Using machine learning to discover candidate localised transcripts from microscopy and genome-wide bioinformatics data



Maria Kiourlappou

Linacre College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2023

To my parents, George and Chrysanthi Kiourlappou.

Abstract

Messenger RNA (mRNA) localisation is an important layer of post-transcriptional regulation. In polarised cells such as glia, mRNA localisation is especially significant as glial cells are long with protrusions that can be far away from the cell body, where mRNA is produced. Thus, mRNA transport and localised translation are essential to ensure that mRNAs and proteins reach their intended destination in a timely and coordinated manner. Proper mRNA localisation is vital for the correct functioning of the entirety of a glial cell and dysregulation of mRNA localisation in glial cells has been implicated in neurodegenerative and neurological diseases and some types of cancer. Therefore, gaining a deeper understanding of the mechanisms that mediate mRNA localisation is of paramount value.

Experimental techniques such as single molecule fluorescence in-situ hybridization (smFISH) or RNA sequencing (RNA-seq) are often utilised to study mRNA localisation, the former to view a visual map of the mRNAs' spatial distribution and expression and the latter to study with more quantitative detail their expression in single tissues or bulk tissue. In this thesis, I study mRNA localisation using computational pipelines to systematically interrogate datasets available to study mRNA localisation in glia and additionally I am using supervised machine learning models to predict candidate genes that could potentially be localised in glial cell protrusions.

Firstly, I present a YFP protein trap microscopy screen conducted collaboratively by members of the Davis lab for 200 genes in the *Drosophila melanogaster* larval nervous system. In this screen we explore the transcriptomic and proteomic spatial distribution for the 200 genes using smFISH, for seven compartments in the larval nervous system. Next, I present the innovative pipeline I use to systematically manage the output raw microscopy images and to create figures (approximately 1400 figures). Next in the pipeline is the use of a scoring application I co-produced with Dr David Pinto in-house for expert scorers to annotate the figures created. Three expert scorers use the application to annotate the figures with labels describing mRNA and protein expression patterns and levels. After the production of labels, I present the figures using a novel visualisation tool which allows a user to explore the images alongside with the labels describing localisation for each gene and the different compartments examined for each gene. Additionally, I incorporated genome-wide bioinformatic datasets in such a way so that a user can use the visualisation tool to discover associations between the figures, the localisation labels and the bioinformatics datasets for different genes or groups of genes.

Next, I focus on one tissue of interest: glial cells. I create localisation labels for *Drosophila melanogaster* and *Mus musculus* using the smFISH microscopy screen and RNA-seq analysed

datasets identified from literature, by Dr Jeff Lee and Dalia Gala. I then create a conglomerate of heterogeneous datasets, all related to mRNA localisation. I then transform the datasets into interoperable and compatible, gene-centric datasets ready to be used by machine-learning algorithms.

Finally, I explore different types of machine learning algorithms to assess their suitability for my datasets. I choose Random Forest (RF) as the most compatible algorithm and adapt a pipeline based on RF for sleep candidate gene selection. Using the adapted pipeline I predict 66 genes highly likely to be localised in glial cells protrusions and assess their functional and cellular attributes using GO enrichment. I find that the group of predicted genes is highly enriched in terms involving localisation, transport and cell protrusions. Additionally, I discover that the most influential factor in the training of the RF model is the RNA binding proteins (RBPs) target site motifs. I further investigate the specific RBPs that contribute the most to the model and find a large percentage of them have established roles in mRNA localisation and RNA methylation.

The work completed in this thesis, is an exciting starting point for using novel visualisation tools to explore data and machine learning techniques to study mRNA localisation. The results I described in the previous paragraph are the first indication of the success of using a machine learning approach.

Declaration

I hereby declare that this thesis is my own work and the data analysis I present in the following pages was performed by myself, unless otherwise stated. The experimental analysis was carried out in the Department of Biochemistry, University of Oxford. The primary imaging data was generated by others as stated explicitly in each case. The microscopy data was assembled into figures using OMERO.figure by many Davis lab members including myself. This thesis has not been submitted for any other degree or professional qualifications.

Maria Kiourlappou
May 2023

Acknowledgements

I cannot begin to express my thanks to my supervisors, Ilan and Steve. Ilan, I am extremely grateful to you for giving me the opportunity to take on this project. Thank you for trusting me with such an interesting and unique project. Although the work has been challenging at times, with your constant support, encouragement, your perseverance and trust in my ability to complete the project, I am here finishing my DPhil. Steve, this endeavour would not have been possible without you either. I am extremely grateful for your guidance, for your nonstop encouragement and for always discussing captivating new ideas which were instrumental in shaping my research work.

I gratefully acknowledge the Biotechnology and Biological Sciences Research Council (BBSRC), Zegami Ltd and the A. G. Leventis Foundation for funding this project.

I am thankful to the Davis lab. I could not have joined a more supportive and welcoming environment. Special thanks to Aino Jarvelin, not only for being my day-to-day supervisor but for becoming my friend, thank you to Darragh Ennis for always helping me with lab and non lab related problems and to Jeff Lee; I could not have worked with a more helpful colleague, you always answered every question I had instantly. I also would like to thank Josh Titlow, MK Thompson and Dalia Gala for the excellent collaboration and guidance.

Thank you to my friends, Marilena, Despoina, Vasilis and Nikos, for being my second family here in Oxford when I first moved. Without you paving the way I wouldn't even have thought of applying to do a DPhil. To my brother Andreas thank you for helping me feel like home in Oxford. To my friends I made throughout the PhD, Greek and non-Greek, I am extremely grateful for spending the last 5 years in your company; for all the good times we shared and the laughs that made my life in Oxford beautiful. Last but not least, thank you Jean-Christophe for supporting me through the last difficult months of the write-up period, you have made this time enjoyable.

I would like to express my heartfelt gratitude to all the teachers and professors who have been part of my academic journey. Their kindness, dedication and support has been invaluable. You have inspired me and taught me to love learning as much as you did. To my first research supervisor Suchitra Sebastian thank you for giving me the opportunity to participate in my first research project and develop my passion for research. To my academic director of studies and supervisor Owen Saxton, your mentorship and support during my undergraduate degree was invaluable and your belief in my abilities made me want to strive for excellence.

Finally, and most of all, thank you to my parents, George and Anthi. Thank you being my biggest and warmest supporters always supporting me constantly throughout all stages of my life. I owe who I am and all my achievements to your unconditional love and endless encouragement. You have been my rock. Without you I would not have been here.

Contents

Abstract iii

Declaration v

Acknowledgements vi

List of Abbreviations xi

1 Introduction 1

1.1 RNA localisation . . . 2

1.1.1 *Drosophila* . . . 2

1.1.2 *Drosophila* third instar larval nervous system . . . 3

1.1.3 Importance of mRNA localisation . . . 5

1.1.4 Mechanisms underlying RNA localisation . . . 5

1.1.5 Experimental methods for studying RNA localisation . . . 8

1.2 Glia . . . 10

1.2.1 RNA localisation and local translation in glia . . . 12

1.2.2 Studying RNA localisation in glia in *Mus musculus* . . . 12

1.3 Big data in Biology . . . 13

1.3.1 Managing data on a local and a global level . . . 13

1.3.2 Making different types of data from heterogeneous sources interoperable 17

1.3.3 Visualising heterogeneous big biological data . . . 19

1.4 Machine learning in Biology . . . 21

1.4.1 A Machine learning project from start to finish . . . 22

1.5 Predicting RNA localisation . . . 31

1.6 The structure of the thesis . . . 34

1.7 Aims of the thesis . . . 36

2 Intermediate-throughput imaging screen of mRNA and protein expression in the *Drosophila melanogaster* larval nervous system 38

2.1 Introduction . . . 40

2.1.1 Aims and objectives . . . 41

2.2 Materials and Methods . . . 41

2.2.1 The Cambridge protein trap insertion (CPTI) collection . . . 41

2.2.2 Imaging and data management using OMERO . . . 45

2.2.3 Reaching consensus and resolving conflict . . . 48

2.2.4 Visualising and merging multiple types of data . . . 55

2.3 Results and Discussion . . . 61

2.3.1 The 200 selected and screened genes from the CPTI YFP protein trap collection are a sample that represents faithfully the whole protein-coding genome . . . 61

2.3.2 There is a high percentage of discordance between mRNA and protein expression, with at least one nervous system region that shows discordance for 195 genes out of 200 genes (97.5 %). . . . 64

2.3.3 mRNA and protein express in the tips of glial cells in the *Drosophila* nervous system . . . 69

2.3.4 Confidence in quality of annotations can be high even when inter annotator agreement is not . . . 71

2.3.5 Can the produced annotations be used as supervised machine learning predictive labels? . . . 77

2.3.6 An adaptable pipeline for assessing gene expression in the *Drosophila* nervous system via simultaneous data visualisation and data exploration that can be generalised and also be used for other purposes . . . 78

2.4 Summary and Conclusions . . . 82

3 Exploration of datasets for mRNA localisation prediction in glia 84

3.1 Introduction . . . 85

3.1.1 Aims and Objectives . . . 87

3.2 Materials and Methods . . . 88

3.2.1 Labels creation . . . 88

3.2.2 Identifying and accessing datasets of interest from online bioinformatics repositories . . . 91

3.2.3 The Gene Ontology Annotations (GOA) database . . . 95

3.2.4 Data preprocessing and feature engineering . . . 102

3.3 Results and Discussion . . . 114

3.3.1 The homologs of *Drosophila* genes' transcripts localising in glia and the *Mus* genes' transcripts found to be present in protrusions are overlapping 114
3.3.2 Assessing missing data values of all datasets used and explaining the potential reasons behind missing values . . . 118
3.3.3 Grouping of GO term annotations and evaluating the genome coverage of the selected groupings . . . 126
3.3.4 Sparsity of features and feature selection using correlation between the features . . . 135
3.4 Summary and Conclusions . . . 140

4 Using supervised and unsupervised learning to predict mRNA localisation in glia 143
4.1 Introduction . . . 145
4.1.1 Aims and Objectives . . . 147
4.2 Materials and Methods . . . 147
4.2.1 Data and data preparation . . . 147
4.2.2 Supervised machine learning . . . 148
4.2.3 Dimensionality reduction . . . 150
4.2.4 Clustering techniques . . . 153
4.2.5 Adapting the approach for identifying candidate sleep regulators to identifying genes with potential to localise . . . 154
4.2.6 Pathway enrichment analysis using REACTOME pathways data . . . 158
4.3 Results and Discussion . . . 159
4.3.1 Exploring the correlation between a dataset's features using machine learning approaches . . . 159
4.3.2 Dimensionality reduction using PCA . . . 161
4.3.3 Candidate gene selection by adapting a sleep-gene-candidates identification approach . . . 173
4.4 Summary and Conclusions . . . 185

5 Discussion and future perspectives 187
5.1 Future perspectives . . . 191
5.1.1 Predicting mRNA localisation patterns based on the microscopy images and the localisation labels for the 200 genes . . . 191
5.1.2 Impact of RNA secondary structure on mRNA localisation . . . 193
5.1.3 Improving of the machine learning algorithm prediction . . . 194
5.2 Conclusion . . . 194

Appendices

A List of CPTI lines screened **197**

B Sensitivity analysis of perturbations to the annotations **203**

C Multi-Dimensional Viewer (MDV): RNA/protein distribution of 200 YFP trap lines **205**

D Datasets used and analysed to get localisation labels **209**

E Clustering results using k-means **210**

References **212**

List of Abbreviations

3'UTR . . . . . 3' Untranslated Region

5'UTR . . . . . 5' Untranslated Region

APIs . . . . . Application Programming Interfaces

AUC . . . . . Area Under the Curve

BP . . . . . Biological Process

CB . . . . . Central brain

CC . . . . . Cellular Component

CNS . . . . . Central Nervous System

CPTI . . . . . Cambridge Protein Trap Insertion

DAG . . . . . Directed Acyclic Graph

DNA . . . . . Deoxyribonucleic Acid

EDA . . . . . Exploratory Data Analysis

EFs . . . . . Evidence Factors

FPKM . . . . . Fragments Per Kilobase of exon per million fragments Mapped

GO . . . . . Gene Ontology

GOA . . . . . Gene Ontology Annotations

GUI . . . . . Graphical User Interface

IC . . . . . Information Content

IDR . . . . . Image Data Resource

MB . . . . . Mushroom Body

MDV . . . . . Multi-Dimensional Viewer

MF . . . . . Molecular Function

MGI . . . . . Mouse Genome Informatics

MLV . . . . . Multi Locus View

mRNA . . . . . messenger RNA

NBs . . . . . Neuroblasts

ncRNA . . . . . non-coding RNA

NMJ . . . . . Neuromuscular Junction

NN . . . . . Neural Network

OL . . . . . Optic Lobe

OMERO . . . . . Open Microscopy Environment's Remote Objects

ORF . . . . . Open Reading Frame

oRNAment . . . . o RNA motifs enrichment

PCs . . . . . Principal Components

PCA . . . . . Principal Component Analysis

PFAM . . . . . Protein Families

PNS . . . . . Peripheral Nervous System

PU . . . . . Positive and Unlabelled

RBP . . . . . RNA-Binding Protein

RF . . . . . Random Forest

RNA . . . . . Ribonucleic acid

RNA-seq . . . . . RNA sequencing

ROC . . . . . Receiver Operating Characteristic

scRNA-seq . . . . single-cell RNA sequencing

smFISH . . . . . single-molecule Fluorescence *In Situ* Hybridisation

TF . . . . . Transcription Factor

TPM . . . . . Transcript Per Million

URL . . . . . Uniform Resource Locator

VIF . . . . . Variance Inflation Factor

VNC . . . . . Ventral Nerve Cord

WCSS . . . . . Within-Cluster Sum of Square

YFP . . . . . Yellow Fluorescent Protein

1 Introduction

An "RNA World". This phrase, first coined by the Nobel laureate Walter Gilbert in 1986, has been used through the years to convey the importance of the many roles of the ribonucleic acid (RNA). On the most fundamental level, RNA is the key component in the transfer of genetic information from the deoxyribonucleic acid (DNA) to proteins, and on a parallel but equally significant level, RNA is a key regulator of gene expression. Post-transcriptional regulation of gene expression is a type of regulation at the RNA level and has recently gained traction as an important influence in many functional and biological cell processes (Buccitelli & Selbach, 2020). Localised information transfer via localisation of messenger RNA (mRNA) and on-site protein synthesis, is a conserved mechanism (Holt et al., 2019). Additionally, mRNA localisation is one of the hallmarks of post-transcriptional regulation.

In the 21st century, advancements in RNA sequencing (RNA-seq) (Emrich et al., 2007; Lister et al., 2008; Nagalakshmi et al., 2008), single-cell RNA sequencing (scRNA-seq) (Tang et al., 2009) and imaging technologies, which now enable imaging at single-molecule resolution sensitivity, such as single molecule fluorescence *in situ* hybridization (smFISH) (Femino et al., 1998; Raj et al., 2008) have revolutionised the way we study RNA. These methods produce vast amounts of data which can result in a huge data mountain. Unless the right tools and methodologies are developed and used to extract insights from these data, the colossal accumulation of data can lead

to a biological cul-de-sac. Fortunately, technological innovations and algorithmic breakthroughs, such as online repositories and management tools (Eliceiri et al., 2012), novel visualisation software (Pavlopoulos et al., 2015) and machine learning algorithms (Greener et al., 2022a) have enabled the handling, displaying, interrogation and extraction of meaning from the data.

In this Chapter, I discuss mRNA localisation, especially the methods used to study it and its underlying mechanisms. I introduce the *Drosophila melanogaster* (*Drosophila*) third instar larval nervous system where I observed localisation in glial cells and in addition I discuss instances of localisation in mammalian glial cells. Next, I examine key elements necessary to generate, manage, interrogate and visualise big biological data. Following this, I illustrate the typical steps required to complete a machine learning project and explore machine learning algorithms I will utilise in the rest of my thesis. Subsequently, I present a literature review on methods used to predict mRNA localisation. Closing the Introduction chapter, I outline the aims and structure of this thesis.

1.1 RNA localisation

1.1.1 *Drosophila*

Drosophila is a widely studied organism. Through the years many techniques have been developed to provide numerous different ways to genetically modify flies. Additionally *Drosophila* is a well annotated organism, with a lot of genomic, transcriptomic and proteomic information available openly online (Matthews et al., 2005), with the availability of public information and the desire for collaboration enabling a fruitful environment for research to thrive amongst the fly research community. Examples include the Cambridge Protein Trap Insertion (CPTI) line collection, a transgenic fly line collection which provides a protein trap library to study the behaviour and properties of proteins in vivo (Lowe et al., 2014). These lines can be freely utilised by the community as they are all available as stocks at the Drosophila Genetic Resource Center at the Kyoto Institute of Technology (see here: `http://www.dgrc.kit.ac.jp/`). Flybase (Gramates et al., 2022), Flymine (Lyne et al., 2007), UniProt (Apweiler, 2004) and Entrez Gene (Maglott et

al., 2011) are only some of the websites and online repositories that host genomic and proteomic sequences, annotations, interactions and literature regarding *Drosophila*.

1.1.2 *Drosophila* third instar larval nervous system

The *Drosophila* nervous system is a versatile model and much simpler compared to the human nervous system. Around 75% of genes related with disease in humans have a *Drosophila* homolog (Reiter et al., 2001), providing the possibility of investigating human-associated disorders using the *Drosophila* brain. The *Drosophila* third instar (L3, the final larval stage before pupation) brain is compact and simple, and facilitates the use of microscopy to study mRNA and protein expression at single-molecule resolution in the intact brain. Additionally, the fruit fly's quick generation time (4-5 days to reach third instar larval development (Tennessen & Thummel, 2011)) allows for quick larval acquisition and fast turnaround of experiments. The *Drosophila* nervous system comprises the central nervous system (CNS) and the peripheral nervous system (PNS). Key regions of the CNS and PNS are shown in Figure 1.1.

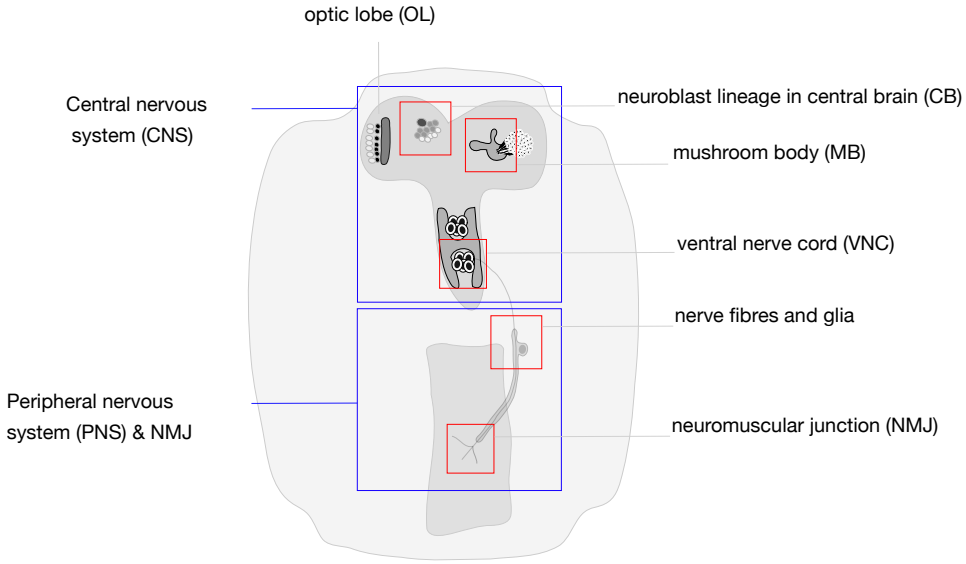


Figure 1.1: Overview of the third instar larval (L3) nervous system, showing important compartments both in the CNS and PNS. Schematic representation of the third instar larval nervous system of a dissected larva, showing the compartments optic lobe, central brain, mushroom body and the ventral nerve cord in the central brain. In the PNS, we can see the neuromuscular junction and the nerve fibres and glia. Figure adapted from Josh Titlow.

The optic lobe (OL) is where neurons responsible for sensory modalities have output synapses, and as such has many olfactory and visual projection neurons (Tsubouchi et al., 2017). The mushroom body (MB) is equivalent to the mammalian hippocampus and the principal neurons of MB are named kenyon cell somata. Neuroblasts (NBs) which are the stem cells of the brain, can be found in the Central Brain (CB), the MB and the Ventral Nerve Cord (VNC). The VNC plays an important role, as it serves as the central hub for receiving and transferring signals to the muscle (Court et al., 2020) to control locomotive behaviour of the fly. The VNC does this by utilising the nerve cord (neuron) and the glia and sends signals to the Neuromuscular Junction (NMJ), where nerve and muscle meet at the synapse. The NMJ is also a good system for studying synaptic function and plasticity as well as motor neuron related diseases. mRNA transport and localisation at the protrusions of the neurons at the NMJ is important for the proper development and functioning of the nervous system. mRNA localisation in glia (another important cell type

which is adjacent to neurons) is also of immense importance and it is strongly hypothesised that it contributes to glia-muscle communication (Gala et al., 2022). Little is known about these molecular processes in neurons and they have only been studied to a very limited extent in glia. In section 1.2 I will try to elucidate what is known regarding mRNA localisation in glia, but first, in the following subsection I will introduce mRNA localisation in more detail.

1.1.3 Importance of mRNA localisation

RNA localisation is one of the most important post-transcriptional regulation mechanisms in the nervous system and it has a vital role in many biological functions. Localisation of transcripts has a significant potential in allowing local translation of mRNAs to proteins according to the temporal and spatial needs of the cell. In long cells, with many protrusions, such as neurons, proteins may be required at a long distance away from the cell body (Holt et al., 2019). Regulated on-site translation is utilised when proteins are required for immediate use (Donlin-Asp et al., 2021; Holt & Schuman, 2013), sometimes in response to a stimuli at the synapses, providing synaptic specificity (Rajgor et al., 2021). Moreover, transferring one single mRNA molecule can lead to production of several protein molecules on the site of translation and compared to transferring multiple proteins it is more cost effective (Medioni et al., 2012). Additionally, the localised mRNA has been shown to be transferred in between cells (Holt & Schuman, 2013), an observation we have also made in the Davis lab (Titlow et al., 2023).

1.1.4 Mechanisms underlying RNA localisation

What are the molecular mechanisms that influence mRNA localisation? mRNA localisation is an important layer of post-transcriptional regulation and mechanisms that impact post-transcriptional regulation also influence mRNA transport and localisation, mRNA abundance and protein production. RNA-binding proteins (RBPs) are key regulators of these processes (Payne et al., 2018). RBPs are a class of proteins that bind RNA through RNA binding domains and can manipulate the behaviour or fate of the bound RNA (Hentze et al., 2018). *Fne* a paralog of the RNA binding protein family ELAV in *Drosophila* is a prime example of an RBP, where changes

in *fne* expression in neurons and glia have affected the synaptic transmission and plasticity of the larval NMJ (W.-Y. Lin et al., 2022). Transcription factors (TFs) are another group of proteins, acting on the DNA level, that recognise and bind specific sequences of DNA and subsequently regulate which genes are transcribed and at what rate (Alberts, 2002). The inter-regulation of RBPs and TFs has been studied and there are numerous examples where RBPs and TFs contribute to post-transcriptional regulation. For example, Prospero (pros), a key conserved regulator of stem cells, is a TF that prevents the glial and neuronal progeny of neural stem cells from retaining their stem cell properties thus driving cell differentiation in the *Drosophila* brain. It has been shown that the stability of *pros* mRNA is regulated by the RBPs Syp and Imp (Samuels et al., 2020), both conserved in mammals and implicated in a number of human diseases. A gene having multiple isoforms can also be an indication of the gene's role in post-transcriptional regulation. Different isoforms can have differing 3' untranslated region (3' UTR) lengths. An important finding that leads us to appreciate the importance of isoforms and their difference in their 3' UTR lengths, is that the isoform of *pros* mRNA that is post-transcriptionally regulated by Syp contains an unusually long 3'UTR (Samuels et al., 2020). This variation in 3' UTR lengths across isoforms will be referred to as 3' UTR extension length in this and subsequent chapters, and the 3' UTR lengths of isoforms of transcripts of genes that are different will be mentioned as alternative 3' UTR lengths. The 3' UTR contains regulatory regions that post-transcriptionally influence gene expression. Studies (Hilgers et al., 2011, 2012) have shown that genes expressed in neural tissues often contain long alternative 3' UTR, intermediate mRNA isoforms appear during early stages of neurogenesis and longer isoforms at later stages. Below, I summarise some of the mechanisms highlighted through the given examples and explain other potential mechanisms as well:

3' UTR length and 3' UTR extension length: Previous studies have established that elongation of 3' UTR mediates mRNA localisation, through producing an unusually long alternative 3' UTR. This has been observed in *Drosophila* transcriptomic datasets for at least 30 genes, preferentially enriched in neural tissues (Hilgers et al., 2011). Additionally, it was shown that truncation of the transcript isoform containing a long 3' UTR, resulted in impaired dendritic localisation of the gene's transcript (An et al., 2008). This research was conducted in mice.

5' untranslated region (5'UTR) length and upstream open reading frame (ORF) 5' of the translation start site: Research in *C. elegans* (Theil et al., 2018) has shown that single 5' UTR sites can be strong cis-regulatory elements and important in translational regulation. An upstream ORF within the 5'UTR of an mRNA molecule, as seen in Figure 1.2, can regulate eukaryotic gene expression. It represses translation by restricting ribosome access to the correct start codon and it can impact gene expression by triggering mRNA decay or by regulating translation (Barbosa et al., 2013).

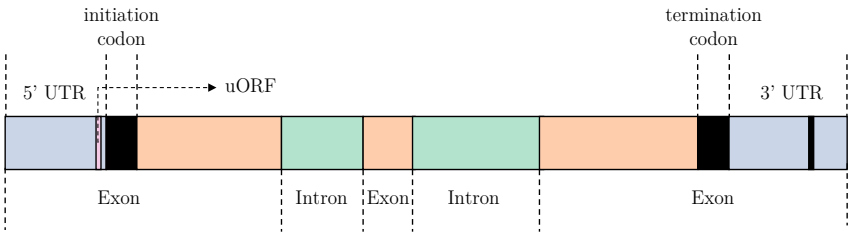


Figure 1.2: A diagram showing the structure of a eukaryotic gene. In this schematic a gene is shown, including the exons (Segment of a eukaryotic gene that consists of a sequence of nucleotides that will be represented in mRNA), introns (noncoding region of a eukaryotic gene) and the positions of 5' UTR and 3' UTR relative to the initiation and termination codons within the mRNA.

Number of splicing isoforms: Alternative splicing is a regulated process that allows a single gene to code for multiple protein variants. Genome-wide transcriptome analysis has revealed a high level of alternative splicing in the mammalian brain (Yeo et al., 2004). It has also been reported that specific alternatively spliced isoforms are expressed in Drosophila neurons (Norris et al., 2014).

RNA sequence: According to research, cis-acting regions known as "zipcodes" or "localisation elements" can specify the cellular "address" of mRNA molecules (Martin & Ephrussi, 2009) and they have a significant part in controlling transcript localisation. These can be contained within single 3' UTR sites (Martin & Ephrussi, 2009), but also within 5' UTR sites as well as within the coding sequence. Research in *C. elegans* has shown that zipcodes, or short sequence segments within an RNA molecule, that are present in the 5' UTR are important in regulating translation (Theil et al., 2018). Localisation elements are recognised by certain RBPs, which, as previously stated, are critical in determining localisation. Although there is no "localisation"

sequence which is clearly responsible for localisation (Martin & Ephrussi, 2009), evidently the sequence of a gene is crucial in establishing the temporal and spatial map of transcript localisation.

RNA structure: The secondary structure of RNA molecules, such as hairpins and stem-loops, can affect post-transcriptional regulation (Wan et al., 2011). The secondary structure can act as a localisation signal and as a recognition motif for RBPs (Hamilton & Davis, 2007). RNA hairpins in the 3' UTR in the RNA molecules of the gene *bicoid* have been shown to regulate mRNA localisation in *Drosophila* oocytes (Svoboda & Cara, 2006).

RNA stability: The stability of RNA molecules is crucial in deciding the molecules' availability for translation and whether the molecule has reached the region within a cell where and when it is required (Thompson et al., 2023). This is particularly profound in long cells like glia or neurons where the mRNA molecule might need to travel considerable distances from the cell body. Research has shown that mRNAs that have been identified as enriched in neuronal projections in the *Drosophila* brain have high stability (Thompson et al., 2023). Therefore regulation of a molecules' stability is an important post-transcriptional regulation mechanism contributor.

RNA editing: RNA editing is a process through which the nucleotide sequence specified in the genomic template is modified to produce a different nucleotide sequence in the transcript. It can lead to the production of altered protein products from a single gene. Editing can also introduce new splice sites (Rueter et al., 1999), and editing that occurs in 5' and 3' UTRs alter the stability, localisation, or translatability of the target mRNA (Prasanth et al., 2005).

1.1.5 Experimental methods for studying RNA localisation

What methods can we use to elucidate the mechanism and function of mRNA localisation and post-transcriptional regulation? Many approaches exist to do so mostly through quantifying gene expression in cells or tissues. These methods include scRNA-seq (Tang et al., 2009) and bulk RNA-seq (Emrich et al., 2007; Lister et al., 2008; Nagalakshmi et al., 2008). Such sequencing methods have the advantage of enabling differential expression analysis to identify enriched or unenriched gene sets. For example, RNA-seq has been used in conjunction with a Boyden chamber to physically isolate oligodendrocytes, a type of glial cell, and calculate the enrichment

of transcripts in the protrusions compared to the somata of the glial cells (Azevedo et al., 2018). A similar technique to separate protrusions from cell somata was used with astrocytes (another type of glial cell with long protrusions) originating from a mammalian cell line (Thomsen et al., 2013). This was followed by high-throughput direct RNA-seq to examine enrichment of mRNAs in protrusions versus somata.

These methods, although very informative about transcript abundance levels in single cells or tissues, do not elucidate the spatial and temporal distribution of transcripts. This is either because single cells need to be physically isolated to be used in scRNA-seq, so information regarding abundance within the cell is lost or bulk RNA-seq is used where an average expression of the population of cells is taken during the time of the measurement (X. Li & Wang, 2021). To retrieve this kind of information imaging techniques are more suitable. smFISH (Femino et al., 1998; Raj et al., 2008), which allows imaging at single-molecule resolution sensitivity is one of those techniques. Because smFISH allows detection of individual mRNA molecules it can provide a map of the location of mRNA within cells. It can also be used to examine intact tissues, such as a *Drosophila* brain to identify mRNA localisation within nervous system structures. smFISH can also be used in unison with immunofluorescence against a target protein of choice to visualise protein and mRNA distributions in parallel (Titlow et al., 2023) both at single molecule resolution. This can be used to examine another layer of post-transcriptional regulation, "discordance". Discordance or discorrelation between the mRNA and protein distribution has been found to be common in the *Drosophila* nervous system (Titlow et al., 2023). This concerns both intercellular and intracellular discordance.

smFISH, an inherently spatial imaging technique, is the foundation of the modern imaging-based spatial transcriptomics (C. G. Williams et al., 2022). Spatial transcriptomics, a method to quantify mRNAs in their spatial context, is a fast evolving field and there are two broad methods of spatial transcriptomics; image-based methods and sequence-based methods (C. G. Williams et al., 2022). Both methods utilise barcodes to spatialy resolve the transcriptomic information (Marx, 2021). For example a sequence-based spatial transcriptomic method, is the commercial

method Visium which is based on the work of Ståhl et al. (2016). In this method, tissue is fixed, stained and then imaged; after is permeabilised and the mRNAs attach to an array of barcoded oligos. The cDNAs produced from mRNAs that are reversed-transcribed, are sequenced. The sequencing is computationally reconstructed using the barcoded oligos to spatially resolve the transcriptomic information (Ståhl et al., 2016). Likewise imaged-based techniques such as seqFISH (Lubeck et al., 2014) use sequential rounds of hybridisation to barcode mRNAs and each round of distinct hybridisation helps to spatially resolve transcriptomic data. Spatial transcriptomics methods have not yet achieved single cell sensitivity however there is optimism that this will be achieved in the near future.

1.2 Glia

The nervous system of all animals is composed of two different cell types, these are neurons and glia. Glia have been thought for years to only play a supporting role to neurons and provide protection to neurons. Glia means glue in Greek, which reflects what has been thought of their function; being the glue of the nervous system, holding things together. However recent understanding has shown that glia are equally as important to neurons for the correct functioning of the nervous system. Glia are classified into seven categories in the *Drosophila* nervous system. These are: astrocyte-like glia, ensheathing glia, midline glia, cortical glia, perineural glia, subperineural glia and wrapping glia. If we examine the CNS and the PNS separately, we can see that wrapping glia are unique to the *Drosophila* PNS, while astrocyte-like glia, ensheathing glia, midline glia, and cortical glia are exclusive to the CNS. Perineural glia and subperineural glia are two forms of glia found in both the PNS and the CNS (Bittern et al., 2021). The *Drosophila* glial categories are shown in Figure 1.3. Figure 1.3 shows a schematic of the *Drosophila* (Panel A) CNS and (Panel B) PNS. In both panels the typical glia present in the part of the nervous system examined are highlighted.

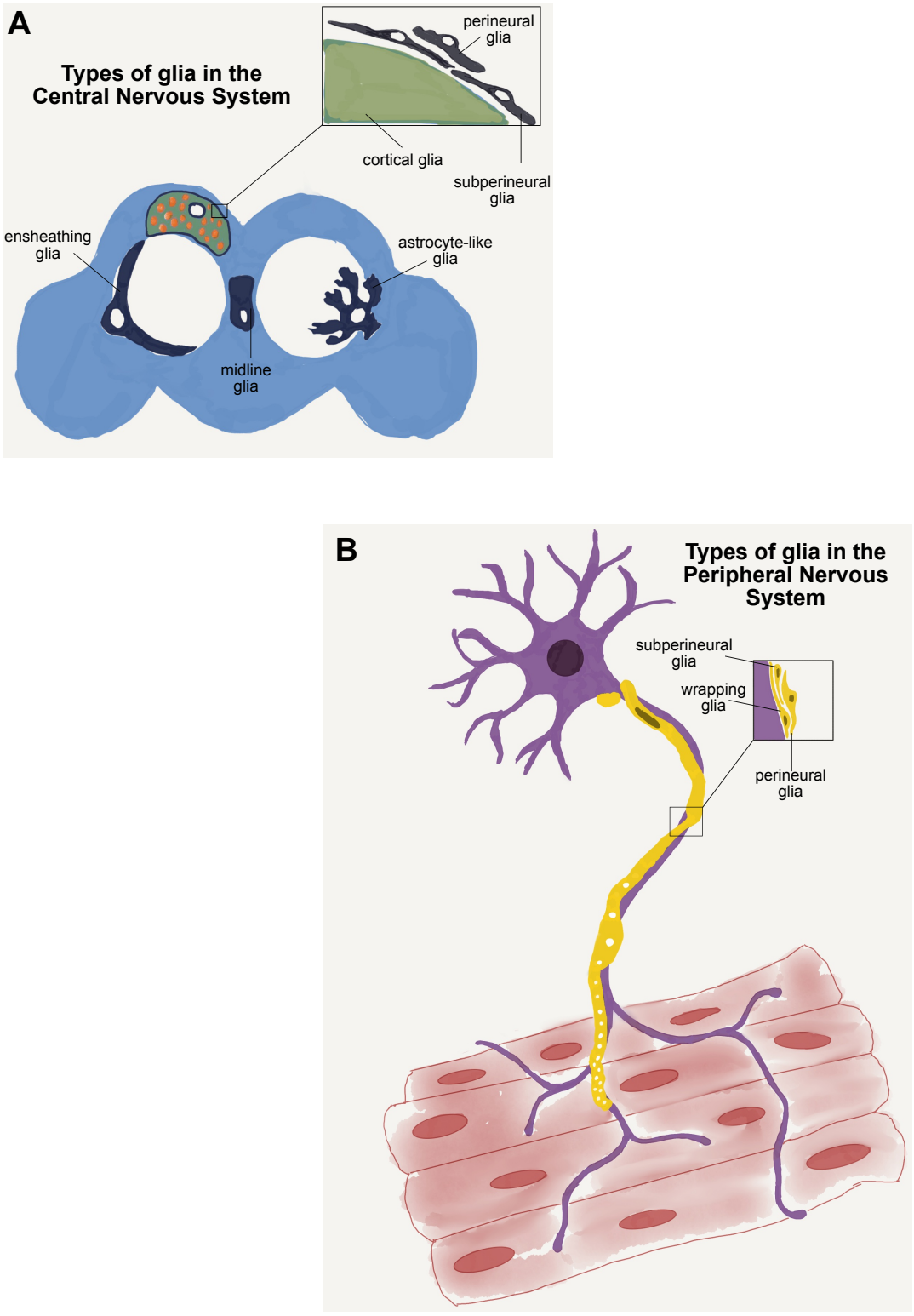


Figure 1.3: Schematic showing the types of glia in the *Drosophila* nervous system. (A) Schematic of the *Drosophila* CNS glia. The inset shows the subtypes of the surface glia. Based on (Ou et al., 2014). (B) Schematic of the NMJ and the PNS glia. The inset shows the subtypes of the peripheral glia. The neuron is shown with purple colour and the glia with yellow.

1.2.1 RNA localisation and local translation in glia

Glia and neurons are both very polarised cells with asymmetrical morphology. Such polarisation and unevenness suggest that for these types of cells tight spatial regulation is required, with mRNAs and proteins likely required at the tips of the cell (Chekulaeva & Landthaler, 2016; Holt et al., 2019). Therefore mRNA localisation, mRNA/protein transport and local translation have a key role as post-transcriptional regulation mechanisms in polarised cells such as glia. Although glia outnumber neurons in the nervous system, very little is known about mRNA localisation in glia compared to the research available for neurons. Research has shown that in a healthy CNS there is evidence of glial mRNA localisation and localised translation (Blanco-Urrejola et al., 2021). It is also shown that neurological and neurodegenerative diseases can be caused by misregulation of mRNA localised translation and mRNA transport. Blanco-Urrejola et al. (2021) have identified at least 50 transcripts whose perturbation of localisation is suggested to be related to the onset of neurodegenerative and neurological diseases, such as Alzheimer's disease and autism spectrum disorder. As a result, further studying glial cell mRNA localisation is important for filling gaps to better understand nervous system related diseases.

