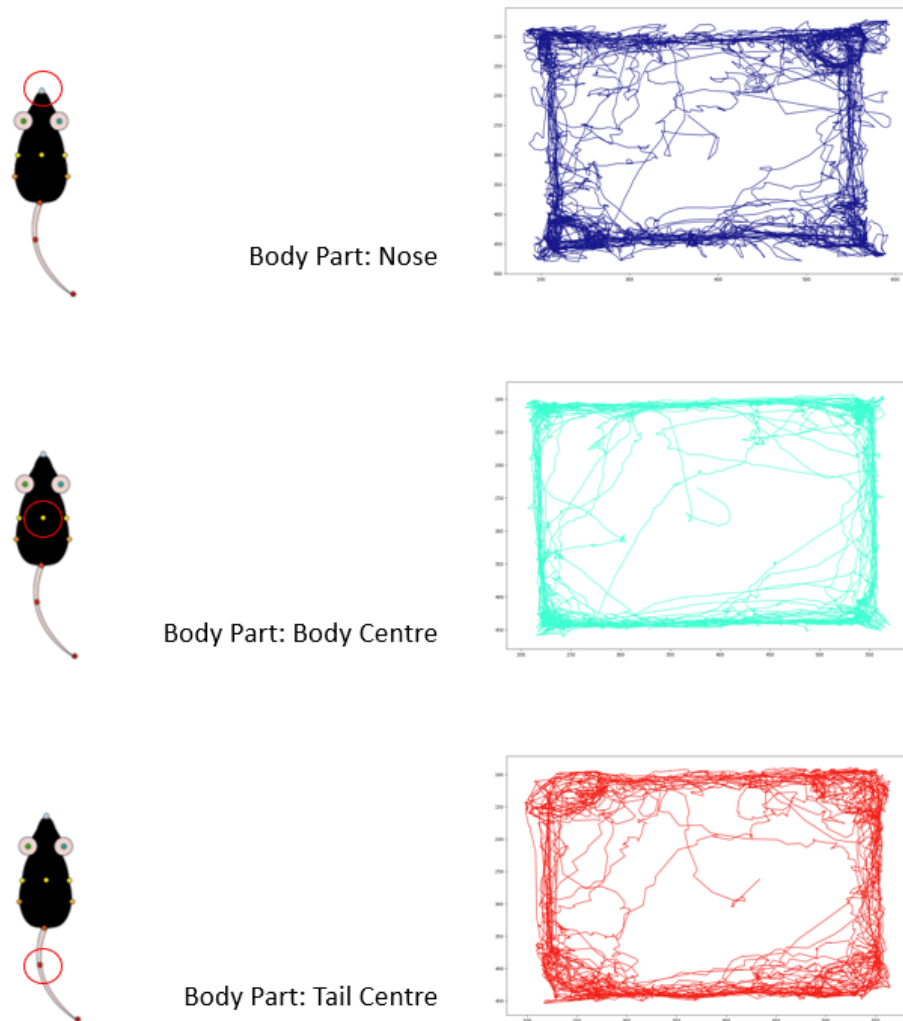


Figure 6.6: Point tracking trajectory of three tracked body parts of a Homo YAC PBS mouse. Trajectories of nose, body centre, and tail centre, from top to bottom, respectively.

anxiety. The graphs presented in Appendix Figure D.2 illustrate the traveled distance, mean speed and resting time for each body part during an open field test conducted on a Homo YAC PBS mouse.

As anticipated, the Homo YAC PBS mouse displayed a notable reduction in both distance traveled and average velocity compared to the wild-type mouse, while exhibiting higher resting time values. Despite the lower average velocity for individual body parts in the Homo YAC PBS mouse, with most values around 4 cm/s as opposed to approximately 5 cm/s in the wild-type mouse, the overall average speed for all tracked points remained relatively consistent, similar to the observed in Appendix Figure D.1 for the wild-type mouse.

6.3.3 Body Parts Keypoint Tracking

As previously explained, we thought it would be insightful to explore the differences in activity between nose, body centre and tail centre. Our results are revealed in Figures 6.7, 6.8 and 6.9. In Appendix D we summarized the results pertaining to these keypoints in Tables D.1 and D.2. Interestingly, both the WT and Homo PBS YAC groups exhibit similar patterns of behaviour across all three body parts.

In the analysis of Figure 6.7 it is visible that wild-type mice exhibited greater distances covered by their nose ($18\,488 \pm 4570$ cm) than the Homo YAC PBS mice group (9035 ± 1554 cm). The average velocities were recorded as 8.98 ± 2.66 cm/s for wild-type mice and as 5.84 ± 0.52 cm/s for Homo YAC PBS mice. Regarding resting times, wild-type mice displayed a mean value of 3.42 ± 1.86 min, while Homo YAC PBS mice showed resting times of 5.84 ± 1.72 min.

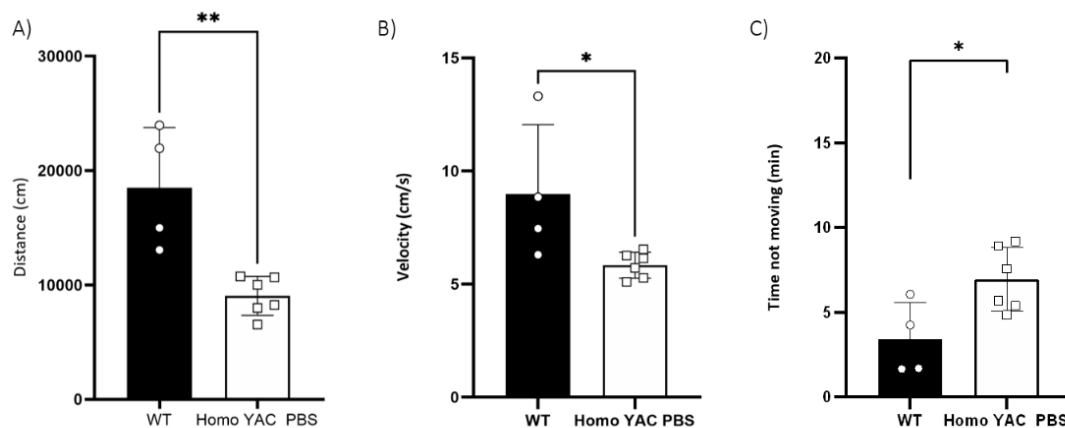


Figure 6.7: Open field test performance of nose point across Wild-Type and Homo YAC PBS using DeepLabCut. Comparison of phenotypic features in wild-type and Homo YAC PBS. Locomotor activity was analysed by measuring: (a) total distance travelled in cm, (b) average velocity in cm/s and (c) resting time in min. The data are presented as the mean \pm SD. Statistical analysis was performed using the unpaired Student's t-test, * $p < 0.05$, ** $p < 0.01$.

Figure 6.8 presents the measurements for the body centre keypoint, that we have discussed in Section 6.2. For the purpose of the present section, our focus was on isolating and comparing the wild-type group and the Homo YAC PBS group. The wild-types traveled a mean distance of 12972 ± 3591 cm as opposed to 9035 ± 1554 cm traveled by the Homo YAC PBS. As for the mean velocity, the wild-type and the Homo YAC PBS group showed values of 6.58 ± 1.46 cm/s and 5.41 ± 0.54 cm/s, respectively. Lastly, the wild-types exhibited a resting time of 9.8 ± 4.26 min, while the Homo YAC PBS reported 15.02 ± 2.99 min for the body centre.

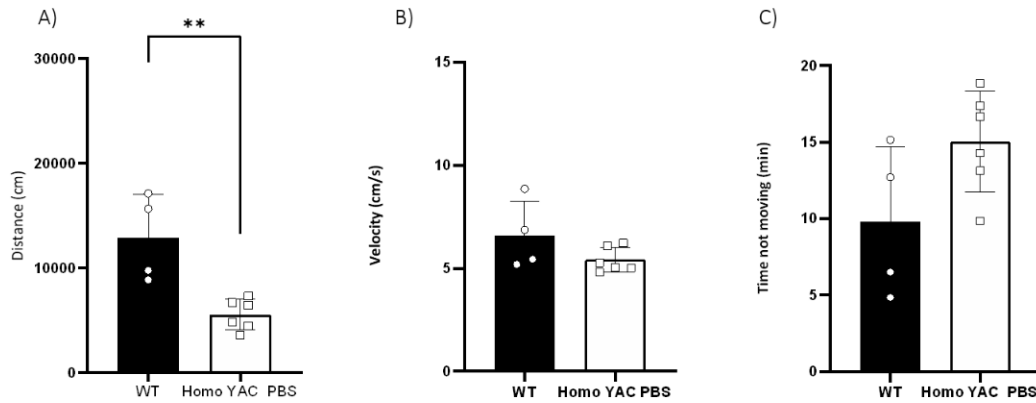


Figure 6.8: Open field test performance of body centre point across Wild-Type and Homo YAC PBS using DeepLabCut. Comparison of phenotypic features in wild-type and Homo YAC PBS. Locomotor activity was analysed by measuring: (a) total distance travelled (cm), (b) average velocity (cm/s) and (c) resting time (min). The data are presented as the mean \pm SD. Statistical analysis was performed using unpaired Student's t-test, ** $p < 0.01$.