1.2.2 Studying RNA localisation in glia in *Mus musculus*

There are many similarities but also differences between mammalian glial cells and vertebrate glial cells. Compared to their mammalian counterpart, *Drosophila* glia are much simpler and they only constitute 10%-15% of the cells in the CNS compared to the 50% and 90% of cells that glial cells constitute in mice and human respectively (Kim et al., 2020). This suggests an increased importance of glial functions as the nervous system expands. There are morphological similarities however for *Drosophila* and mammalian glia, such as that glia provide electrical insulation for neurons and metabolic support, as well as adjusting the neuronal activity by controlling neurotransmitter homeostasis (Zuchero & Barres, 2015). Additionally, it has been reported that there are many conserved molecular signatures between the *Drosophila* and mammalian glia such as mediating clearance of apoptotic cells and participating in axon pruning (Freeman et al., 2003; MacDonald et al., 2006).

However, we observed that there are more available RNA-seq datasets for mammalian glial cells compared to *Drosophila* glial cells in online repositories, likely due to the greater volume of research in mammalian models. To be able to perform sequencing on cells, there is a certain amount of starting material that needs to be provided for sequencing. One reason that could explain the limited number of *Drosophila* RNA-seq datasets compared to *Mus* RNA-seq datasets, is that the *Drosophila* brain is smaller than the *Mus*', and the glia are not as well separated in flies compared to mice, so it is harder to extract enough initial material for the experiment. One example is the use of the Boyden chamber to extract the protrusions of astrocytes in mammalian glial cells (Thomsen & Lade Nielsen, 2011) where it has been used extensively for mice but not flies. One example of a mammalian system with a well-defined anatomy is the rodent olfactory system where astrocytes localise at synapse-rich areas of the olfactory bulb (Ung et al., 2021).

1.3 Big data in Biology

With the advancement of technology and the ability to produce a high-throughput of data, biology has been transformed into a "big data" science. "Big data" is precisely what the phrase describes; data big in volume, big in their production rate and big in complexity (Pal et al., 2020). To manage this seemingly bottomless data reservoir and capitalise on it through scientific discovery, we need to bridge the gap between high-throughput data generation and managing, integrating and visualising the datasets in a constructive way.

1.3.1 Managing data on a local and a global level

Managing data on a local level refers to managing data generated within a research group or a research institution, whereas managing data on a global level involves managing data generated by multiple groups and institutions, accessing data available publicly on online repositories and making data generated globally available for others to use. In this section, I discuss resources available to manage imaging data locally as well as available resources to deposit locally generated data and make them available online. Additionally, I present the status quo regarding big bioinformatics repositories that provide instrumental biological information. I

focus on databases that provide genomic, transcriptomic and proteomic knowledge. I discuss how the different databases are structured and how this organisation diversification is expected as the datasets are inherently heterogeneous.

Managing imaging data with associated metadata

One of the hardest types of data to manage are imaging data. Imaging modalities are growingly multiparametric and voluminous, with accompanying metadata that are often vital to describe and correctly interpret the images (Eliceiri et al., 2012). For example, for raw microscopy data, the acquisition conditions such as the numerical aperture, the focal distance and the wavelength corresponding to each channel are of immense importance in recreating the experimental conditions. Additionally, if during preprocessing the raw image is recontrasted, the contrasting measurements would make each image unique and likewise the metadata associated with the image are also unique.

The Open Microscopy Environment's (OME) Remote Objects (OMERO) (Allan et al., 2012; Burel et al., 2015; S. Li et al., 2016) is an image management repository, which is open source and can be deployed locally for researchers to save and manage their images. The repository can also be used as a foundation for public image data repositories (Allan et al., 2012) and it can accommodate microscopy and medical imaging modalities (Kuhn Cuellar et al., 2022). Multi-channel microscopy images, 2D and/or 3D imaging data with different focal distances resulting in z-stack composite images can also be managed and stored using OMERO (Kuhn Cuellar et al., 2022). A user can create a project and datasets within the project; the user can store images belonging to the same project and dataset together. The images can be retrieved using the image names and/or tags the user has annotated the images with. Metadata can be viewed and accessed, but they have to be related to the way the images were acquired and be associated with the instrument. The underlying processes convert the proprietary data into a format OMERO can read. However, the data that can be viewed alongside the images are limited to metadata that define the images and cannot be any additional genomic or functional information attached to the images post-acquisition. Therefore we observe an inflexibility in the type of data that

can be stored and accessed alongside the imaging data and OMERO provides limited support to -omics data (Liu & Demosthenes, 2022). Image Data Resource (IDR) is a flavour of OMERO that allows access to datasets deposited by individuals and can be searched publicly. It accommodates addition and exploration of limited omics datasets alongside the imaging datasets (E. Williams et al., 2017) but it still has limitations and a user cannot explore a dataset very flexibly. There are predefined requirements for structuring the metadata and the search functionality within the IDR website is restricted. To my knowledge, imaging repositories that are able to support storing additional datasets other than metadata have not been sufficiently developed.

Accessing bioinformatic datasets on public repositories with a global reach

To accelerate research and collaboration between multiple institutions in different parts of the world, transparency is required for the managing and sharing of experimentally produced data. This is especially important for disciplines such as biology where discovery is based on observation and evidence, rather than theoretical formulation. Public repositories enable access to reliable and most of the times curated datasets and provide a centralised location for researchers to share their data and download data from other groups to use in their research. ENSEMBL (Kinsella et al., 2011), Flybase (Gramates et al., 2022), Intermine (R. N. Smith et al., 2012), Mouse Genome Informatics (MGI) (Blake et al., 2021) and UniProt (Apweiler, 2004) are databases that play a pivotal role in allowing free access to genomic, transcriptomic and proteomic datasets.

There are barriers to retrieving information in a useful way from these repositories. Each of the databases has a unique organisation of the data and structure. A user needs to learn how the information is arranged; what is the unique entity that the information is organised around. For example a researcher should know if a database is gene-centric (such as ENSEMBL, Flybase), transcript-centric or protein-centric (such as UniProt). Additionally, for each of the databases a user needs to invest time to understand how to access the data, for example if they need to browse the website to download it, or if they have to access it using a local machine terminal or through application programming interfaces (APIs). Furthermore, each bioinformatics library could be focused on a different organism. If a researcher wants to analyse data for many

organisms this requires access to different libraries for the same type of data. This could introduce discrepancies to the data when they are integrated together. Another challenge is that libraries that provide different data for the same organism might not use the same genome version. As a result individualised ways to access the libraries are required most of the time and a lot of additional data curation to make even data with similar data types interoperable.

Accessing data from databases structured in the form of ontologies

An ontology is a description, a formal specification of the concepts defining a domain (Staab & Studer, 2004). It is like a controlled vocabulary for the language of a branch of a science domain. The common vocabulary the ontologies provide, facilitates domain knowledge sharing. For example in the field of bioinformatics one of the most well-known and used ontologies is the Gene Ontology (GO) (Ashburner et al., 2000) provided by the Gene Ontology Consortium. GO provides a set of concepts that applies to all eukaryotes and describes the functional and biological roles of genes or genes products (Ashburner et al., 2000). GO is (as most ontologies are) structured in a hierarchical way which introduces additional challenges to researchers wanting to access the concepts within the ontology. For GO specifically, the structure is made of nodes and edges, with a GO term being a node and the type of relationship between two GO terms being an edge. The hierarchy of the structure comes in the form of parent-children pairs. A child term is contained within a parent term, and the meaning of a parent term also describes a child term. There is a root term in the tree of the GO where all relations stem from. Currently, there are more than forty three thousand GO terms (Carbon & Mungall, 2023) and therefore going through the terms and understanding the relationship between them to find important terms is a huge task. Additionally, the GO term dataset is centred on GO terms and not genes or genes products, it is therefore GO-term-centric and not perhaps gene-centric. Each GO term is associated with zero or more genes/gene products. Each gene is annotated and the annotations are the GO terms. However a gene might be annotated both with the parent and child term if they are related and this causes a bottleneck of annotations. Finding the right level of annotation, where a gene is described accurately and not too generally, is one of the challenges the GO annotations users

face. Furthermore, GO is reliant on curation and although there are thousands of GO terms, the curation and the annotation of genomes or proteomes, are incomplete and sometimes can be biased by the amount of research on certain topics. Therefore sometimes it is unclear if a gene is not annotated by a term because the term is not descriptive of the role of the gene or if just the annotation of that gene has not been examined yet by the experts.

1.3.2 Making different types of data from heterogeneous sources interoperable

The main purpose of data integration is to allow for data mining and pattern identification from combined multiple datasets, in ways that are not possible when single individual datasets are interrogated (Gligorijević & Pržulj, 2015). To reach this goal, integration strategies have to overcome many obstacles, some of which were discussed in the previous section. Obstacles include obtaining the data, ensuring that the retrieved data are of the same genome version and modifying the data so that they are centralised along the same entity, e.g. a gene, a transcript, a protein or a GO term. Additional challenges include making different data modalities interoperable and integrating heterogeneous and multidimensional datasets.

BioMart (Smedley et al., 2009) and pybiomart (Ruiter, 2016) are two programming packages that enable fast retrieval of data using queries easy to implement. BioMart is an R package and pybiomart a python package. They can integrate together data from libraries such as ENSEMBL, UniProt and REACTOME. These two packages provide a way of collecting and integrating datasets together but they are limited to text or numerical data and cannot handle imaging data. When used for retrieving data in the form of ontologies, they convert the hierarchical structure of the ontologies to a more flat structure and consequently some information is lost. As a result although the integrative power of these packages is high it is also limited to specific forms of data.

Methods used to address the multidimensionality of data are inspired from machine learning techniques such as dimensionality reduction algorithms (Gligorijević & Pržulj, 2015). For example, sequence data can be represented as k-mers; a k-mer being a contiguous subsequence of length “k” formed by extracting “k” consecutive nucleotide bases from the original sequence. The k-mer size can introduce a high dimensionality in the dataset, as the higher the length k the bigger the number

of the possible k-mer combinations. Here methods such as principal component analysis (PCA) (Abdi & Williams, 2010) have been used to reduce the dimensions. I will elaborate on this approach in the next section, which is dedicated to machine learning methods for interrogating datasets.

In the next subsection I discuss a way of making imaging data interoperable with text data. This is using the help of expert annotators to score the images depending on a topic or question of interest and therefore associate text data with the image.

Experts visually scoring images in order to classify them

A way of extracting meaning from images while making them interoperable with text data, is the classification of the images via an expert visually scoring the images. High-quality annotations are time-consuming but also very valuable. The labels can contribute to data integration but also they can be used as labels to train predictive machine learning models. These labels can also be considered ground-truth when the scoring is done by experts and is blind (i.e. no biases are introduced by the scorer).

To my knowledge, there are relatively few methodologies developed to enable experts to score images and label them. One of them is the software developed by Cothren et al. (2018). Cothren et al. (2018) developed a scoring programme to classify muscle mitochondrial morphology of *C. elegans* subjected to different conditions (Hartman et al., 2018). This is freely available, has a practical user interface and can be used to anonymise the images as well. However it only works on Windows. One other limitation of the software is that an annotator can only label an image for one task at a time; if there are many questions to be answered for an image this is not possible to be done using the software. Additionally the label can only be in the form of a number and not a text label perhaps making it more difficult for the scorer to remember to which text label each number corresponds to.

Expert scoring is the optimal option for producing high-quality annotation but it is expensive, hard to organise and time-consuming. The pool of experts is often small, so the time burden on each expert can be high for any large and complicated dataset. Additionally, to increase the

quality of the annotations it is good practice to have multiple experts score each image to eliminate potential bias that could be introduced, exacerbating the time-consuming nature of this approach.

An alternative approach to annotate images when annotation by experts is not possible or viable is online citizen science. In online citizen science projects, online platforms are set up that are accessible for volunteers to score images. The tasks are simplified enough so that volunteers without expert knowledge can answer accurately. The results of the annotation are summarised and analysed using statistical techniques to remove errors introduced by the non-specialist scorers. A prime example of a platform set up for online annotating is the Zooniverse platform (Cox et al., 2015). Zooniverse allows any volunteer to access and contribute to research while answering as many questions as they have time for. Additionally, it provides researchers a platform where they can publish their research for crowd annotation while not having to worry about the mechanics of setting up a public-facing server to contain their research.

1.3.3 Visualising heterogeneous big biological data

The need for developing visualisation tools capable of supporting big data visualisation is stronger than ever. This holds especially true for the field of biology which is experiencing a "big data" era. To meet the requirements of big data, the visualisation tools should be capable of visualising big volumes of data at the same time (be scalable) and also heterogeneous data high in complexity (Cruz et al., 2019). Heterogeneous data often include multimodal data such as microscopy data and expression data. These are data of different types and visualising images, text data and numerical data in a meaningful way is a real challenge in the field. Allowing a coordinated interrogation between the heterogeneous datasets in the form of multiple integrated visualisation models is an optimal solution (Cruz et al., 2019). This approach promotes knowledge discovery and identification of patterns otherwise difficult to spot.

Most biological software applications are often tailored to address a specific biological problem and are not easily generalisable. Cruz et al. (2019) and Dunn et al. (2016) performed an extensive review on interactive and coordinated visualisation approaches for biological multidimensional data. The two reviews list in total more than sixty tools. From those, I highlight iGPSe (Ding et al.,

2014) and Caleydo (*Caleydo*, n.d.; Gratzl et al., 2014) as I believe they are the most adaptable and could potentially accommodate datasets I describe in this thesis. iGPSe integrates gene expression datasets (microarray experiments) with molecular features such as genetic variance and expression levels for selected genomic regions (Ding et al., 2014). It provides a clustering capability and network capability. The tool is focused on patient data related to cancer. However iGPSe has not been developed to be accessible to the public. Caleydo has been used by different research groups in several studies accommodating different types of data. In a similar manner to iGPSe it has been used to visualise mRNA and microRNA expression datasets obtained from patients with cancer (Lex et al., 2012). It features Sankey diagrams and flexible linked axes.

Although the aforementioned visualisation tools provided are useful in data visualisation and provide a powerful means for data analysis they are restricted to accepting text or numerical data only. These tools do not provide a means for viewing imaging data and perform analysis at the same data, combining imaging and text/numerical data. Zegami ("Imaging AI Software Company," n.d.; Taylor & Noble, 2014) and the Multi-Dimensional Viewer (MDV) (COSMIC – COVID-19 Lung Single Cell Mass Cytometry Imaging Consortium STUDY et al., 2022) are two tools that have the advantage of integrating imaging data. The two tools allow the user to search, interact and interrogate datasets associated with imaging data and the image can be at the centre of investigation. Zegami is a commercial software and MDV is an open-source and freely accessible tool. One asset of Zegami[1] is that it can display multiple images at once (at the order of thousand images) in a scalable fast way. A user can zoom in to the images and when they select an image can see all the metadata and data associated with it on a side-panel. Although MDV is not as good at displaying multiple images at once, it accommodates more data analysis views whereas Zegami restricts the users to five different data views.

MDV is an example of the concept of coordinated and multiple views in visualisation. This is a concept that has been developed through the years and allows the users to access multiple representations of different sections of data at the same time (Roberts, 2007). While using MDV the user can select which views to add from a drop down box and additionally select the subset

[1]Zegami is at the moment not available, but it is in the process of becoming open source.

of data the user wants to display using these views. Furthermore MDV can be extended as users can write their own add on packages for visualisations they would like to include in the tool. This capability provides great flexibility on the data analysis front. MDV has been based on another software, Multi Locus View (MLV) by the same authors (Sergeant et al., 2021) which allows users to explore genomics datasets at multiple scales. MLV has been extended to allow visualisation and analysis of multidimensional heterogeneous datasets and images.

1.4 Machine learning in Biology

In recent years there has been a growing interest in the application of machine learning algorithms to solve biological problems. Broadly, machine learning has applications in drug design (Dara et al., 2022), in disease diagnosis (Ahsan et al., 2022), in protein structure prediction (AlQuraishi, 2021; Jumper et al., 2021), in the identification of novel cell types (Z. Li et al., 2022) and in RNA localisation prediction (Garg et al., 2020; Yuan et al., 2023; Zhang et al., 2021) (I will discuss RNA localisation prediction in more detail in Section 1.5). This does not come as a surprise given that there was an explosion in biological data production and in the worldwide availability of biological big data (Pal et al., 2020). Machine learning techniques have the potential to extract insights from the vast amounts of data in novel ways which traditional data analysis tools and experiments cannot compete with. Machine learning has the potential of transforming our understanding of complex biological systems and help us answer unresolved questions (Xu & Jackson, 2019). However, the application of machine learning in biological problems is fraught with its own challenges. Production of good quality data to train supervised machine learning algorithms and missing valuable data which are difficult to generate are some of those. Additionally interpreting the outcomes of a machine learning project (Lim et al., 2023) as well as generalising the algorithms to bigger datasets are some hurdles that need extra attention to be overcome.

Additional challenges associated with using machine learning are the understanding of the importance of features contributing to the model, the interpretability and transparency regarding how the model works and the explainability of the results produced by a machine learning algorithm. Biologists or other physical scientists, are often not predominantly interested in

improving computational efficiency of the model or developing a state-of-the-art model. They are primarily interested in understanding the results of the model and how it can advance their understanding of the field they are passionate about. Explainability in machine learning can be reached when explanations that are understandable by a human can be provided to understand how the model works (Markus et al., 2021). Interpretability is when a human looks at the model and can directly understand how the model works (Markus et al., 2021).

This lack of transparency on how a machine learning model works or treating it as a black-box model can make researchers wary of the predictions made by a model. Biologists usually have a deep understanding of the underlying complexity of biological systems and when a model predicts something that contradicts their intuition without a clear reasoning it is difficult to accept. Lack of interpretability in predictive models can undermine trust in these models (Petch et al., 2022). Therefore until this level of trust is reached, it is important not only to further develop the explainability and interpretability of machine learning algorithms but to also verify the predictions using experimental or other computational methods, such as bioinformatics methods.

In the next subsections I discuss the required components of a machine learning project, I outline the typical steps a scientist has to go through to complete such a project and I give a literature review of some of the key machine learning algorithms that will be used in the rest of this thesis. Additionally I discuss some of the challenges arising at each step of a machine learning project.

1.4.1 A Machine learning project from start to finish

When machine learning projects involve complex and heterogeneous datasets, it is of immense importance to follow a set of best practises in order to ensure the success of the project. Following a set of best practises facilitates the computational efficiency of the project (Adadi, 2021), contributes to quality control (data validation and testing of the model) (Lones, 2023) and mitigates any risks that could lead to wrong predictions such as model overfitting or data leakage which could undermine generalisability (Lones, 2023).

In summary, the key steps to follow when carrying out a machine learning project (based on the steps presented by Geron (Géron, 2017) in his book) are:

1. Frame the problem and look at the big picture
2. Get the data
3. Explore and visualise the data to gain insights
4. Prepare the data to better expose the underlying data patterns to Machine Learning algorithms
5. Explore and train different models and shortlist the best ones
6. Fine-tune the selected models and combine them into one solution
7. Present the solution

Frame the problem and look at the big picture

A clear and well-posed problem is a cornerstone of a successful machine learning project. Identifying what needs to be predicted helps firstly in choosing the right model and secondly in defining how the success of the model will be measured. Additionally, it also helps ensure that the problem can be solved using machine learning.

Get the data

Data is the building block which machine learning models use to be constructed. The quality of a machine learning model's predictions is limited by the quality of the data (Budach et al., 2022). As discussed, the volume and the speed of the big data biological generation is ever-increasing, but that does not guarantee high data quality. Low quality data, missing data and polluted data can drive a model to predict poorly and will require a lot of time and manual curation to clean the data. Planning the way data should be generated, defining the requirements regarding quality and the amount of data before hand is vital for the efficiency of the whole project.

As discussed in Section 1.3.2, there is a lot of heterogeneity in the biological data produced and furthermore the data can be multimodal, introducing another level of complexity. Therefore, an important and time-consuming part of this step is to make the data generated interoperable.

A way to make imaging and text data interoperable is the annotation of imaging data by experts or crowd workers. In both cases methods should be devised for the annotations to be as close to the ground truth as possible so that the annotations' errors are as small as possible and are not propagated to the model. Examples of the inaccuracy introduced in the model because the annotations were noisy can be seen in the work of Budach et al. (2022). Production data bias is another challenge sometimes overlooked which is especially true in problems trying to use a predictive model to diagnose a disease. For some diseases, there are more data for unhealthy people as usually it is then that a patient gets examined using highly specialised equipment to take measurements. A healthy person will not usually have such specialised exams, therefore we have a biased and imbalanced dataset. On the contrary there are cases that there is lack of sufficient unhealthy data compared to healthy controls such as when using functional magnetic resonance imaging (fMRI) for identification of functional brain networks (Ismaila et al., 2022).

Moreover, the scientist working on the project, should convert the data to a format that is easily manipulated (Géron, 2017) without altering the data itself. This requires a specific approach for each dataset and datatype which can be a labor-intensive work.

Explore and visualise the data to gain insights

The data can be separated into categories: features and labels. Each example or instance of data can be described by a label and multiple different features. Additionally, the data can be separated into labelled examples and unlabelled examples. Labelled examples are usually used to train a model and then the trained model is used to predict the labels for the unlabelled examples.

This holds true if the goal of a machine learning model is to predict a certain outcome. In case the goal of a machine learning model is to identify patterns within the data, labelled examples are not necessary; the only required component of this type of dataset are the features.

In more detail, in this step, the target variable needs to be identified within the data. This can be a continuous variable or a categorical variable. The target variable (which as a term is interchangeable with the term labels), should describe what we want to predict and solve the original problem. Therefore, for the instances of data where we have knowledge of the labels,

we can treat the labels as ground truth. For the instances of data that are not labelled we use the model to predict the labels and depending on the accuracy and quality of the model, the labels can be close to the ground truth.

After identifying the labels, we can focus on the features. Features are attributes of the instances of data and characterise them. Understanding and exploring the features using exploratory data analysis (EDA) is an essential step in every machine learning project. The purpose of EDA is to generate insights and identify potential issues with the data. The issues could include missing data values for certain features or classes of data (Ghosh et al., 2018) or outliers (Konopka et al., 2018; Richardson, 2022). Additionally, training using insufficient labelled data might lead to models that are not able to generalise well and predict accurately on never-before-seen data. Furthermore, the labelled dataset could be biased towards one class of data (Richardson, 2022) (representation bias in data capture). Missing values can also result in biased analysis of data and predictive algorithms (Ghosh et al., 2018).

Common tools used during the EDA process to identify missing values or outliers include visualisation of the data using heatmaps, histograms, distributions and boxplots. Additionally, summary statistics such as mean or median to check unexpected values and also the interquartile range (John, 2021) can also be used. Furthermore, clustering values together using clustering techniques is also a common technique, as by grouping data together one can identify data that fall out of these clusters and therefore could be different or outliers.

The size of the labelled samples relative to the size of the features is also an important parameter to consider when creating a machine learning dataset. A good practise is that if the number of labelled samples is N the number of the features should be N-1 (Hua et al., 2005). In cases where the features are highly correlated the number of features should be approximately the square root of N (Hua et al., 2005). The relative size of the dataset also depends on the model that will be used. In case we use Neural Networks (NNs) as the algorithm of choice the labelled sample should be at least 10 times bigger than the number of features (Alwosheel et al., 2018).

The issue of missing values in the data can be dealt with in two ways. Firstly by dropping instances of data with many missing values or features that for most of instances there are missing values. Secondly, the missing values can be imputed. Imputation is a technique for inferring the true value of an instance of data (Richardson, 2022). However both techniques can introduce editing bias (Richardson, 2022) as described by Richardson (2022). The preferable way of action can be influenced by the distribution of the missing values. For example, when in a dataset most missing values fall under one category it is best to remove all the data related to the specific category (Richardson, 2022).

The problem of representation bias, that is when one class of data is overrepresented compared to the other classes, can cause issues with the prediction of a machine learning algorithm. These type of data are unbalanced data. There are ways to manage this, including data sampling or weighting the under-represented class. Random oversampling (from the minority class) and under sampling (from the majority class) aim to balance the skewed data distribution (Kumar et al., 2021). Weighting contributes to the under-represented class in having more significance in the training of a model compared to the over-represented class.

Prepare the data to better expose the underlying data patterns to Machine Learning algorithms

Armed with the dataset-related knowledge and insights gained in the previous step, in this step a scientist tries to curate the dataset while keeping the insights in mind. Filtering out the best possible subset of features within the dataset is key in training a machine learning model to reach a successful outcome. When the dataset is highly dimensional, a way to reduce the number of dimensions is by using feature selection techniques or feature extraction techniques (Zebari et al., 2020).

Selecting the most informative features, is extremely important as data are generated at an accelerating pace. Removing redundant or noisy features from datasets to ameliorate the dataset training a machine learning model can be achieved with finding correlated or collinear data and removing part of them or using feature selection techniques based on impurity or entropy

reduction. Correlation metrics such as Pearson and Spearman can be used to calculate correlation. Which one should be used, depends on the distribution of the features and whether they are expected to have a linear relation between them or not (Schober et al., 2018). Pearson is used when the distribution is normal and the features to be linearly depended on each other. When these conditions do not hold true, Spearman is expected to be used (Schober et al., 2018). The two methods can be applied both in finding the correlation between the feature and the labels which is also called the "relevance" of the feature or to find correlation between two features to find the redundancy between pairs of features (Tsanas et al., 2010).

Feature selection techniques can be based on machine learning algorithms as well. Packages to find which features contribute the most or the least to a machine learning model can provide a ranking of the features, based on the accuracy of the models when a feature is included or excluded in the prediction. These include python package such as rfpimp (Parr, n.d.) or inbuilt functions to machine learning models such as impurity or permutation importance implemented in scikit-learn (Pedregosa et al., 2011).

Feature extraction techniques, in comparison to feature selection techniques, reduce the number of features by transforming the features into a new set that somehow captures the variability of the full dataset (Zebari et al., 2020). Feature extraction techniques aim to preserve the variance and variability of the dataset, therefore conserve the original information captured by the dataset. However the transformation of the data might introduce loss of the ability of identification of single features and understanding the contribution of each feature. PCA (Jolliffe & Cadima, 2016; Lever et al., 2017) is one of the techniques that can be used for feature extraction.

Feature selection techniques, can conditionally preserve the same summary statistics as the original feature set. That depends on certain factors such as feature relevance, feature redundancy and the statistical metric we are examining. For example, if the initial feature set consisted of noisy features that were deemed irrelevant by the feature selection algorithm and were dropped, that could alter the summary statistics such as variance. On the other hand, a feature selection technique called "variance thresholding" would drop features with low or nearly zero variance -

so that the feature selection technique would not alter variance substantially but still the number of features will be reduced (Sadeghyan, 2018). The ultimate goal of feature selection techniques is to find the subset of features to improve the accuracy and performance of the classifier.

Explore and train different models and shortlist the best ones

There are numerous machine learning models in the field introducing the dilemma of which one is the most suitable to use for answering the question in mind and for the available data. Firstly, based on the data at hand, and whether the dataset contains labels or not, a user should select between supervised machine learning or unsupervised machine learning algorithms. Supervised machine learning requires training data that contain some element of ground truth (Greener et al., 2022a) whereas unsupervised machine learning does not. Unsupervised learning is able to find associations between the data without guidance or the need of predetermined labels for part of the dataset.

Once the type of machine learning required is identified, different machine learning models should be trained and the accuracy of their predictions should be assessed. The way to accurately evaluate the performance of a machine learning model is by providing to the model new, never-before-seen data and assessing the accuracy of its predictions. This is to see if a model generalises to new data well.

Typical machine learning models include Decision Trees (DT) (Rokach & Maimon, 2005), Random Forest (RF) (Breiman, 2001), Naive Bayes (NB) (Webb, 2016) and NNs (Gurney, 1997). These are all supervised machine learning models that have been used in unique biological applications and have also been used in my thesis project, their use will be described in more detail in Chapter 4. NNs can be part of a special category of machine learning called Deep Learning (Greener et al., 2022a) while the other models given above are part of more traditional machine learning methods (Greener et al., 2022a). A NN is characterised as a Deep Learning algorithm when it consists of at least three hidden layers and this is also why Deep Learning is called “deep”, because of the depth of the NNs used in this approach. NNs consist of nodes organised into various layers, aiming to simulate the connectivity of neurons within the brain.

NNs are suitable for large and highly structured datasets (clear relations between the features) (Greener et al., 2022b) and typical biological applications of NNs include drug discovery and protein structure prediction. One of the disadvantages of NNs is that they need large training sets to be trained thus do not predict accurately when there are smaller datasets (less than a thousand data points). The more traditional methods such as RF have been used in biology to discover disease associated genes (Greener et al., 2022a) or as gene candidate identification approaches (Y. Y. Lee et al., 2023). RF is suitable for high-dimensional data, both numerical and categorical. RF is a good choice for smaller datasets (less than a thousand data points) as it can be trained with fewer data points and still perform well. The multiple layer architecture of certain types of NNs contributes to their algorithmic complexity; a multi-layer NN is much more complex than DT or RF as NNs consist of many algorithmic layers. More complex algorithms have the ability to capture intricate patterns within large datasets. As the complexity of the data increases more sophisticated and scalable algorithms are required to extract meaningful insights from multidimensional and big datasets. For example, drug discovery through machine learning algorithms is only possible because multilayered advanced NNs are able to learn the multifaceted relationships of the interactions between molecules and their targets. Simpler machine learning algorithms are not capable of tackling complicated problems such as drug discovery.

Unsupervised machine learning model methods such as clustering have been used to extract insights or patterns from complex datasets. Examples include k-means clustering (Ikotun et al., 2023) which has been used to find natural groupings within the dataset based on similarity between the points in a dataset. K-means has been applied in clustering genes based on their expression profiles in order to uncover genes with ideally similar phenotypes. I have used k-means clustering in my work too (described in Chapter 4), to evaluate whether the clustering algorithm can separate genes that their mRNAs localise within cells in a similar fashion.

Fine-tune the selected models and present the solution

Once a model or more have been identified as the most suitable choice, fine-tuning of the models is the final step to ensure the performance of the models is the best possible. This can be

done by changing the implementation parameters until its performance is optimised. This is also called "hyperparameter tuning" and can be automated. GridSearchCV and RandomSearchCV are some of the most common strategies for automating the tuning of the hyperparameters (Bergstra & Bengio, 2012). Hyperparameters include among others, the number of trees or the tree depth for RF and number of layers, the batch size and the epochs for NNs. In GridSearchCV we start by defining the space of the hyperparameters i.e. all the different values the hyperparameters can take and that we want to test, the "Grid". After that the GridSearchCV will construct different versions of the model with all possible combinations of hyperparameters and select the best performing model. However this method can be very computationally costly. RandomSearchCV does not go through all the combinations but only a limited number of them, in a randomised fashion. On the other hand for k-means, an optimised hyperparameter tuning can take the form of selecting the right number of clusters for k-means, as this approach of clustering requires the a priori selection of the number of clusters. There are methods to help a user find the optimal value such as the within-cluster sum of square (WCSS) metric (Kiruba Nesamalar et al., 2021). But it can also be approached through a trial and error method.

In the previous section, I mentioned that one should test their model with never-before-seen data. This is alternatively referred to as the testing set. The testing set of the data should be defined in the beginning of the project and separated from the training set part of the data. This is to make sure that the model has not interacted with this part of the data. Additionally, a researcher should also set aside the validation set of the model and this is to help fine-tune the model and set the hyper parameters to their optimal value. Typical training-testing-validation data splits are 70-80% of the data to be used for training, with 10-20% for testing and 10-20% of the data for validation. When using RF, a researcher can use the out-of-bag score to assess the prediction performance of the model. To calculate the out-of-bag score, the training dataset is used to train the model and then part of the same training dataset to calculate the score. For each trained classifier different parts of the training dataset are used to ensure model blindness and therefore it is not required to split the dataset in advance into training and validation parts (Janitza & Hornung, 2018). This is especially useful when only small datasets are available to train an algorithm. Additionally, this

saves both computational time and computational memory as the same RF classifier which has already been trained is used for the error estimation (Janitza & Hornung, 2018).

Another method that can be used to evaluate the performance of a model and therefore help fine-tune it is cross-validation. In contrary to the out-of-bag score cross-validation is a more general approach that can be used to assess many models and not only RF or other ensemble methods. Furthermore, cross-validation does require splitting the dataset into training and validation/testing sets (Janitza & Hornung, 2018). In general, cross-validation involves splitting the dataset into different parts or "folds". The different folds constitute the training and the validation/testing sets. There are different cross-validation methods and their defining characteristic is mainly the number of folds that the dataset is split in. One of these methods is k-fold cross validation (Kohavi, 1995). Here, the dataset is split into k folds each of the same size. k-1 folds are used as the training set and the other fold as the validation/testing set. This is repeated k times so that k models are trained with different data. The performance of all models is averaged and this is the finalised performance metric. Another cross-validation approach is the leave-one-out cross-validation (Kohavi, 1995). This is similar to the k-fold cross validation but instead of grouping together the data into folds each data point is considered one fold. So the number of models that are trained equals the number of data points. The leave-one-out method is more computationally expensive than the k-fold method but more robust as each model is trained using all the data bar one. Another approach is the holdout cross-validation (Kohavi, 1995) whereas a dataset is split into training set, validation set and testing set. The model is only trained once and the validation set is used for hyper parameter tuning and the testing set for performance evaluation.

After following the well-defined steps explained, the machine learning model(s) of choice are trained and ready to be used to make predictions!

1.5 Predicting RNA localisation

Experimentally studying mRNA localisation and directly observing single molecules of mRNA using state-of-the-art imaging techniques, has enabled scientists to study mRNA localisation in

depth and make novel and often serendipitous discoveries. However, imaging techniques like smFISH (Femino et al., 1998; Raj et al., 2008) require very extensive design and preparation which is both time consuming and expensive. Additionally, research groups that would like to screen large numbers of genes for example, often take years to finish designing and conducting the experiments. So experimental techniques as such, often have limited throughput.

The use of machine learning and artificial intelligence has gained significant momentum in the last few years. The application of predictive algorithms in biology has been accelerated by the availability of huge amounts of experimental data which the algorithms can use to be trained. Unlike experimental methods machine learning methods can produce high throughput as they require a lot of data to be trained but can be very easily scalable, to produce more and more predictions. In a similar fashion to experimental techniques, if a scientist is using unsupervised prediction methods (e.g. clustering), they can identify unusual patterns between the data unexpectedly. However, machine learning prediction is reliant on relevant and very high quality data which are hard and time expensive to obtain. Nevertheless, many methods have been developed to predict mRNA localisation in different cell types using distinct input data, some of which I describe below.

RNAlight: To infer RNA subcellular localisation in human cells, focused on nuclear and cytoplasmic localisation, Yuan et al. (2023), built RNAlight. RNA localisation labels in human HEK293T cells were used to train the model to identify localisation. It is focused on predicting mRNA and lncRNA subcellular localisation but can be generalised to other RNA types such as snoRNA and ciRNAs. There is a feature selection algorithm incorporated into RNAlight, the Tree SHAP algorithm (Lundberg et al., 2020), which identifies features that contribute most to the prediction and hence can highlight sequence features of interest that are important to nuclear and cytoplasmic localisation.

mRNALoc, **iLoc-mRNA**: These two methods, developed by different groups, use the same data to create labels and train a model to predict mRNA localisation. Both mRNALoc (Garg et al., 2020) and iLoc-mRNA (Zhang et al., 2021) use data from RNALocate (Cui et al., 2022)

which is a database with manually curated localisation information assembled together from literature. iLoc-mRNA is designed specifically to predict human mRNA subcellular localisation prediction, whereas mRNALoc can be generalised to other eukaryotic cells too. Both however, are trained to predict localisation in five specific locations within the cell. These are: cytoplasm, endoplasmic reticulum, extracellular region, mitochondria and the nucleus. imRNALoc uses support vector machines (SVMs) which is a type of machine learning classification algorithm. SVMs traditionally can only be used in binary classification, that is distinguishing between two classes. In this case, where five classes needed to be predicted, independent SVMs were trained to predict mRNA presence or absence in each of the five cell types. As a result an RNA can only be predicted to be in one location, which might not be accurate as it might be present in more than one cell types. Additionally these methods do not work as well for *Drosophila* and they do not have predictive power for polarised cell types that have protrusions such as glia or neurons. For polarised cells, localisation is potentially biologically more relevant.

mLoc-mRNA: Likewise, mLoc-mRNA (Meher et al., 2021) sources data from RNALocate (Cui et al., 2022) and is a direct competitor of both mRNALoc and iLoc-mRNA. It uses a RF machine learning algorithm (Breiman, 2001) to make predictions and can make predictions for nine potential localisation positions of the RNA transcripts, for eukaryotic cells. Evaluation shows (Meher et al., 2021) that mLoc-mRNA, outperforms both mRNALoc and iLoc-mRNA.

All of these methods have a common characteristic; they use gene sequence as one of the features to train the algorithms. Apart from training using sequence k-mers those methods do not combine any other datasets that could potentially influence or explain localisation, limiting perhaps the predictive power of the methods.

There are several advantages of using predictive algorithms to make scientific discoveries. When the two methods, the computational predictive methods and experimental methods are used in unison they can complement each other. For example, an experimentalist can be guided by the hypothesis generated and design more focused experiments on candidate genes suggested by machine learning algorithms. The idea of using the insights and knowledge generated by

machine learning algorithms to inspire and guide scientists to design experiments and increase their understanding of the problem has been discussed in this review (Friederich et al., 2021), with applications in many different areas of science. I aspire to do the same in this thesis; draw insights regarding mRNA localisation from relevant to localisation data used to train machine learning algorithms. This is one of the aims of this thesis which I discuss in Section 1.7.

1.6 The structure of the thesis

In this Chapter, I gave a brief overview of mRNA localisation. I discussed the mechanisms underlying mRNA localisation and how to study it experimentally. I focused on transcript localisation in the protrusions of polarised cells such as glial cells in *Drosophila* and *Mus*. I introduced a major challenge arising in the dawn of the 21st century, the fact that the high-throughput technologies produce an enormous amount of data which are difficult to manage and derive insights from. This applies to data generated to study mRNA localisation too. I provided a brief overview of the repositories and cloud-solutions developed to manage the masses of data, both the ones that can be deployed locally and the ones that are accessible globally. I elaborated on selected publicly and globally accessible bioinformatics repositories and present ones that give access to useful biological information, including mRNA localisation related datasets. I discussed the structure of these repositories and the challenges of integrating data of different structures and making the datasets interoperable. Many of the datasets come in the form of imaging data which introduces additional challenges, as imaging data and other types of data are hard to interrogate and visualise simultaneously. Hence, I introduced visualising software tools with the ability to incorporate imaging data and other associated different types of data allowing the user to find correlations in the data and draw conclusions. Next, I introduced machine learning as a means of interrogating the streams of gigantic biological data sets. I presented the steps that a researcher should follow to achieve a successful outcome when trying to use machine learning predictive algorithms. Essential elements of such a pipeline are data collection, data preprocessing, label creation or target variable assignment, feature engineering, model selection and training and model evaluation. I then focused on clustering algorithms and the RFs algorithm as these

methods will be used later in the thesis. Finally, I provided a literature review on established methods to predict subcellular mRNA localisation.

In this thesis, I focus on shedding light on potential distinctive features of mRNAs that localise in glial cell protrusions and additionally find candidate transcripts that are anticipated to be present in glial cell protrusions. Chapter 2 describes the results of the exploration of the transcriptomic and proteomic profiles of 200 genes in the *Drosophila* nervous system. Initially, we explore the transcriptomic and proteomic profiles of the genes using smFISH probes to visualise the mRNA and yellow fluorescent protein (YFP) to visualise their encoded protein. To manage the microscopy images we produce and translate the imaging information into text information we established a pipeline which I present in Chapter 2. This valuable workflow's first step, is to automate the managing and conversion of the raw data acquired in the form of microscopy images into figures in a systematic way. Next in the workflow, is annotating the figures by a group of experts using bespoke software which I co-created, named "Annotate.OMERO.Fig". The software has a friendly user interface and is customisable according to the needs of the annotation step. The final part of the annotation is resolving any conflicts arising from disagreements between the scorers and converting the finalised annotations into machine learning labels. The last segment of the workflow is the visualisation of the figures together with the scoring the expert annotators assembled.

Having explored mRNA localisation in the *Drosophila* nervous system in Chapter 2 and produced definite mRNA localisation labels for 200 genes for twelve tissues of interest, in Chapter 3, I select one of the tissues of interest to focus on. This is the peripheral glial tissue in *Drosophila*. In this chapter, I consolidate large datasets reported in literature to be relevant to mRNA localisation in glia and try to make the heterogeneous datasets interoperable. I do this for both *Drosophila* and *Mus*, as *Mus* in the aspect of nervous system glial mRNA localisation is more well studied than *Drosophila* and there is a greater availability of datasets. Moreover, I discuss how each of the datasets can be converted and engineered into a fit-for-machine-learning dataset. I also discuss potential machine-learning complications arising from the characteristics of the dataset.

In Chapter 4, I use machine-learning approaches to investigate mRNA localisation in protrusions of glial cells. I start by investigating the correlation within groups of similar features using a python package called rfimp which finds the dependency of a feature on the rest of the features using machine-learning techniques. Next, I investigate if the high dimensionality of both the *Drosophila* and *Mus* datasets can be reduced using PCA. I then attempt to understand which features within the dataset contribute the most to the variance of the complete dataset, for both *Drosophila* and *Mus*. I do that using machine-learning feature-selection techniques. I then present the results from adapting a machine learning pipeline (used to predict candidate genes related to sleep) found in literature (Y. Y. Lee et al., 2023) to my data. The final goal of the adapted pipeline is to identify candidate genes and using this I predicted 66 candidate genes that it seems are highly likely to have a transcript localised in glial cell protrusions. Finally, I perform REACTOME and GO terms enrichment analysis to study the enrichment of the candidate genes in functional pathways, biological processes and cellular components to identify if there is enrichment in terms related to mRNA localisation in glia.

Finally, in Chapter 5, I discuss and summarise the results of this work. I provide the conclusions I have drawn and discuss potential extensions to the project and future work I am planning to do after the submission of this thesis.

1.7 Aims of the thesis

The overarching aim of this thesis is to understand mRNA localisation in glial cell protrusions using computational and machine learning approaches. Specific aims include:

1. Creating a novel paradigm for exploration of the mRNA and protein expression in the *Drosophila* larval nervous system visualised in the form of microscopy images along with rich bioinformatics metadata and annotations describing the distribution and patterns of mRNAs and proteins. My aim is to make it possible to provide the vast data domain in the form of a gene-centric browsable platform where the user can interrogate data associated with the gene of interest and examine the co-visualised mRNAs and proteins.

2. Exploring the online bioinformatic and biological public repositories to identify potential mRNA localisation datasets and extracting information in a systematic and reproducible way. Investigating the structure and hierarchy of information within each dataset and converting them into features that are machine-learning-ready.

3. Identifying candidate genes whose transcripts are highly likely to be localised in protrusions of glial cells and highlight which factors, in the form of functional, structural or molecular features, contribute the most to the selection of the candidate genes. After predicting the candidate genes, examining their enrichment and assessing their localisation potential using bioinformatics methods.

2

Intermediate-throughput imaging screen of mRNA and protein expression in the *Drosophila melanogaster* larval nervous system

In this Chapter, I set out to tackle a general data problem using the exemplar of a systematically acquired gene expression dataset in the fruit fly *Drosophila melanogaster (Drosophila)*. An intermediate-throughput microscopy imaging screen has been generated by at least ten Davis lab members over seven years, identifying the mRNA and protein expression distributions in the *Drosophila* intact brain and PNS. The goal of this chapter is to describe how we annotated and classified systematically the data and made the annotations interoperable with rich bioinformatics data, with the final goal being to make it ready to be used by machine learning algorithms.

I present a complete, adaptable and generalisable workflow and computational pipeline for creating labelled microscopy data suitable for training a supervised machine learning algorithm. I begin by analysing the presence and nature of biases in the selection of the genes of interest from a pool of YFP trap lines across the entire *Drosophila* nervous system. I show that the selection of the 200 genes used is not biased towards any of the physical properties that we believe are related to mRNA localisation. I present the imaging data management procedure set-up using OMERO and the consistent producing of figures for the overview and for the six compartments within the nervous system for the 200 genes. I then present an adjustable in-house user-friendly program

for annotations and I summarise the results produced by our three expert annotators labelling the figures. I analyse the quality of annotations using the inter-annotator agreement and Cohen's kappa coefficient. Once I produce a table containing the annotations for all ~1,300 figures I present them online in Zegami, a data visualisation platform. Zegami is a cloud-based solution where a user can upload huge datasets, view and explore large numbers of images quickly and in parallel.

Finally, I demonstrate how Zegami, an interactive visualisation and exploration tool can be used to aid biological discovery and uncover patterns in the image and text/numerical data. I conclude that post-transcriptional regulation is common amongst the 200 genes and that 18 genes show mRNA localisation in the tips of glia with the corresponding protein also localised at the tips. I also produce sets of labels for the 200 genes. The sets are of high quality, credible and trustworthy to use to train an algorithm. On the negative side, I find that the production of labels is lengthy, laborious and to ensure the high quality of labels a significant investment of resources is required.

Part of the work presented in this chapter exists as an article which has just been published by the Journal of Cell Biology. I am a co-first author of this article and the full reference (Titlow et al., 2023) is given below. I am also a primary author of a paper, accompanying the preprint above, which is published in Zenodo (Kiourlappou et al., 2022). It explains how to recreate a Zegami instance, provides all the data underlying a Zegami instance and a self-contained zip folder containing the code for recreating the graphical user interface (GUI) annotation software "Annotate.OMERO.Fig" presented in the paper and later in this chapter.

Full references for the two articles:

1. Joshua S. Titlow, Maria Kiourlappou, Ana Palanca, Jeffrey Y. Lee, Dalia S. Gala, Darragh Ennis, Joyce J. S. Yu, et al. 2023. **"Systematic Analysis of YFP Traps reveals common mRNA/protein discordance in neural tissues."** J Cell Biol 5 June 2023; 222 (6): e202205129. `https://doi.org/10.1083/jcb.202205129`.

2. Maria Kiourlappou, Stephen Taylor, and Ilan Davis. 2022. **"Zegami User Manual for Data Exploration: 'Systematic Analysis of YFP Gene Traps Reveals Common**

Discordance Between mRNA and Protein Across the Nervous System'." Zenodo. https://doi.org/10.5281/ZENODO.7308444.

2.1 Introduction

In recent years, the sheer size of biological data produced has become increasingly challenging to manage both manually and offline. It has become very challenging to analyse biological datasets in the context of the existing body of knowledge or in the context of similar studies. Even conventional computational methods are unsatisfactory to manage, organise and annotate the vast volumes of biological data produced daily, especially bioimage data (Eliceiri et al., 2012). Image-based genome-wide 'high-content' screens are a prime example of biological datasets that require more advanced and automated ways of browsing. The OMERO (Allan et al., 2012; Burel et al., 2015; S. Li et al., 2016) and the Bio-Image Semantic Query User Environment (BISQUE) (Kvilekval et al., 2010) projects are two examples of biological image repositories where users can store their images. The images are searchable using the image names and/or tags the user added online using the software. Although these repositories usually allow for metadata to be viewed and accessed, they usually have to be in a specific format and contain specific and complete information; often associated with the way images were acquired. Even when the images depict gene-related phenomena, the metadata associated with the images do not typically involve genomic and functional information from a rich variety of sources. Therefore, there is a general inflexibility and size limit in the way images and their metadata are stored and annotated. There is also an inflexibility in the way images are presented and annotated in publications; most of the systems developed for automatic annotation are still in early stages of development and quality control methods are at an even earlier stage. Therefore, there remains a major technological gap and still a need for a flexible manual annotation method which allows the user to annotate images easily.

Here, I present a complete workflow for firstly creating a user specified vocabulary for annotations and annotating images that are stored in OMERO, secondly assessing the quality of annotations produced by a number of experts, discarding annotations of low quality and presenting the annotations together with the images. Moreover, using the workflow I present, the user has

the capability of adding metadata in the form of ontologies where the data can span the whole genome. As expected, in image repositories, the information presented is image-centric rather than entity-centric. A user is therefore restricted in searching for specific information via searching for the images first. Here I use Zegami, a cloud-based visualisation tool which allows the user to have information not directly associated with an image. As a result, I provide a tool that allows the presentation of multimodal data in a text-information-centric rather than image-information-centric way and that allows interaction with the dataset as a whole, from a genome-point of view.

2.1.1 Aims and objectives

1. To create a generalisable, scalable, flexible, user-adjustable and easily deployable workflow for annotating images, analysing and exploring them along with metadata and ontology data, allowing the user to examine the data in its entirety (whole genome) or to focus on subsets of the data (particular genes or groups of genes).
2. To study the mRNA and protein expression and intracellular localisation of 200 genes across the *Drosophila* larval nervous system.
3. To generate high quality annotation datasets related to mRNA and protein expression across the *Drosophila* larval nervous system ready to be used as training labels for training machine learning algorithms.

2.2 Materials and Methods

One of the aims of this project is the systematic analysis of the distribution and the expression levels of the mRNAs and proteins across the *Drosophila* larval nervous system for 200 genes. In the Davis lab, we worked collaboratively to achieve that and as I stated above produced a manuscript that is accepted for publication by the Journal of Cell Biology. Here I describe my own work and when I present work of other members of the Davis lab I mention them explicitly.

2.2.1 The Cambridge protein trap insertion (CPTI) collection

In this project, we wanted to create a spatial visualisation microscopy image collection of the co-localisation of the mRNA and protein expression levels in the *Drosophila* larval nervous system. To do that we decided to utilise a very useful resource, the CPTI collection (Lowe et al., 2014). The CPTI collection is constructed so that the fly lines have an intron incorporated fluorescent protein

reporter in each gene, which is flanked by splice donor and acceptor sites (Lowe et al., 2014). We take advantage of the fluorescent protein reporter, to co-visualise the mRNA alongside the protein. We created a common smFISH probe to target the mRNA sequence encoding the reporter - the genetically encoded YFP sequence. Transcription in the nucleus of the cell is reported, as well as the smFISH probe is inserted in the normal locus of genes and can detect primary transcripts. We used confocal microscopy to capture the fluorescent protein tag and the smFISH probe in whole tissues in different compartments of the *Drosophila* larval brain. A schematic of the procedure can be seen in Figure 2.1. We imaged the whole larval brain and six compartments of the larval nervous system highlighting specific tissues in each compartment. We imaged the CB and NBs (neural stem cells), the OL, the MB (the *Drosophila* equivalent to mammalian hippocampus), the VNC, the larval NMJ neurons, muscles and associated glia and the segmental nerves (Figure 2.2).

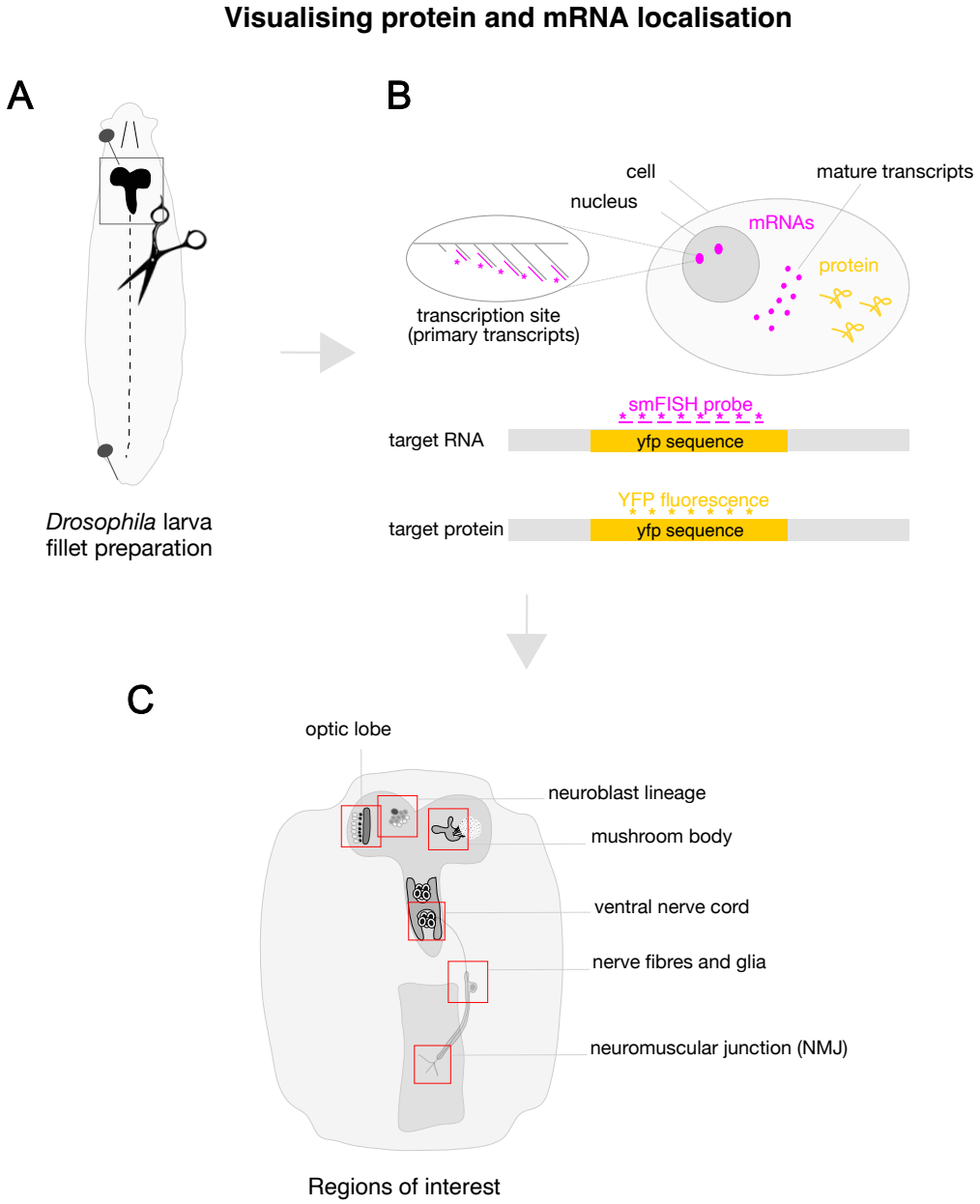


Figure 2.1: Workflow of visualising protein and mRNA expression in *Drosophila* larva. General strategy for simultaneously visualising RNA and protein in fluorescent protein trap lines. (A) After isolating the brain (including both the CNS and NMJ) from *Drosophila* 3rd instar larvae and (B) following a protocol to attach the smFISH probes to the specimen (described previously in (Titlow et al., 2018)), we image the nervous system using confocal microscopy. (C) We focus on six regions of interest in the nervous system and we also obtain brain overviews. (B) To achieve the imaging we take advantage of the YFP reporter inserted at the endogenous gene locus, whose function is not disrupted by the insertion. We construct common smFISH probes that bind to the YFP reporter sequence visualising both primary nascent transcripts produced in the nucleus and transcripts in the cell. A much brighter fluorescent focus is observed at the nucleus where the primary transcripts are highly concentrated. Figure based on Titlow et al. (2023) Figures 1 and 2.

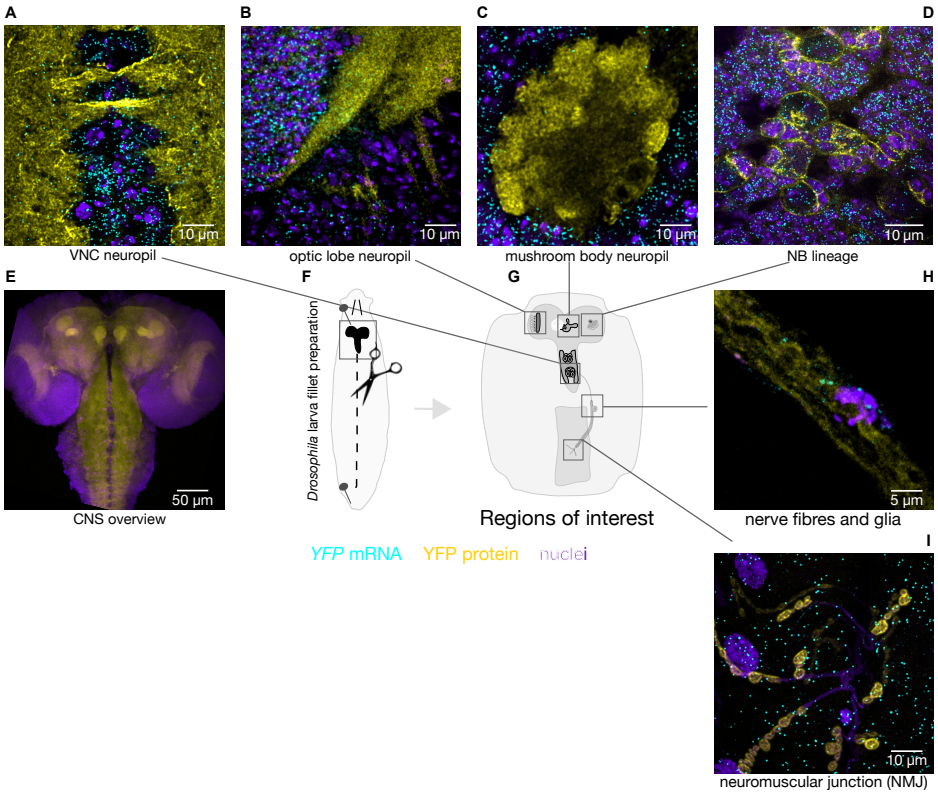


Figure 2.2: Nervous system regions of interest imaged for our microscopy YFP protein trap screen. (F, G) Seven compartments in the *Drosophila* nervous system were imaged. (A-E, H, I) Representative images are shown for each compartment focusing on the tissues of interest. The YFP protein is visualised with yellow, the YFP mRNA with cyan and the nuclei in the tissues with magenta. Figure based on Titlow et al. (2023) Figure 2.

For the construction of the CPTI lines a piggyBac vector transposon was used as piggyBac reportedly inserts randomly into the genome (Häcker et al., 2003; Thibault et al., 2004) and this approach provides greater genome coverage compared with previous methods such as P element insertions. The CPTI collection contains 575 lines, and 526 out of those, carry insertions into introns between protein-coding exons generating protein traps in 346 unique *Drosophila* genes (Lowe et al., 2014). For our microscopy YFP protein trap screen, we selected 200 lines corresponding to 200 unique genes. Our goal was to select lines that their insertion sites capture the most RNA isoforms. A full list of the CPTI lines selected, along with genes IDs and gene names are included in Appendix A.

2.2.2 Imaging and data management using OMERO

Imaging hundreds of individual *Drosophila* brains using smFISH is time-consuming as it involves many preparatory steps, including sample preparation, probe labelling and imaging. In this project we take advantage of the CPTI collection as it consists of expression lines that are fused with YFP where we can design a common smFISH probe to detect the YFP sequence for all selected genes. By using the same smFISH probe for all selected genes we minimise the time needed for the probe preparation step, reducing significantly the length of the whole procedure. Thus more brain samples can be imaged using confocal microscopy in a shorter time period.

Managing at scale high-content screening data is a notoriously challenging task. Multiple and heterogeneous types of information such as imaging data and acquisition related metadata that are essential in the interpretation of the imaging data need to be integrated together for efficiently analysing and evaluating them as a whole. Making them interoperable and consistent requires advanced software tools that can be deployed on-premises or accessed through the cloud (S. Li et al., 2016). For our study, it was required to image a three-dimensional volume of the tissue to retain the plethora of rich and diverse information that can be extracted from a probed *Drosophila* brain. Important metadata include the wavelength associated with each probe (or colour used to visualise the tissues), Z-stack information and other microscope related information. To store the raw data we used a version of the OMERO (Burel et al., 2015) software which we deployed locally in our lab server. We imaged 400 *Drosophila* samples corresponding to 200 unique genes (for each gene, one sample for the CNS and one for the PNS). Scanning the tissue three-dimensionally results in each raw image containing at least 50 z-stack layers producing large and rich data files. For each gene, we examined multiple tissues and compartments and took an overview of the brain. For each gene and for each tissue there was one sample imaged resulting in one image replicate per gene. For some genes, we repeated the experiment using a new sample, as we decided upon examination of the image that the quality of the image was not of the level required. The old image was replaced by the new, better quality image. The genes that were screened twice are given in Appendix A in table A.5. As this measure was taken to ensure that all the images are of the same

quality and clarity, this does not affect downstream annotation of the images or introduces any bias to the image analysis. Negative control experiments were performed using wild type fruit flies (no YFP insertion), to examine the baseline background expression.

As the overall number of raw images is in the scale of thousands, it was essential to find a way to summarise the imaging data in an accessible and meaningful way. I used the OMERO.figure web application (Allan et al., 2012), to create figures that display selected overlapping Z-stacks that capture the full complexity of expression, for each tissue of interest. Each image has four intensity channels corresponding to different probes. These included the mRNA expression channel (white), the protein expression channel (green), the blue channel showing the location of all the nuclei and the red channel showing the location of the NBs (Figure 2.3). We used the OMERO.figure app to contrast the images so that the distinguishing features stand out clearly. The web app allows the user to save copies of the images in the form of figures while preserving the unchanged raw data. In cases where irregularities were observed in the images, for example if the anticipated structure of the compartment examined was not observed we repeated the experiment and the imaging of the gene, to obtain a second image replicate. PDFs of the figures were exported together with links to the whole 3D volume of image data stored in OMERO. 1,361 figures with 1,896 raw image sources were exported. Each figure consists of panels for the tissues examined (as seen in Figure 2.3 A1, A3, B1, B3, C1, D1, E1, E2, F1, F3, G1 and G3) as well as panels for negative controls (wild type - no YFP inserted) (Figure 2.3 A2, A4, B2, B4, F2, F4, G2 and G4) of the corresponding tissues for comparison. Representative figures for each compartment and an example figure of the brain overview are shown in Figure 2.3.

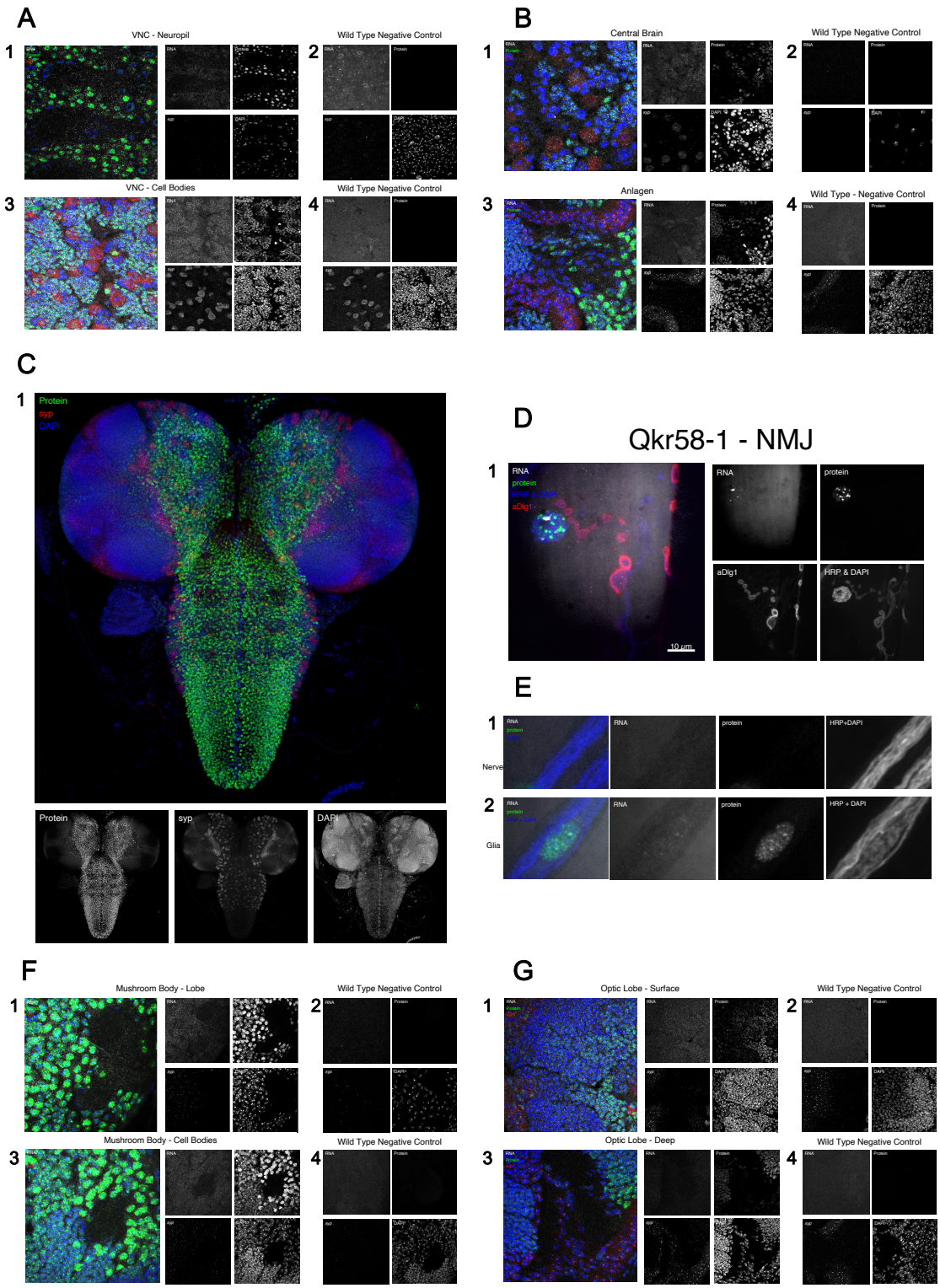


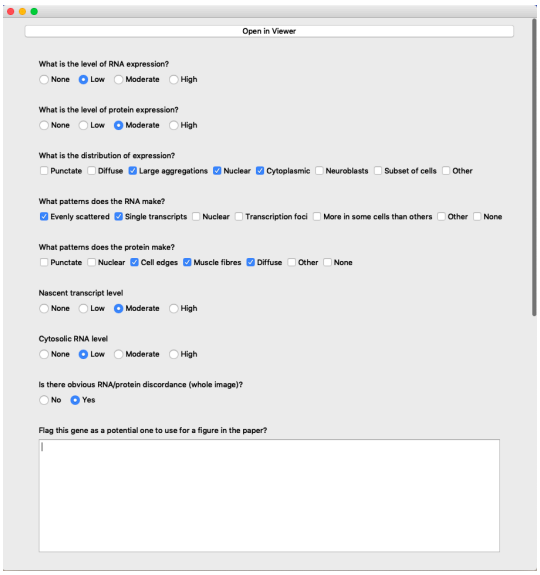
Figure 2.3: Representative figures for each compartment for the gene *qrk58E-1*. Figures for the (A) VNC (B) CB (C) overview (D) NMJ (E) Nerve (F) MB (G) OL compartments. In all figures, the DNA stain DAPI is shown in blue, indicating the location of all the nuclei in the brain and syncrip is shown in red indicating the location of NBs. The mRNA distribution is shown in white and protein distribution is visualised with green. All distributions are shown superimposed on the left panels of each figure (A1, A3, B1, B3, C1, D1, E1, E2, F1, F3, G1, G3) and separately on the right panels (A2, A4, B2, B4, F2, F4, G2, G4). A wild type (no YFP inserted) negative control for each tissue of interest is visualised.

2.2.3 Reaching consensus and resolving conflict

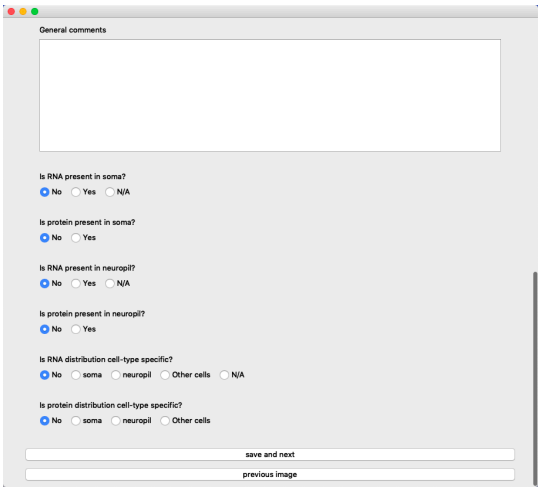
To systematically, efficiently and methodically annotate the figures created from the microscopy images we decided to create our own in-house scoring application. This gives us the ability to create our own controlled vocabulary to annotate the images and have a consistent description of the image metadata. With the help of Dr. David Pinto, I created an application to streamline the annotation process consisting of several scripts using python for the core scripts and the PyQt5 python library for the GUI (Kiourlappou et al., 2022). After conducting a short user experience survey between members of the lab it was deemed very important for the application to have a GUI (Figure 2.4 (A) and (B)) so that it is user-friendly and to make it as easy to run from each person's machine. Based on the user-experience testing feedback, I decided to provide the application as a zip file and make it possible to run the application by only typing a single line of code in the command line. The application, named "Annotate.OMERO.Fig" is available at the GitHub repository here: `https://github.com/mkiourlappou/Annotate.OMERO.Fig` and it is also archived at the Zenodo repository (Kiourlappou et al., 2022). I also wrote very clear instructions for the users on how to use the software. After producing the software and the instructions, I conducted usability testing to evaluate the overall user experience and cross-platform testing to assess the compatibility with different operating systems. The users were able to run the software successfully when following my instructions and the software was robust to different operating systems (macOS, Linux) and versions (macOS Catalina, macOS Big Sur).

Key files in our application include "questions.py", a file that the user can modify with their own set of questions and answers, i.e. the "vocabulary" or "taxonomy". The user can specify the type of question they want to ask, for example "CheckQuestion", "RadioQuestion" or "TextQuestion". The different types of questions get different types of answers, with the "CheckQuestion" allowing the user to select multiple answers, the "RadioQuestion" only allowing one answer and the "TextQuestion" providing a text box for the user to give a text answer (Figure 2.4 (A) and (B)). The ability to specify the content and type of question, makes our application fully adaptable to different requirements other researchers might have.

Part of the testing I conducted was to test if Annotate.OMERO.Fig can be used to annotate other datasets. I used the published dataset "BashTheBug" (Fowler et al., 2022) which comprises of photographs of 96-well broth micro dilution plates with each well containing a pre-determined amount of an antibiotic. The dataset needs to be annotated to determine the minimum inhibitory concentration for 13 drugs (Fowler et al., 2022). The annotators need to choose which photograph describes that for each drug for each plate. I created specific questions and answers to address this problem with Annotate.OMERO.Fig and successfully annotated the images and exported the labels for each figure. In the original "BashTheBug" project, Fowler et al. (2022) used Zooniverse (Cox et al., 2015), a science citizen platform and a crowd of volunteers to annotate the images. An alternative to Zooniverse, is to use my pipeline for a local and small-scale deployment of the annotation process.



(A)



(B)

	A	B	C	D	E	F	G
1	figure_id	Is RNA present in muscle?	Is protein present in muscle?	Is RNA present in post synaptic density?	Is protein present in post synaptic density?	Is RNA present in neuron?	Is protein present in neuron?
2	2125613	No	No	No	No	No	Yes
3	2125621	Yes	Yes	No	No	No	Yes
4	2137763	Yes	Yes	Yes	No	Yes	No
5	2137797	Yes	Yes	Yes	No	Yes	No
6	2137799	Yes	Yes	No	No	No	No
7	2137802	Yes	Yes	No	No	No	No
8	2137819	Yes	Yes	Yes	No	No	No
9	2137821	Yes	Yes	Yes	No	No	No
10	2137822	No	No	No	Yes	No	No
11	2137825	Yes	Yes	No	No	No	No
12	2137826	Yes	Yes	Yes	No	Yes	No
13	2137828	Yes	No	Yes	No	Yes	Yes
14	2137829	No	No	No	No	Yes	Yes
15	2137830	Yes	No	Yes	No	Yes	No
16	2137831	Yes	Yes	Yes	No	No	No
17	2137832	Yes	Yes	Yes	No	No	No
18	2137833	Yes	No	Yes	No	Yes	No
19	2137834	Yes	Yes	Yes	No	Yes	No
20	2137835	Yes	No	Yes	No	Yes	Yes
21	2137836	No	Yes	No	No	No	No
22	2137837	Yes	Yes	Yes	No	Yes	No
23	2137839	No	No	No	No	No	No
24	2137840	No	No	No	No	No	No
25	2137841	Yes	No	No	No	No	No
26	2137842	Yes	Yes	No	No	No	No
27	2137843	Yes	No	No	No	No	Yes
28	2137844	Yes	Yes	Yes	Yes	No	No
29	2137845	Yes	No	Yes	No	No	No
30	2137847	Yes	Yes	No	No	No	No
31	2137849	Yes	Yes	Yes	No	No	No
32	2137852	Yes	Yes	No	No	No	No
33	2137853	Yes	No	Yes	Yes	Yes	No

(C)

Figure 2.4: Snapshots of the scoring program GUI used to annotate our YFP protein trap screen. Panel (A) shows the first half of the program and Panel (B) the second, after the user scrolls down. The "Open in Viewer" button opens one of the figures (seen in Figure 2.3) and one can score the figure by clicking on the answer that best describes the figure. By clicking "save and next", shown in (B), the answers are exported in a .csv file (C) ready to be analysed. The different types of questions and answers are demonstrated in this version of the GUI. This version is used to score both the OL and the VNC compartments. The user can use the text box to comment on important findings or identified errors such as duplicated images or genes that exhibit interesting behaviour. These are also exported in the .csv file and can be revisited to make further conclusions or improve the study. (C) Representative .csv file showing part of the annotations for the NMJ compartment. The figures' IDs are shown but not the gene IDs or names to ensure anonymity.

A requirement for Annotate.OMERO.Fig, is that the images should be provided as jpgs or pngs in the "figures" folder. Therefore I converted the OMERO exported PDFs to jpgs. After cycling through the jpgs, annotating and completing the scoring, the output is compiled as a .csv file. The file contains each individual image name in a different row and the questions as column names. The answers are stored in cells with each column corresponding to the question and each row to the image ID (Figure 2.4 (C)).

After conducting more usability and user interface testing on Annotate.OMERO.Fig, I identified common problems that might occur and devised contingency strategies. For example, I identified that a user might not close an image and then get confused when a new one opens and not know which one to score so I made sure each image is automatically closed after the user clicks the "save and next" button and the new image opens. Additionally I conducted error handling and logging tests. For example, if there is already an answer in the "answers" folder for a specific image, the program handles this by not running for this image and this is to ensure that every image is only scored once in each iteration of scoring. If a user needs to re-score an image they need to delete the file containing the scoring for that image.

Our annotators were experts from the lab and as a result have prior knowledge on how some genes are expected to behave and express mRNA and protein. To avoid the risk of introducing any prior knowledge or bias into the current annotations, I took the measure of anonymizing the images. In this way the annotators do not know which gene they score so they score objectively and impartially. I also added a "General comments" text box in the questionnaire (as seen in the GUI in Figure 2.4 (B)). The expert can note any comments they have for example if an image is duplicated and they have noticed they scored this already, or if a figure seems that it has not been contrasted correctly or if simply the expression observed in the specific microscopy image is very interesting. The "General comments" text box was necessary for quality control purposes, as there was initially one image replicate per gene. However, if one of the expert annotators noted that there is a discrepancy with the structure or expression expected in the image (e.g. after

comparing this image with the negative control image) the experiment and the imaging were repeated followed by re-annotation of the figure by the three experts.

We produced figures corresponding to six distinct *Drosophila* nervous system compartments and an overview figure showing an image of the intact brain for the 200 protein-coding genes. We scored the six compartments only and not the overview figures. The compartments include the following: CB, OL, MB, NMJ, VNC and nerve. The nerve compartment is not screened for the entirety of the 200 genes and this is the reason I do not present full results for the nerve compartment annotations in Figure 2.8, where I present the summary of annotations. I therefore have six sets of images, that is approximately 1,200 images that need to be scored. For each compartment the questions were slightly modified depending on the number and type of tissues present in each compartment, so that each tissue was scored separately for mRNA and protein expression levels, but also each compartment was scored as a whole. The maximum number of questions asked was 20 for NMJ and the minimum was 14 for the OL and VNC.