In Figure 6.9, the wild-type group traveled an average distance of $18\,921 \pm 5118$ cm, whereas the Homo YAC PBS group covered a distance of 8238 ± 1625 cm. In terms of mean velocity, the wild-type group exhibited a value of 9.54 ± 2.64 cm/s, while the Homo YAC PBS group had a mean velocity of 7.41 ± 0.62 cm/s. Finally, the resting time was 8.91 ± 3.59 min and 12.63 ± 2.19 min for the wild-type and the Homo YAC PBS groups, respectively.

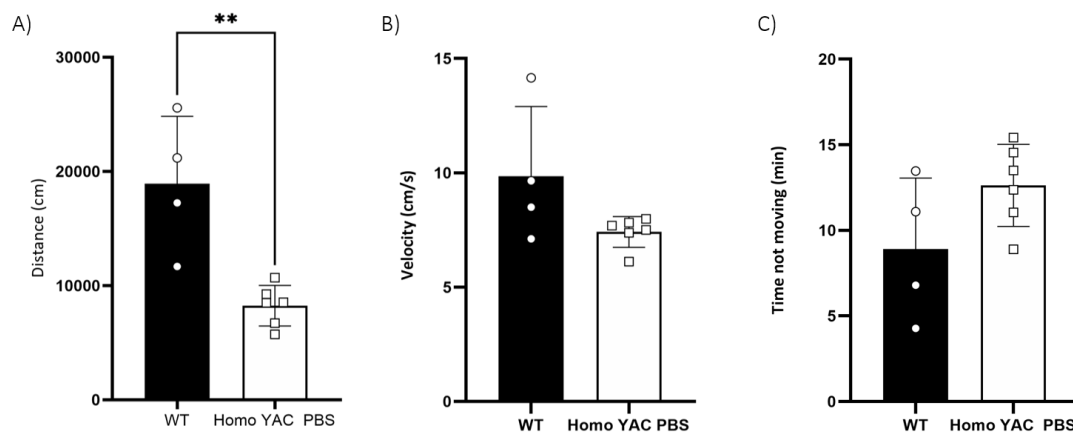


Figure 6.9: Open field test performance of tail centre point across Wild-Type and Homo YAC PBS using DeepLabCut. Comparison of phenotypic features in Wild-type and Homo YAC PBS. Locomotor activity was analysed by measuring: (a) total distance travelled in cm, (b) average velocity in cm/s and (c) resting time in min. The data are presented as the mean \pm SD. Statistical analysis was performed using the unpaired Student's t-test, ** $p < 0.01$.

6.4 Final Discussion

To conclude this chapter, we examine and compare our findings from the two employed strategies. Moreover, we believe it is crucial to delve into the validity of our results in comparison to those reported in the existing literature, as well as some of the limitations we considered this work has encountered.

Comparing the two strategies

Our analyses yielded congruent results between the Bonsai and DeepLabCut tracking methods. Following this dissertation, the question arises regarding which tool is better suited for our purposes. Both tools have their distinct advantages and disadvantages. The choice between them largely depends on the analysis' specific objectives.

Bonsai RX may be the preferred option when the main focus revolves around obtaining overall measurements (such as distance travelled, velocity and resting time) pertaining to the entire mouse's behaviour without the need for specific body part tracking.

Conversely, DeepLabCut is particularly valuable when a more detailed analysis is required. An additional drawback of DeepLabCut, and of most pose estimation algorithms, is they typically require high-quality recordings to effectively capture the mouse's body parts and therefore poses. Our behaviour assessment took place in a dark room illuminated with red light, which undeniably had an impact on the quality of our video recordings. This, in turn, may have affected our model estimations. Concerning both approaches, a significant advantage is the flexibility to adapt the measures and customize novel ones to be extracted, using the existing workflows.

In summary, these tools facilitate the investigator's work and provide greater control over their experiments, which ultimately leads to more informed decision-making.

Benchmarking against existing literature

Locomotor and exploratory activity is commonly assessed through an open field behaviour test. However, the conditions and criterion employed in the analysis of such activity vary considerably across different studies. Some commercially available systems divide the arena into quadrants using photobeams and count the number of times the mouse breaks them (e.g. San Diego Instruments). This approach provides a qualitative analysis representing the total distance covered by the mouse [Tsimpanouli et al., 2022, Bushart et al., 2021, do Carmo Costa et al., 2013]. In contrast, other apparatuses operate through video tracking and provide quantitative measurements of distances and velocities. Another variable factor is the duration of the test, with some researchers opting for a 30-minute test [Tsimpanouli et al., 2022, Bushart et al., 2021, do Carmo Costa et al., 2013], while others prefer 1, 5 or 10 minutes [Li et al., 2018, Lin et al., 2022, Wu et al., 2022, Rajamani et al., 2017]. Furthermore, the time points chosen for behaviour assessments vary widely among studies, as highlighted in our meta-analysis research [Silva*

et al., 2023]. These discrepancies in testing protocols contribute to challenges in comparing our research findings.

Consequently, we found it challenging to identify articles that conducted the open field test on YAC transgenic mice under similar conditions for meaningful comparisons. Notably, the measurement of mouse distance travelled was particularly difficult due to prevalence of qualitative analyses in articles employing 30-minute tests, such as the number of beam breaks. Only a few articles provided information on mouse speed and we found none that quantified resting time. To calculate locomotor activity, it is essential to select a movement threshold, which represents the minimum amount of movement required for the animal to be considered "in motion". As previously mentioned, we adopted a specific movement threshold of 2cm/s. The threshold value is seldom shared in the existing literature. Instances with movement below this threshold were excluded from velocity calculations, and distance travelled was not accumulated. Accordingly, resting time was added when the mouse's movement fell below this threshold.

To ensure that we were quantifying movement consistently and capturing slower movements as well, we purposefully decided on the value of 2cm/s. However, it is crucial to remember this value deeply influences our results and any alteration to this threshold would lead to modifications in our quantified result findings.

Despite a lack of consensus regarding the specific assessment criteria for the locomotor function tests, a recurring conclusion emerges: YAC transgenic mice manifest diminished locomotor and exploratory activity at an early stage when compared to wild-type littermates performance [Tsimpanouli et al., 2022, Bushart et al., 2021, do Carmo Costa et al., 2013, Li et al., 2018, Lin et al., 2022, Wu et al., 2022, Rajamani et al., 2017].

Conclusion

The present study demonstrates that both tested open source approaches, Bonsai RX and DeepLabCut, may be used as a viable replacement for the existing commercial approach in the evaluation of the open field behaviour assay related to MJD research or other SCAs.

Our work commenced with the utilization of a pre-established script for open field created with Bonsai RX. In addition to adapting and optimizing it to our experimental specifications, we sought to enrich the insights on the open field assay and introduce pose estimation capabilities into our workflow. To achieve this, we employed the DeepLabCut pose estimation toolbox, which involved image labelling and refinement, network evaluation and parameter selection in order to train a CNN tailored for within our context. From a general standpoint, the performance of both utilised strategies presented reliable results. We believe this thesis constitutes an elaborate explanation of this entire process, including the toolboxes used, parameter selection, calculated metrics and the steps undertaken, which we hope may be used as an initial framework for further behaviour tests automation. We have also created a repository with all the detailed information concerning our work, hoping we contribute to the relative standardization of behaviour analysis in mouse research [Angeja, 2023].

Within the scope of our work, we conducted a meta-analysis research that provided valuable insights into the motor abnormalities observed in the YAC84.2 mouse model of Machado Joseph disease [Silva* et al., 2023]. Our findings contribute to our understanding of the disease and emphasise the critical importance of careful experimental design and the importance of standardised assessment protocols.

Regarding the systems we employed, one clear advantage of the Bonsai workflow is that it does not involve frame labelling, a step that can be time-consuming. Conversely, as previously explained, DeepLabCut operates on the foundation of pre-trained networks, making this step relatively expeditious. Even though both approaches oblige a certain degree coding and programming knowledge, they benefit from active and supportive communities to facilitate the resolution of questions and challenges that may arise, mitigating this particular concern.

One of our initial objectives was to expand upon our existing metrics within the open field assessment. We successfully extracted measurements of the nose, body, and tail centres, which represented an advancement in the test's analysis. Nevertheless, it is important to recognize that it remains much to be exploited within this test. For future research, we recommend the

implementation of a camera under the arena to record the open field test from below and use a transparent arena. This setup would allow the recording of paw movements and hence the calculation of stride length, overlap and velocity. These inclusions would further extend the utility of the open field test.

Although not the primary focus of this study, our workflow has the capability of segmenting the extracted measurements into time slots, potentially offering insights into mouse activity patterns. Furthermore, we divided the arena into different regions of interest, yet we did not have the opportunity to explore the segmented data thoroughly. As part of future work, we suggest the exploration of these data regions.