Six expert annotators from the Davis lab research group with deep knowledge of the larval nervous system tissue and smFISH signal interpretation were asked to annotate the sets of images, and combinations of three of them annotated each set of images. No expert consulted another expert when annotating so that the scoring was consistent throughout the dataset and impartial between the annotators. I then compared the output of the application for each one of the three experts between them to spot discrepancies and inconsistencies between the scoring. These were expected as different experts might have different opinions or scoring approach. This study generated more than 18,000 annotations.

As the questions are modified for each compartment I analyse the annotations per compartment. There were different types of questions with various responses, so I performed my analysis based on the type of question; e.g. single-selection and multi-selection questions. Single-selection questions are questions where the annotator can select only one answer from a predetermined list of answers and multi-selection questions where the expert can select one or more responses. For each compartment, there are two multi-selection questions, these are: “If there is RNA

expression, what patterns does the RNA make?" and "If there is protein expression, what patterns does the protein make?". These questions are set up as "RadioQuestion" so that they can take multiple answers. In this specific case, the aforementioned questions can take up to seven pattern answers. Single-select annotations can be broken down to binary class and multi-class annotations. Most of the questions have a simple "Yes"/"No" answer but others have levels for example like the "what is the level of RNA/protein expression?" which can take up to four answers, "High", "Moderate", "Low", "None".

Initially I compared the scoring of each annotator and as it was expected I observed a dissimilarity between the scoring. I decided to use majority voting (Sabou et al., 2014 ; Davani et al., 2022) to reach consensus and resolve annotator disagreements. To democratise the process, all experts were treated as equals and consequently, all experts' annotations carried the same weight. The final annotations are presented in section 2.3.4, where I also examine metrics for judging annotator agreement and present the results of the analysis.

In cases where consensus could not be achieved and there was conflict a fourth, independent and more senior expert was recruited to resolve the disagreement (Davani et al., 2022; Waseem & Hovy, 2016). To accomplish this as efficiently and as quickly as feasible I decided to employ the data visualisation software, Zegami ("Imaging AI Software Company," n.d.; Taylor & Noble, 2014). It's use is described in section 2.2.4 where I introduce the software and present usecases. The data for the cases where a fourth expert were required are available along with the finalised annotations in my GitHub repository (`https://github.com/mkiourlappou/thesis-project/tree/main/scoringResults`).

I also conducted a sensitivity analysis to assess the robustness of the annotation consensus to perturbations in the annotations. In a collection of annotations generated by a large number of annotators, the effect of changing one annotation may be not as pronounced compared to collection of annotations created by fewer annotators. This is because when there are fewer annotators and annotations each annotation is more significant in determining the overall consensus. However

this also depends on the level of agreement between the annotators. If there is high agreement, changing one annotation may not alter consensus.

I conducted my analysis, by first identifying the type of perturbation I wanted to simulate, e.g. changing a single annotation. I also identified all the possible values each annotation could take, the probability that the annotations change to the new value and rerun the consensus analysis. Finally, I assessed how the results deviate from the original perturbations. I present the full results in the form of a table in Appendix B and discuss them in section 2.3.4.

Majority voting and arbitrating by an independent expert are methods that have been used for years in corpus acquisition research (Davani et al., 2022). In this type of research, there is usually a high amount of subjectivity, as there might be implicit bias when annotating a text with a sentiment. In our project we tried to make the questions and answers as objective as possible, by binarising our answers and introducing levels instead of continuous scales. Additionally, biological annotation in contrast to sentiment analysis is a more objective field as annotation concerns visual data rather than the sentiment a text describes. For example, we designed the question regarding protein expression in specific tissues to be "Is protein present in soma?" (soma is the specific tissue in this case) and we designed simple "Yes" and "No" answers.

Subjectivity is inevitable though as different experts might disagree on what makes protein and mRNA expression present. Some might see a single protein spot and label it as present but others might argue that there should be more than one spot to label the figure positive for protein presence. Moreover, sometimes image artefacts might get confused with mRNA molecules and this might result in spurious answers. However the fact that multiple individuals are asked to classify the same subject can mitigate the effects of subjective leading to more objective annotations compared to when only one expert is asked to classify a subject (P. Smith et al., 2023). Additionally, subjective annotations which lead to disagreement between the annotators, might lead to serendipitous discovery, as the different experts might notice different important structures in biological images worth investigating further (P. Smith et al., 2023). However, to

mitigate any hint of subjectivity, I set up an evaluation method for the annotators agreement and disagreement which is described in section 2.3.4.

Protein expression can be measured computationally by quantifying the fluorescence intensity of the YFP protein. In recent work by J. Y. Lee et al. (2022), Cellpose (Stringer et al., 2021), an automated cell segmentation method powered by deep learning algorithms, was used for cell segmentation in 2D tissue to quantify expression of protein in SARS-CoV-2 infected cells. When using Cellpose, nuclei and cytoplasm can be identified and the protein expression compared with the control protein expression to quantify its absolute value. There are some technical limitations associated with using Cellpose. Segmentation of 3D cells is not very accurate and identifying the different cell types that might exist in the 3D volume is not yet possible. Therefore using such automated cell segmentation techniques to quantify intensity would be difficult in my project, as many of the questions asked are tissue specific ("Is protein present in soma?") and the image data are three-dimensional. If such automation was to be introduced experts would still be required to identify the different types of cells and aid with cell recognition in the image data.

2.2.4 Visualising and merging multiple types of data

The sheer volume of biological data generated in this screen has caused many challenges, including data storage and management (mentioned in section 2.2.2) and data visualisation. Effective visualisation of complex biological data can be difficult due to the fact that it often involves multiple dimensions, different types of data and a huge amount of information. When data visualisation is done correctly, it can provide a thorough insight of the biological system under consideration as well as reveal interesting trends in the data. It usually necessitates specialised expertise and software. To aid with data visualisation, Zegami ("Imaging AI Software Company," n.d.; Taylor & Noble, 2014), an image analysis platform with integrated advanced analysis tools that enables artificial intelligence exploration, was used. Zegami is also the company employing the Zegami software which is cloud-based. As part of my DPhil programme I have been collaborating with Zegami which has been acting as my industrial partner.

Zegami allows the user to explore multiple images simultaneously and at high resolution, by utilising a deep zoom functionality when the user focuses on one image. The user can explore and analyse metadata associated with the images with the help of graph visualisation and incorporated machine learning techniques. A Zegami collection is fundamentally a collection of visual tiles. Each tile can be occupied by an image and if the user selects a tile they can access data associated with the tile/image if the data exist. To create a collection the user should upload a group of images and can optionally add associated metadata (Figure 2.5 A). The metadata should be in a tabular format and can take the form of a .csv, .tsv or .xlsx file (Figure 2.5 A). The file should include a column with the images' names so that each row of data is connected with one image and has a unique identifier. If a row does not have an associated image name the data appear in a Zegami collection as a grey tile (Figure 2.5 B1).

I created the collection shown in Zegami using our YFP screen microscopy figures. I created a metadata file containing the annotations for each image for each gene. In the metadata file I included physical, functional and expression information for the whole *Drosophila* genome, including which genes have protein insertion traps. Additionally, I have added links to online bioinformatics libraries like FlyBase (Gramates et al., 2022) and links to the OMERO database where the raw figures are stored. There are 18,203 items in the collection as these correspond to facts about the whole genome and not only the screened genes. In section 2.3, the collection will be described in more detail.

The landing page of the collection can be seen in Figure 2.5 B1. In the same panel the "Snapshots" functionality is also shown. Different views of the same collection can be accessed through this panel. Highlights include the "Data Similarity view" and the "Image Similarity view". These are views that are automatically produced by Zegami as the user uploads the data. They show a clustered version of the data, the former based on similarity between the metadata and the latter based on image similarity. These are useful as they can help identify important relationships and patterns that may be missed by a more narrow or isolated view of the data.

The different filters correspond to the different column names taken from the input table file. The filter types can be numerical or categorical depending on the type of data provided. In Figure 2.5 B2, 1,400 out of 18,203 images have been filtered using the filter "Yes" for the category "Is this gene screened?" as shown on the top left corner. In the same figure in panel C, all the filters available are shown on the filtering panel and in panel D, a filter with many positive selections is shown. The 137 genes filtered out have mRNA expression present in at least one of the MB neuropil, NMJ axon terminal, OL neuropil and VNC neuropil tissues. The aforementioned tissues are collectively referred to in this thesis as synaptic compartments. In panel C we can also see one of the functionalities that makes Zegami so unique, the zoom-in and zoom-out functionality. With the same speed and high quality that the user can see a zoom-out version of the collection with tens of thousands images all at the same time, the user can zoom in quickly to one of the images. In this panel the gene *Smr* is on focus and the user can examine all the metadata associated with the selected image and gene.

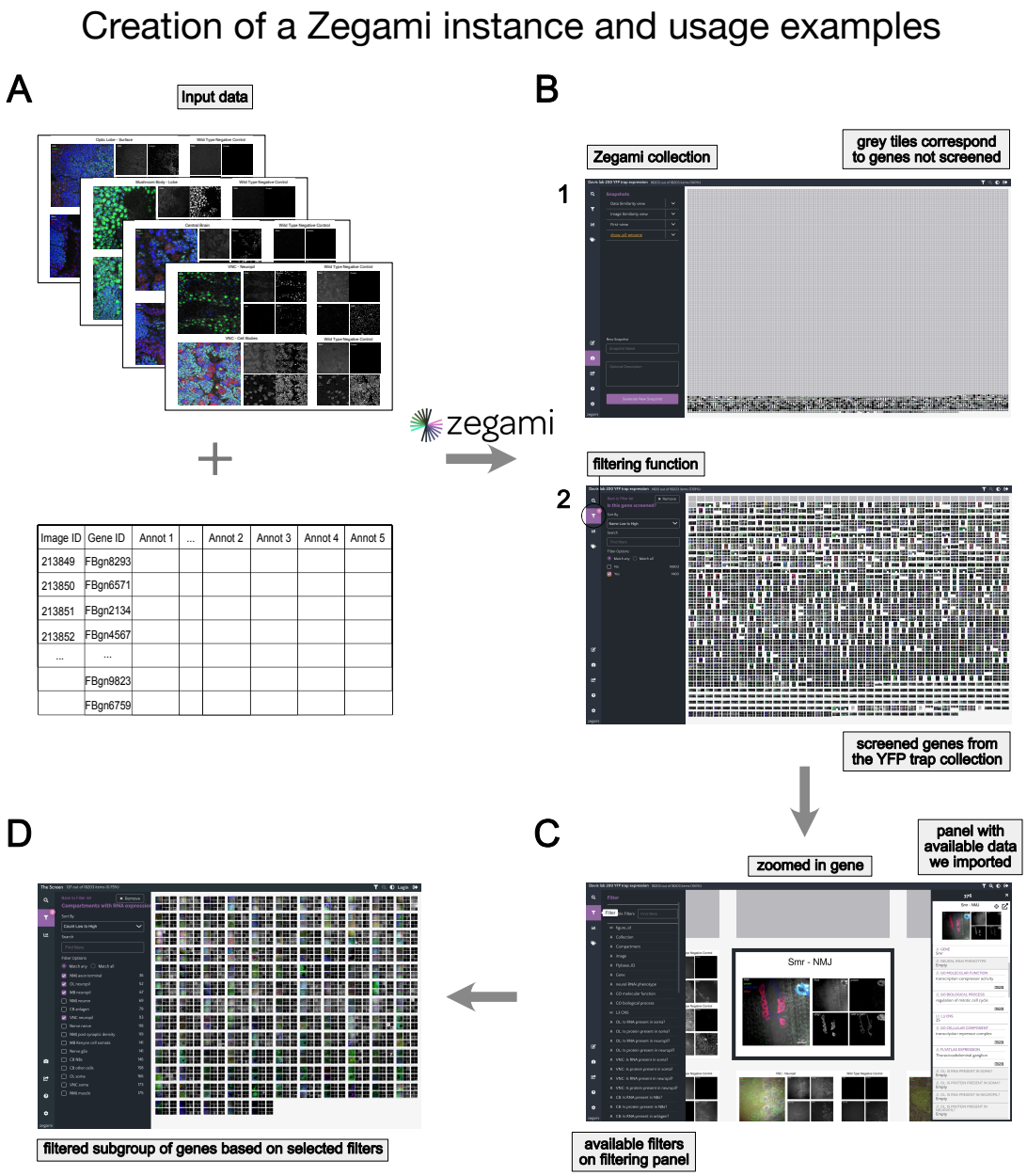


Figure 2.5: Creating and using a Zegami instance based on our 200 YFP protein trap microscopy screen. (A) To create a Zegami instance, images in the form of pngs or jpgs are required. A tabular dataset should be uploaded with the images, if the user wants the instance to have associated metadata. To connect the images with the related metadata, the images should have an associated image ID for each image included in the tabular file. Each row of the file that has an associated ID, contains data that will appear as metadata for the relevant image. The file can also contain rows without an image ID. In this example we used a csv file containing information for every gene in the *Drosophila* genome and for the 200 that were screened there is a figure ID corresponding to them. (B1) Once the data are uploaded to Zegami, a collection is created. There are 18,203 items (total number of grey tiles and images). We can also see the name of the collection on the top left. Grey tiles correspond to genes that were not screened and there is no image corresponding to them. (B2) Using the filtering function on the panel on the left hand side, I can filter and select a subset of genes. Here, I selected the 200 genes we screened from the CPTI collection. (C) Zoomed in version of the selected tile/image for the gene *Smr*. The data related to the image appear on the right hand side panel (metadata panel) and on the left panel the filtering panel appears with the available filters so that the user can select and view other images. (D) In this panel a filtered subsection of the images is shown. In this specific example 137 images are filtered out and they correspond to the 137 genes that have transcripts present in at least one synaptic compartment.

Another useful functionality of Zegami, is the tagging functionality. The user can add a tag and annotate an image interactively, while using the software and browsing the collection. Then the user can search using the search capability for a specific tag and Zegami will filter out all images/tiles associated with the tag. This functionality is used in the annotating figures cycle to aid with conflict resolution when employing an adjudicator, as I mention in the previous section. This is shown visually in Figure 2.6. Panel A of Figure 2.6 shows a Zegami collection containing figures for the NMJ compartment. These are figures for which all three annotators disagreed. Using python, I merged the scoring for each annotator for the questions that there was disagreement and created a .csv file. Each row of the csv corresponds to an event of disagreement, and contains the figure ID, the question and the annotations of each of the annotators. When the user selects any image these data appear in the metadata panel as it can be seen in the blue box of panel A. For the selected image there is no tag added as it can be seen in the red box. However in panel Figure 2.6 (B) there is a tag added and can be added using the "Tag Name" text box and then clicking on "Create". A user can also use the "Search" panel (blue box) to search for an annotation they added previously.

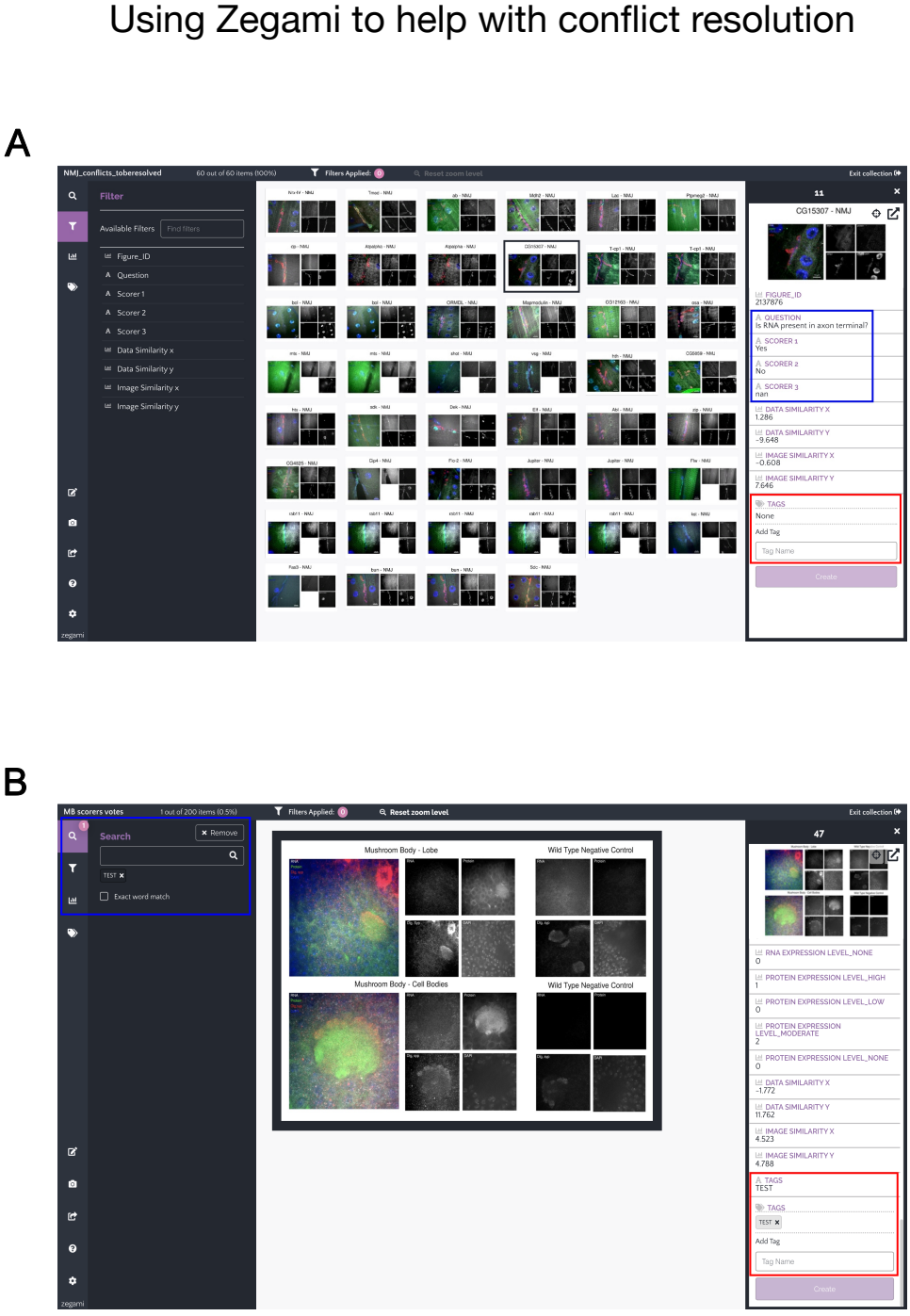


Figure 2.6: The tagging functionality in Zegami is used for helping with conflict resolution. (A) A collection for the NMJ compartment containing 60 instances of disagreement. In the metadata panel for the selected image, the question and the answers the voters gave are enclosed in a blue box. The red box shows the tagging functionality in Zegami and that for the selected image there is no tag assigned to it. (B) A collection for the MB compartment containing the votes the experts gave for a specific question. A user can search for a tag using the search functionality and this is highlighted in the blue box. The tag for the selected image is shown in the red box.

Using Zegami in the conflict resolution pipeline was very beneficial. It saved the adjudicator a significant amount of time as it allowed them to navigate between the images and look at the annotations at the same time; as well as seamlessly adding a tag with what the adjudicator thought was the correct answer. Moreover, it helps the fourth expert to avoid making mistakes by looking at a different image and not the annotations corresponding to the image as Zegami zooms in in the selected image. Additionally, Zegami allows the creator of the collection to export the tags in the form of a csv file thus allowing integration with the data analysis pipelines I set up in python.

2.3 Results and Discussion

2.3.1 The 200 selected and screened genes from the CPTI YFP protein trap collection are a sample that represents faithfully the whole protein-coding genome

To investigate whether the selected lines can be representative of the whole genome and not biased towards specific mRNA and protein subcellular localisation patterns we decided to compare mRNA expression levels, physical structure and gene function for the 200 selected genes to the whole transcriptome. Figure 2.7 presents the comparison of the aforementioned properties in the form of violin plots.

We started by comparing mRNA expression levels of the 200 selected screened genes to all the protein-coding genes constituting the transcriptome using the published 3rd instar larval brain transcriptome (Samuels et al., 2020). I produced split violin plots of the expression level distributions of all protein-coding genes, shown in gray in Figure 2.7 A and of the screened genes, shown in green in Figure 2.7 A. I observe that although the 200 selected screened genes are overall more expressed than the transcriptome average, they express over the whole range of protein-coding genes, with both high and low RNA expression levels. I conclude that they can represent the diversity of gene expression in the larval brain.

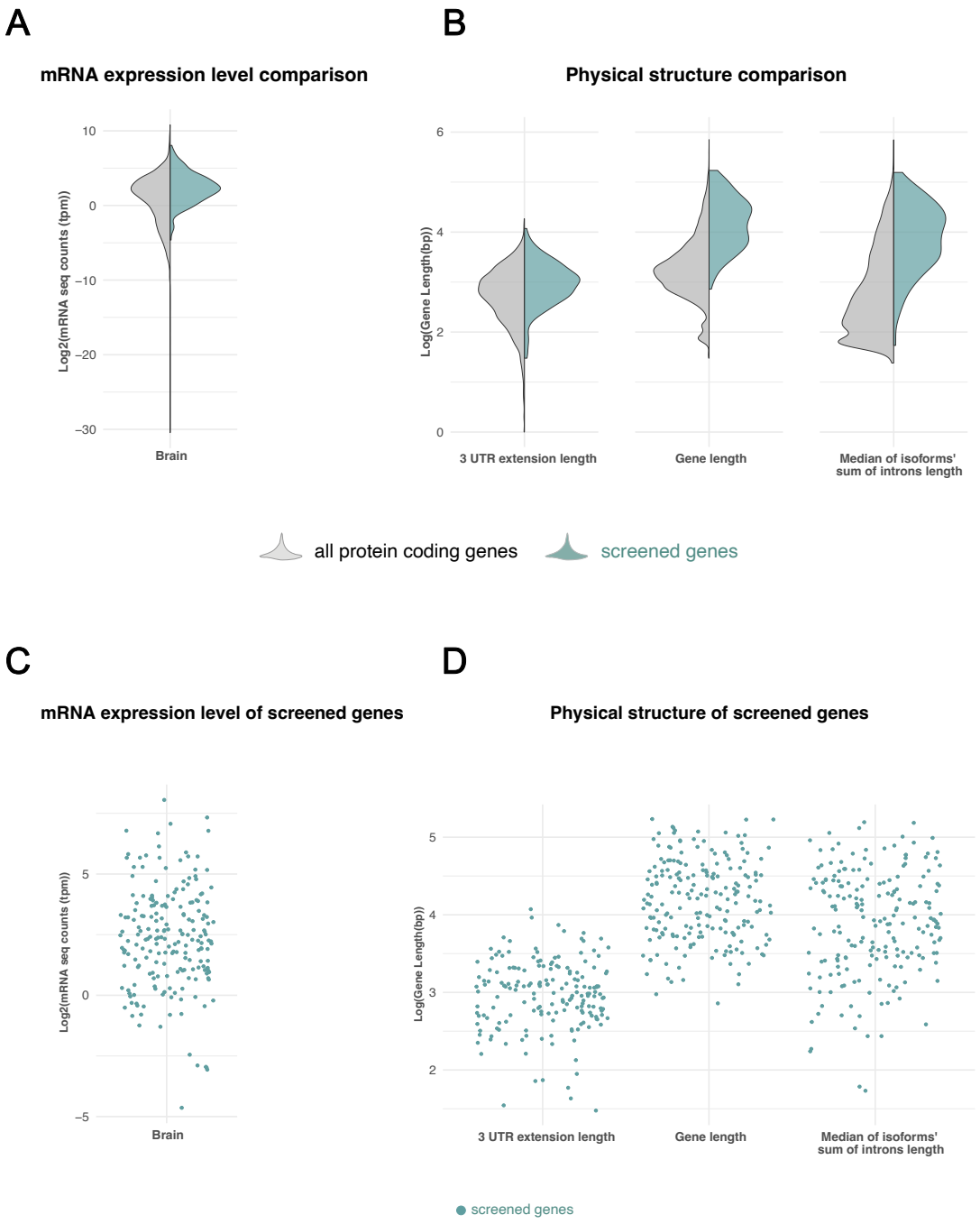


Figure 2.7: The selected screened genes are a valid representation of the whole transcriptome based on mRNA expression levels and physical structure. In panel (A) we can see split violin plot distributions of the mRNA expression in the 3rd instar larval brain (transcript per million) using published data by Samuels et al. (2020). Distributions of the mRNA expression for the screened genes compared to the protein-coding genes are similar. In panel (B), we see the distributions of the physical properties. The 3' UTR lengths are comparable for both groups of genes but the gene lengths and sum of introns length are not. In both panels the whole transcriptome distributions are depicted in gray while the screened genes distributions are presented in green. Panels (C) and (D) show scatter plots of the mRNA expression in the 3rd instar larval brain and the physical properties respectively for the screened genes only. Data from the same sources as for panels (A) and (B).

Next, we focused on comparing the physical properties of the genes. I compared the distributions of gene length, 3' UTR extension length and the median of the isoforms sum of introns lengths using data available from ENSEMBL (Cunningham et al., 2022) (ENSEMBL release 104). The 3' UTR extension length is the maximum difference in 3' UTR length between genes' isoforms. I produced split violin plots of the distributions of lengths in logarithmic scale, as seen in Figure 2.7 B. We note that the distribution of the 3'UTR extension lengths for the protein-coding genes is similar to the distribution for the 200 screened genes. 3' UTRs are known localisation drivers with significant regulatory potential (Tushev et al., 2018). This finding indicates that we can treat our scoring of the RNA expression as representative of the protein-coding genome RNA expression behaviour. The CPTI collection genes are on average longer and have more introns, with their average size significantly larger than the average gene's (Lowe et al., 2014). This is expected as they carry random insertions between protein-coding exons. These findings correlate well with our analysis as we see in the last two violin plots of Figure 2.7 B that both the gene length and the median of isoforms sum of introns lengths distributions are larger than the transcriptome. For clarity, I plot also scatter plots of the distributions of mRNA expression and the physical properties shown in panels A and B of Figure 2.7 separately. I show the scatter plots in panels C and D of Figure 2.7.

To determine if the 200 CPTI collection screened genes are not biased for specific gene functions, Dr Joshua Titlow used GO terms (Ashburner et al., 2000; The Gene Ontology Consortium et al., 2021), and specifically using a selected subset, the GO slims subset acquired using ENSEMBL BioMart (Kinsella et al., 2011). He overlapped the distinctive terms associated with the screened genes and the distinctive terms associated with the protein-coding genes and found that our 200 selected genes overlap by 89.9% and this overlap covers all three aspects of the biological domain (molecular function (MF), cellular component (CC) and biological process (BP)) as defined by the GO (Titlow et al., 2023). Therefore I conclude that the selected genes' function is a fair representation of the whole transcriptome's function, suggesting that the diversity of gene expression is captured by our subset of genes.

2.3.2 There is a high percentage of discordance between mRNA and protein expression, with at least one nervous system region that shows discordance for 195 genes out of 200 genes (97.5 %).

Next, I examine the relative mRNA and protein levels and spatial expression for each gene and for each nervous system region of interest. I summarised the finalised ground truth labels from our annotation data set and analysed the discordance measure for every gene and for each tissue of interest. Discordance is the lack of correlation between the expression of mRNA and protein, for example when the expression of protein is observed but not of mRNA and vice versa. On the contrary, concordance is the lack of expression for both mRNA and protein or the simultaneous expression of both mRNA and protein. Discordance in the mRNA and protein levels and in their distribution is one of the signatures of post-transcriptional control (Buccitelli & Selbach, 2020; Pilaz & Silver, 2015).

In Figure 2.8 we see a heatmap, with each row representing a gene and each column a region of interest examined. The shades of purple/pink in the heatmap indicate the presence of discordance (purple), and the presence and type of concordance observed (pink). Out of the 200 genes we tested we see that 195 out of the 200 genes, show discordant mRNA and protein expression in at least one of the regions of interest examined, as shown in Figure 2.8. This is an exciting finding as typically the levels of mRNA and protein show a substantial across-gene positive correlation, taking into consideration technical limitations (Buccitelli & Selbach, 2020). Additionally, in Figure 2.8 the columns showing discordance per tissue are grouped by the discordance type, intracellular or intercellular. Intracellular discordance happens within a single cell and intercellular between cells. Intracellular discordance can occur in highly polarised cells such as neurons where protein is required at the tips of the cell (Chekulaeva & Landthaler, 2016; Holt et al., 2019). Post-transcriptional regulation mechanisms are employed to help with that such as local translation and mRNA degradation. Intercellular mRNA/protein discordance arises from diversity in transcription and translation. In our dataset and as seen in Figure 2.8, 64.5% of the genes show intercellular discordance and 89.0 % intracellular discordance in at least one tissue examined. As mentioned earlier, I do not present the annotation results for nerve in Figure 2.8 as

there were missing microscopy images for a small part of the genes. Additionally, we noticed that for the anlagen tissue in the CB compartment the screened anlagen structure was not as clear as required for scoring for certain genes. Therefore, I do not include the annotation results in Figure 2.8 either as it would not be a complete set of labels. In summary, I present results for twelve tissues in five compartments of the *Drosophila* nervous system.



Figure 2.8: Heatmap showing mRNA and protein expression discordance results for each gene, per region of interest. Presence of discordance is shown with dark purple and concordance with pink. The shades of pink categorise the type of concordance; darker pink displays concordance arising from genes expressing both mRNA and protein at the same time in the same tissue whereas lighter pink displays concordance arising from absence of both mRNA and protein expression. Each row represents a gene. The data are grouped by intercellular expression (between cells) and intracellular expression (within a single cell). A bar chart on the right hand side sums up the number of tissues where a gene expresses discordantly. I produced this figure, published as Figure 3A of (Titlow et al., 2023).

As we can see in Figure 2.8, the MB, OL, and VNC synaptic neuropils have larger percentages of genes that solely express protein unlike other tissues. We think that this expression pattern is created by either selective protein transport to the synapse or by increased protein stability, which permits it to survive after the mRNA is degraded (Titlow et al., 2023).

Figure 2.9 shows a cumulative percentage of types of expression in all tissues of interest grouped by compartment. It shows the roots of discordance and concordance in more detail. This graph emphasises that in most compartments discordance arises from the presence of mRNA without protein (compare light pink to dark green segments), which is highly indicative of translational inhibition (Titlow et al., 2023).

Figures 2.8 and 2.9 emphasise the importance of presenting detailed views of this dataset as well as aggregated as this helps explore the different types of discordance and compare and contrast regions of interest expected to behave similarly.

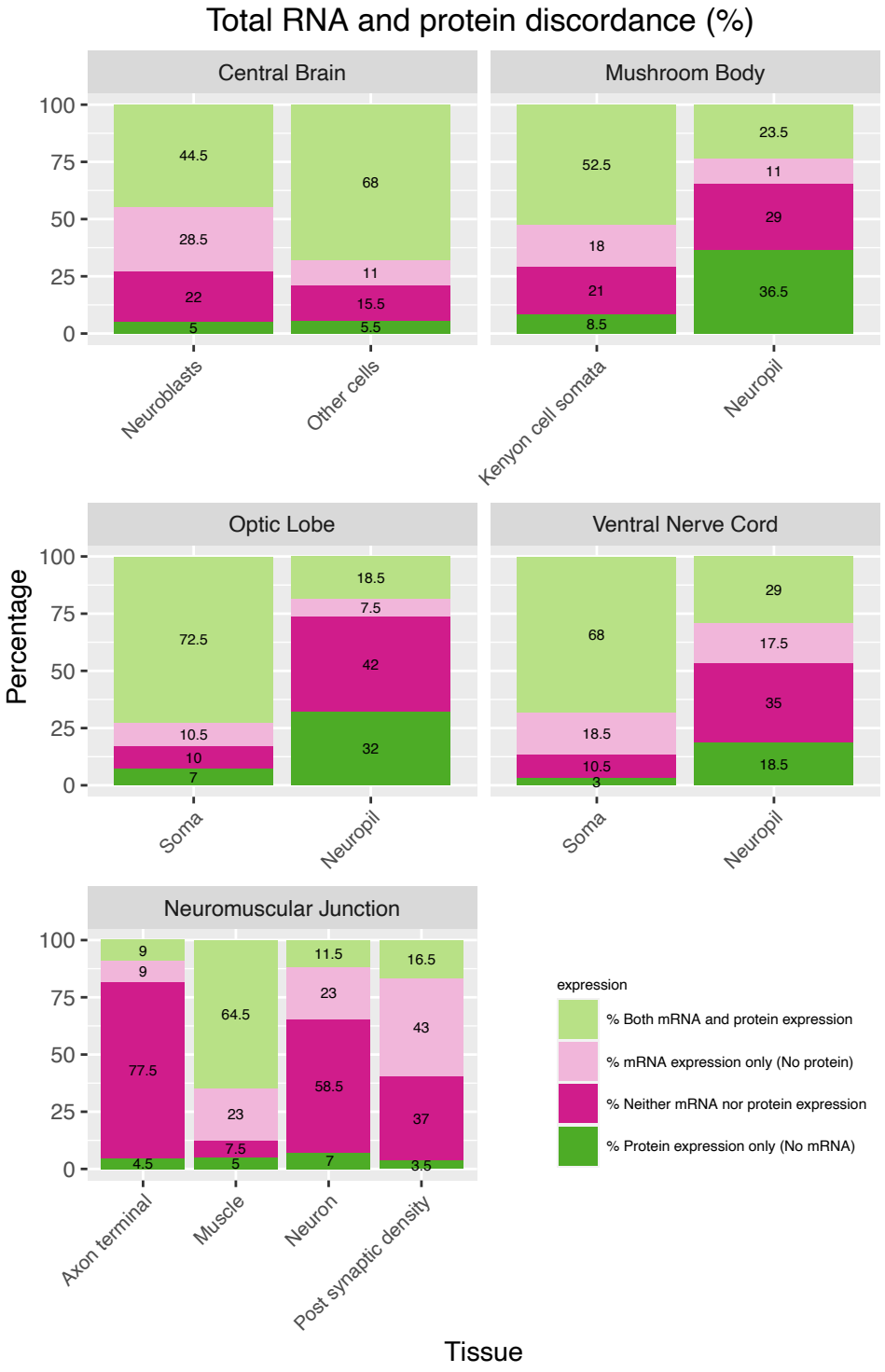


Figure 2.9: Bar charts summarising the annotation labels per compartment and per tissue. The bar charts display percentages of the 200 screened genes that show specific types of expression. The combinations of presence and absence of expression (i.e. mRNA expression in the specific tissue but no protein expression and vice versa) are denoted by the different colours. This figure I produced is also published as Figure 3B of Titlow et al. (2023).

2.3.3 mRNA and protein express in the tips of glial cells in the *Drosophila* nervous system

As previously mentioned, mRNA localisation, mRNA/protein transport and local translation are key post-transcriptional regulation mechanisms in polarised cells such as neurons and glia. There are seven types of glia in the *Drosophila* nervous system, both in the CNS and the PNS. Specific to the *Drosophila* PNS are the wrapping glia and to the CNS astrocyte-like glia, ensheathing glia, midline glia and cortex glia. Perineural glia and subperineural glia are the two types of glia that are common to both PNS and CNS (Bittern et al., 2021). There are examples that suggest that deregulated local translation in oligodendrocytes, a mammalian glia cell type in the CNS, plays an important role in neurological and neurodegenerative diseases (Blanco-Urrejola et al., 2021).

Studying the subcellular distribution of the glial transcriptome and proteome is therefore a primary means to better understand their function and if perturbation of this equilibrium leads to neurological and neurodegenerative diseases. Here, we examined the mRNA and protein expression in the peripheral glia of the *Drosophila* PNS for where the map of expression is unknown. When scoring the NMJ compartment we observed that there were mRNA and protein expression patterns in other cell types wrapped around the neuron, which we identified as peripheral glial cells. These were wrapping glia at the axon terminal, or perineural and subperineural glia of the NMJ (Titlow et al., 2023). To verify that the cell types we observed were indeed glia, Josh Titlow and Dalia Gala used a pan-glial GAL4 driving a membrane mCherry marker (repo-GAL4>UAS-mcd8-mCherry) to mark the glial cells within the nervous system. They then compared the expression of the repo-GAL4>UAS-mcd8-mCherry marker with the expression of a known wrapping glial marker, Nrv2 (Yadav et al., 2019). The Nrv2::YFP insertion is one of the 200 YFP insertion lines tested. Comparison of this expression is shown in Figure 2.10, where the *Drosophila* larval NMJ compartment is imaged. We see a schematic of the peripheral glial and muscle anatomy. In panel A, we see a microscopy image of this, with the nuclei and axons marked with purple. In panel B, shows a zoomed in single optical section of the neuron and the glia with the Nrv2::YFP protein shown with green and the *nrv2::YFP* mRNA in white. Panels C and D show the same optical section as panel B, but only protein is visualised

in panel C and only mRNA in panel D. Glia are marked with the repo (glia) > CD8-RFP marker and we can see an obvious overlap in the inset in panel D.

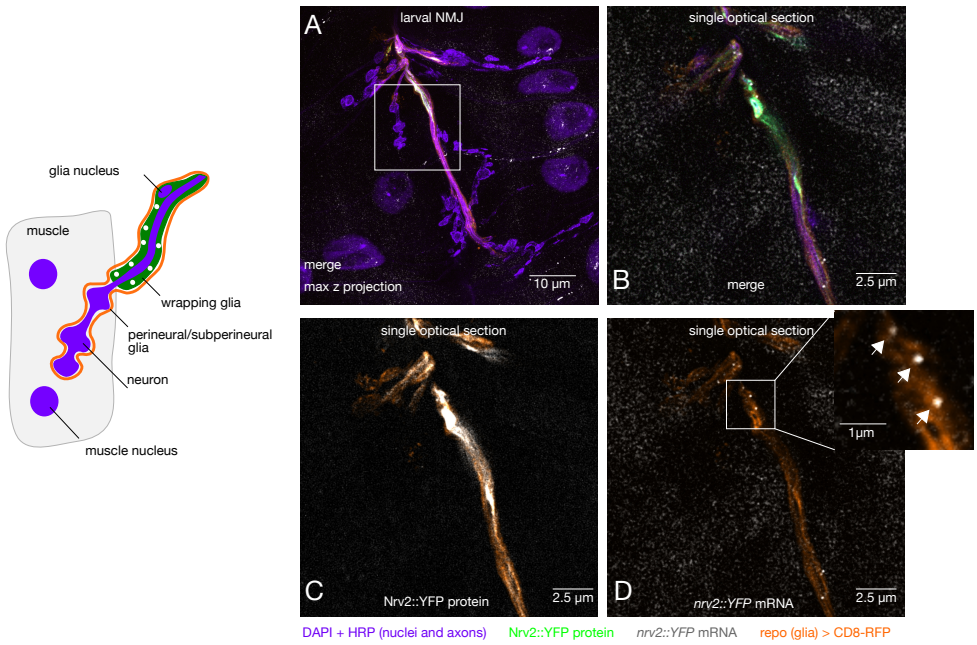


Figure 2.10: Peripheral glia cells mRNA and protein expression in the *Drosophila* NMJ. Schematic overview of the peripheral glia (wrapping, perineural and subperineural) and the neuron in the NMJ. (A) Microscopy overview image of the larval NMJ. (B-D) Single optical sections showing *nrv2::YFP* mRNA and Nrv2::YFP protein expression in glia, merged and separately. The inset is showing single molecules in greater detail. Adapted from Figure 6 from Titlow et al. (2023).

Joshua Titlow was able to identify the pattern of expression of the peripheral glial cells in the NMJ, without highlighting it with a glial marker. He then examined the 200 selected imaged genes looking for this pattern. He found that 18 genes out of the 200 show mRNA and protein expression in the tips of glial cells, the specific genes are shown in Table 2.1. This together with results from (Kügelgen & Chekulaeva, 2020 ; Giandomenico et al., 2022 ; Zappulo et al., 2017) can signify that mRNA localisation at the tips is important in the production of protein in that region of interest, proposing that mRNA localisation and local translation are important mechanisms in the peripheral glia of the *Drosophila* brain.

Table 2.1: List of genes for which there is mRNA and protein expression in peripheral glia in the *Drosophila* larval nervous system

Gene ID	CPTI line ID	Gene name
FBgn0010215	CPTI002342	*alpha-Cat*
FBgn0002921	CPTI002761	*Atpalpha*
FBgn0033446	CPTI100012	*CG1648*
FBgn0259244	CPTI000033	*CG42342*
FBgn0031268	CPTI001277	*cold*
FBgn0264078	CPTI001427	*Flo-2*
FBgn0001987	CPTI002805	*Gli*
FBgn0263594	CPTI002437	*growl*
FBgn0001145	CPTI001918	*Gs2*
FBgn0004167	CPTI001728	*kst*
FBgn0010238	CPTI002408	*Lac*
FBgn0264975	CPTI001714	*Nrg*
FBgn0015777	CPTI001455	*nrv2*
FBgn0013997	CPTI001977	*Nrx-IV*
FBgn0286818	CPTI000688	*Pdi*
FBgn0021764	CPTI000337	*sdk*
FBgn0013733	CPTI001962	*shot*
FBgn0005671	CPTI002645	*Vha55*

In a new publication from the Davis laboratory in which I am also a co-author, Dalia Gala (Gala et al., 2022) for the purposes of the study tested 11 out of the 18 genes identified above with specific glial markers using smFISH and this showed that the genes tested had indeed mRNA present in the peripheral glia.

Identifying mRNA and protein expression patterns in peripheral glia by chance is an interesting discovery that expands our understanding of glial expression. This is especially noteworthy as the understanding of the *Drosophila* glia in the nervous system is still very limited. This discovery will be discussed in the following two chapters, as the identified 18 genes from a pool of 200 genes will be utilised to help prediction of transcript presence in glia in *Drosophila* and *Mus*, another important model organism.

2.3.4 Confidence in quality of annotations can be high even when inter annotator agreement is not

In section 2.2.3 I acknowledged that occasionally there is disagreement between the scoring of the annotators. This could rise due to chance disagreement in case the annotators are not sure what

the right answer is, if they genuinely disagree about the answer or if they interpret the answers differently. I now investigate the inter-annotator agreement between the three expert scorers.

Inter-annotator or interrater agreement is a measure of the extent to which independent judges tend to assign exactly the same rating to each object (Tinsley & Weiss, 2000). One can simply calculate the percentage of agreement between the annotators, called "simple agreement", which is defined as the percentage of annotations that are agreements (Geisler & Swarts, 2019). Simple agreement does not account for the possibility that the annotators' agreement could have been unintentional. Jacob Cohen recognised that some of the experts might select an annotation without being certain it is the correct answer and hence make a random guess. To compensate for this, there are specific metrics that can be used, one of these is Cohen's kappa (κ) (Cohen, 1960). Cohen's kappa (κ) evaluated the chance agreement, using the mathematical formula below:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.1}$$

where p_o is the relative observed agreement between the annotators and p_e is the hypothetical probability of chance agreement. p_e is calculated using the probability of two independent events happening at the same time, i.e. both annotator A and annotator B select the same annotation. Then the probabilities of annotator A and B selecting the same annotation for all available annotations are summed and this is p_e. To use Cohen's kappa to calculate agreement some assumptions should be satisfied such as that the categories/labels of the nominal scale should be independent, mutually exclusive and exhaustive, the judges operate autonomously and the units that are being annotated also independent (Cohen, 1960). These assumptions are all satisfied by the dataset we produced, for example the annotators were blind to each other's judgements ensuring independence and they operated autonomously. Moreover, the labels are of nominal and categorical scale as the labels correspond to levels of mRNA/protein expression, such as "none", "low", "medium" or "high" (nominal) or the labels correspond to the presence or absence of mRNA/protein expression (categorical). These labels are independent too, because there is no overlap between the levels represented by each label. The criterion of exclusivity and exhaustiveness is also satisfied by the

labels as they cover the whole range of possible values of a level. The units are separate from each other and independent as they correspond to data for unique genes.

However, Cohen's kappa applies to two annotators. To calculate the inter-annotator agreement for my dataset and the three annotators answering each question, I decided to follow Light's approach (Light, 1971) for multiple raters. Light's proposal is to calculate Cohen's kappa for all the pairs of combinations of annotators and then take the mean value for the kappa. This method provides an overall index of agreement. Similarly, I calculated the simple agreement between the pairs of annotators and took the mean. Other methods for assessing inter-annotator agreement for more than two annotators, include Fleiss' kappa (Fleiss, 1971). Fleiss' kappa is not suitable for this study as it requires a pool of multiple raters to annotate each question, and for each question the raters should be randomly chosen and therefore not the same for each question. For our dataset and study the same three raters annotate all the images related to the same compartment for the 200 genes. So for the nature of our study the most suitable approach for inter-annotator agreement is Light's.

In Figure 2.11, panel B we can see the simple agreement score and Cohen's kappa score for every question asked for our dataset, to assess the degree to which the annotators consistently assigned expression labels to the figures. The results shown in Figure 2.11 A, show the agreement of the annotators for the annotations presented in Figure 2.8. In Figure 2.11, I present the agreement for the nerve compartment scoring and the CB anlagen tissue scoring as we did score these tissues as well.



Figure 2.11: Agreement between annotators answering questions for every gene and inter-annotator agreement. (A) Heatmap showing the agreement level between the annotators for each question, if it is a complete agreement (all three annotators agree), partial agreement (only two annotators agree) and no agreement. (B) Metrics calculated for agreement of the three annotators per question. All annotators were independent and different combinations of annotators from a pool of six annotators answered the questions.

In panel A we can see a heatmap, in the y axis we have the questions asked and in the x axis each gene. The tile corresponds to the agreement between the annotators. For example, for gene *14-3-3epsilon*, which is the gene that appears in the first row of the heatmap, for the question "CB: Is there protein in anlagen?" the colour of the tile is dark green, which means that all three experts assigned the same answer to the question (complete agreement). If only two annotators gave the same answer (partial agreement) the tile's colour is light green and if none answered in the same way the tile colour is almost white. The three answers each question in the heatmap can take is "Yes", "No", "Non applicable". The "Non applicable" was used when the annotator might have thought that there is something wrong with the image for example the tissue of interest is missing or not identifiable due to the contrasting of the figure when this was created. As mentioned in the previous section, I treated all the annotators as having equal expertise but in reality some annotators might have identified some nervous system structures more easily than others depending on their work or years of experience. We can see that the bulk of the tiles are dark green, and practically all of the others are light green, with only a few almost white squares. In panel B, when looking at the simple agreement index, it is always above 0.50 with very few exceptions and in most of the cases it reaches above 0.70. The Cohen's kappa index for every question is most often between 0.30 and 0.59 indicating fair and moderate agreement according to Table 2.2. These ratings were deemed as adequate for use in this study in particular because in cases where the disagreement occurred a fourth expert was employed to resolve the disagreement. In cases where the kappa was very low (< 0.20) the individual kappa's were assessed and it was seen that it was one of the three annotators that had a higher rate of disagreement with the other two so the average kappa was lowered and not because all three pairs had low agreement. This is likely caused as one of the experts might have misinterpreted the question/did not recognise the structures well. I expected this to happen as I did not want to prepare the annotators with guidance on how to score so that the annotators scoring was impartial.

Table 2.2: Interpretation of the kappa statistic: Labels assigned to ranges of kappa statistic to describe the relative strength of the agreement

Kappa Statistic	Strength of Agreement
<0.00	Poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost Perfect

After performing an inter annotator agreement analysis I am more confident about the validity of our annotations. In cases where the Cohen's kappa is low and therefore indicating that part of the annotations are not correct as there is disagreement I assessed the individual annotations with the help of a fourth expert as mentioned earlier. Additionally when there was partial agreement (two annotators agreed) but the Cohen's kappa is low, I checked the pairwise kappa values. As noted earlier, some pairs of annotators had high agreement and others not resulting in a lower kappa, indicating that the raters determining the final label (the two raters that agree) highly agree. These two assessments show that although some of the initial annotations might be wrong the majority vote or the fourth annotator return the correct vote. Therefore I treat the labels as ground-truth as the variance was in acceptable levels.

Another aspect of the validity of our annotations I explored, is the robustness of the annotations to perturbations, as described in section 2.2.3. The robustness is different for each question and I present the complete results in Appendix B. Taking two extreme scenarios, the most robust-to-perturbations question is the "OL:Is RNA present in soma?" as for 147/200 gene annotations the classification for each gene stays the same even with small perturbations. For 46 annotations there is a 22.5 % probability of the class of the annotation to change to a wrong class. For the 7 remaining gene annotations that there was no initial consensus, consensus is achieved if there is a small perturbation, with a 33.3% probability that is the right class. As a conclusion there is an expected value of 13 genes out of the 200 to be classified wrongly with small perturbations to the annotations. The least robust question is the "CB:Is RNA present in other cells?" for which there is an expected value of 41.5 genes out of the 200 to be assigned the wrong class when small

perturbations happen during annotating the entirety of the 200 genes. The rest of the questions have robustness values evenly distributed between the two extreme cases. This analysis assumed that for each question, a perturbation happens to all annotations. Given that the frequency of the perturbations will be lower, I conclude that the annotations are robust and acceptable in this study.

2.3.5 Can the produced annotations be used as supervised machine learning predictive labels?

Most of the machine learning algorithms used today are based on supervised learning; this type of learning requires input labelled training data in order for the algorithm to learn and be predictive. Labelling training data is heavily reliant on humans or experts labelling them (Monarch & Manning, 2021). In the context of machine learning, "annotation" is the process of labelling raw data so that it becomes training data for machine learning (Monarch & Manning, 2021). It is the most time consuming part of a machine learning pipeline and it takes even more time than building the machine learning model.

Likewise the lack of labels to train predictive models, both classical machine learning algorithms such as RFs (Breiman, 2001) or SVMs (Cortes & Vapnik, 1995) and deep NNs (Goodfellow et al., 2016) has hindered the great progress that the field of machine learning could have achieved in biology. Biology is a data-rich discipline, but the data are unstructured, often difficult to understand and categorise. Hence machine learning techniques may be particularly good at resolving biological problems (Ching et al., 2018).

Research to infer gene expression of target genes based on gene expression of landmark genes exists; that is using expression data from micro arrays (Y. Chen et al., 2016; Ching et al., 2018). Additionally Imbert et al. (2022) produced and used FISH-quant v2, a tool for smFISH image analysis to analyse and classify the expression patterns of single HeLa cells. The HeLa cells were also manually scored so that they could be used as a usecase to test the software.

Thus, the sets of annotations I created for mRNA and protein expression can be used as machine learning labels and are a significant contribution to the scientific community. They have a double impact; they are labels that can be used in association with the microscopy image

corresponding to the gene, so that an algorithm can learn image features related to gene expression or mRNA localisation. Also they can be utilised in prediction of gene expression in affiliation with other features like structural or physical gene features, without using an image as each label is linked to a gene. They are enabling the use of supervised machine learning in transcriptomics and proteomics research, as they can train tools such as FISH-quant v2 (Imbert et al., 2022) to recognise expression patterns and improve future tools available to use.

2.3.6 An adaptable pipeline for assessing gene expression in the *Drosophila* nervous system via simultaneous data visualisation and data exploration that can be generalised and also be used for other purposes

In this section, I further relate and interlink every part of the work already presented to each other. I also show the data acquisition, annotation, analysis and visualisation complete workflow and explain how it can be used and be generalised to other projects once the data are in the right format (Figure 2.12). In the end, I discuss the great value of having and using the Zegami instance.

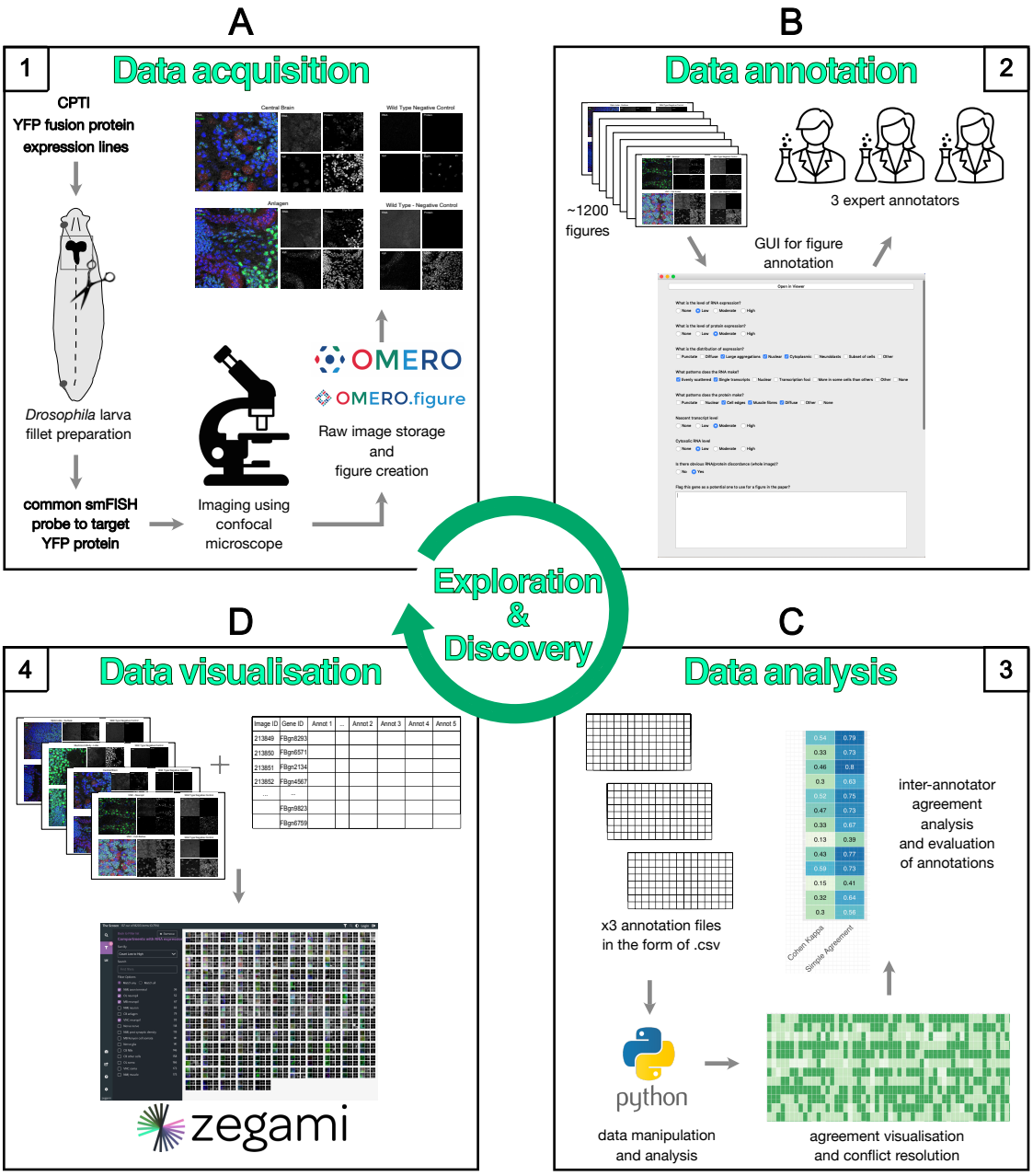


Figure 2.12: Overview of the complete workflow followed to acquire, annotate, analyse and visualise the data. The numbers on each of the panels show the direction of the workflow and the sequence of the procedures (A) Preparing the *Drosophila* larva fillets corresponding to different CPTI lines and capturing the expression using a confocal. Managing the raw data and creating figures using OMERO and by utilising the OMERO web applications. (B) Three expert scorers annotating each of the produced figures using the Annotate.OMERO.Fig application. (C) Using python I manipulated the data generated during step 2 and created heatmaps for the conflict between the annotators' answers and the inter-annotator agreement. (D) Finalised figures and annotations visualised in Zegami.

The workflow consists of four stages:

1. Data acquisition (Figure 2.12 A);
2. Data annotation (Figure 2.12 B);
3. Data analysis (Figure 2.12 C);
4. Data visualisation (Figure 2.12 D).

For the data acquisition part (Figure 2.12 A), in this project, we imaged *Drosophila* brains using a confocal microscope. We then uploaded our microscopy data to our locally hosted OMERO (Allan et al., 2012) server. The microscopy data were compatible with OMERO. We then used OMERO.figure web application to produce figures, and these were exported as PDFs. We locally converted them into jpegs. If another researcher would like to adopt our pipeline for a different project they could very easily image another organism or another part of the *Drosophila* larva and upload and manage their images using OMERO (Allan et al., 2012) as it is an open source tool. Alternatively, if their project is entirely different from ours, they could prepare images in the form of jpgs which they can feed into our annotation tool, as shown in Figure 2.12 B.

The second part is the "Data annotation" part, as seen in Figure 2.12 B. Here, we annotated approximately 1,200 images using the help of three expert annotators. The annotators used the application Annotate.OMERO.Fig (Kiourlappou et al., 2022) to score the figures. The application is accessible on GitHub (`https://github.com/mkiourlappou/Annotate.OMERO.Fig`) with a toy dataset for a user to experiment with and clear instructions on how to set up and run. As explained earlier, the application is fully customisable thus an external user can change the questions and answers for scoring depending on their research and the images they would like to score.

The next stage is the data analysis of the annotations (Figure 2.12 C). Once the annotation process was finalised, I analysed the scoring to see the agreement between the scorers. The python code I wrote to do that is freely available on my GitHub repository (`https://github.com/mkiourlappou/thesis-project/blob/main/Inter-Annotator-Agreement.ipynb`) in the form of a Jupyter Notebook (Loizides & Schmidt, 2016). The functions I wrote to do the analysis

and produce the heatmaps are using the output data produced from the Annotate.OMERO.Fig application and thus, are widely applicable. Likewise the Jupyter Notebook with the python code to resolve conflict using majority voting is also available on the same GitHub repository (`https://github.com/mkiourlappou/thesis-project/blob/main/Resolving_conflict_pipeline.ipynb`). The output is a csv file with just one finalised annotation (instead of three) corresponding to each question. The files are available in the same repository in the scoringResults folder (`https://github.com/mkiourlappou/thesis-project/tree/main/scoringResults`). Multiple annotations correspond and describe each figure.

Finally after the data analysis is completed the data, images and annotations, are uploaded in Zegami (Figure 2.12 D). The Zegami instance I created is a collection that contains not only the annotations produced for each gene screened, but also physical properties of the gene, GO annotations (The Gene Ontology Consortium et al., 2021), expression data and links to online bioinformatics libraries like FlyBase which are useful and rich-in-information resources regarding each gene. I also provide the additional information not only for the screened genes but also for the rest of the genome to build a simple knowledge base for the *Drosophila* genome.

One of the discoveries made using the produced Zegami instance, which I discussed briefly in section 2.2.4 is that 137 genes out of the collection of 200 have mRNA expressed in at least one of their synaptic compartments. This is also shown in Figure 2.5 D, where I filtered for the synaptic regions (NMJ axon terminal, OL neuropil, MB neuropil and VNC neuropil) and I can easily see the figures that correspond to the genes with the expression and examine them all together or in detail one by one. These synaptic regions are close to the periphery of the cells and this is a very important finding which further provides insight into the mechanisms of post-transcriptional regulation.

The Zegami collection I created used to be publicly available and accessible through a uniform resource locator (URL) for anyone to browse until November 2022. Since then, the Zegami company has entered liquidation. The company however is working on a solution to provide the Zegami software in a new form and to create an open source solution. During this time we have been looking into alternative solutions and found the MDV solution (COSMIC – COVID-19

Lung Single Cell Mass Cytometry Imaging Consortium STUDY et al., 2022) an open source resource which provides multiple views for tabular data such as table views, bar plots, heatmaps and more. It can also associate the tabular data with images and the user can browse them in a similar fashion to Zegami but more limited in terms of the zoom in and out functionality. We have implemented our collection of images and associated metadata in MDV and this collection can be accessed through this link: `https://mdv.molbiol.ox.ac.uk/projects/mdv_project/7012?view=RNA+%2F+Protein+Distribution`. We created a user manual on how to explore our collection in MDV and is available at the Zenodo repository referenced here (Kiourlappou et al., 2023). Figures showing how the data can be interrogated to gain insights along the microscopy figures, are shown in Appendix C.

2.4 Summary and Conclusions

In this Chapter I described the automated workflow I produced for image acquisition, figure creation, data annotation, visualisation and multi-type data integration, for our intermediate-throughput microscopy imaging screen of the *Drosophila* larval nervous system.

We selected 200 *Drosophila* lines from the CPTI YFP fusion collection and created smFISH probes against the YFP sequence to image protein and mRNA expression in the larval brain. We imaged seven regions of interest and examined tissues of interest within the compartments. We produced more than 1300 figures for the 200 genes to highlight the distribution and levels of mRNA and protein expression.

I created software for annotating these images and three experts from the lab answered predefined questions to score the images for presence and absence of single molecules of mRNA or protein. I used majority voting to reach consensus and inter-annotator agreement metrics to evaluate the validity and variance of the annotations. I generally found fair to moderate agreement between the raters for most of the questions and in cases where the agreement was poorer I assessed the individual rater reliability and revisited disagreements.

2. Chapter 2

I summarised the annotations for the compartments and regions of interest and found that discordance between mRNA and protein expression is very common amongst the 200 genes; only 5 genes out of the 200 show correlation between the distribution of mRNA and the encoded protein in all regions of interest examined. The rest of the screened genes show discordance in at least one compartment of interest and that is a common sign of post-transcriptional control. Moreover, I noticed significant and exciting localisation patterns in glial cells; this adds to the scientific understanding regarding glial cells in *Drosophila* as this is an area that required much further study. I found that 18 genes out of the 200 show patterns of mRNA and protein localisation in peripheral glia in the NMJ tissue.

I constructed a data set containing the annotations for each figure showing the presence or absence of mRNA and/or protein single molecules per gene, and I added bioinformatics information such as GO terms, physical structure and mRNA expression information. I also added links for every gene to FlyBase (Gramates et al., 2022), a repository of genetic and molecular data for *Drosophila*. I added this type of information for the whole genome and not just for the 200 genes screened to extend the utility of the dataset and make it a useful resource for those who wanted to search information about genes in the rest of the genome or genes with known protein trap insertions.

Finally I created a Zegami instance, using the Zegami software containing our ~1,400 figures and the dataset I constructed. This enables us and any user who would like to explore the instance, to search our large dataset seamlessly, fast, efficiently and gain high level insight. Additionally this useful resource allows for any findings of interest to be projected and help predict genes with analogous expression profiles and phenotypes in the rest of the genome.

The products of this Chapter are valuable in many ways; firstly because I produced a user-friendly browsable information-rich resource for the expression patterns of mRNAs and their encoded proteins in the *Drosophila* larval nervous system. Secondly, I created an adaptable and generalisable method for similar screens that can be used by other laboratories to present and interrogate their data in a similar fashion.

3

Exploration of datasets for mRNA localisation prediction in glia

In Chapter 2, I explored the transcriptomic and proteomic profiles of 200 genes in the *Drosophila* nervous system. I did that by creating a novel pipeline for acquiring microscopy images in the form of raw data for those 200 genes, managing the huge volume of the raw data using online repositories, creating figures in a systematic way and annotating the figures using bespoke software I contributed to producing. Part of the pipeline I present, is the novel visualisation of the figures together with the annotations expert scorers from the Davis lab produced. Resulting from this pipeline are the annotations for the 200 genes which are used as labels of mRNA and protein presence and absence. I generated sets of labels for six compartments and twelve tissues of interest.

In this Chapter, I focus on one tissue of interest, the peripheral glial tissue in *Drosophila*. The aim of this chapter is to explore and combine big datasets found in literature that are related to mRNA localisation in glia and make the heterogeneous datasets stemming from various sources interoperable. Additionally, in every section of the chapter I explain how different aspects of the complex and diverse datasets could give rise to data science and machine learning related complications, as machine learning is used in the next Chapter to predict candidate genes that have localisation potential in glia.

Equipped with the set of annotations describing mRNA presence and absence in peripheral glial cells for the 200 genes of interest, I divide the genes into two classes, creating predefined labels for the 200 genes. I discuss potential implications in my goal to predict localisation as a pool of 200 genes with only 18 localised is a small sample to be used for machine learning training purposes. Subsequently, I also examine genes that are found to be localised in the periphery of glial cells in mice. Glial transcript localisation has been studied more in mammals and there are available RNA-seq datasets characterising the localised transcriptomes and translatomes. Additionally, I investigate the relationship between the *Drosophila* 200 genes and the mammalian genes examined.

I provide a parallel, genome-wide study of datasets that could provide insight into mRNA localisation for both the fruit fly and the laboratory mouse. I discuss how I identified, acquired and explored the datasets and managed any missing values, how I feature-engineered and encoded the datasets and understood correlations between the features. The datasets I used include the GO terms, the protein families (Pfam) dataset, RNA binding protein information, mRNA half-life (stability) measurements, structural and physical gene information, secondary structure information, REACTOME pathway annotations and more.

3.1 Introduction

A significant challenge that modern day biology faces is managing the multitude and complexity of data types, the hierarchical nature of the field, and the requirement to gather data through a diverse range of methods. This has resulted in the need for centralised repositories to store, manage and access those data. Previously discussed is the difficulty of managing data arising from image-based genome-wide 'high-content' screens at a lab-scale. Managing image-based data together with other types of data, mostly heterogeneous, at a global scale, making them interoperable and accessible, is an even greater challenge.

The online available repositories come in many guises; some are focused on one organism only such as Flybase (Gramates et al., 2022) and the MGI Database (Blake et al., 2021). These are dedicated to the fruit fly and the laboratory mouse organisms respectively. Others are not dedicated

to one organism only, such as the GO (Ashburner et al., 2000) provides unified information across many species. Flybase is a repository integrating genomic data, expression data, RNA seq data, microarray data and phenotypic data. The MGI database is a similar database to Flybase but for mouse and it is the community model organism knowledge base. GO is a repository that provides an ontology of defined terms mapping genes to molecular functions, biological processes and cellular components. ENSEMBL (Kinsella et al., 2011), on the other hand is a genome browser and a database which contains information from Flybase, MGI, GO and more databases, both species specific and species agnostic.

One other big difference is the way datasets are structured and how repositories encompass the hierarchy of those datasets. A way that biological databases are organised to highlight hierarchy between values of datasets are in the form of ontologies. Biological ontologies define basic terms and relations in biological domains and they are effectively a vocabulary. Providing such a large and broad vocabulary in the form of a flat list is not very efficient and easy to browse (Staab & Studer, 2004). Therefore the "words" of the vocabulary or the concepts and the relations between them are organised into hierarchies. A hierarchy is a sort of taxonomy with parent terms and child terms that are related between them.

A prime example of a biological database that is organised as ontology is the Gene Ontology Annotation (GOA) database (Ashburner et al., 2000). GO is organised in the form of a directed acyclic graph (DAG) (Hill et al., 2002; Lord et al., 2002; Thulasiraman & Swamy, 1992). It is composed of three biological domains, BP, MF and CC as defined in Chapter 2. These all are attributes of genes or gene products (Ashburner et al., 2000). Each node in the DAG refers to a GO term and the edges between the nodes refer to the relations between the terms. A GO term closer to the root of the graph is referred to as a parent term and the GO terms closer to the leaf nodes as child terms. The closer the GO term is to the root of the graph, the more GO terms it has as child terms and the higher up it is in the hierarchy of the terms and the more general it is.

REACTOME is another open-source taxonomic information library available online (Jassal et al., 2019; Sidiropoulos et al., 2017) providing access to a knowledge base of biomolecular

pathways. It contains molecular information for cellular processes such as signal transduction, transport, DNA replication and metabolism.

In addition to the diversity, the heterogeneity of the data itself and the model organism they focus on, the repositories also differ in the options given for acquiring the data. There are many ways to access the data in these databases and these include web applications, python or R packages, APIs or other frameworks that integrate heterogeneous datasets together. For example FlyMine (Lyne et al., 2007) and MouseMine (Motenko et al., 2015), both under the umbrella of InterMine (R. N. Smith et al., 2012) are powered by the InterMine framework and populated by Flybase and MGI respectively. They facilitate the building of biological databases as they integrate together complex datasets. A user can access data from these repositories by the web application or via the RESTful web services. There are python queries that can be produced from the InterMine website for accessing FlyMine and MouseMine but the user needs to visit the website and produce the query or learn the specific syntax query.

In this Chapter, I aimed to understand how data are structured in all the aforementioned repositories (and more) and access and acquire these data. I did that for both *Drosophila* and *Mus* data. I then explored the data, converted datasets from transcript-centric to gene-centric where appropriate and feature engineered them to produce features. Through the literature and other Davis lab members' collated RNA-seq datasets of glial localisation in the mouse brain I produced labels for glial localisation in mice.

3.1.1 Aims and Objectives

1. Explore the huge amount of datasets found in online biological repositories, understand the structure of different datasets and their organisation, acquire the data and make them interoperable.
2. Prepare predefined labels, collect and prepare datasets with relevancy to mRNA localisation and encode them, to make them suitable for machine learning algorithm training.

3. Gain an understanding of how typical machine learning project related issues arise in my datasets from a data science point of view so that I can develop a reliable machine learning model capable of predicting mRNA localisation.

3.2 Materials and Methods

3.2.1 Labels creation

In Chapter 2, with the help of expert annotators we produced multiple sets of labels that describe mRNAs and their encoded protein expression patterns for selected 200 genes. We discussed the high quality of these labels and their suitability for being used as machine learning labels to train supervised predictive algorithms. However, the selection of labels I wish to work with to define the presence/absence of transcripts in peripheral glia includes 18 genes with mRNA presence and 182 genes with the transcript absent. In machine learning terms and in binary classification, there are 18 positive labels (9% of the dataset) and 182 negative labels (91% of the dataset). This class ratio suggests an imbalanced data distribution. The issue with imbalanced data distribution appears when the number of significant data occurrences is small (Kumar et al., 2021) which in this case holds true. When there are imbalanced labels, the majority classes are strongly represented, hence there is a strong bias towards them. This results in a strong inclination for the model to predict the majority class thus reducing the models capacity to predict the minority class. There are methods to counterbalance this when the labels are used in machine learning prediction and the main one is data sampling. Random oversampling (from the minority class) and under sampling (from the majority class) aims to balance the skewed data distribution (Kumar et al., 2021).

Additionally to using this set of labels to predict mRNA presence in peripheral glia together with techniques to compensate for this, I decided to produce another set of labels using mammalian genes that are known to be localised in glial protrusions. The mammalian dataset is far more balanced and larger, in the order of thousands, so this should eliminate any problems arising in the future due to the class imbalance and the small size of the *Drosophila* dataset. Knowledge of mRNA expression in glial cells in mammals is significantly greater compared to knowledge

of mRNA localisation in *Drosophila*, where very little is known about localisation in polarised *Drosophila* cells. To address this disparity and to elucidate this, Dr Jeff Lee and Dalia Gala, members of the Davis lab, identified (to their knowledge) all the published mammalian glial peripheral or synaptic transcriptomes and translatomes (Gala et al., 2022). These are twelve datasets, eleven datasets for the model organism *Mus musculus* and one for the organism *Rattus norvegicus*, the full list of the datasets is given in Appendix D. Dr Jeff Lee developed an RNA-seq analysis pipeline and processed the transcriptomic datasets in such a way so that they are comparable. Using ENSEMBL BioMart annotations (Kinsella et al., 2011) *R. norvegicus* genes were transformed into *M. musculus* genes. More details on how the RNA-seq analysis works, can be found in (Gala et al., 2022).

I used Dr Jeff Lee's RNA-seq analysis results to identify transcripts that are present in glial protrusions. In his analysis results, there is information about each transcript for the full transcriptome (the complete set of all RNA molecules in glial protrusions in this case), such as how many datasets out of the twelve provide evidence for the transcript to be present in glial protrusions. If the transcript is present in more than one study, the mean transcript per million (TPM) count is calculated over all the studies. Figure 3.1 shows box plot distributions for the mean TPM counts over all the transcripts grouped by the number of studies they appear in, in logarithmic scale. Outliers are shown in red. For genes that are supported by zero, one, two or three studies the median of the box plot is below 10 mean TPM and there are outliers that have mean TPM lower than 1. However for genes that are supported by at least four studies the median is above 10 mean TPM and there are no outliers with TPM lower than 1. Based on these observations I decided to filter genes that have mean TPM greater than 10 and are supported by at least 4 studies as genes that their transcript is considered present in glial protrusions (cut-off shown in Figure 3.1 with blue dotted lines). The filtering produced a set of 7,583 genes.

Additionally, I examined if including mouse orthologs of *R. norvegicus* genes diminishes the quality of the produced labels. I did that by finding the number of *R. norvegicus* genes with their transcripts in glial protrusions having a mean TPM above 10. I used that cutoff as it is the same

criterion I used to filter the genes earlier. I found that there are 6,743 *R. norvegicus* genes that satisfy this criterion. Out of those, 5,696 have *M. musculus* orthologs, which is approximately 15% less genes. This is likely due to the fact that the *R. norvegicus* genome is less well studied and annotated compared to the *M. musculus* genome. Therefore, there are 15% less genes in one of the 12 datasets taken into consideration when finalising the gene labels. I conclude that the number of genes that are present in protrusions are slightly undercounted compared to having 12 *M. musculus* datasets as a starting point versus 11 *M. musculus* datasets and 1 *R. norvegicus* dataset. Nonetheless, when I analyse and compare all 12 datasets to finalise the gene labels, I estimate that this underrepresentation is less than 15%. This is because genes that are missing from the rat dataset could be present in any of the other 11 datasets, therefore still supported by the studies and likely passing the required threshold.

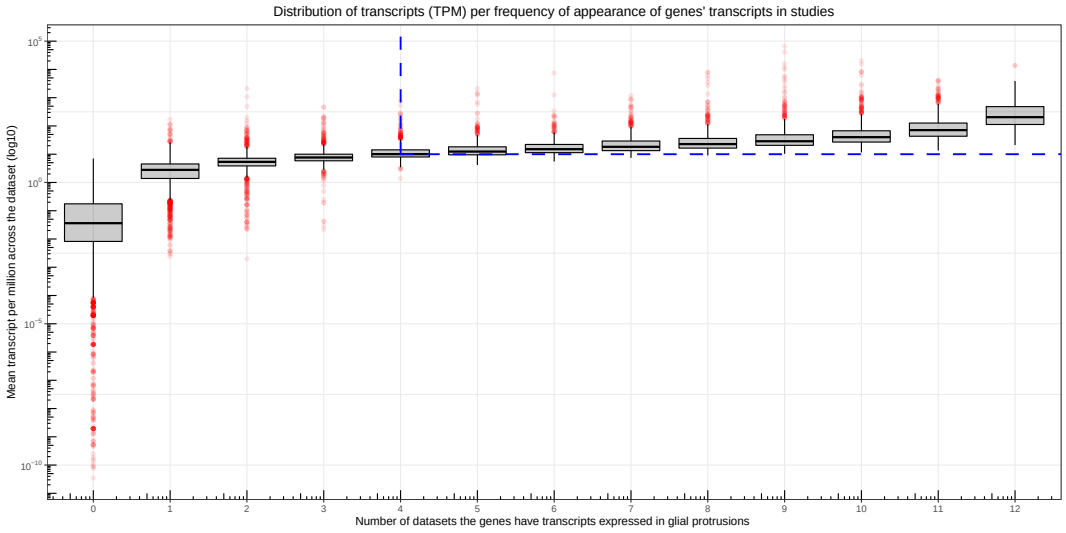


Figure 3.1: Box plot distributions of the mean transcripts per million (TPM), grouped by the amount of datasets the gene's transcript is supported to be localised in glial protrusions. The x axis shows the number of datasets that support that the mRNA of a specific gene is localised in glial protrusions. There are 12 datasets in total supporting this study, and 11 concern the model organism *Mus musculus* and 1 the organism *Rattus norvegicus*. The mean TPM for each gene's transcript, found when taking the mean of the TPM of the transcript in each study is shown as a box plot distribution over all genes that appear in the *n* number of datasets. The outliers are shown in red. The blue dotted lines are showing the cut-off to which I chose to use to select transcripts and label them as localised. This is when a gene's transcripts has at least 4 supporting datasets and when it has at least the value of TPM is at least.

To summarise, the method I used to select the localised genes was based on the meta-analysis described above and shown in Figure 3.1. The numbers of the localised/non localised genes will

be presented in the first subsection of the Results and Discussion section.