It is worth emphasizing that the outcomes and insights presented within this thesis are not confined to the specific study of the open field test. Rather, they represent a versatile body of work that offers insights and methodologies applicable to a diverse spectrum of behavioural assessment tests. Hence, for future projects, we believe it would be interesting to extend the work to embrace the beam walking test. For instance, this behaviour test holds substantial relevancy in assessing locomotor function in MJD mice, and its current methodology still involves arduous manual calculations and scoring by researchers. In the beam walking test, the mouse traverses a beam and measurements such as latency to cross and number of slips on the beam are scored by hand. In our opinion it would be valuable to explore the possibility of training a neural network using DeepLabCut to automate and expedite these measurements.

Lastly, an aspect we were unable to explore was real-time tracking or pose estimation. In fact, there is a project that allows the two tools to be brought together, where Bonsai acts as an interface for DeepLabCut enabling real-time markerless pose estimation using pre-trained models. Therefore, we propose the integration of the model trained within this thesis, tailored for the open field test, with the Bonsai RX custom script. This would generate a workflow with enhanced user-friendliness, ultimately benefiting investigators in their research endeavors.

Thus, we conclude that the present study provides insights into the automation of behavioural analysis and serves as a foundation for the continuation of work in the field. Ultimately, we aspire for this thesis to have created a valuable and practical tool to be used in investigations at our research centre.

References

- [Akam et al., 2022] Akam, T., Lustig, A., Rowland, J. M., Kapanaiiah, S. K., Esteve-Agraz, J., Panniello, M., Márquez, C., Kohl, M. M., Kätzel, D., Costa, R. M., et al. (2022). Open-source, python-based, hardware and software for controlling behavioural neuroscience experiments. *ELife*, 11:e67846.
- [Alves et al., 2008] Alves, S., Nascimento-Ferreira, I., Auregan, G., Hassig, R., Dufour, N., Brouillet, E., Pedroso de Lima, M. C., Hantraye, P., Pereira de Almeida, L., and Déglon, N. (2008). Allele-specific rna silencing of mutant ataxin-3 mediates neuroprotection in a rat model of machado-joseph disease. *PloS one*, 3(10):e3341.
- [Alves et al., 2010] Alves, S., Nascimento-Ferreira, I., Dufour, N., Hassig, R., Auregan, G., Nobrega, C., Brouillet, E., Hantraye, P., Pedroso de Lima, M. C., Déglon, N., et al. (2010). Silencing ataxin-3 mitigates degeneration in a rat model of machado-joseph disease: no role for wild-type ataxin-3? *Human molecular genetics*, 19(12):2380–2394.
- [Amini, 2023] Amini, A. (2023). Mit introduction to deep learning. MIT 6.s191. January 9, 2023.
- [Angeja, 2023] Angeja, L. (2023). Pipeline for openfield analysis in mjd mice. <https://github.com/lauraangeja/open-field-DLC-pipeline-MJD>.
- [Arac et al., 2019] Arac, A., Zhao, P., Dobkin, B. H., Carmichael, S. T., and Golshani, P. (2019). Deepbehavior: A deep learning toolbox for automated analysis of animal and human behavior imaging data. *Frontiers in systems neuroscience*, 13:20.
- [Azure, nd] Azure, M. (nd). What are machine learning algorithms? <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms>.
- [Bettencourt and Lima, 2011] Bettencourt, C. and Lima, M. (2011). Machado-joseph disease: from first descriptions to new perspectives. *Orphanet journal of rare diseases*, 6(1):1–12.
- [Bichelmeier et al., 2007] Bichelmeier, U., Schmidt, T., Hübener, J., Boy, J., Rüttiger, L., Häbig, K., Poths, S., Bonin, M., Knipper, M., Schmidt, W. J., et al. (2007). Nuclear localization of ataxin-3 is required for the manifestation of symptoms in sca3: in vivo evidence. *Journal of Neuroscience*, 27(28):7418–7428.

- [Boy et al., 2010] Boy, J., Schmidt, T., Schumann, U., Grasshoff, U., Unser, S., Holzmann, C., Schmitt, I., Karl, T., Laccone, F., Wolburg, H., et al. (2010). A transgenic mouse model of spinocerebellar ataxia type 3 resembling late disease onset and gender-specific instability of cag repeats. *Neurobiology of disease*, 37(2):284–293.
- [Boy et al., 2009] Boy, J., Schmidt, T., Wolburg, H., Mack, A., Nuber, S., Böttcher, M., Schmitt, I., Holzmann, C., Zimmermann, F., Servadio, A., et al. (2009). Reversibility of symptoms in a conditional mouse model of spinocerebellar ataxia type 3. *Human molecular genetics*, 18(22):4282–4295.
- [Bradski, 2000] Bradski, G. (2000). The opencv library. *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, 25(11):120–123.
- [Burnett et al., 2003] Burnett, B., Li, F., and Pittman, R. N. (2003). The polyglutamine neurodegenerative protein ataxin-3 binds polyubiquitylated proteins and has ubiquitin protease activity. *Human molecular genetics*, 12(23):3195–3205.
- [Bushart et al., 2021] Bushart, D. D., Zalon, A. J., Zhang, H., Morrison, L. M., Guan, Y., Paulson, H. L., Shakkottai, V. G., and McLoughlin, H. S. (2021). Antisense oligonucleotide therapy targeted against atxn3 improves potassium channel-mediated purkinje neuron dysfunction in spinocerebellar ataxia type 3. *The Cerebellum*, 20:41–53.
- [Cemal et al., 2002] Cemal, C. K., Carroll, C. J., Lawrence, L., Lowrie, M. B., Ruddle, P., Al-Mahdawi, S., King, R. H., Pook, M. A., Huxley, C., and Chamberlain, S. (2002). Yac transgenic mice carrying pathological alleles of the mjd1 locus exhibit a mild and slowly progressive cerebellar deficit. *Human molecular genetics*, 11(9):1075–1094.
- [Chen et al., 2017] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.
- [Chou et al., 2008] Chou, A.-H., Yeh, T.-H., Ouyang, P., Chen, Y.-L., Chen, S.-Y., and Wang, H.-L. (2008). Polyglutamine-expanded ataxin-3 causes cerebellar dysfunction of sca3 transgenic mice by inducing transcriptional dysregulation. *Neurobiology of disease*, 31(1):89–101.
- [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- [do Carmo Costa et al., 2013] do Carmo Costa, M., Luna-Cancelon, K., Fischer, S., Ashraf, N. S., Ouyang, M., Dharia, R. M., Martin-Fishman, L., Yang, Y., Shakkottai, V. G., Davidson, B. L., et al. (2013). Toward rna therapy for the polyglutamine disease machado-joseph disease. *Molecular Therapy*, 21(10):1898–1908.
- [do Carmo Costa and Paulson, 2012] do Carmo Costa, M. and Paulson, H. L. (2012). Toward understanding machado-joseph disease. *Progress in neurobiology*, 97(2):239–257.