Next, I wanted to compare the mammalian glial localisation data found in literature with the smFISH *Drosophila* data produced by Davis lab, to examine the potential overlap of mammalian to fly genes that are considered localised in glia. To do so, I found the *Drosophila* genes' orthologs of the mouse genes. I converted the mouse to fly orthologs using the DIOPT Ortholog Finder version 9 (Hu et al., 2011 ; Hu et al., 2021). To find orthologs with a high degree of confidence, I only considered orthologs with a DIOPT score 8 or higher. Some fly genes map to more than one mouse gene and vice versa, I kept only the ortholog with the highest DIOPT score available (max DIOPT score was 18) to produce a one-to-one relationship and simplify the mapping. I then mapped the whole mouse genome using the selection of the orthologs to fly genes, which produced an *n* number of genes. Finally I compared this subset with the 200 screened *D. melanogaster* genes. The results of this analysis are also shown in subsection 3.3.1.

For the purposes of these examinations, the genome assembly version used for *M. musculus* was GRCm39 and for *D. melanogaster* was BDGP6.32, sourced from the ENSEMBL genome assembly version 106 (Cunningham et al., 2022). The genome assemblies versions used for the individual species are the same in ENSEMBL genome assembly version 104 which was used in earlier work described in Chapter 2. The data manipulation performed above and in the rest of the chapter is all done using Python (Rossum & Drake, 2010) and the Python package Pandas (McKinney, 2010).

3.2.2 Identifying and accessing datasets of interest from online bioinformatics repositories

D. melanogaster and *M. musculus* are both extensively studied model organisms with a large amount of biological data available in openly accessible biological databases and repositories. As discussed in the Introduction, section 1, examples of such databases and repositories include Flybase (Gramates et al., 2022) and the MGI Database (Blake et al., 2021), as well as the GO database (Ashburner et al., 2000). I used ENSEMBL (Kinsella et al., 2011), a genome

browser that provides a centralised resource to genomic information to access other repositories and have a single point of reference.

I used the python package pybiomart (Ruiter, 2016) to gather together information from ENSEMBL and Pandas (McKinney, 2010) for data manipulation (Figure 3.2 B). Pybiomart retrieves and queries ENSEMBL datasets using the server interface. I chose ENSEMBL and pybiomart as my main method of accessing biological and bioinformatics libraries, as pybiomart facilitates direct access of many genomes with the change of just one line of code in python where I specify the species I want. Moreover compared to other packages, the commands are very simple to use and can directly pull data into Pandas dataframes which are easy to manipulate. Multiple dataframes can be merged together to create a bigger gene-centric biological dataset. Additionally, these data can be fetched in real-time allowing for future users of my work to rebuild these databases without the costly and difficult transfer of the large datasets involved through data storage repositories. Very recently published research (Luebbert & Pachter, 2023) also highlights the usefulness of the pybiomart package in making genomic data more accessible through python with minimal coding required.

Through ENSEMBL, I was able to access Pfam (Mistry et al., 2021) information for every protein corresponding to every gene, in both the fly and mouse genomes (Figure 3.2 B). The Pfam database is currently hosted by the InterPro database (Paysan-Lafosse et al., 2023), a bioinformatics resource which not only categorises protein sequences into families but also predicts the presence of valuable domains in proteins and conserved sites. In like manner via ENSEMBL, I also retrieved the sequence of every gene for both model organisms, for the coding region (if protein-coding), for the 3' UTR and for the 5' UTR.

Furthermore, I successfully accessed the GO database using pybiomart, retrieving how many GO terms are related to each gene. As there is inherent hierarchy built into the structure of GO I will elaborate on the retrieved dataset and related information in the GO dedicated subsection.

As my focus is to try to find correlations between datasets that are related to mRNA localisation in protrusions of glial cells with the genes that I identified as localised, I mainly look for datasets

that can be related to post-transcriptional regulation and localised translation. The datasets retrieved from ENSEMBL are expected to be linked with post-transcriptional regulation as are RNA binding proteins (RBPs) related datasets. RBPs are key regulators of post-transcriptional regulation and localised translation (Payne et al., 2018). ENSEMBL does not provide information for RBPs and so I had to access the oRNAment (o RNA motifs enrichment in complete transcriptomes) database (Benoit Bouvrette et al., 2019) directly and download the source data manually, for both species (Figure 3.2 A). I then used Pandas to manipulate the dataset and integrate it with my bioinformatics library retrieved from ENSEMBL. oRNAment provides information in a transcript-centric way, by giving all the putative RBP binding sites per transcript. oRNAment contains data for 223 distinct RBPs produced by using either the RNAcompete (Ray et al., 2013) or the RNA Bind-n-Seq (Lambert et al., 2014) experimental methods. The data result to 453 binding site motifs and they are presented in the form of position weight matrices. Position weight matrices show the probability of finding a certain nucleotide at each position within the motif. Both RNAcompete (Ray et al., 2013) and RNA Bind-n-Seq (Lambert et al., 2014) are methods that characterise the sequence and structural specificity of RBPs and analyse RNA-protein interactions.

Finding pathways that can group together genes expected to be localised and genes from the rest of the genome where there is no obvious expectation can provide insights into the mechanisms that mediate the regulation of those genes as well as confirm if pathway terms such as mRNA translation and mRNA stability regulation are enriched in the groups of genes that are localised. This information is contained within the REACTOME, a taxonomic information library available online (Jassal et al., 2019; Sidiropoulos et al., 2017). To access the REACTOME pathways for both species, I had to query FlyMine and MouseMine through the InterMine framework to find the pathways associated with each species (Figure 3.2 C). In addition I exported the REACTOME pathways network of relations between the annotators through the REACTOME web application so that I can group pathways to the hierarchy level I chose.

Finally, using the MGI Database (Blake et al., 2021) phenotype search, I obtained lists of genes that have abnormal glial differentiation or morphology in different types of glia in the mouse

brain. These lists are curated by MGI from different studies found in the literature. The size of the gene sets is relatively small (in the range of a few hundreds) however these gene collections are informative so I decided to include them in my database. Such lists do not exist for flies as *Drosophila* glial cell research is much more limited compared to mammalian glial cell research.

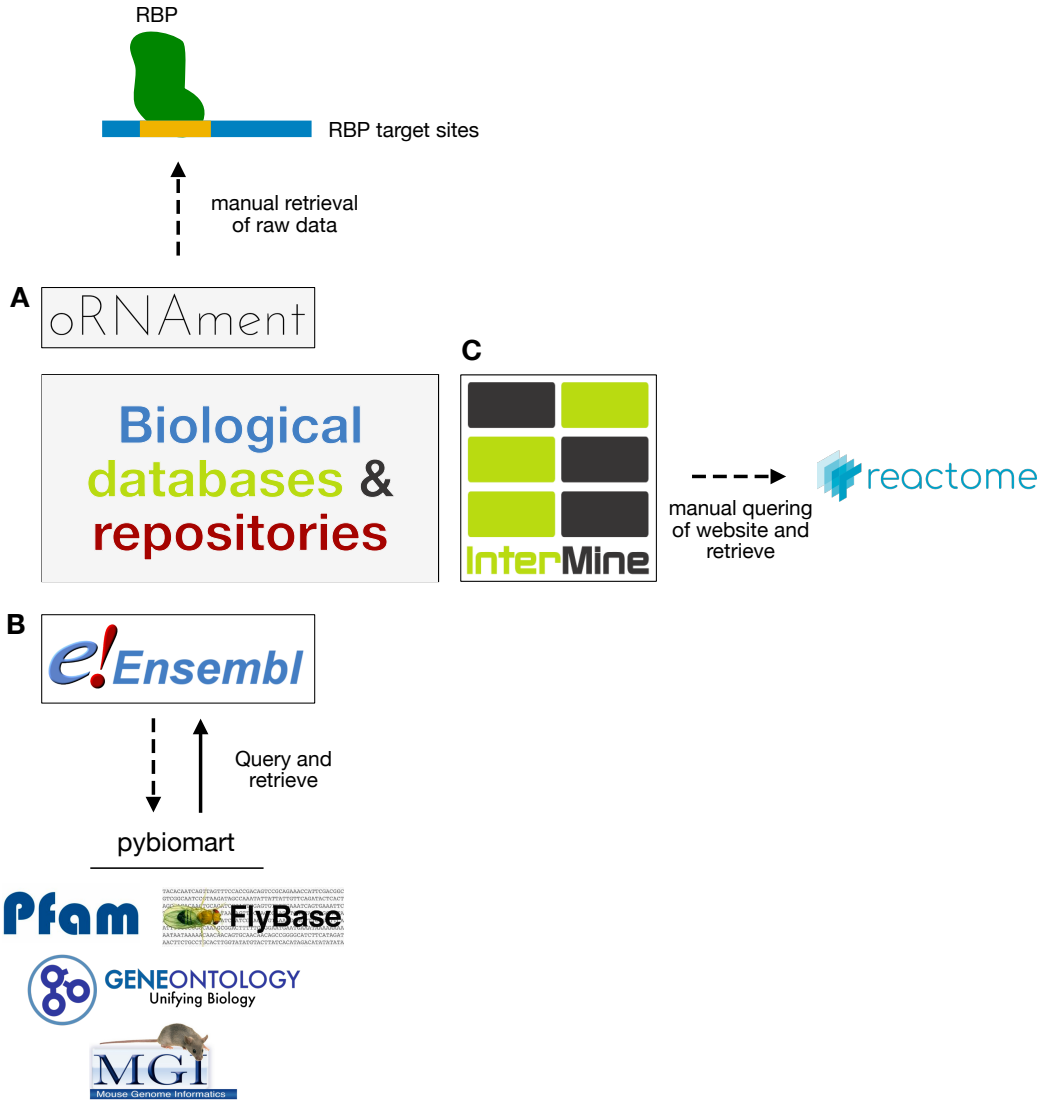


Figure 3.2: Overview of the ways I accessed each public bioinformatics repository and the databases I retrieved. (A) The oRNAment database was accessed manually. The putative RNA binding protein target site raw data were downloaded for both fruit fly and mouse. (B) I used the python package pybiomart to access ENSEMBL, through querying the genome browser and retrieving the datasets requested. These include datasets found in the repositories Flybase, MGI, GO Ontology and Pfam, which contain physical, structural functional and protein classification information. (C) Important information such as REACTOME pathways information not being able to be retrieved using pybiomart (B) was accessed through manually querying InterMine.

The ENSEMBL and other biological repositories contain numerous layers of data combinations. Information can be categorised according to different entities such as gene, transcript, GO term, or protein. However, comprehending the complex interplay between these different forms of data and transforming them to create a gene-centric database that makes biological sense posed a significant challenge in this project. It involves the integration of data from different sources to create a comprehensive picture of each gene. This process was time-consuming and challenging, as the data came from multiple repositories and required extensive curation to ensure accuracy and consistency.

3.2.3 The Gene Ontology Annotations (GOA) database

The GOA (Ashburner et al., 2000) database is a resource that provides information on the functional characteristics of genes across different organisms. GO is structured as a DAG (Hill et al., 2002; Lord et al., 2002; Thulasiraman & Swamy, 1992) comprising the three biological domains mentioned above (BP, MF, CC) which are all attributes of genes or gene products (Ashburner et al., 2000). Each node in the DAG represents a GO term, and the edges between nodes denote the relations between the terms. GO terms closer to the root of the graph are parent terms, while those closer to the leaf nodes are child terms. Terms closer to the root have more child terms and are higher up in the hierarchy, indicating a broader scope.

Using the ENSEMBL release version 106, for *Drosophila* there are more than 9,500 GO terms and for the laboratory mouse *M. musculus* more than 19,000 GO terms assigned to their respective genomes. It has to be noted that the GO terms annotation coverage is not complete and it constantly changes and improves (Huntley et al., 2014). Moreover the difference in the organism coverage is obvious from the number given earlier but this could also be related to the much smaller genome of *D. melanogaster* compared to *M. musculus*. Each gene, transcript or protein can be assigned multiple GO terms describing it. The fewer edges the GO terms (nodes) have separating them, the closer they are in meaning and the more genes or gene products they share. Additionally, the genes or gene products that have been attributed one child GO term, by definition are also attributed the parent GO term of the child GO term. This is because GO

follows the "true-path-rule" which means if a gene is described by a GO term it has to also be correctly described by the ancestor terms as well (Huntley et al., 2014).

Since the creation of the GO annotation database, GO terms have been used in many ways. This includes GO enrichment analysis, the identification of GO terms that are significantly overrepresented in a target group of genes against a background group of genes (Khatri & Draghici, 2005) or the identification of lists of genes related to a biological question via the GO terms annotations. However there are a few problems identified with the GO annotation database and its hierarchical structure. Some GO terms can be very similar to each other or some child GO terms to their parent GO terms. Additionally as the annotation tree is following the "true-path-rule" all the descendant terms of the parent term map to genes that are subsets of the original parent GO term to gene mapping. This introduces a high level of redundancy in the GO terms with many of those overlapping and introducing little additional information. To tackle this and to make sure analysis tools using GO terms are providing insightful information, similarity measures can be used to group together similar GO terms (Mazandu et al., 2016). Most prominent is the use of semantic similarity which measures the likeness between two concepts based on the distance between them in the geometric space (in the ontology tree) (W.-N. Lee et al., 2008). Deploying semantic similarity measures for grouping together GO terms has helped in deploying GO terms in more areas of biological discovery such as predicting and validating cellular localisation (Lei & Dai, 2006; Mazandu et al., 2016), protein–protein interactions (Jain & Bader, 2010) and protein functions (Ye et al., 2005), as pointed out in the survey by Mazandu et al. (2016). Additionally similarity has been used to impute missing values in micro arrays experiments based on expected similarity which correlates with semantic similarity of GO terms (Yang et al., 2016).

In my project, one of the datasets I use to produce features and to train my predictive algorithm are the GO terms. To be able to do so, I have to engineer the GO terms into features. In section 3.2.4, "Data preprocessing and feature engineering", I describe in more detail the concept of featurization for each of the datasets, and more information will be given regarding the use of GO terms as well. However, as the GO terms datasets come in the form of a complex

hierarchy I elaborate here on the structure and explain the need for grouping together GO terms and reducing their number.

Each of the three GO domains: BP GO terms, MF GO terms and CC GO terms can be a feature and selected GO terms from each domain the values of the categorical features. As GO terms are commonly strings and comprise categories, the feature type should be categorical. Each gene gets assigned one or more of these values which are the selected GO terms or sets of GO terms grouped together. As previously stated there are almost ten thousand terms describing fruit fly genes and gene products and almost twenty thousand terms describing mouse genes and gene products. If these were to be converted directly to categorical features, the values that the feature could take would be very large and some feature values would correspond to only one gene from the genome. Additionally, there would be the possibility that the GO terms that were feature values for the genes used as training genes would be present only in the training phase and not in the testing/validating phase. That would cause the model to predict on completely new feature values that it has not been trained on and this could affect the model predictions as the genes to be predicted to be localised/not localised would not have similar qualities to the genes that we already know are localised. In machine learning terms the problem of a feature having many possible unique values is defined as "high cardinality".

The GO consortium releases high-level views of each ontology, these terms are known as GO Slims and are available via the GO website (*Guide to Go Subsets*, n.d.). These are subsets of the GO ontology and they are very broad. For *Drosophila* the general set of GO Slims contains 139 GO terms. Compared to the tens of thousands of GO terms annotating the fly genome, 139 used instead as the unique values of the categorical features is a much better condition for training an algorithm. However, these categories are general and they do not provide a lot of insight into how genes are correlated together via the GO annotations. Instead of using GO Slims I decided to use semantic similarity to group together related GO terms. There are many packages developed for grouping similar GO terms together based on semantic similarity, such as GOATOOLS (Klopfenstein et al., 2018), GOGO (Zhao & Wang, 2018) and GOMCL (Wang

et al., 2020). These packages provide ways to calculate semantic similarity between GO terms. GOGO has an online web service where a user can calculate the semantic similarity between two or more GO terms or calculate the semantic similarities between pairs of genes. The source code which is available to use however is written in perl which is not a language I am familiar with. Both GOMCL and GOATOOLS are written in python and open-source available at GitHub, but I used GOATOOLS in the end as it provided comprehensive jupyter notebook (Loizides & Schmidt, 2016) examples and it was easier to implement and introduce in my coding workflow.

The semantic similarity in GOATOOLS is calculated using Lin's similarity (D. Lin, 1998). Lin's similarity is information content (IC) based. IC is defined as the negative log of the probability of the concept (based on the observed frequency counts of the concept), as shown in equation (3.1).

$$IC(term1) = -\log(P(term1)) \tag{3.1}$$

Once this is calculated the Lin's similarity is given by the formula below:

$$sim_{lin} = \frac{2 * IC(lcs(term1, term2))}{IC(term1) + IC(term2)} \tag{3.2}$$

where lcs is the least common subsummer which is the most specific term two terms share as a common ancestor (McInnes & Pedersen, 2013).

In this paragraph, I describe the method I devised to group together GO terms based on Lin's similarity and the genome coverage of the GO terms. The complete pipeline I followed is visualised in Figure 3.3. Sailem et al. (2020) in her work trained GO term classifiers, but because the annotated genes for each GO term vary, she filtered GO terms with the number of genes associated with each term to be in the range of 100 to 500. The number of genes for each GO term depends on the position of the GO term in the annotation tree and how specific the GO term is. I followed a similar approach in filtering the GO terms I used before grouping them together. Like Sailem et al. (2020), I produced GO term per gene distributions and vice versa. Besides producing the GO term per gene distributions, I calculated the GO term genome coverage as my

ultimate goal is to group together GO terms and drop redundant terms while preserving as high genome coverage as possible. I did this for each of the three GO annotations domains.

I define genome coverage as the number of unique genes that have any GO term annotated to them ($n_{genes\,GO\,annotations}$) divided by the number of genes in the genome ($n_{genes\,genome}$), as defined in equation (3.4).

$$Genome\,Coverage = \frac{n_{genes\,GO\,annotations}}{n_{genes\,genome}} \tag{3.3}$$

Likewise, protein-coding genes coverage is defined as the number of unique protein-coding genes that have GO term annotations associated with them over the number of protein-coding genes in the genome, as seen below.

$$Genome\,Coverage\,Protein\,Coding = \frac{n_{protein\,coding\,genes\,GO\,annotations}}{n_{protein\,coding\,genes\,genome}} \tag{3.4}$$

I produced the distribution of how many GO terms have an x number of genes associated with them and I calculated the coverage if only the set of GO terms that have x genes associated with them was included. Step 1 of Figure 3.3, shows an an indicative example, the line plot of cumulative coverage for the CC domain for the protein-coding genes for flies. A bigger more detailed version will be shown in the Results and Discussion section, where I describe it with more detail. In an effort to group together GO terms that individually have low genome coverage, I filter GO terms that have fewer than 200 genes assigned to them. This is supported by the fact that the cumulative coverage line plot has the same slope for up to 200 genes, but the slope changes after that. The x axis is set as a logarithmic axis, which means it linearises exponential growth and hence the stability in the slope for the first 200 genes shows the exponential increase in coverage for those 200 genes. This is followed by a more stable increase in coverage when more GO terms are included. Therefore the GO terms that add to the total coverage this much should be grouped together to be retained and not dropped. This is described in the second step of the process in Figure 3.3.

Reasoning for this approach is the assumption that if a GO term has many genes assigned to them relative to other terms, that means that it is too general. Vice versa, if a GO term has too few terms assigned to it, that means it is too specific. If however many "too specific" GO terms are put together and have a high coverage that means that the GO terms should be retained. Additionally if a "too general" GO term does not add much to the coverage (as other GO terms that are more specific already cover most of the genome), that means this GO term could be dropped.

Next (shown in Figure 3.3 Step 2), I found all possible combinations of GO terms to find all GO term pairs of the selected subset of GO terms. Without limiting the number of GO terms (one step earlier), there are around 300 million possible pairs of GO terms. Following this step, I used the GOATOOLS package (Klopfenstein et al., 2018) to calculate Lin's similarity for every plausible combination of two GO terms. To test the GOATOOLS package and make sure the GOATOOLS package was working as expected, I reproduced some of the examples given in the accompanying to the package jupyter notebooks. A threshold of Lin's similarity of 0.8 was used to define GO term pairs as similar, shown in step 3 of the pipeline (Figure 3.3). Subsequently, using the python package networkx (Hagberg et al., 2008), I created a network graph with the GO terms as the nodes and the Lin similarity as the edges, example given for *Drosophila* in Figure 3.3 Step 3, for the CC domain. I found all the sub-graphs that were not connected to each other and treated the GO terms within these sub-graphs as belonging to the same group. Therefore, I created groups of GO terms that were treated as different categories that are going to be used as values of the categorical feature for the GO terms. Finally, some GO terms might not be semantically similar to any other GO term but still be important as they have a large number of genes associated with them. Therefore, in the final part of the pipeline, (Figure 3.3, Step 4) I retain those terms and add them to the groups of GO terms. After plotting the distribution of genome coverage I noticed that for all the domains and for both organisms, GO terms with fewer than 100 genes assigned to them do not contribute a lot to the percentage coverage and so I drop them. The finalised GO term groupings will be presented and assessed in the Results and Discussion section. The pipeline is robust to different dataset sizes as it was tested and used in six different scenarios (*Drosophila* and *Mus* datasets, for the BP, MF and CC domains).

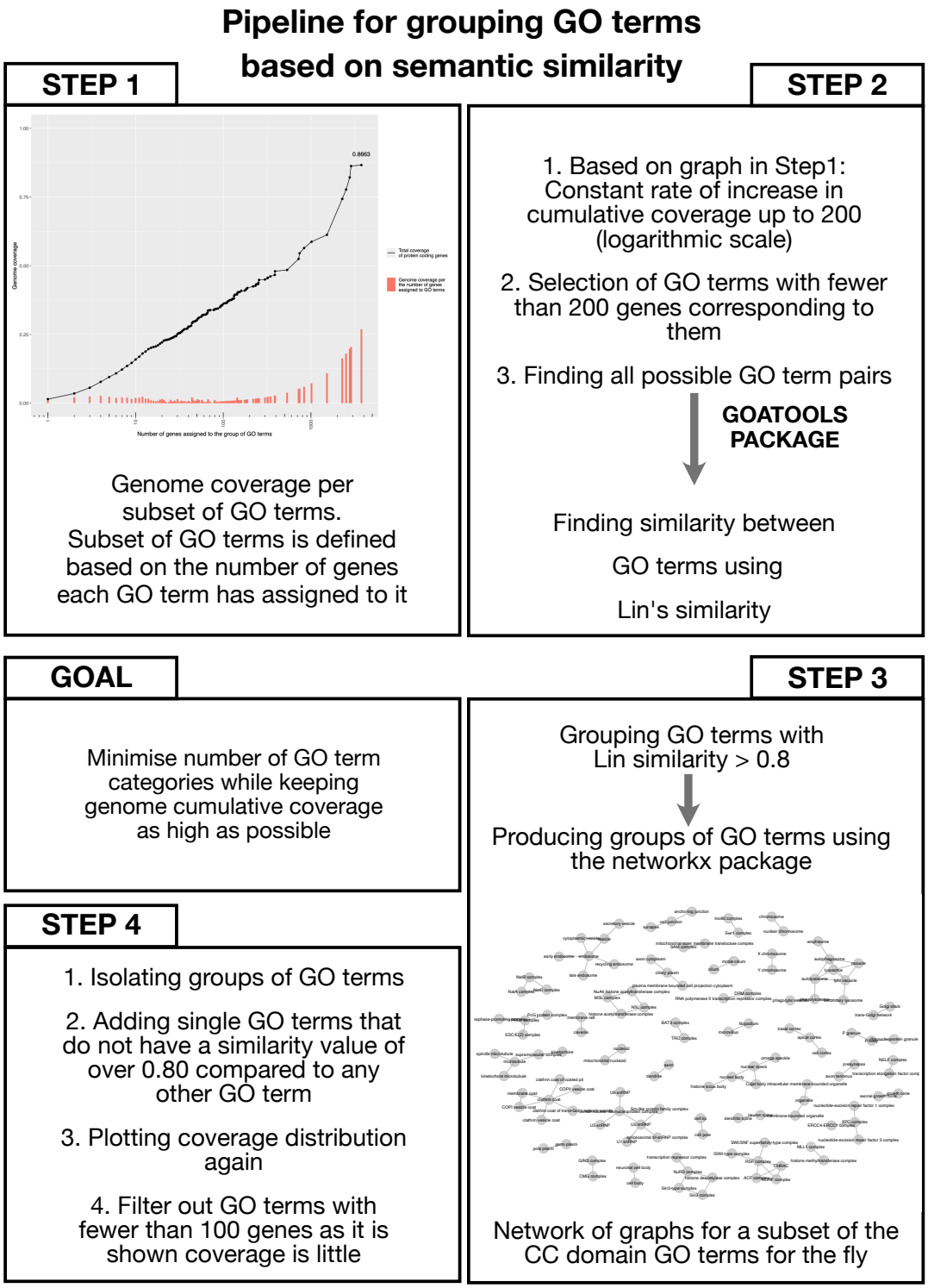


Figure 3.3: Pipeline for filtering and grouping GO terms based on semantic similarity. (Step 1) Line and bar plots showing the cumulative and component genome coverage of GO terms depending on the number of genes each GO term has assigned to it. The x axis shows the number of genes a GO term has assigned to it and the y axis shows the genome coverage for protein-coding genes for those GO terms. (Step 2) GO terms with less than 200 genes assigned to them are selected. All the possible combinations of the GO terms within the selected list are found. GOATOOLS is used to find the semantic similarity between the pairs using the Lin's similarity metric. (Step 3) Pairs with Lin's similarity more than 0.80 were considered as similar and they were grouped together. Using networkx an undirected network graph was produced to visualise the similarity between the groups and find the disconnected groups of GO terms. (Step 4) The groups of GO terms are isolated and GO terms that have more than 200 genes assigned to them are added to the list of groups of GO terms. Gene count is calculated for the groups and if the count is lower than 100 these groups are dropped.

3.2.4 Data preprocessing and feature engineering

A feature is a numeric representation of raw data and there are many ways to convert raw data to numeric representations; always depending on the type of data, the type of model they are going to be fed to and the problem that we are trying to solve. This process is defined as "feature engineering" (Zheng & Casari, 2018).

The selection of features is not an easy process. Care should be taken to select informative features so that the model is trained with the proper input to achieve its goal. There are problems arising from both the number of features and the relevancy of them (Zheng & Casari, 2018). If there are too many features, high dimensionality in the input space is introduced. This in turn requires more time to train the model and also requires a lot of computational resources and advanced infrastructure to train the model which sometimes might not be available or accessible due to the high costs involved. If the features selected are not providing relevant or useful information, noise can be introduced in the predictive model which may compromise the quality of predictions.

To help identify a subset of features there are non-subjective approaches to feature selection. According to their relation to the learning algorithms the feature selection methods can be separated into three categories (Cai et al., 2018): 1. Filter methods; 2. Wrapper methods; 3. Embedded methods. Filter methods are independent of the learning algorithm and act as a preprocessing step to firstly prioritise the features and then select the higher ranking ones (Chandrashekar & Sahin, 2014). Examples include correlation methods which their use is described in the next sections of this thesis. Wrapper methods are related to the learning method, as they use different subsets of the features to train a learning algorithm and they find the subset of the features with the highest algorithm performance (Chandrashekar & Sahin, 2014). Different wrapper methods include recursively adding or removing features to subsets of features like forward selection or backward elimination. Embedded methods are a combination of filter and wrapper methods as they include a combination of pre-training feature selection and training with the selected subset to optimise the learning model performance. In terms of computational complexity and increase in training time with data complexity/size, wrapper methods are the computationally

more complex. Filter methods are the computationally most efficient as filter methods evaluate features independently. Amongst filter and wrapper methods there is a trade-off between model accuracy and computational/time efficiency (filter methods might lead to a less accurate model compared to wrapper) (Bolón-Canedo et al., 2015). Embedded methods reduce the computational time required by wrapper methods but they might be less effective when trying to identify smaller feature sets. The choice of the appropriate feature selection method should be done taking into consideration the computational power available and the dataset complexity. Once the appropriate feature selection method is selected, the optimal choice of features can be identified.

In Table 3.1, I show information about the features created, as well as their source, the type of each feature and the organism that the feature is available for. Figure 3.4 shows the raw data each feature was created from to show the level association of each feature with the raw data. The data features are available at my GitHub repository (`https://github.com/mkiourlappou/thesis-project/tree/main/data_matrices`).

Table 3.1: Table of features used in this work, with feature type and feature source information availability as well as information about the organism the feature applies to

Name of feature	Source of feature	Type of feature	Organism feature is available for
Gene length	ENSEMBL	Numeric	Both
3' UTR mean length	ENSEMBL	Numeric	Both
3' UTR min length	ENSEMBL	Numeric	Both
3' UTR max length	ENSEMBL	Numeric	Both
3' UTR extension length	ENSEMBL	Numeric	Both
5' UTR mean length	ENSEMBL	Numeric	Both
5' UTR min length	ENSEMBL	Numeric	Both
5' UTR max length	ENSEMBL	Numeric	Both
5' UTR extension length	ENSEMBL	Numeric	Both
Isoform mean length	ENSEMBL	Numeric	Both
Isoform min length	ENSEMBL	Numeric	Both
Isoform max length	ENSEMBL	Numeric	Both
Isoform extension length	ENSEMBL	Numeric	Both
No of isoforms	ENSEMBL	Numeric	Both
Mean intron/Exon ratio	ENSEMBL	Numeric	Both
Min intron/Exon ratio	ENSEMBL	Numeric	Both
Max intron/Exon ratio	ENSEMBL	Numeric	Both
3' UTR GC content	ENSEMBL	Numeric	Both
5' UTR GC content	ENSEMBL	Numeric	Both
Coding sequence GC content	ENSEMBL	Numeric	Both

Name of feature	Source of feature	Type of feature	Organism feature is available for
3' UTR k-mers (k=3,4,5)	ENSEMBL	Numeric	Both
5' UTR k-mers (k=3,4,5)	ENSEMBL	Numeric	Both
Coding sequence k-mers (k=3,4,5)	Numeric	ENSEMBL	Both
3' UTR mean proportion single stranded content	ENSEMBL	Numeric	Both
5' UTR mean proportion single stranded content	ENSEMBL	Numeric	Both
Coding sequence mean proportion single stranded content	ENSEMBL	Numeric	Both
RNA binding protein targets sites	oRNAment	Categorical	Both
GO terms	ENSEMBL	Categorical - ordinal	Both
Pfam domains	ENSEMBL	Categorical	Both
Half life	Literature	Numeric	Fruit fly
Abnormal glia	MGI	Categorical (0/1)	Mouse
Abnormal CNS glia	MGI	Categorical (0/1)	Mouse
Abnormal PNS glia	MGI	Categorical (0/1)	Mouse
Glial cell differentiation	MGI	Categorical (0/1)	Mouse
Glial cell differentiation regulation	MGI	Categorical (0/1)	Mouse
Glial cell proliferation	MGI	Categorical (0/1)	Mouse

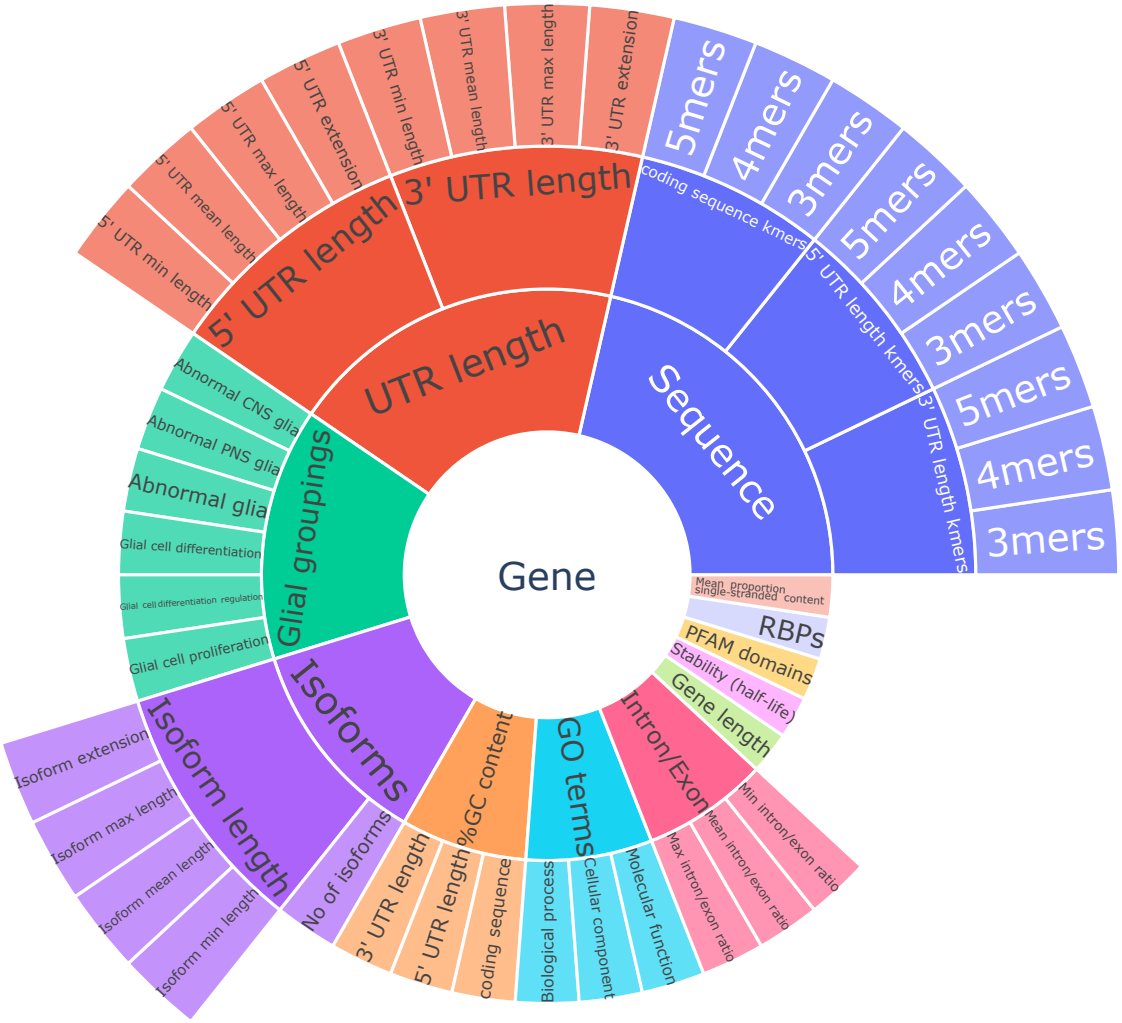


Figure 3.4: Sunburst diagram of the features produced. The sunburst diagram shows the features created from the raw data (inner circle). Each further circle or ring shows how the raw data gets split into more features with feature engineering and data manipulation. The outer pieces represent the finalised features produced which are contributing in the same way to training the model, hence the equal sizes of the pieces.

In the next subsections, I describe the techniques I used to convert each of the raw data sources into the features shown in Table 3.1. These depend on the type of the feature created, numeric or categorical, the raw data and the structure of the database I retrieved the data from.

Numerical features

The numeric features are those that their values are on a continuous scale. In this work the inarguable numerical features are those that are derived from gene length, are percentages of a quantity or ratios of two quantities. As shown in Table 3.1 all the numeric features are retrieved from ENSEMBL. Feature "gene length" was calculated by subtracting the genomic location that the gene starts from the genomic location on the chromosome that the gene ends. To construct the "%GC content" feature for the coding sequence, the 5' UTR sequence and the 3' UTR sequence, I retrieved the sequences from ENSEMBL and then calculated the content of the sequence in guanine and cytosine nucleotides.

For all the features related to isoform length and UTR length, either the 3' UTR length or the 5' UTR length, manipulation of the dataframe was required. This is because each gene can have multiple isoforms and consequently each isoform of the same gene can have different UTR lengths. Firstly, I had to find the difference between the genomic coordinates of each region of each gene to find the length of the isoform, like I did for the gene length. To address the variability of the isoforms, I assigned all the rows corresponding to different isoforms of the same gene to that gene to index the isoform length information to the gene the isoforms correspond to. In that way, I converted a dataset that was structured around isoforms to be structured around genes. I also found the isoform minimum and maximum lengths and kept them as different features. For the "Isoform min length" I used the isoform with the smallest size and if there was only one isoform corresponding to a gene I used this isoform length for both "Isoform min length" and "Isoform max length". Additionally, I calculated the mean length for both the isoform and the UTR regions and used it as a feature. The "Isoform extension", the "3' UTR extension length" and the "5' UTR extension length" features are calculated by subtracting the length of the region

of interest of the isoform with the biggest length from that of the smallest length. Moreover, I calculated how many isoforms each gene has and treated this as a feature.

Finally, I found how many introns and exons each transcript of a gene has by again starting with retrieving data from ENSEMBL using pybiomart. I did that by pulling the available data which are the exon IDs for each transcript. I then summed the number of exon IDs per transcript to find the total number of exons. Introns are non-coding sequences and exons are sequences that are "EXpressed". As by definition introns are the sequences lying between exons, "INtervening" sequences and hence the name (Hubé & Francastel, 2015), I found the number of introns by subtracting 1 from the number of exons. I then found the intron/exon ratio. It was a necessity to find the intron/exon ratio and not vice versa, as in cases where there is only 1 exon and no intron the intron/exon ratio is 0. If the exon/intron ratio was calculated instead that would introduce division by 0, so I chose to avoid this. As there were multiple transcripts for each gene, I tried to keep the information per gene by finding the average intron/exon ratio for each gene, the maximum and the minimum, as I did for the UTR lengths and isoform lengths too.

Categorical features

Gene sequence - k-mers production Research has shown that single 3' UTR sites contain cis-acting elements called "zipcodes" or "localisation elements" which can define the cellular "address" of mRNA molecules (Martin & Ephrussi, 2009) and they have a key role in regulating transcript localisation. Cis-regulatory elements can be found in the 5' UTR as well as in the coding sequence as research in *C. elegans* has shown and they are important in regulating translation (Theil et al., 2018). Localisation elements are recognised by specific RBPs which in turn are crucial in deciding localisation. Although there is no "localisation" sequence which is clearly responsible for localisation (Martin & Ephrussi, 2009), evidently the sequence of a gene is crucial in defining the where and when of transcript localisation.

In this work, I decided to use gene sequence as one of the features to predict localisation. Gene sequence has been used to predict mRNA localisation by many groups who developed and trained machine learning algorithms. Yuan et al. (2023) created RNAlight to determine

RNA subcellular localisation using nucleotide k-mers and to identify which of those are the most influential in nuclear and cytoplasmic localisation. The model was trained using datasets of RNA localisation labels in human HEK293T cells. Two more in-silico tools for predicting mRNA localisation exist in literature, mRNALoc (Garg et al., 2020) and mLoc-mRNA (Meher et al., 2021). mRNALoc uses data from RNALocate (Cui et al., 2022) as training data. RNALocate (Cui et al., 2022) is a database with manually curated localisation information assembled together from literature. As a resource, it has the goal to summarise RNA subcellular localisation on a genome-wide scale. However it lacks *Drosophila* information and so such a model would not generalise when trying to predict information regarding *Drosophila*. When interrogating RNALocate for *Drosophila* data, I only found NMJ localisation information for only one gene which is not included in the 200 screened genes and hence not in the labelled dataset. Likewise mLoc-mRNA is trained using the RNALocate dataset in a similar manner to mRNALoc producing very similar results and these two tools are comparable.

One of the differences between the RNAlight approach (which in terms of training input is the most similar to mine) and my approach, is that its sole feature inputs are created from the gene or transcript sequence, whereas in my case the sequence-based features produced are only some of the input feature sources. Secondly this model was trained using data from human cell-lines, whereas in this work I am using *Drosophila* data produced from intact brains and *Mus* data produced from mouse tissue after being sequenced. Additionally it was trained to predict localisation in the nucleus and the cytoplasm, unlike the glial protrusions I examine. The other two models were trained using the RNALocate resource data (Cui et al., 2022) which mainly contains information about human and mammalian RNA localisation but not *Drosophila*. While they predict more locations of localisation, they are not trained to predict localisation in protrusions.

To produce the features for training my machine learning algorithm, the sequence of each gene for both organisms was retrieved using pybiomart (Ruiter, 2016) through ENSEMBL (Kinsella et al., 2011). The pybiomart python package allowed me to retrieve the sequence for the 3' UTR, 5' UTR and the coding regions separately. Once I retrieved those I organised them as different

columns in a Pandas dataframe (McKinney, 2010), also shown in Figure 3.5 A. Then, I started featurising the datasets by producing all the possible combinations of the k-mers with k = 3,4,5 using the four nucleotides (A, C, G, T) as bases. I created a dataframe with all the possible combinations of k-mers as columns and the genes' IDs as rows, as illustrated in Figure 3.5 B. I then (using code I wrote in python) traversed every gene's sequence (separately for each gene region) and found how many times each combination appears in the sequence. I normalised the frequencies by length divided by the size of the k-mer (3, 4 or 5 base pair); this is to normalise by how many k-mers can fit in the gene. Each gene might appear multiple times; this is to account for the different isoforms each gene has. After, I normalised the frequency by the length of the sequence, I added the frequencies of all isoforms and divided by their number. I thus created a k-mer frequency matrix after the normalisation (Figure 3.5 B). This characterised each gene with 1,344 (= 4^3 + 4^4 + 4^5) times 3 (for each region), giving a total of 4,032 features. There are a few reasons behind selecting to featurise the sequence into k-mers with k = 3,4,5 and separately for the 3' UTR, 5' UTR and the coding region. Firstly, I wanted to capture all the combinations of possible k-mers and the number of possible k-mers grows exponentially with base number 4. With the selection of k=3, 4 and 5 the number of possible combinations is already 1,344, as there are four nucleotides as bases. Consequently the increase of computational power complexity required to find and test all the combinations of bigger size, have limited my considerations to those k-mer sizes. I selected to separate the k-mers into the three regions so that I can retain spatial information regarding the position of a k-mer within the gene. Moreover, I thought that in case certain k-mers (of size k = 3,4,5) were strongly correlated with localised genes, they could be part of bigger motifs or k-mers. In such a case, it would be easier to check combinations of k-mers that are already expected to be significant rather than searching the whole high-dimensional space of k-mer (k >5) combinations.

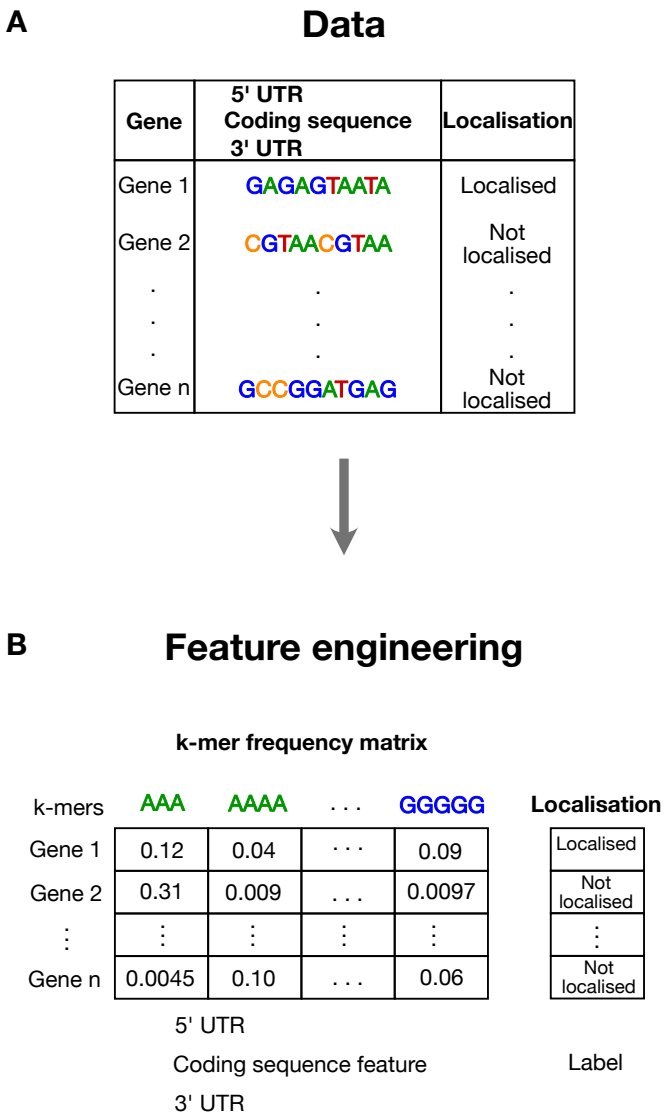


Figure 3.5: Schematic representation of the sequence datasets and the process of feature engineering them. (A) Schematic representation of the dataframe containing the sequence information per gene for all coding sequences, 3' UTR sequence and 5' UTR sequence alongside the localisation labels. (B) k-mer frequency matrix per gene diagram for each gene for the three types of sequence. 3-mers, 4-mers and 5-mers are all computed alongside their frequencies and localisation labels for each gene.

Secondary structure - mean proportion single-stranded content To construct secondary features for the genes, I used the sequences I retrieved earlier via ENSEMBL (Kinsella et al., 2011), for the 3' UTR, 5' UTR and the coding regions separately. I used the function RNAplfold from the Vienna RNAfold package (Lorenz et al., 2011), to find the probability of contiguous unpaired segments and to find this the base pairing probability is averaged over all sequence

windows that contain the putative pair. I produced two different sets of probabilities corresponding to different starting parameters, for each part of the gene (3' UTR, 5' UTR and coding sequence). Firstly I tried the conditions: window size = 70, span = 70 and then the conditions: window size = 100, span = 100. Next, I produced histograms of the distribution of the nucleotides for both conditions to define a threshold for what constitutes an unpaired nucleotide. I expect that most nucleotides will be paired, therefore I expect the histogram to be skewed towards lower probabilities (as the output of the package is the probability of unpaired segments). However the histogram was mostly flat, except at very high probabilities. Therefore I decided to use a range of thresholds and calculate the mean proportion single-stranded content per gene part for different thresholds, starting from probability = 0.5 to probability = 0.95. I then used all the values as numerical features for my algorithms in the next section.

RNA binding proteins (RBPs) domains A similar method to the method used for the feature engineering of the k-mers was also used in the featurisation of RBP target sites. I downloaded the target sites directly from the oRNAment database. These were in the form of two files, one for coding DNAs and one for non coding RNAs. Coding DNAs refer to DNAs that encode for proteins so they certainly produce coding RNAs too. Non coding RNAs do not encode proteins. There are cases where non coding DNA does not encode any non coding RNA hence the distinction. I concatenated the two files, as the coding DNAs file and the non coding RNAs file contain different sets of genes that are not overlapping and are complementary. The files provide information in a transcript-centric way by design but also contain information about the gene the transcript corresponds to. As before, I used the Pandas python package to manipulate the dataset. I firstly normalised the presence of each motif in each transcript by the length of the transcript. I then added the number of motif instances found in each transcript for each motif for the transcripts corresponding to the same gene. I did that so that I can get the number of motif instances for each motif present in each gene. I then normalised this by the number of transcripts corresponding to each gene. The final outcome was a RBP motif frequency matrix for each gene. Moreover, the way I structured and organised the dataset ensured that the transcript information

was preserved and carried forward. In this way I converted the database from transcript-centric to gene-centric. I did this for both mouse and fly RBPs. As I mentioned earlier, converting the different libraries and datasets to gene-centric is a step that takes significant time and thought in the design of the conversion process. Each distinct dataset is organised in a different way so I had to make different assumptions that would be equivalent each time so that the datasets can be comparable as much as possible.

Pfam domains The Pfam domain information for both organisms was retrieved using pybiomart and the information was structured in a gene-centric way. When I retrieved the data in a dataframe, the Pfam domains present in the protein produced by each gene were given in one column and the gene ID corresponding to that protein was given in the next column. Through trial and error, I noticed that when I retrieved the coordinates of the Pfam domains along the protein product of the gene together with the Pfam domain ID for each gene, the dataframe was much bigger than when I only retrieved the Pfam domain ID along the gene ID. I concluded that this is because each gene could have the same Pfam domain appearing in the gene product of the gene multiple times. As a result when I included the coordinates of each Pfam domain this information was not lost. After this, the frequency matrix production method was once more utilised to featurise protein family information and convert the categorical data (each gene having multiple categories (Pfam domains) corresponding to it) into numerical data. As the data were organised in a gene focused way it was simpler to construct a frequency matrix of Pfam domains present in protein products for each gene. As each Pfam domain has different length I could not normalise the frequency in the same way I did with k-mers (normalising by the length of each gene divided by the k-mer size) so I only normalised the frequency matrix for gene length.

GO annotations As discussed in the previous section, I devised and employed a method to group GO terms together. The finalised GO terms groupings will be presented in the Results and Discussion section. In this section I want to highlight the method that is used to produce features from the GO groupings. I will use one-hot encoding (Chollet, 2018) to produce features from

the GO terms or groupings, as this is a well-known way of converting categorical variables so that they can be provided to machine learning algorithms to be trained.

Half-life measurements - stability Research published by my colleague Dr. Mary Kay Thompson (Thompson et al., 2023) shows that the stability of mRNA molecules is an important layer of post-transcriptional regulation, as if an mRNA is not stable it might not be able to travel the distance needed to be at the position at which is required. Therefore I decided to incorporate stability information in the form of half-life measurements in my datasets. Thompson et al. (2023) produced a half-life dataset for genes in the whole cultured *Drosophila* larval brain which I have used. For genes that their half-value was missing, I imputed the missing values by calculating the average half-life of the existent values and using this in cases the half-life was missing.

Gene sets retrieved from MGI Part of the MGI database is the "Gene expression database", which I was not able to access through ENSEMBL. So I manually searched the website for gene expression datasets in glia. As mentioned earlier, there are more studies in the literature regarding glial expression in mammals and mouse rather than in flies. I searched this database for all types of data except RNA-seq and found genes expressed in glia or contributing to abnormal glial development when knocked-out. This was done to avoid overlap with the RNA-seq datasets used through the work of Dr. Jeff Lee and Dalia Gala (Gala et al., 2022). These were used to deduce the localisation labels of transcripts in mouse. The datasets found are demonstrated in Table 3.2, together with how many genes are included in these datasets. I encoded these datasets, by keeping the list of gene IDs and merging it with the whole mouse genome genes IDs. For the genes that were present in the list I added "1" in the feature column and "0" for the ones not present. In this way I converted the categorical text data to numerical features. Unfortunately, I did not find similar results for *Drosophila* so this dataset does not have an analogous one to be used with the fly model; it is only available for the mouse model.

Table 3.2: Table of the datasets retrieved from MGI with direct links to the datasets and information about the size of the dataset

Name of dataset	Link to dataset	Size of dataset
Abnormal glia	link	404 genes
Abnormal CNS glia	link	320 genes
Abnormal PNS glia	link	46 genes
Glial cell differentiation	link	258 genes
Glial cell differentiation regulation	link	94 genes
Glial cell proliferation	link	70 genes

3.3 Results and Discussion

3.3.1 The homologs of *Drosophila* genes' transcripts localising in glia and the *Mus* genes' transcripts found to be present in protrusions are overlapping

In section 3.2.1, I presented the analysis I did to identify the relevant sets of *Mus* genes that are considered localised and not localised. Significant steps of this analysis together with the results are shown in Figure 3.6. As a reminder of the initial parts of the analysis, Panel (A) of Figure 3.6 depicts the mammalian datasets identified from literature, panel (B) shows the boxplots that were used to visually inspect the data mean TPM counts per gene per number of datasets the gene was present in. Panel (C) shows that the use of the certain cut-off described in section 3.2.1, yielded 7,583 genes considered to be localised/present at protrusions, out of the 52,136 genes found in the mouse genome. I considered the rest as not localised and absent from the protrusions, that is 44,553 genes.

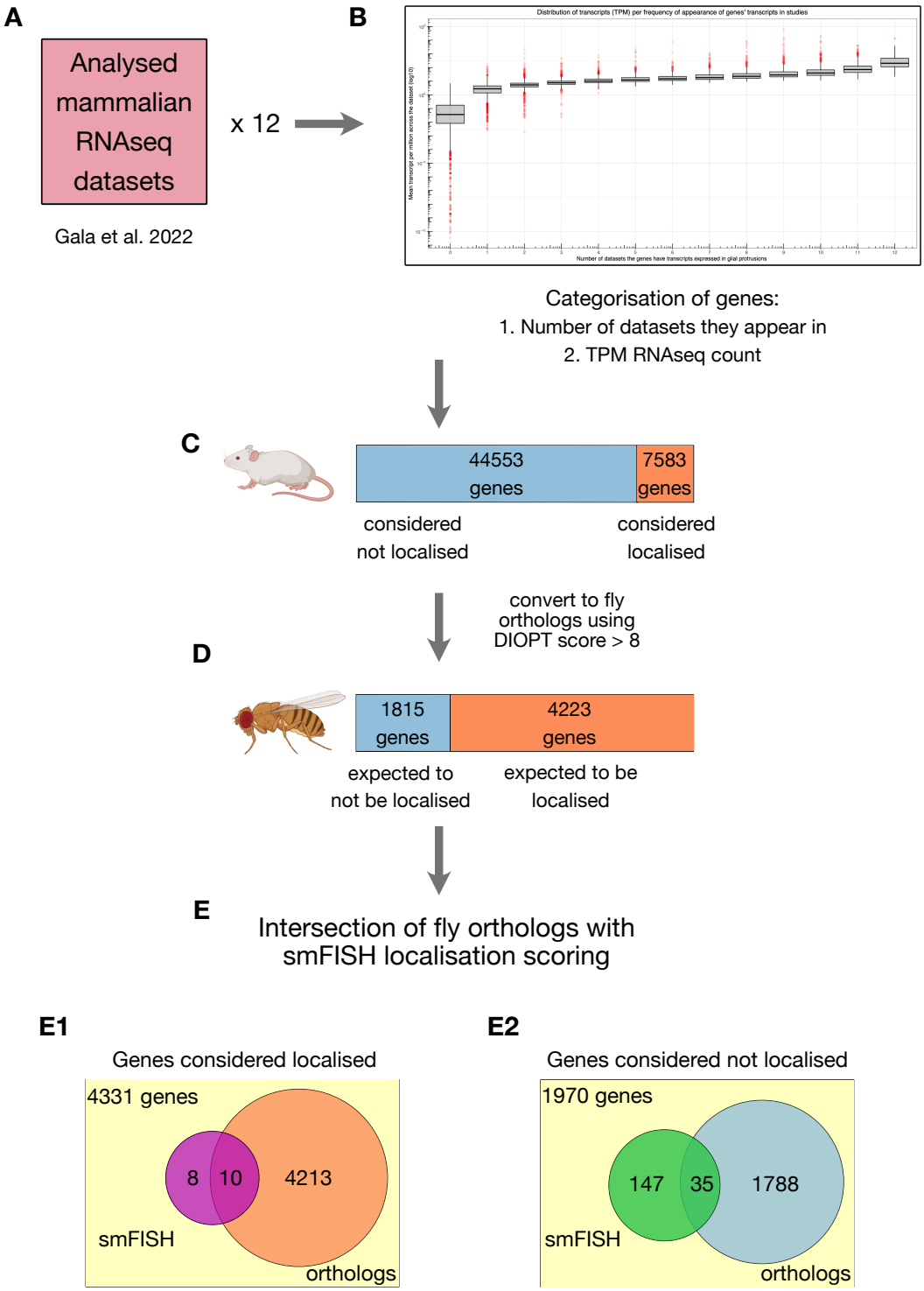


Figure 3.6: Associating expressed genes in mammalian glia to genes with transcripts present in fly glia. (A) Twelve mammalian expression in glia datasets assembled from literature by Gala et al. (2022), analysed by Dr Jeff Lee to be comparable to each other. (B) Transcripts per million (TPM) values for genes found in 1 to 12 datasets in the form of a boxplot distribution was used to select a TPM cut-off for transcripts present at the periphery of glial cells. (C) Numbers of *Mus* genes considered as localised/not localised at glial protrusions. (D) DIOPT scores were used to find the fly orthologs to the mouse orthologs and find *Drosophila* genes that are potentially present in protrusions. (E1) Venn diagram showing overlap of *Drosophila* genes that are known to be localised at glial protrusions in NMJ (visualised with the help of smFISH) with the orthologs that are expected to be localised. (E2) Similar principle as in (E1) but for genes expected not to be localised.

A significant part of Dr Jeff Lee's analysis is the calculation of the relative enrichment when comparing soma and protrusion compartments where he evaluated differential expression of transcripts. I decided to not use this part of the dataset for my analysis and use the mRNA presence/absence TPM counts instead. The reason behind this choice is that I wanted the data to be comparable to the smFISH data for *D. melanogaster* where we examine transcript presence in protrusions and classify it as localised without being concerned about presence in soma at the same time.

After identifying the sets of mouse genes considered localised and not localised (Figure 3.6C), I found (as described in section 3.2.1) that the number of fruit fly orthologs of those genes is 6,038 genes (Figure 3.6D). In Figure 3.6D we can see that the proportion of available *Drosophila* orthologs of genes that are expected to not be localised is much smaller compared to the genes that are expected to be localised. This is contrary to the proportions of the original dataset shown in Figure 3.6C. This might arise because the DIOPT ortholog finder primary works with only protein-coding genes. In Figure 3.6C we can see that 44,553 genes are not considered to be localised and out of those, only 14,490 are protein-coding genes. Therefore half out of the 14,490 *Mus* genes, 1,815 (Figure 3.6D) have *Drosophila* orthologs. Likewise, from the 7,583 *Mus* genes, 7,343 are protein-coding and therefore more than half, 4,223 genes (Figure 3.6D) have *Drosophila* orthologs. If we compare the two proportions of ortholog genes, the *Mus* genes that are considered to be localised have many more *Drosophila* orthologs, which could potentially suggest that localised genes are more conserved. As discussed in the Introduction chapter, mRNA localisation is crucial for the correct functioning of polarised cells such as glia and localised mRNA enables local translation and precise regulation of protein production.

Next, I compared this subset with the 200 screened genes and the overlap is visualised in Figure 3.6E.

In his analysis (Gala et al., 2022), Dr Jeff Lee filtered genes that are present in 8 or more datasets and with TPM bigger than 10, compared to my analysis where I chose genes that are present in at least 4 datasets and with the value of TPM greater than 10. His selection outputted

4,801 genes which mapped to 3,513 *Drosophila* genes when using the DIOPT ortholog finder. He then used the Fly Cell Atlas data, a single-cell nuclear sequencing atlas of the adult fly (H. Li et al., 2022) to dissect the set of 3,513 genes based on cell type. He filtered for the following glial cell types: wrapping/ensheathing, perineural and subperineural glia. These kinds of glia have been reported to produce distal protrusions in the larval NMJ (Yildirim et al., 2019) which is more similar to our observation of transcripts using smFISH in the fly glial cells. His finalised list of genes contains 1,700 genes. Results of the intersection of this subset with the 200 screened genes are shown in Figure 3.7. Out of the 1,700 identified, the 1,612 genes are a subset of the 6,038 I filtered and this difference is attributed to two factors. Firstly, I used a different version of the *D. melanogaster* genome and secondly, I used only the DIOPT top scoring ortholog for each gene rather than all of the ortholog genes with a DIOPT score above 8.

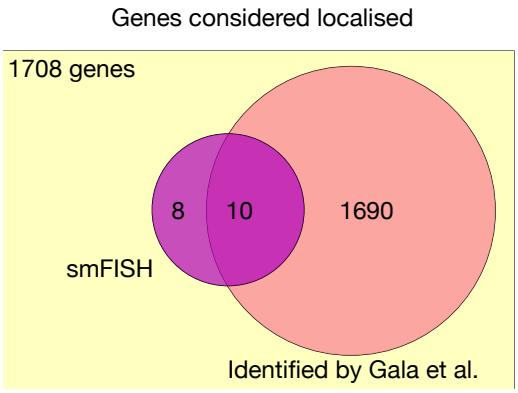


Figure 3.7: Overlap of predicted genes to be localised in glial protrusions by Gala et al. (2022) with genes expressed in NMJ glial protrusions in *Drosophila* larval brain identified using smFISH. Venn diagram showing the overlap between the two sources. 10 genes from the predicted localised genes are confirmed to be present in glial *Drosophila* protrusions using smFISH.

There are now different sets of genes identified as having their transcripts present in glial protrusions, in two model organisms, *M. musculus* and *D. melanogaster*. Going forward in this project, I will be using the following sets of genes for training machine learning models, shown in Table 3.3 :

Table 3.3: Table with sets of genes to be used a training labels for machine learning algorithms

Dataset ID	Organism	Total number of genes	Number of genes with their transcript regarded localised	Number of genes with their transcript regarded not localised
D1	*D. melanogaster*	23,932	18	182
D2	*D. melanogaster*	23,932	1,700	-
M1	*M. musculus*	52,136	7,583	44,553

Dataset D1 is the set of labels identified using smFISH for the glial cells in the NMJ. This is considered ground truth for reasons explained in Chapter 2. Dataset D2 is the dataset identified by Dr Jeff Lee with the 1,700 *Drosophila* genes that are predicted to be localised using orthologs of mouse genes found to be localised using RNA-seq. My analysis yielded 4,223 using slightly different methods. However if we compare Figures 3.6 E1 and 3.7 we can see from the Venn diagrams that when the sets of genes are intersected with Dataset D1, in both instances there are 10 genes overlapping. However 1,700 genes are a much smaller set compared to 4223 and there is the exact same overlap with the ground truth. Hence, I conclude that it is a more localisation focused dataset and I choose to use this for more accurate results. Datasets D1 and D2 are related to *Drosophila* and the results from correlating the sets of labels with other datasets will be used comparatively and complementary to each other. Dataset M1 will be used to predict transcripts localisation in glial protrusions of *M. musculus* brain and since it is a bigger dataset and derived directly from experimental data, part of it will be used for training and part of it for testing and evaluating the trained model.

3.3.2 Assessing missing data values of all datasets used and explaining the potential reasons behind missing values

Assessing missing values in features is an essential step in any machine learning project because missing data can significantly affect the performance of machine learning algorithms. Missing data can make it difficult to identify patterns and trends in the data, which can limit the scope of research and prevent the discovery of potentially important relationships between biological

variables. Additionally in the machine learning prediction part of the project, missing data can limit the performance of a machine learning model and introduce biases to the predictions, affecting the reliability of the results (Nijman et al., 2022). For example, if the approach to handle missing data is to delete the instances containing missing data this will introduce bias to the training of the model, especially if the data are missing not at random. An alternative solution to removing incomplete instances of data (instances with missing data) is to impute the missing data and try to model the missing values (Nijman et al., 2022). Additionally, tree-based models have built-in capabilities for handling missing data and these can be used during the development or implementation of those algorithms to address the issue of missing values (Nijman et al., 2022).

To develop a strategy to deal with missing data, I need to firstly assess their distribution and frequency of missing data across the datasets and identify which datasets have missing data. To understand the nature and extent of missing data in the dataset, I produced heatmaps of the missing values for all datasets. These are shown in Figure 3.8A for the fruit fly and in Figure 3.9A for the mouse. In cases where different features were produced from the same dataset the missing values in the features are universal for all the features produced from the same dataset. Hence, in Figures 3.8A and 3.9A I only show some of the features as features that are produced from the same datasets (such as all the k-mers and the secondary structure which are produced from the same sequence or the UTR lengths mean, minimum and maximum lengths) are omitted for clarity.

In the bottom panels of the two figures (Figures 3.8B and 3.9B) we see the total number of the missing values for each dataset. The total numbers are given in the form of a percentage of the missing values over the whole genome for each organism.

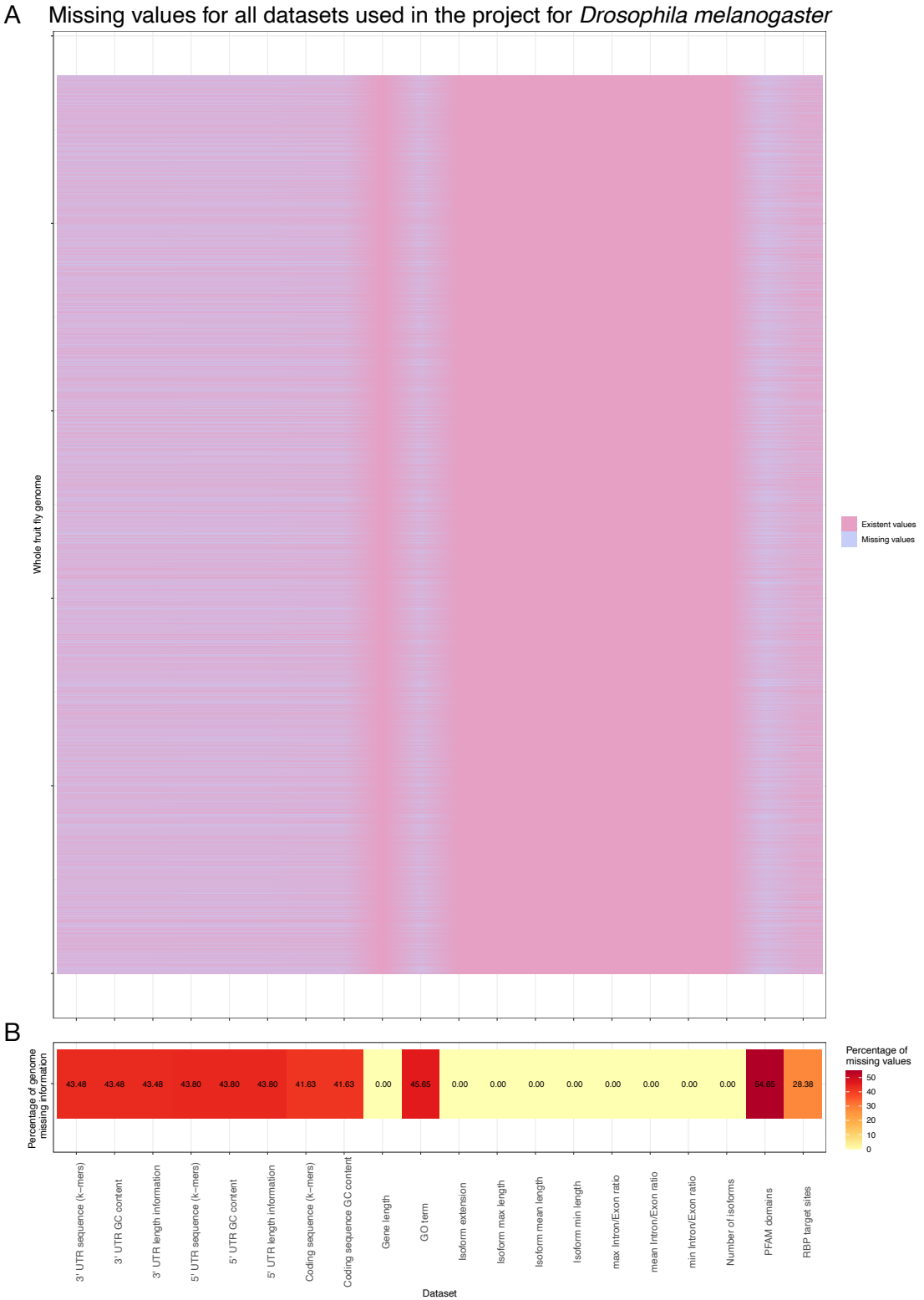


Figure 3.8: Missing values for all datasets used in this work for *Drosophila*. (A) Heatmap of the missing values and the existent values across all datasets for the whole genome. Each row corresponds to one gene condensed together to represent the genome. The missing values are shown with lilac and pink shows values for each gene that exist in each dataset. Some features produced from the same dataset are omitted for clarity. The name of each dataset in the horizontal axis can be seen in panel (B) as the two panels are aligned. (B) The total of missing values for each dataset in the form of percentage compared to the whole genome is shown as a heatmap. Datasets with more missing values have darker red values compared to datasets with no missing values which are shown with yellow.

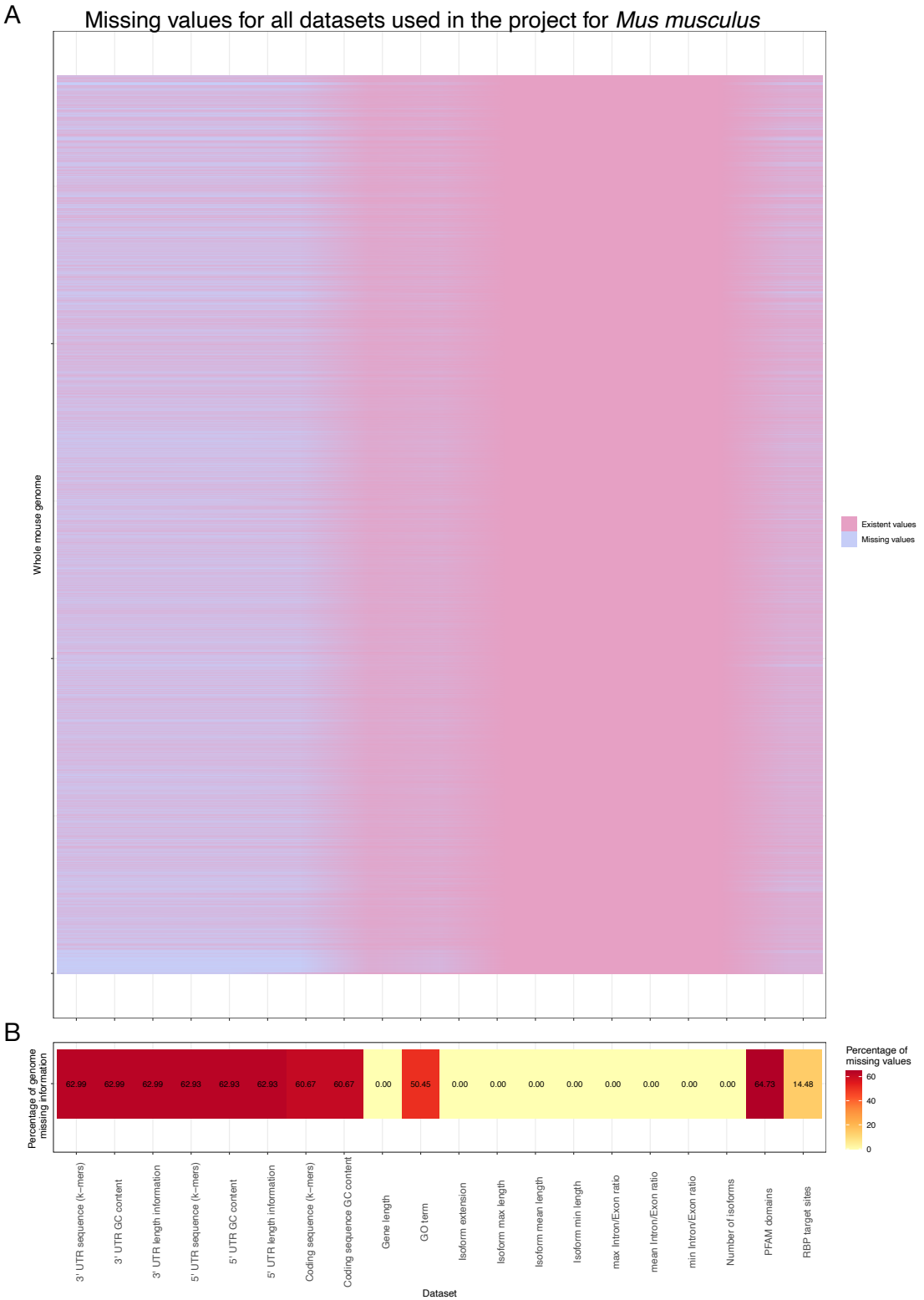


Figure 3.9: Missing values for all datasets used in this work for *Mus*. (A) Heatmap demonstrating the distribution of missing and existing values for the entire mouse genome across all datasets. Every row represents an individual gene. Lilac indicates missing values, while pink represents existing values for each gene in each dataset. To avoid clutter, some features generated from the same dataset have been omitted. (B) Total percentage of missing values for each dataset compared to the entire genome, shown as a heatmap. Datasets with a greater number of missing values are denoted by deeper shades of red, whereas those without any missing values are indicated by yellow.

As I can conclude from Figures 3.8 and 3.9, some datasets relating to the length of the genes (gene length, isoform length) have no missing values. Additionally, the Intron/Exon ratio and the number of isoforms have values for all the genes too. To find the non existent values for Pfam domains, RBP target sites and GO terms I found how many genes have no Pfam domains, RBP target sites and GO terms associated with them and treated them as genes with missing values. From the graphs plotted (Figures 3.8B and 3.9B) we can see that for the fruit flies, 54.65% of genes have no Pfam domain associated with them and for the mice this number is bigger, it is 64.73 %. There are 45.65% and 50.45% genes with no GO terms for flies and mice respectively and these numbers are 28.38% and 14.48% for RBP target sites. For the data related to 3' UTR and 5' UTR (both sequence type data and length) the numbers are very similar (43.48% and 43.80%) for *Drosophila* and likewise for *Mus*, with 62.99% and 62.93% respectively. The coding sequence dataset is missing for 41.63% of fruit flies genes and for 60.67% of mice genes. As the GC content is derived from the sequence data the missing values percentage for the GC content is for 3' UTR, 5' UTR and the coding sequence for *Drosophila* are 43.48%, 43.80% and 41.63% and 62.99%, 62.93% and 60.67% for the mouse respectively.

The reasons for missing values in specific genes can vary depending on the dataset and its source. In the case of GO terms, missing annotations could be due to the fact that a gene's function has not yet been studied, and therefore no GO term has been assigned to it. Therefore, a missing GO term annotation for a gene may not necessarily mean that the gene has no function, but rather that it has not been studied yet. Conversely, a missing GO term could indicate that the gene has no functional role in biological processes. So in this case, the reasoning of missing values is a bit unclear. Likewise, regarding missing values within Pfam domains, some genes might have not yet been well characterised, and the protein encoded by the genes may be very small or have unique domain architecture that is not well-represented in the Pfam database, making it difficult to detect any significant sequence similarity to other known domains. Additionally, it might depend on the biotype of the gene, for example protein-coding genes are more likely to have at least one Pfam domain assigned to them. This is one of the reasons, that I decided to examine the missing values for each dataset per gene biotype to further understand the emerging of missing values.

A gene biotype is an annotation of the general category of a gene. Figures 3.10 and 3.11 display in the form of a bubble plot for each dataset, the present and the missing total values per gene type. Figure 3.10 refers to the fruit fly and Figure 3.11 to the mouse. For fruit flies, ENSEMBL has annotated the genes in-house and for the mouse it sources this type of information from HAVANA (Thomas, 2009). This explains why there are fewer categories of genes for *Drosophila* compared to *Mus*; some of the gene types displayed for *Mus* are parts of the gene types displayed for *Drosophila*. More information on which subtypes of genes belong to each wider category can be found in the Vega ENSEMBL archived website[1].

The assignation of a gene biotype is helping in understanding probable functions of the genes and why some features could be missing. However, a single gene often has more than one transcript, and these transcripts can be of different classes. The convention that is used by ENSEMBL and HAVANA in the process of gene annotation is that, the classification of the gene as a whole is defined by the transcript with the 'highest' level of classification. The different levels of classification are defined by the type of experimental evidence (RT-PCR-seq, manual annotation and computational prediction) (Harrow et al., 2012) that supports them, with the highest being a gene's class was both given by manual annotation and RT-PCR-seq. That means that a gene might have several transcripts with different biotypes but for example only given the protein-coding label.