- [Durr et al., 1996] Durr, A., Stevanin, G., Cancel, G., Duyckaerts, C., Abbas, N., Didierjean, O., Chneiweiss, H., Benomar, A., Lyon-Caen, O., Julien, J., et al. (1996). Spinocerebellar ataxia 3 and machado-joseph disease: clinical, molecular, and neuropathological features. *Annals of neurology*, 39(4):490–499.
- [Ellenbroek and Youn, 2016] Ellenbroek, B. and Youn, J. (2016). Rodent models in neuroscience research: is it a rat race? *Disease models & mechanisms*, 9(10):1079–1087.
- [Eto et al., 1990] Eto, K., Sumi, S., Bird, T. D., McEvoy-Bush, T., Boehnke, M., and Schellenberg, G. (1990). Family with dominantly inherited ataxia, amyotrophy, and peripheral sensory loss: spinopontine atrophy or machado-joseph azorean disease in another non-portuguese family? *Archives of neurology*, 47(9):968–974.
- [FFmpeg, nd] FFmpeg (n.d.). FFmpeg Website. <https://ffmpeg.org/>.
- [Gomez-Marin et al., 2014] Gomez-Marin, A., Paton, J. J., Kampff, A. R., Costa, R. M., and Mainen, Z. F. (2014). Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nature neuroscience*, 17(11):1455–1462.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- [Goti et al., 2004] Goti, D., Katzen, S. M., Mez, J., Kurtis, N., Kiluk, J., Ben-Haïem, L., Jenkins, N. A., Copeland, N. G., Kakizuka, A., Sharp, A. H., et al. (2004). A mutant ataxin-3 putative-cleavage fragment in brains of machado-joseph disease patients and transgenic mice is cytotoxic above a critical concentration. *Journal of Neuroscience*, 24(45):10266–10279.
- [Gould et al., 2007] Gould, V. F. C., Goti, D., Pearce, D., Gonzalez, G. A., Gao, H., de Leon, M. B., Jenkins, N. A., Copeland, N. G., Ross, C. A., and Brown, D. R. (2007). A mutant ataxin-3 fragment results from processing at a site n-terminal to amino acid 190 in brain of machado-joseph disease-like transgenic mice. *Neurobiology of disease*, 27(3):362–369.
- [Graving et al., 2019] Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., and Couzin, I. D. (2019). Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8:e47994.
- [Gu et al., 2004] Gu, W., Ma, H., Wang, K., Jin, M., Zhou, Y., Liu, X., Wang, G., and Shen, Y. (2004). The shortest expanded allele of the mjd1 gene in a chinese mjd kindred with autonomic dysfunction. *European neurology*, 52(2):107–111.
- [He et al., 2019] He, K., Girshick, R., and Dollár, P. (2019). Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927.
- [He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Howard et al., 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.
- [Hübener et al., 2011] Hübener, J., Vauti, F., Funke, C., Wolburg, H., Ye, Y., Schmidt, T., Wolburg-Buchholz, K., Schmitt, I., Gardyan, A., Drießen, S., et al. (2011). N-terminal ataxin-3 causes neurological symptoms with inclusions, endoplasmic reticulum stress and ribosomal dislocation. *Brain*, 134(7):1925–1942.
- [IBM, 2020] IBM (2020). An introduction to deep learning. Updated 9 de novembro de 2020 — Published 3 de março de 2020.
- [Ikeda et al., 1996] Ikeda, H., Yamaguchi, M., Sugai, S., Aze, Y., Narumiya, S., and Kakizuka, A. (1996). Expanded polyglutamine in the machado–joseph disease protein induces cell death in vitro and in vivo. *Nature genetics*, 13(2):196–202.
- [Insafutdinov et al., 2016] Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). Deepcut: A deeper, stronger, and faster multi-person pose estimation model. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14*, pages 34–50. Springer.
- [Isik and Unal, 2023] Isik, S. and Unal, G. (2023). Open-source software for automated rodent behavioral analysis. *Frontiers in Neuroscience*, 17:1149027.
- [Iwabuchi et al., 1999] Iwabuchi, K., Tsuchiya, K., Uchihara, T., and Yagishita, S. (1999). Autosomal dominant spinocerebellar degenerations. clinical, pathological, and genetic correlations. *Revue neurologique*, 155(4):255–270.
- [Jégou et al., 2017] Jégou, S., Drozdal, M., Vazquez, D., Romero, A., and Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19.
- [Katsuno et al., 2008] Katsuno, M., Banno, H., Suzuki, K., Takeuchi, Y., Kawashima, M., Tanaka, F., Adachi, H., and Sobue, G. (2008). Molecular genetics and biomarkers of polyglutamine diseases. *Current molecular medicine*, 8(3):221–234.
- [Kawaguchi et al., 1994] Kawaguchi, Y., Okamoto, T., Taniwaki, M., Aizawa, M., Inoue, M., Katayama, S., Kawakami, H., Nakamura, S., Nishimura, M., Akiguchi, I., et al. (1994). Cag

- expansions in a novel gene for machado-joseph disease at chromosome 14q32. 1. *Nature genetics*, 8(3):221–228.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- [Lang et al., 2020] Lang, J., Haas, E., Hübener-Schmid, J., Anderson, C. J., Pulst, S. M., Giese, M. A., and Ilg, W. (2020). Detecting and quantifying ataxia-related motor impairments in rodents using markerless motion tracking with deep neural networks. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 3642–3648. IEEE.
- [LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
- [Li et al., 2018] Li, T., Liu, Y., Yu, L., Lao, J., Zhang, M., Jin, J., Lu, Z., Liu, Z., and Xu, Y. (2018). Human umbilical cord mesenchymal stem cells protect against sca3 by modulating the level of 70 kd heat shock protein. *Cellular and Molecular Neurobiology*, 38:641–655.
- [Lima and Coutinho, 1980] Lima, L. and Coutinho, P. (1980). Clinical criteria for diagnosis of machado-joseph disease: report of a non-azorean portuguese family. *Neurology*, 30(3):319–319.
- [Lin et al., 2022] Lin, Y.-S., Cheng, W.-L., Chang, J.-C., Lin, T.-T., Chao, Y.-C., and Liu, C.-S. (2022). Igf-1 as a potential therapy for spinocerebellar ataxia type 3. *Biomedicines*, 10(2):505.
- [Livingstone and Sequeiros, 1984] Livingstone, I. R. and Sequeiros, J. (1984). Machado-joseph disease in an american-italian family. *Journal of neurogenetics*, 1(2):185–188.
- [Lopes et al., 2015] Lopes, G., Bonacchi, N., Frazão, J., Neto, J. P., Atallah, B. V., Soares, S., Moreira, L., Matias, S., Itskov, P. M., Correia, P. A., et al. (2015). Bonsai: an event-based framework for processing and controlling data streams. *Frontiers in neuroinformatics*, 9:7.
- [Luxem et al., 2023] Luxem, K., Sun, J. J., Bradley, S. P., Krishnan, K., Yttri, E., Zimmermann, J., Pereira, T. D., and Laubach, M. (2023). Open-source tools for behavioral video analysis: Setup, methods, and best practices. *ELife*, 12:e79305.
- [Lysenko et al., 2010] Lysenko, L., Grewal, R. P., Ma, W., and Peddareddygari, L. R. (2010). Homozygous machado joseph disease: a case report and review of literature. *Canadian journal of neurological sciences*, 37(4):521–523.
- [Maciel et al., 2001] Maciel, P., do Carmo Costa, M., Ferro, A., Rousseau, M., Santos, C. S., Gaspar, C., Barros, J., Rouleau, G. A., Coutinho, P., and Sequeiros, J. (2001). Improvement in the molecular diagnosis of machado-joseph disease. *Archives of neurology*, 58(11):1821–1827.
- [Maciel et al., 1995] Maciel, P., Gaspar, C., DeStefano, A. L., Silveira, I., Coutinho, P., Radvany, J., Dawson, D. M., Sudarsky, L., Guimarães, J., Loureiro, J. E., et al. (1995). Correlation between cag repeat length and clinical features in machado-joseph disease. *American journal of human genetics*, 57(1):54.

- [Mathis et al., 2021] Mathis, A., Biasi, T., Schneider, S., Yuksekgonul, M., Rogers, B., Bethge, M., and Mathis, M. W. (2021). Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1859–1868.
- [Mathis et al., 2018] Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. (2018). Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289.
- [Mathis et al., 2020] Mathis, A., Schneider, S., Lauer, J., and Mathis, M. W. (2020). A primer on motion capture with deep learning: principles, pitfalls, and perspectives. *Neuron*, 108(1):44–65.
- [Mathis and Warren, 2018] Mathis, A. and Warren, R. (2018). On the inference speed and video-compression robustness of deeplabcut. *BioRxiv*, page 457242.
- [Moeslund et al., 2006] Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3):90–126.
- [Nath* et al., 2019] Nath*, T., Mathis*, A., Chen, A. C., Patel, A., Bethge, M., and Mathis, M. W. (2019). Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nature Protocols*.
- [Newell et al., 2016] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer.
- [Nóbrega and de Almeida, 2012] Nóbrega, C. and de Almeida, L. P. (2012). Machado-joseph disease/spinocerebellar ataxia type 3. *Spinocerebellar Ataxia*.
- [Nóbrega et al., 2014] Nóbrega, C., Nascimento-Ferreira, I., Onofre, I., Albuquerque, D., Déglon, N., and Pereira de Almeida, L. (2014). Rna interference mitigates motor and neuropathological deficits in a cerebellar mouse model of machado-joseph disease. *PLoS One*, 9(8):e100086.
- [Ogun et al., 2015] Ogun, S. A., Martins, S., Adebayo, P. B., Dawodu, C. O., Sequeiros, J., and Finkel, M. F. (2015). Machado–joseph disease in a nigerian family: mutational origin and review of the literature. *European Journal of Human Genetics*, 23(2):271–273.
- [Orphanet, 2022] Orphanet (January 2022). Orphanet report series - prevalence of rare diseases: Bibliographic data.
- [Oue et al., 2009] Oue, M., Mitsumura, K., Torashima, T., Koyama, C., Yamaguchi, H., Furuya, N., and Hirai, H. (2009). Characterization of mutant mice that express polyglutamine in cerebellar purkinje cells. *Brain research*, 1255:9–17.

- [Paulson, 2012] Paulson, H. (2012). Machado–joseph disease/spinocerebellar ataxia type 3. *Handbook of clinical neurology*, 103:437–449.
- [Peleh et al., 2019] Peleh, T., Bai, X., Kas, M. J., and Hengerer, B. (2019). Rfid-supported video tracking for automated analysis of social behaviour in groups of mice. *Journal of Neuroscience Methods*, 325:108323.
- [Pereira et al., 2019] Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S.-H., Murthy, M., and Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nature methods*, 16(1):117–125.
- [Pereira et al., 2022] Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., Papadoyannis, E. S., Normand, E., Deutsch, D. S., Wang, Z. Y., et al. (2022). Sleep: A deep learning system for multi-animal pose tracking. *Nature methods*, 19(4):486–495.
- [Pishchulin et al., 2016] Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., and Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937.
- [Poppe, 2007] Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer vision and image understanding*, 108(1-2):4–18.
- [Rajamani et al., 2017] Rajamani, K., Liu, J.-W., Wu, C.-H., Chiang, I.-T., You, D.-H., Lin, S.-Y., Hsieh, D.-K., Lin, S.-Z., Harn, H.-J., and Chiou, T.-W. (2017). n-butylidenephthalide exhibits protection against neurotoxicity through regulation of tryptophan 2, 3 dioxygenase in spinocerebellar ataxia type 3. *Neuropharmacology*, 117:434–446.
- [Ramani et al., 2015] Ramani, B., Harris, G. M., Huang, R., Seki, T., Murphy, G. G., Costa, M. d. C., Fischer, S., Saunders, T. L., Xia, G., McEachin, R. C., et al. (2015). A knockin mouse model of spinocerebellar ataxia type 3 exhibits prominent aggregate pathology and aberrant splicing of the disease gene transcript. *Human molecular genetics*, 24(5):1211–1224.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- [Sakai et al., 1983] Sakai, T., Ohta, M., and Ishino, H. (1983). Joseph disease in a non-portuguese family. *Neurology*, 33(1):74–74.
- [Sandler et al., 2018] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.
- [Sanworks LLC, nd] Sanworks LLC (n.d.). Bpod. <https://sites.google.com/site/bpoddocumentation/home?authuser=0>. Accessed on April 13, 2023.
- [Scherzed et al., 2012] Scherzed, W., Brunt, E., Heinsen, H., De Vos, R., Seidel, K., Bürk, K., Schöls, L., Auburger, G., Del Turco, D., Deller, T., et al. (2012). Pathoanatomy of cerebellar

- degeneration in spinocerebellar ataxia type 2 (sca2) and type 3 (sca3). *The Cerebellum*, 11:749–760.
- [Schmidt and Schmidt, 2018] Schmidt, J. and Schmidt, T. (2018). Animal models of machado-joseph disease. *Polyglutamine Disorders*, pages 289–308.
- [Schöls et al., 2004] Schöls, L., Bauer, P., Schmidt, T., Schulte, T., and Riess, O. (2004). Autosomal dominant cerebellar ataxias: clinical features, genetics, and pathogenesis. *The Lancet Neurology*, 3(5):291–304.
- [Shakkottai et al., 2011] Shakkottai, V. G., do Carmo Costa, M., Dell’Orco, J. M., Sankaranarayanan, A., Wulff, H., and Paulson, H. L. (2011). Early changes in cerebellar physiology accompany motor dysfunction in the polyglutamine disease spinocerebellar ataxia type 3. *Journal of Neuroscience*, 31(36):13002–13014.
- [Siegle et al., 2017] Siegle, J. H., López, A. C., Patel, Y. A., Abramov, K., Ohayon, S., and Voigts, J. (2017). Open ephys: an open-source, plugin-based platform for multichannel electrophysiology. *Journal of neural engineering*, 14(4):045003.
- [Silva* et al., 2023] Silva*, A., Angeja*, L., Gabriel, G. L., Henriques, C., Gaspar, L. S., Pereira de Almeida, L., and Nobre, R. J. (2023). Motor performance in mouse models of spinocerebellar ataxia type 3: a meta-analysis. *Submitted to "Journal of Translational Medicine"*.
- [Silva-Fernandes et al., 2010] Silva-Fernandes, A., do Carmo Costa, M., Duarte-Silva, S., Oliveira, P., Botelho, C. M., Martins, L., Mariz, J. A., Ferreira, T., Ribeiro, F., Correia-Neves, M., et al. (2010). Motor uncoordination and neuropathology in a transgenic mouse model of machado-joseph disease lacking intranuclear inclusions and ataxin-3 cleavage products. *Neurobiology of disease*, 40(1):163–176.
- [Silva-Fernandes et al., 2014] Silva-Fernandes, A., Duarte-Silva, S., Neves-Carvalho, A., Amorim, M., Soares-Cunha, C., Oliveira, P., Thirstrup, K., Teixeira-Castro, A., and Maciel, P. (2014). Chronic treatment with 17-dmag improves balance and coordination in a new mouse model of machado-joseph disease. *Neurotherapeutics*, 11:433–449.
- [Simoes et al., 2012] Simoes, A. T., Gonçalves, N., Koeppen, A., Deglon, N., Kügler, S., Duarte, C. B., and Pereira de Almeida, L. (2012). Calpastatin-mediated inhibition of calpains in the mouse brain prevents mutant ataxin 3 proteolysis, nuclear localization and aggregation, relieving machado-joseph disease. *Brain*, 135(8):2428–2439.
- [Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [Sturman et al., 2020] Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., Grimm, C., Thieren, L., Zerbi, V., Grewe, B., et al. (2020). Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology*, 45(11):1942–1952.

- [Sun et al., 2022] Sun, J. J., Ryou, S., Goldshmid, R. H., Weissbourd, B., Dabiri, J. O., Anderson, D. J., Kennedy, A., Yue, Y., and Perona, P. (2022). Self-supervised keypoint discovery in behavioral videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2171–2180.
- [Switonski et al., 2015] Switonski, P. M., Szlachcic, W. J., Krzyzosiak, W. J., and Figiel, M. (2015). A new humanized ataxin-3 knock-in mouse model combines the genetic features, pathogenesis of neurons and glia and late disease onset of sca3/mjd. *Neurobiology of disease*, 73:174–188.
- [Takiyama et al., 1993] Takiyama, Y., Nishizawa, M., Tanaka, H., Kawashima, S., Sakamoto, H., Karube, Y., Shimazaki, H., Soutome, M., Endo, K., Ohta, S., et al. (1993). The gene for machado-joseph disease maps to human chromosome 14q. *Nature genetics*, 4(3):300–304.
- [Tan and Le, 2019] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.
- [Taroni and DiDonato, 2004] Taroni, F. and DiDonato, S. (2004). Pathways to motor incoordination: the inherited ataxias. *Nature Reviews Neuroscience*, 5(8):641–655.
- [Torashima et al., 2008] Torashima, T., Koyama, C., Iizuka, A., Mitsumura, K., Takayama, K., Yanagi, S., Oue, M., Yamaguchi, H., and Hirai, H. (2008). Lentivector-mediated rescue from cerebellar ataxia in a mouse model of spinocerebellar ataxia. *EMBO reports*, 9(4):393–399.
- [Toshev and Szegedy, 2014] Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660.
- [Tsimpanouli et al., 2022] Tsimpanouli, M.-E., Ghimire, A., Barget, A. J., Weston, R., Paulson, H. L., Costa, M. d. C., and Watson, B. O. (2022). Sleep alterations in a mouse model of spinocerebellar ataxia type 3. *Cells*, 11(19):3132.
- [Wu et al., 2020] Wu, X., Sahoo, D., and Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, 396:39–64.
- [Wu et al., 2022] Wu, Y.-L., Chang, J.-C., Sun, H.-L., Cheng, W.-L., Yen, Y.-P., Lin, Y.-S., Chao, Y.-C., Liu, K.-H., Huang, C.-S., Liu, K.-L., et al. (2022). Coenzyme q10 supplementation increases removal of the atxn3 polyglutamine repeat, reducing cerebellar degeneration and improving motor dysfunction in murine spinocerebellar ataxia type 3. *Nutrients*, 14(17):3593.

Appendices

A

DeepLabCut Parameter Comparison

This appendix provides an in-depth comparison of CNN parameter values, demonstrating their impact on the outcomes. To reduce the search domain, we ran comparisons by modifying only one selected parameter while maintaining all other parameters configured with the initial values, considered as an appropriate starting point. These are referred to as the standard values, illustrated in Table A.1. We expected that, by doing so, we could reduce the variability of the comparisons and provide more objective results.

Table A.1: Standard Values for DeepLabCut Hyperparameter Comparison.

Hyperparameters	Batch Size	P-Cutoff	Crop	Global Scale
Values	2	0.1	False	0.8

A.1 Batch Size

The overall difference between the train errors across the networks with different batch sizes is observable in the graph below (Figure A.1). It is most observable in iterations 100 000 and 150 000 although still being less than a pixel's difference. Regarding the test errors of the different batch sizes, they are very close together occasionally with even less than 0.5 pixels of variation between them.

At iteration number 200 000, the test and train errors for all batch size networks were nearly identical. While some minor differences of up to 1.5 pixels were observable in the snapshots, these findings suggest that batch sizes ranging from 1 to 4 do not significantly impact the network's inference model.

Although not significant, we can see that the difference between train and test error is slightly larger for the batch size 4 network, when compared to the remaining two. It is also noticeable that the test and novel set errors for this network appear nearly coincidental, which suggests this network works very well when fed novel data, despite the bigger error difference for the train error. Because these error variations are occurring within the same range for all the tested networks, ranging between a minimum of 1.65 and a maximum of 4.20 pixel errors, we decided either one of them offered a good option for implementation.

Furthermore, at iteration number 200 000, we can see that the three different training sets

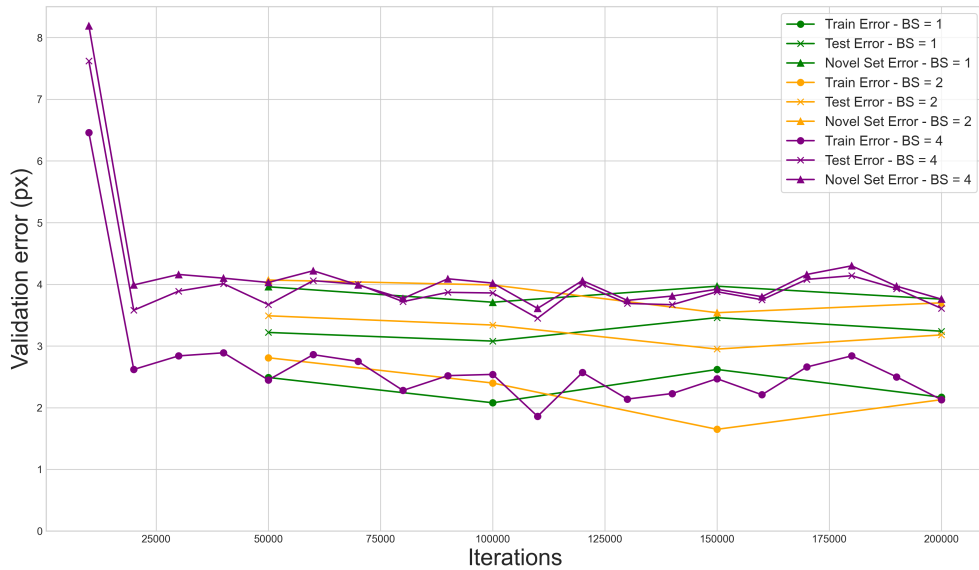


Figure A.1: Comparison of Train and Test error of Batch Size 1, 2 and 4.

obtain nearly identical values for the training error, as well as for the novel set error. The same results can be observed for the test errors, with a slight deviation of the network with batch size 4. These findings inspire confidence in all three networks.

Bigger batch sizes were not able to be tested due to hardware limitations, i.e. the computer GPU’s 8GB dedicated memory was not able to withstand larger batch sizes. Furthermore, we noticed the training time was incrementing very significantly with the batch size increase, presumably for the same reason. For example, training 200 000 iterations with a batch size of 1 took approximately 3.5 hours and 6 hours for the same number of iterations with a batch size of 2. For a batch size a 4, the network took around 12 hours to reach 200 000 iterations.

Considering all the above, we decided to move forward with the network of batch size 2, choosing the weights of iteration 150 000, as this was the snapshot, of all the saved snapshots, that obtained the best results, as we can see in the graph. Although the network with a batch size of 1 was the fastest to train until a stable plateau, we decided to use the batch size of 2 network due to yielding slightly better error values in training, testing and the novel set. Having said that, we consider that using the default batch size of 1 would be a good decision for a problem such as ours, if time for testing this parameter is a constraint.

A.2 Global Scale

We initially set the *global_scale* (GS) parameter to the default value of 0.8. However, given the significance of understanding the behaviour of our CNN and considering the resolution of our images, we sought to explore alternative values for the global scale. Consequently, we conducted tests with a value of 0.9 for this parameter, aiming to take full advantage of our video resolution,

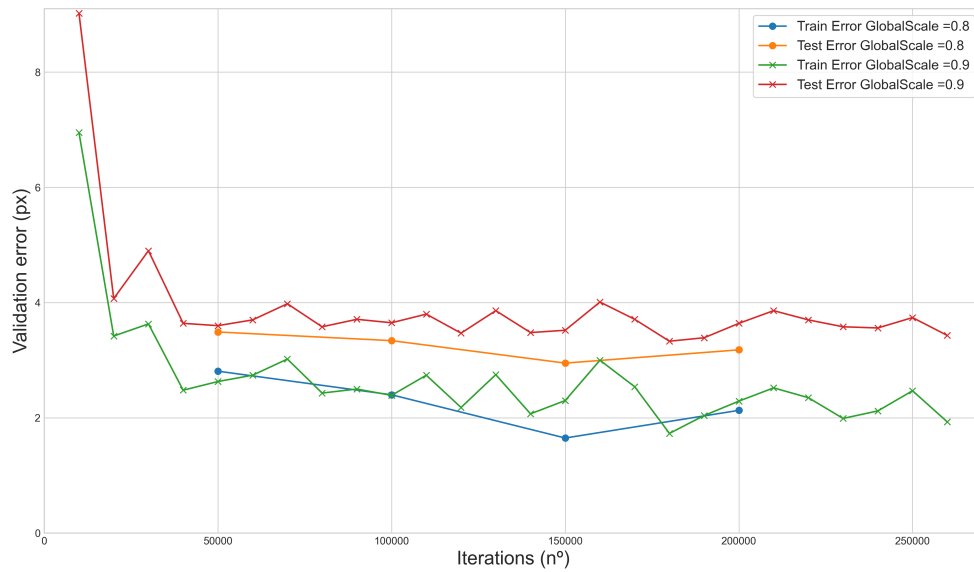


Figure A.2: Comparison of Train and Test error of Global Scale = 0.8 and Global Scale = 0.9.

which is not excessively high.

Upon analysis, we observed that adjusting the *global_scale* parameter did not lead to a substantial impact on the network or its training time. Nevertheless, there was a minor increase in both the train and test errors associated with the modified value (Figure A.2).

After carefully considering the results, we ultimately decided to revert to the default value of 0.8 for the *global_scale* parameter. This choice was made to ensure optimal performance and stability in our CNN, aligning it more effectively with the specific requirements of our work.

A.3 Data Augmentation - Cropped Images

We conducted training on two different networks, one with the cropping parameter set to TRUE and the other with the parameter set to FALSE (Figure A.3).

Remarkably, we found that both the train and test errors exhibited remarkably similar patterns for the CNN, regardless of whether the cropping parameter was set to TRUE or FALSE. However, when the cropping parameter was set to FALSE, the error exhibited a more rapid decline, continuing to decrease throughout the iterations until reaching 150 000. However, at iteration 200 000, we observed a slight increase in the error, suggesting potential overfitting of the CNN.

Conversely, for the CNN with the cropping parameter set to TRUE, the error stabilization occurred at a slightly higher value with the increment of the iterations.

Based on our comprehensive analysis, we concluded that, within the scope of our problem,

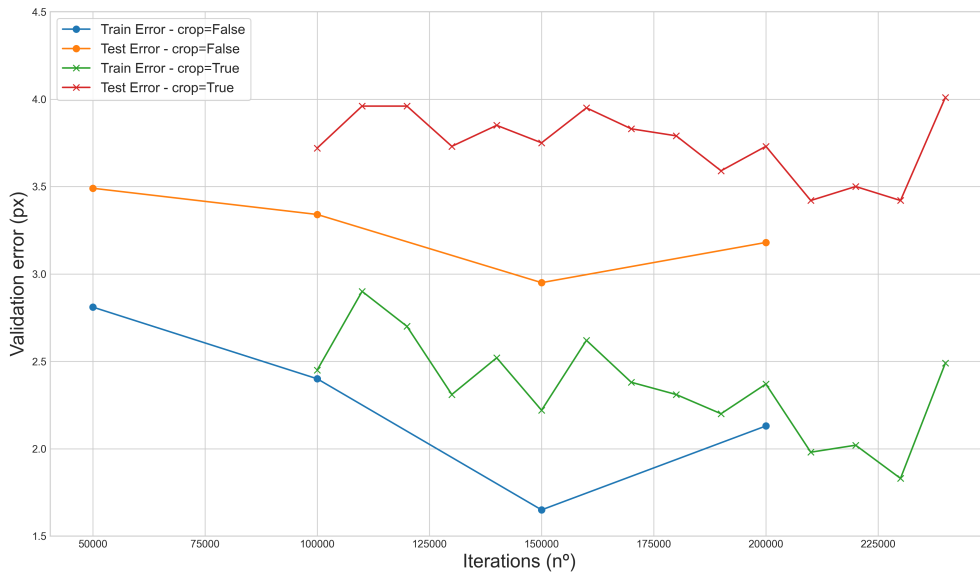


Figure A.3: Comparison of Train and Test error of Crop = TRUE and Crop = FALSE.

the cropping parameter did not play a fundamental role. Consequently, we made the decision to retain the parameter setting as FALSE.

A.4 Likelihood Threshold - *p-cutoff*

In our CNN training process, we adjusted the *p-cutoff* parameter, which controls the confidence level of our network’s predictions. The default value for the parameter *p-cutoff* was 0.1, indicating that body parts with a likelihood above 0.1 were considered for video labeling. We experimented with setting this value to 0.8, meaning that only body parts with a high confidence level (above 0.8) in each frame were used in constructing the labeled video.

When comparing the two training sessions (0.1 and 0.8), we observed only a small increase in the testing error with the higher *p-cutoff* (Figures A.4 and A.5). Additionally, in both training sessions, the discrepancy between the network’s train and test errors, both with and without the defined *p-cutoff*, was minimal. The differences were nearly imperceptible in the training sets, while in the testing sets they amount to only a few decimal points.

Our choice of 0.8 as the *p-cutoff* value was driven by the understanding that confidence improves as the network undergoes more iterations. Setting a high *p-cutoff* initially could lead to inaccuracies due to low confidence in the initial training iterations. However, due to the proximity of the two CNNs trained with the *p-cutoff* values and keeping in mind we aim to achieve the truest results for the x and y values, we decided to set this value to 0.8.

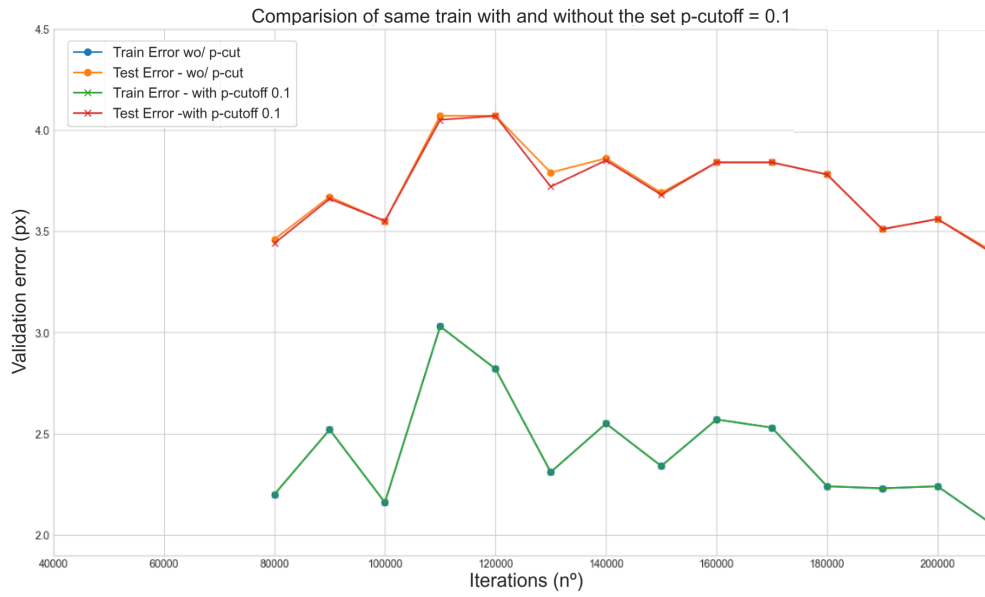


Figure A.4: Comparison of Train and Test error with and without p -cutoff = 0.1.

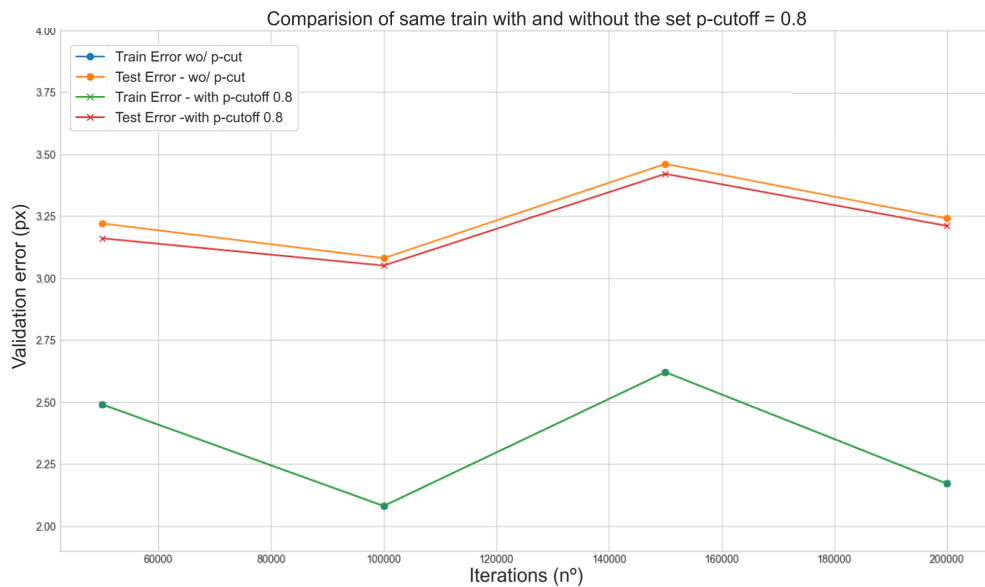


Figure A.5: Comparison of Train and Test error with and without p -cutoff = 0.8.

B

Supplementary Figures of Bonsai Analysis

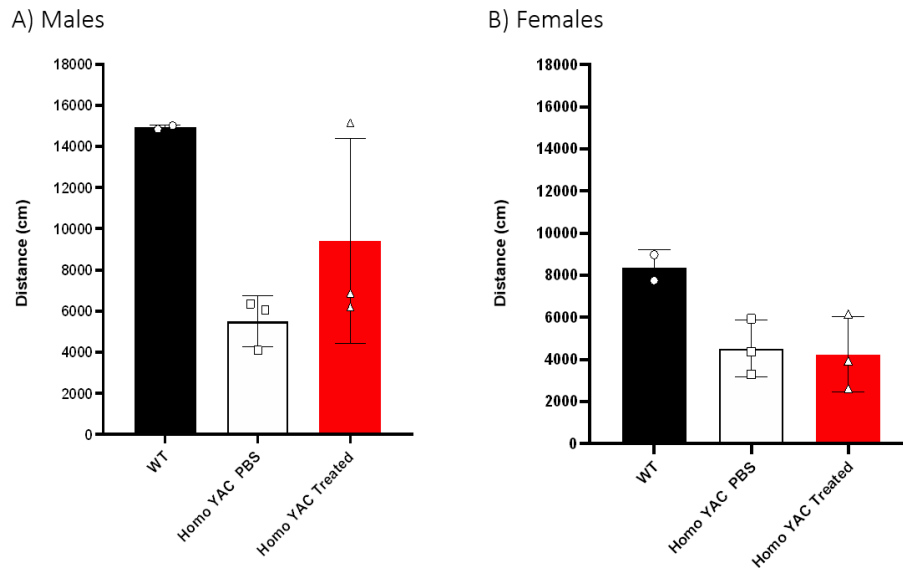


Figure B.1: Comparison of distance travelled across gender using Bonsai. Gender separated analysis of distance travelled in wild-type, Homo YAC PBS and Homo YAC Treated groups. (a) total distance travelled (cm) among males and (b) total distance travelled (cm) among females. The data are presented as the mean \pm SD. Statistical analysis was performed using the ANOVA test.

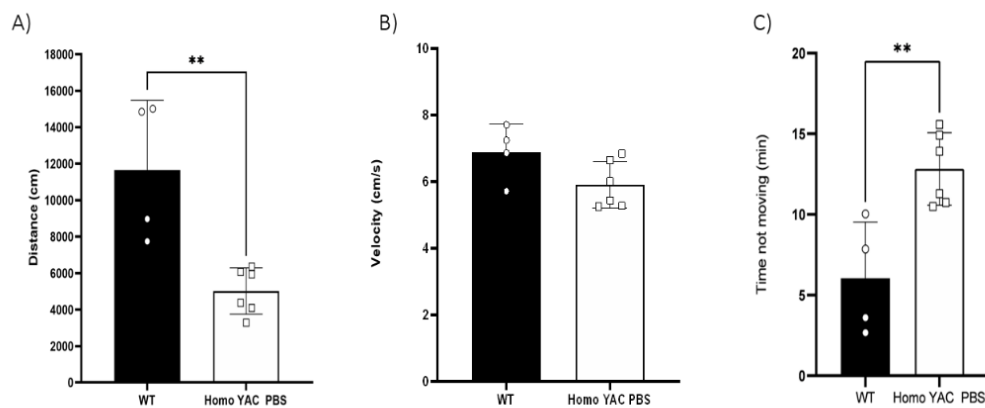


Figure B.2: Comparison of open field test performance across Wild-Type and Transgenic Homo YAC PBS-treated using Bonsai. Comparison of phenotypic features in wild-type and Homo YAC PBS. Locomotor activity was analysed by measuring: (a) total distance travelled (cm), (b) average velocity (cm/s) and (c) resting time (min). The data are presented as the mean \pm SD. Statistical analysis was performed using the unpaired Student's t-test, ** $p < 0.01$.

C

Supplementary Figures of DeepLabCut Analysis

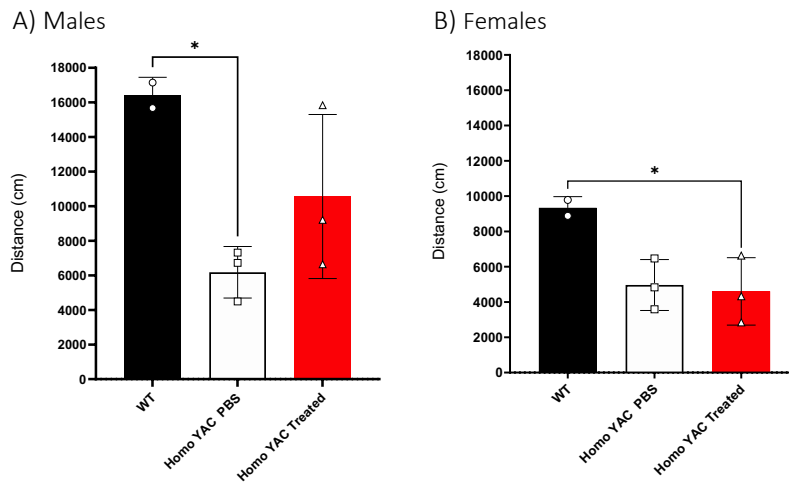


Figure C.1: Comparison of distance travelled across gender using DeepLabCut. Gender separated analysis of distance travelled in wild-type, Homo YAC PBS and Homo YAC Treated groups. (a) total distance travelled (cm) among males and (b) total distance travelled (cm) among females. The data are presented as the mean \pm SD. Statistical analysis was performed using the ANOVA test, * $p < 0.05$.

D

**Supplementary Figures of
DeepLabCut Keypoint Tracking
Analysis**

Wild-type Animal

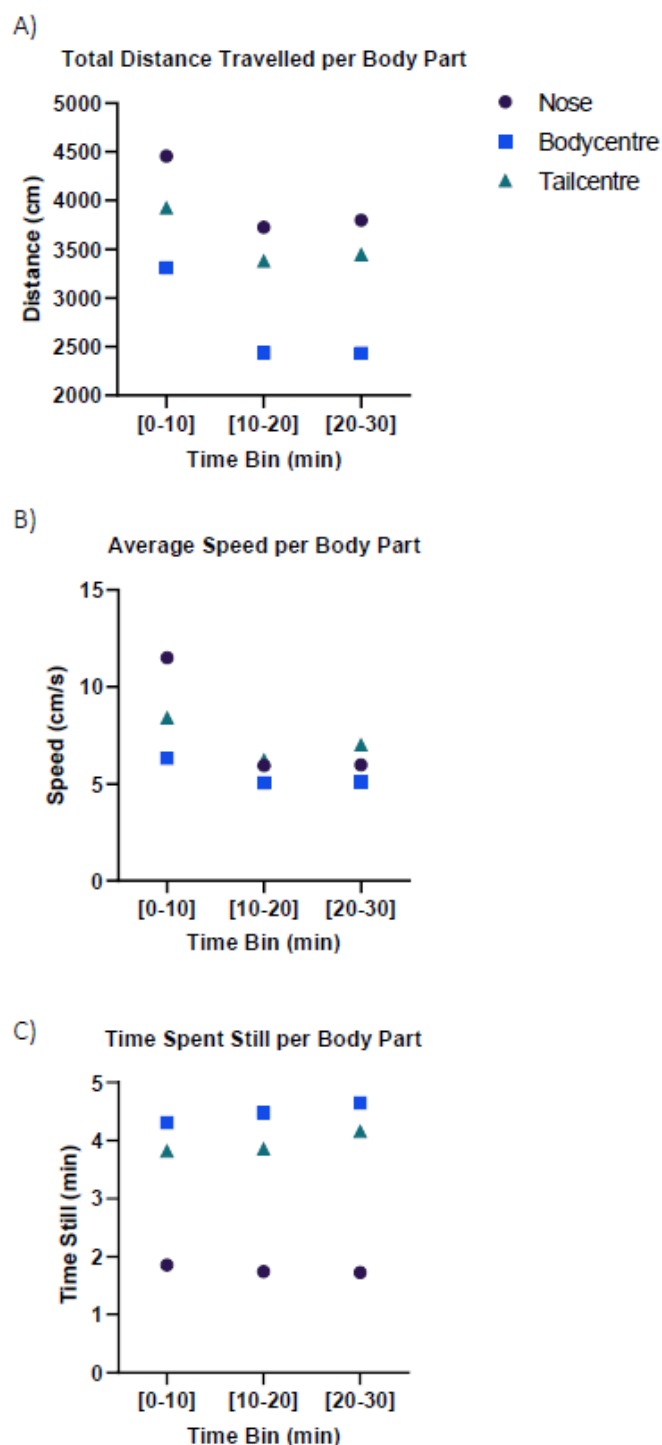


Figure D.1: Total Distance detailed by a wild-type per body part, nose, body centre and tail centre. Locomotor activity was analysed by measuring: (a) total distance travelled (cm), (b) average velocity (cm/s) and (c) resting time (min). Time is divided into 3 time bins of 10 minutes.

Homo YAC PBS Animal

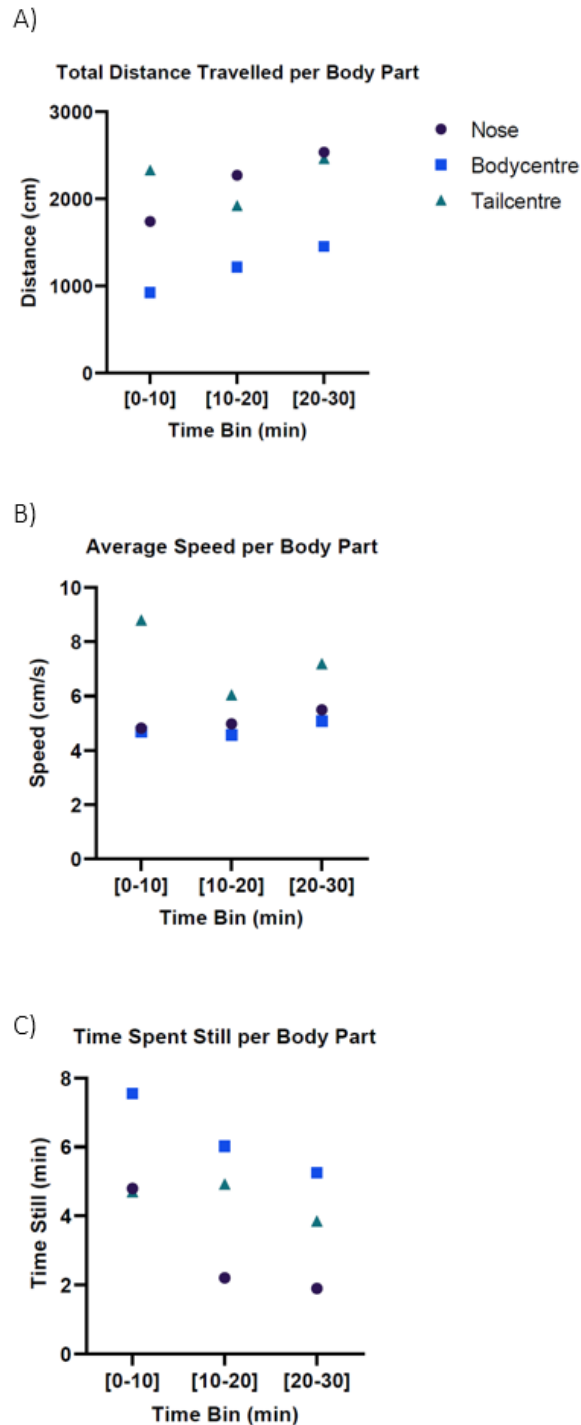


Figure D.2: Total Distance detailed by a Homo YAC PBS per body part, nose, body centre and tail centre. Locomotor activity was analysed by measuring: (a) total distance travelled (cm), (b) average velocity (cm/s) and (c) resting time (min). Time in divided into 3 time bins of 10 minutes.

Keypoint Tracking Results

Table D.1: Values of the measurements for Wild-Type group.

	Distance Travelled (cm)	Velocity (cm/s)	Resting Time (min)
Nose	18 488 \pm 4570	8.98 \pm 2.66	3.42 \pm 1.86
Body Centre	12 972 \pm 3591	6.58 \pm 1.46	9.8 \pm 4.26
Tail Centre	18 921 \pm 5118	9.54 \pm 2.64	8.91 \pm 3.59

Table D.2: Values of the measurements for Homo PBS YAC group.

	Distance Travelled (cm)	Velocity (cm/s)	Resting Time (min)
Nose	9035 \pm 1554	5.84 \pm 0.52	5.84 \pm 1.72
Body Centre	5570 \pm 1343	5.41 \pm 0.54	15.02 \pm 2.99
Tail Centre	8238 \pm 1625	7.41 \pm 0.62	12.63 \pm 2.19