[1] `https://vega.archive.ensembl.org/info/about/gene_and_transcript_types.html`, last accessed 11/03/2023

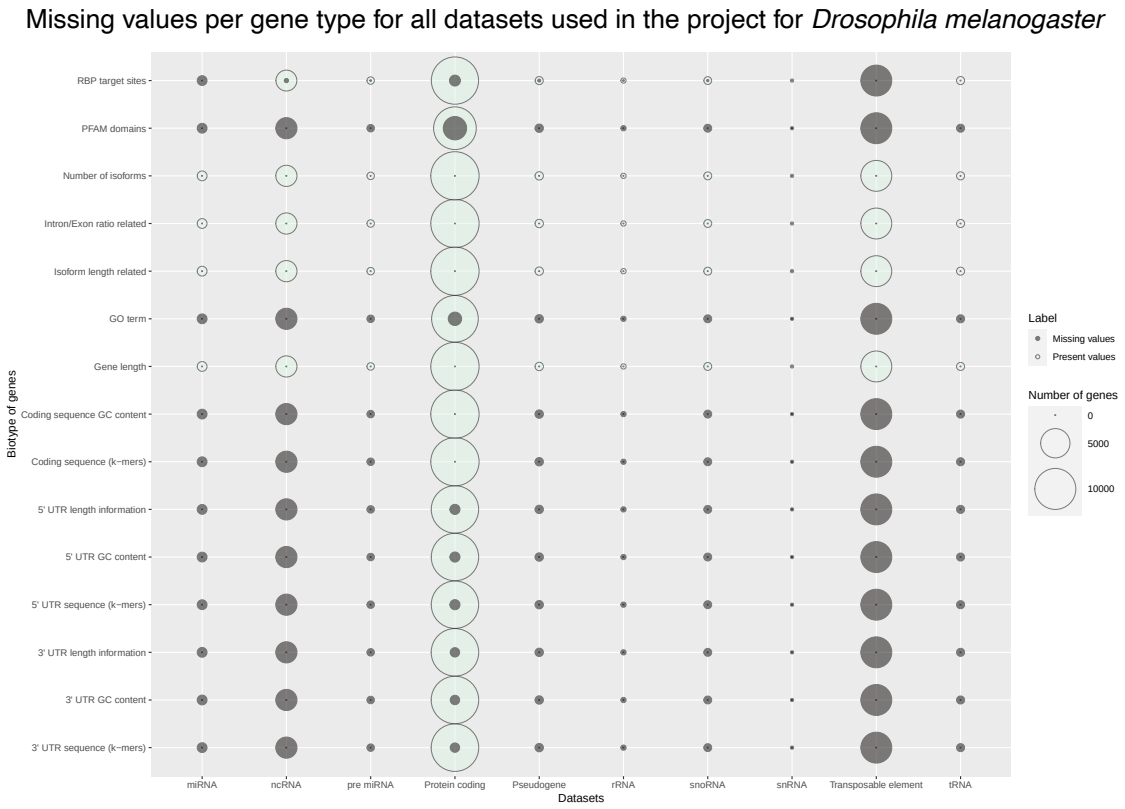


Figure 3.10: Missing values per gene type for all datasets used in the project for *Drosophila.* A bubble plot showing the total missing and existent values for each dataset. This is grouped by gene biotype shown on the y axis and the datasets are shown in the x axis. The size of circles in the form of a grid correspond to the number of genes there is in each category. Light green circles show the total present values for each category and dark grey shows the total absent values.



Figure 3.11: Missing values per gene type for all datasets used in the project for *Mus.* A bubble plot presenting the total amount of missing and present values for each dataset. The data are organized by gene type on the y-axis and by the datasets names on the x-axis. The circles on the chart show the number of genes in each category. The bigger the circle, the more genes there are. The light green circles show the total number of present values for each category, while the dark grey circles show the total number of missing values.

Figures 3.10 and 3.11 demonstrate that for both organisms the protein-coding genes are one of the biotypes of genes that have the least amount of missing values for both datasets. Additionally, during the selection process of the 200 genes to be screened (described in Chapter 2) we highlighted that those genes are all protein coding, as the YFP is fused to the encoded protein. Therefore all the labels produced for *Drosophila* correspond to protein-coding genes. For these two reasons, I have decided to limit the subset of genes of interest to that of the protein-coding genes, for both *Drosophila* and *Mus*. This is to avoid introducing unseen bias to the model by

training it with only protein-coding genes and requesting it to predict labels for non protein-coding genes as well. I use the term unseen as the model will not be trained using the biotype as a feature. This type of bias is documented in the literature (Richardson, 2022) as bias in data capture.

As evident, retrieving genes from various websites involves numerous considerations and introduces inconsistencies. I conducted an extensive study to ensure comparability and inter-operability among the different data sources. This part of the work has been challenging and time-intensive. To account for these challenges and highlight them explicitly, I have openly stated every underlying assumption to account for these. As long as these assumptions are transparent to the reader, I believe the results are appropriate.

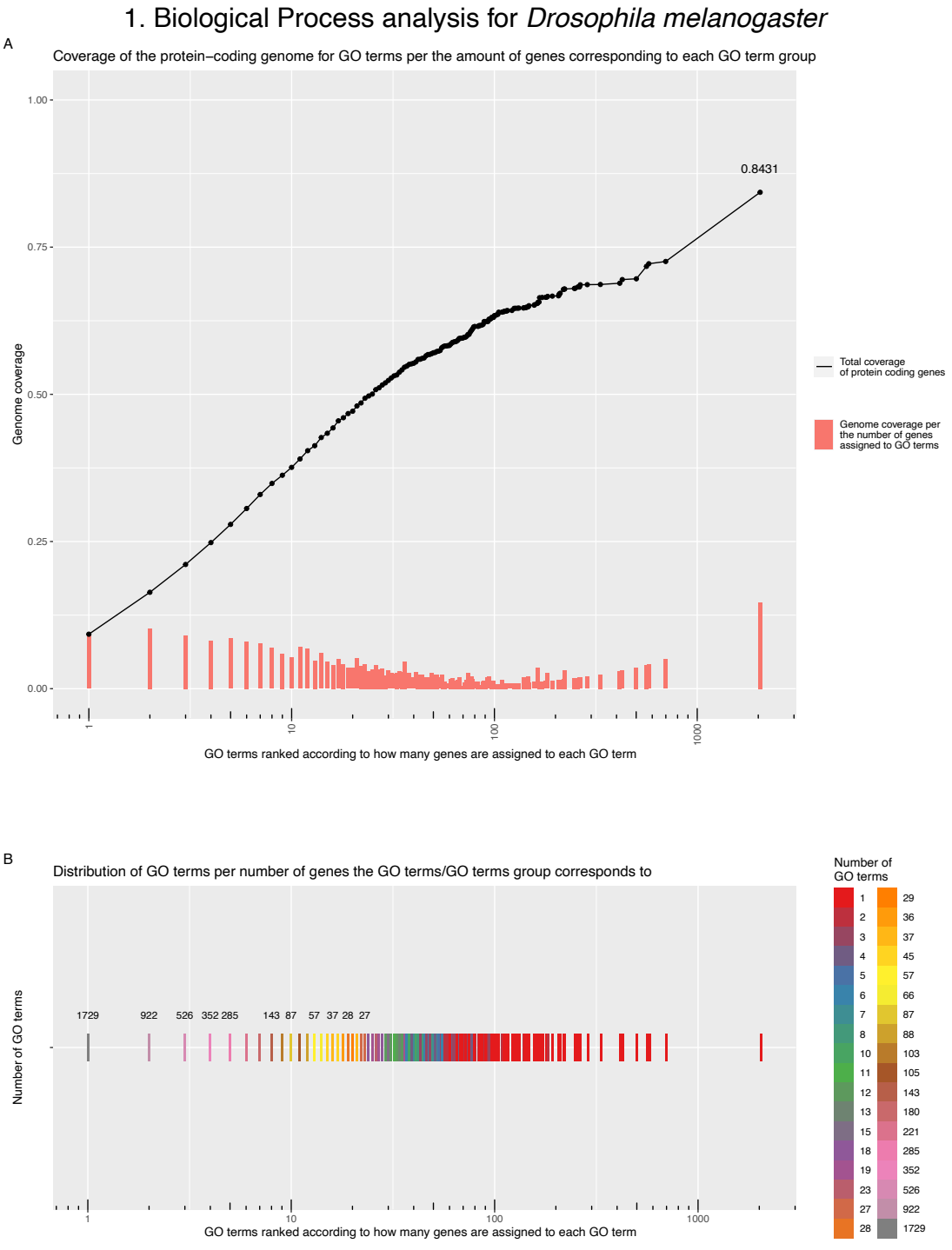
3.3.3 Grouping of GO term annotations and evaluating the genome coverage of the selected groupings

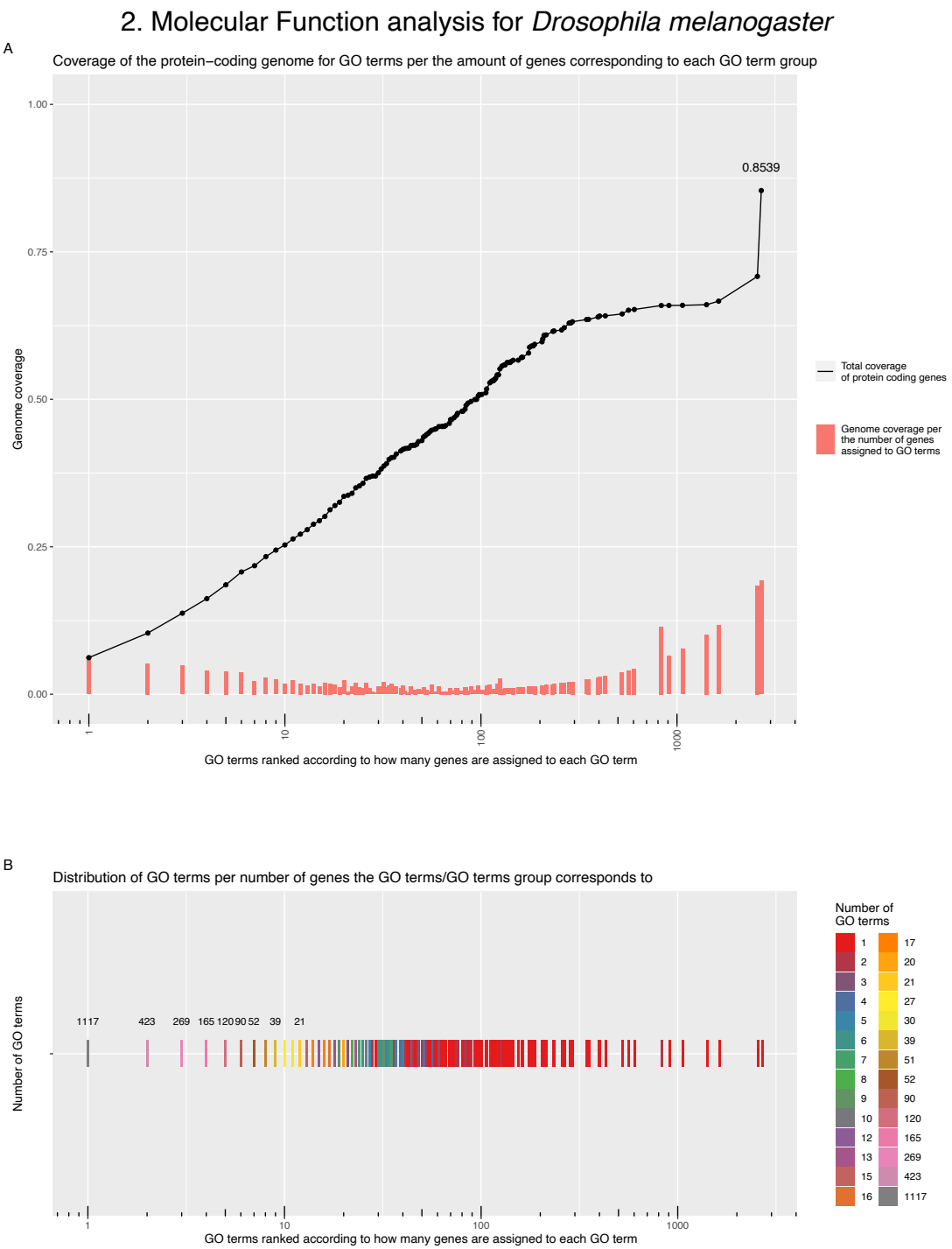
In this section, I firstly present the results of the genome coverage analysis I conducted for both the fly and the mouse, and all three GO domains (BP, MF and CC). Next, I provide the parameters used to produce the groupings of GO terms and finally demonstrate how the groupings of genes produce a high genome coverage even when GO terms dropped.

As discussed before, coverage for a subset of GO terms represents the proportion of the total genes that have at least one GO annotation in the selected subset of GO terms. This depends on the organism and how extensively has been studied and annotated. So, I expect coverage for GO terms for flies and mouse to be different. I focus my analysis on protein-coding genes only as protein-coding is the biotype I chose to examine in this work for reasons explained earlier.

Figure 3.12, panels 1A, 2A and 3A, show the protein-coding genes GO term annotation coverage for *D. melanogaster* for the domains BP, MF and CC respectively and Figure 3.13, panels 1A, 2A and 3A for *M. musculus*. The barplot shows the coverage if a subset of GO terms is used. The subset is defined by how many genes (x axis) correspond to a GO term. For example the subset of GO terms that has only 1 gene assigned to them is used to calculate the coverage (y axis) for the first bar from the left shown in panel A. The black points show the cumulative coverage as we go along the x axis, and for the first point the value of the cumulative

coverage is equal to the value of the bar. For *Drosophila* the total coverage shown in Figure 3.12, panels 1A, 2A and 3A, are 0.8431 (BP), 0.8539 (MF) and 0.8663 (CC) (also shown in Table 3.4) and for *Mus* the corresponding values are 0.9439, 0.942 and 0.9506 (shown in Figures 3.13 and Table 3.4). Figure 3.12, panels 1B, 2B, 3B and Figure 3.13, panels 1B, 2B, 3B show heatmaps of the distribution of how many GO term annotation (frequency of distribution) exist that only have an x number of genes associated with them (value of variable of distribution), for the fly and mouse respectively. As we can see, for all the domains and organisms there is quite a large number of GO terms that have only one gene related to them and only a few GO terms with more than a 1,000 genes associated with them.


1. Biological Process analysis for Drosophila melanogaster
A
Coverage of the protein–coding genome for GO terms per the amount of genes corresponding to each GO term group
Genome coverage
1.00
0.75
0.50
0.25
0.00
0.8431
1
10
100
1000
GO terms ranked according to how many genes are assigned to each GO term
Total coverage of protein coding genes
Genome coverage per the number of genes assigned to GO terms
B
Distribution of GO terms per number of genes the GO terms/GO terms group corresponds to
Number of GO terms
1729
922
526
352
285
143
87
57
37
28
27
GO terms ranked according to how many genes are assigned to each GO term
Number of GO terms
1
2
3
4
5
6
7
8
10
11
12
13
15
18
19
23
27
28
29
36
37
45
57
66
87
88
103
105
143
180
221
285
352
526
922
1729


2. Molecular Function analysis for Drosophila melanogaster
A
Coverage of the protein−coding genome for GO terms per the amount of genes corresponding to each GO term group
1.00
0.75
0.50
0.25
0.00
Genome coverage
0.8539
Total coverage of protein coding genes
Genome coverage per the number of genes assigned to GO terms
1
10
100
1000
GO terms ranked according to how many genes are assigned to each GO term
B
Distribution of GO terms per number of genes the GO terms/GO terms group corresponds to
Number of GO terms
1117
423
269
165
120
90
52
39
21
Number of GO terms
1
2
3
4
5
6
7
8
9
10
12
13
15
16
17
20
21
27
30
39
51
52
90
120
165
269
423
1117
1
10
100
1000
GO terms ranked according to how many genes are assigned to each GO term

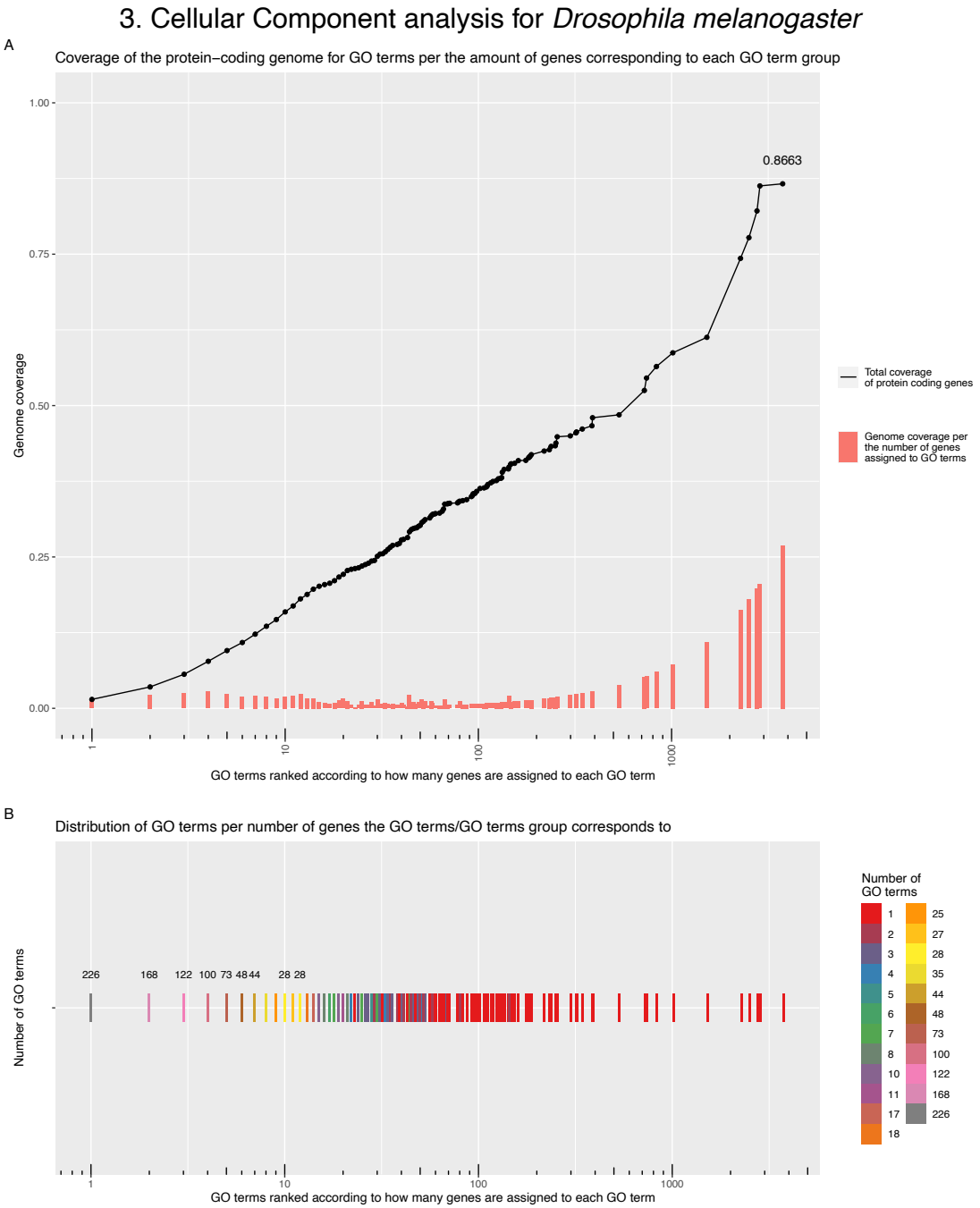
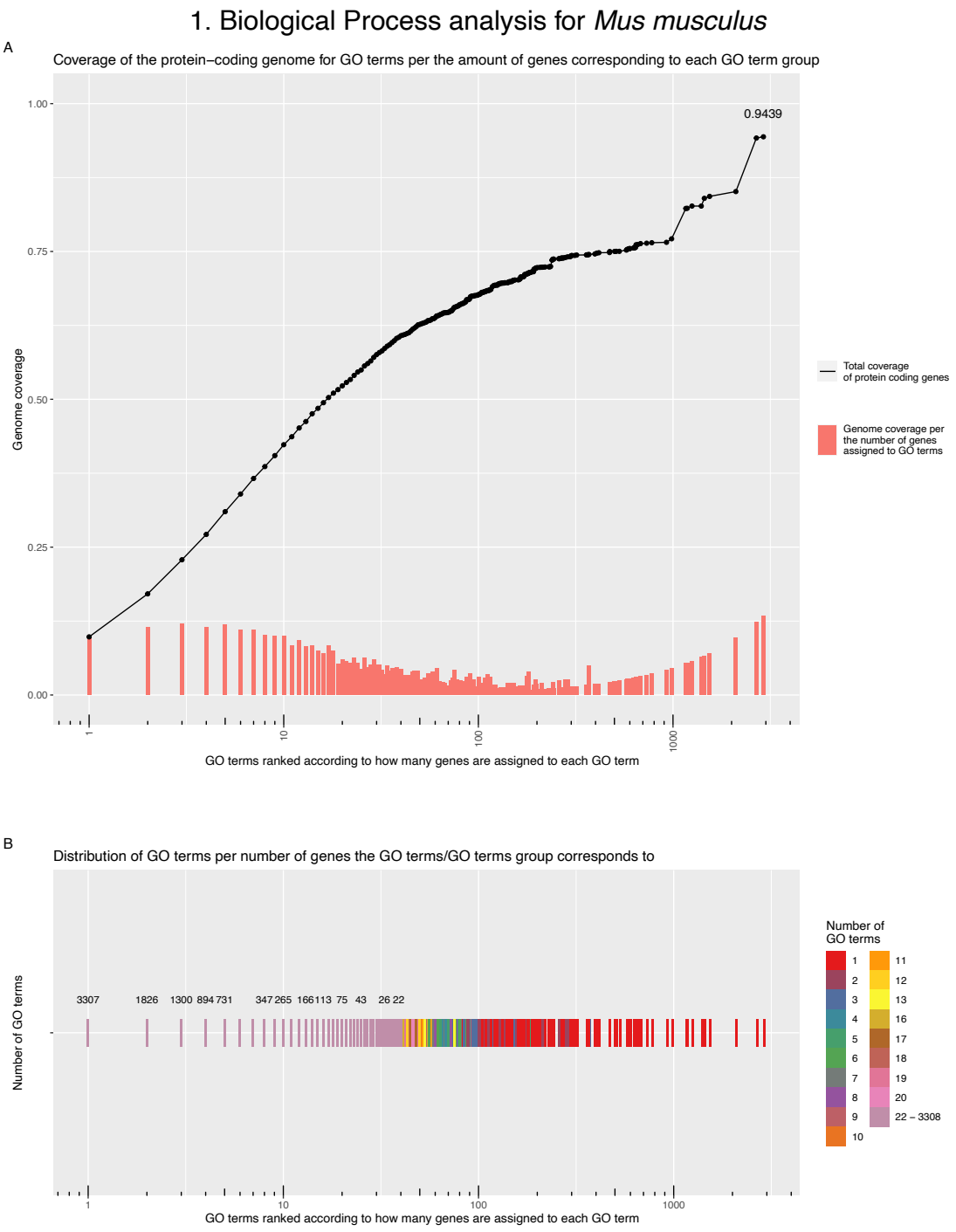
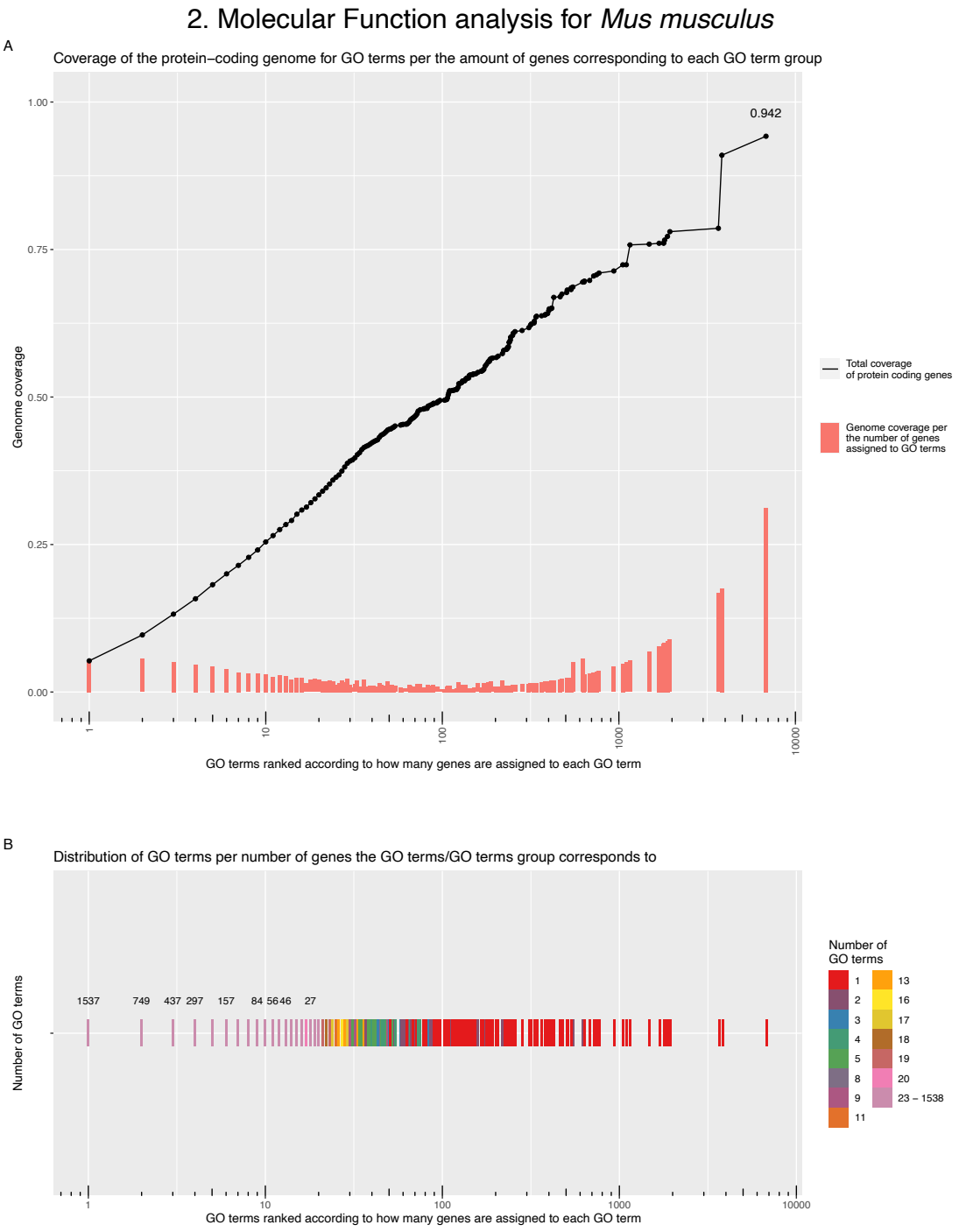


Figure 3.12: Protein-coding genes GO terms annotation coverage analysis for *Drosophila*. (1-3) All sub-figures show the same analysis for the GO terms annotation coverage but for different GO domains for *Drosophila*. Sub-figure 1 presents analysis for the BP domain, sub-figure 2 for the MF domain and sub-figure 3 for the CC domain. Panel A in every sub-figure, presents the protein-coding genes GO terms coverage. Each pink bar represents the coverage obtained if only GO terms that have *n* number of genes assigned to them were included in the calculation of the coverage. The genome coverage is given in the form of a ratio. The coverage is calculated separately for each of the domains and it is given in the different sub-figures. The black line shows the cumulative coverage as we progressively move to a higher number of genes annotated to GO terms across the x axis. The highest point shows the total highest possible coverage if all GO terms are included for protein-coding genes. Panel B in every sub-figure, is aligned with panel A, showing for each *n* number of genes assigned to GO terms how many GO terms there are that only have *n* genes associated with them. For example in 1B, there are 1,729 GO terms annotations that are only related to one gene, but there is only 1 GO term that is associated with 1,100 genes. All possible numbers of genes are shown with different colours. For clarity, labels explicitly indicating how many GO terms correspond to number of genes are shown in the heatmap for a limited number of genes. The number of genes in the x axis is given in base 10 logarithmic scale in both panels.


1. Biological Process analysis for Mus musculus
A
Coverage of the protein–coding genome for GO terms per the amount of genes corresponding to each GO term group
0.9439
Genome coverage
1.00
0.75
0.50
0.25
0.00
1
10
100
1000
GO terms ranked according to how many genes are assigned to each GO term
Total coverage of protein coding genes
Genome coverage per the number of genes assigned to GO terms
B
Distribution of GO terms per number of genes the GO terms/GO terms group corresponds to
Number of GO terms
3307
1826
1300
894
731
347
265
166
113
75
43
26
22
Number of GO terms
1
2
3
4
5
6
7
8
9
10
11
12
13
16
17
18
19
20
22 – 3308


2. Molecular Function analysis for Mus musculus
A
Coverage of the protein–coding genome for GO terms per the amount of genes corresponding to each GO term group
0.942
1.00
0.75
0.50
0.25
0.00
Genome coverage
Total coverage of protein coding genes
Genome coverage per the number of genes assigned to GO terms
1
10
100
1000
10000
GO terms ranked according to how many genes are assigned to each GO term
B
Distribution of GO terms per number of genes the GO terms/GO terms group corresponds to
Number of GO terms
1537
749
437
297
157
84
56
46
27
Number of GO terms
1
2
3
4
5
8
9
11
13
16
17
18
19
20
23 – 1538
1
10
100
1000
10000
GO terms ranked according to how many genes are assigned to each GO term

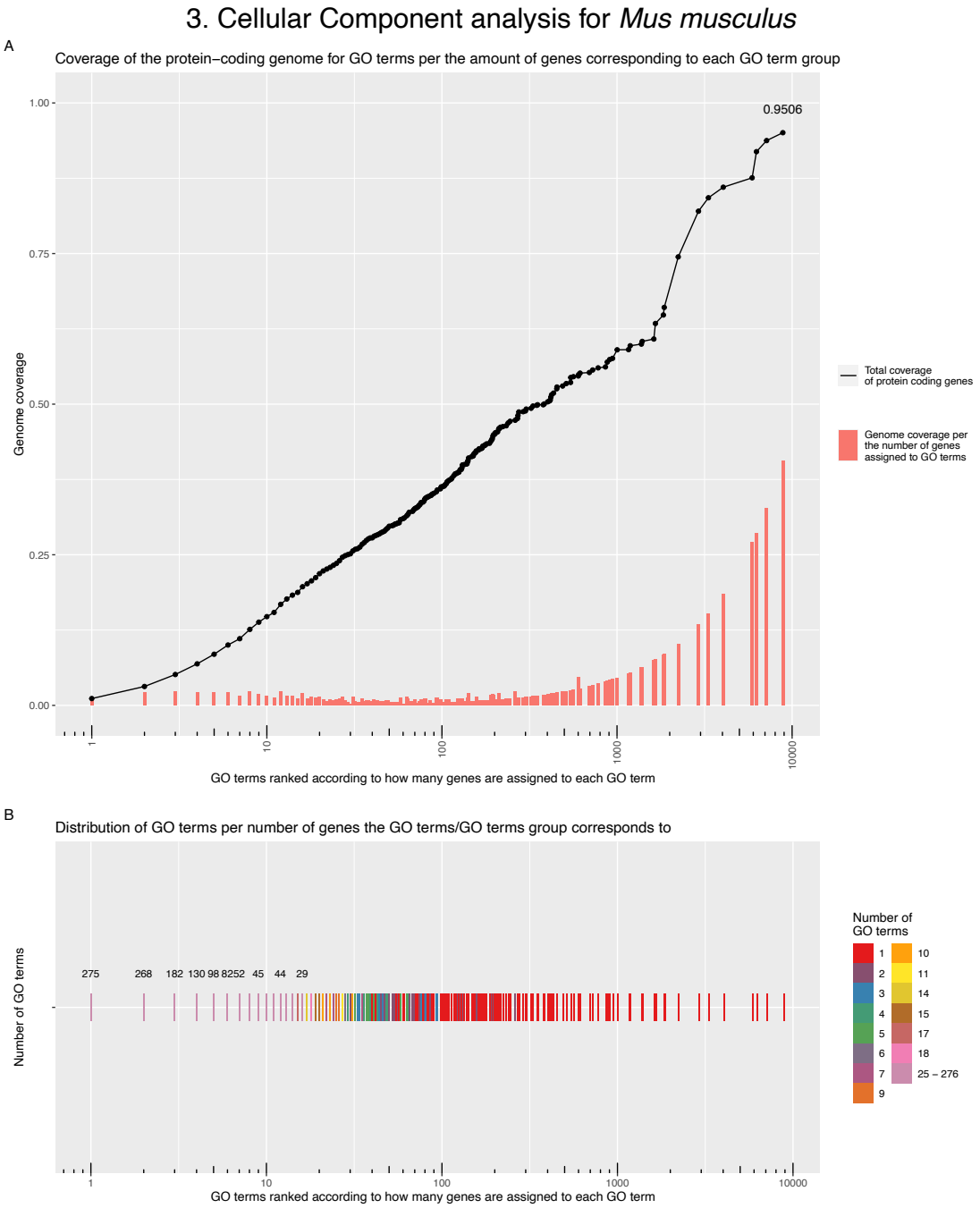


Figure 3.13: Protein-coding genes GO terms annotation coverage analysis for *Mus*. (1-3) All sub-figures depict the same assessment for GO terms annotation coverage for *Mus* with each sub-figure corresponding to the distinct GO domains. Sub-figure 1 shows the evaluation of the assessment for the BP domain, sub-figure 2 for the MF domain and sub-figure 3 for the CC domain. Panel A in every sub-figure, shows a bar plot of the protein-coding genes GO terms coverage for subsets of GO terms and a line plot of the cumulative coverage. Each pink bar represents the coverage obtained if only GO terms that have an *n* number of genes assigned to them were included in the calculation of the coverage. In the y axis, the genome coverage is given in the form of a ratio. The black line shows the cumulative coverage as we move to a higher number of genes annotated to GO terms across the x axis. The maximum point shows the highest possible coverage if all GO terms are included for protein-coding genes. Panels B and A are aligned, showing for each *n* number of genes assigned to GO terms how many GO terms there are that only have *n* genes associated with them. For example in 1B, there are 3,307 GO terms annotations that are only related to one gene. For clarity the smaller numbers (which overlap the most) are shown with different colours and the ones that do not overlap are shown with the same colour but labels explicitly indicating how many GO terms correspond to number of genes are shown above the bars. The number of genes in the x axis is given in base 10 logarithmic scale for both panels.

As it was also described in Section 3.2.3, we can see in the plots in Figure 3.12 1A, 2A, 3A and Figure 3.13 1A, 2A, 3A, that the black line's slope is constant. As the x axis is set at logarithmic scale it means that the coverage (which is depicted by the black line) increases exponentially. Thus the coverage contributed to the cumulative coverage by the GO terms visualised in the beginning of the graph is a big part and hence the GO terms are considered important. In the heatmaps in Figure 3.12 1B, 2B, 3B and Figure 3.13 1B, 2B, 3B, the distribution of the number of GO terms shows the actual number of those GO terms contributing to the high coverage. After following the pipeline, the coverage for the groupings of GO terms and how many groups there are, are documented in Table 3.4.

Table 3.4: Table of the genome coverage before and after grouping for each organinsm and for each GO domain

Organism	Domain	Number of unique GO terms	Total coverage before grouping	Number of groups	Total coverage after grouping
D. Melanogaster	BP	5,637	84.31 %	37	81.89 %
D. Melanogaster	MF	2,759	85.39 %	45	81.50 %
D. Melanogaster	CC	1,184	86.63 %	46	84.95 %
M. Musculus	BP	12,831	94.39 %	131	93.07 %
M. Musculus	MF	4,590	94.20 %	79	92.55 %
M. Musculus	CC	1,820	95.06 %	101	94.54 %

For both organisms and after grouping the GO terms, the coverage for protein-coding genes has dropped for all three domains. The percentage drop ranges between 2-4 % for *Drosophila* and between 1-2 % for *Mus*. This an insignificant change in coverage. When it is combined with the big reduction in the unique categories (5,637 unique GO terms as categories before grouping and 37 groupings afterwards used as unique categories for *Drosophila* for the BP domain and likewise for the *Mus* and the other domains), we can conclude that the goal of this part of the project has been achieved.

3.3.4 Sparsity of features and feature selection using correlation between the features

Here, I investigated the correlation between the features. My goal is to see if there are features that are highly correlated. If so, it could indicate that some of them are redundant and could be dropped. According to literature, there are two typical correlation metrics that are usually used for correlation calculations, Pearson correlation and Spearman correlation. Both metrics fall under the umbrella of the "filter" type feature selection techniques. Pearson correlation is used when the features that are involved in the correlations are expected to have a linear relationship between them and are jointly normally distributed (Schober et al., 2018). On the contrary, Spearman correlation is used when the correlation is expected to be non linear, when the data are non normally distributed and there are relevant outliers in the data (Schober et al., 2018). Both metrics are typically used to calculate correlation between each feature and the target variable (relevance) but they can also be used to calculate the redundancy between pairs of features (Tsanas et al., 2010).

I started by grouping together features stemming from the same source. I anticipated that features in the same group are more similar in nature and more likely to provide comparable information. Therefore I expected them to be more highly correlated than features from different groups. Next, I examined the distributions of the features to understand their shape and choose the correlation metric accordingly. For example, the features related to length (3' UTR extension related, 5' UTR extension related features, isoform extension related features and gene length) were grouped together and the distributions of the variables calculated.

The sparsity of a feature is another important quality of a feature to examine. A sparse feature is a feature that contains a high percentage of zero values. Sparse features are very common in recommender systems. These are systems where a user rates items, such as movies (Idrissi & Zellou, 2020). It is observed that only a few users leave feedback on all available items thus the rating data are usually sparse (Idrissi & Zellou, 2020). A common measure to find similarity between users ratings is the cosine similarity (*Data Mining*, 2012). A user's ratings are considered a vector and even for a very small amount of ratings (only a few non zero values) the cosine rule can find the direction of the vectors and hence the similarity between the two vectors (users)

Table 3.5: Table of the sparsity of grouped features and correlation analysis between features of the same group

Organism	Features grouping	Size of grouping	Sparsity of grouping	Correlation metric used	Number of pairs of features with correlation > 0.9
Drosophila	Length related	13	17.2 %	Spearman correlation	8
Drosophila	5' UTR k-mers	1344	71.3 %	Spearman correlation	0
Drosophila	3' UTR k-mers	1344	63.9 %	Spearman correlation	0
Drosophila	Coding sequence k-mers	1344	33.4 %	Spearman correlation	0
Drosophila	Pfam domains	4634	77.7%	Spearman correlation	51
Drosophila	RBP target sites	453	39.0 %	Spearman correlation	91
Drosophila	MF GO terms	45	77.8 %	Spearman correlation	0
Drosophila	BP GO terms	37	60.4 %	Spearman correlation	0
Drosophila	CC GO terms	46	80.6 %	Cosine similarity	0
Mus	Length related	13	17.6 %	Spearman correlation	0
Mus	5' UTR k-mers	1344	68.5 %	Spearman correlation	0
Mus	3' UTR k-mers	1344	39.3 %	Spearman correlation	0
Mus	Coding sequence k-mers	1344	34.3 %	Spearman correlation	1
Mus	Pfam domains	6598	91.9 %	Cosine similarity	461
Mus	RBP target sites	453	28.0 %	Spearman correlation	608
Mus	MF GO terms	78	66.6 %	Spearman correlation	0
Mus	BP GO terms	131	79.7 %	Spearman correlation	0
Mus	CC GO terms	101	90.5 %	Cosine similarity	0

(Wen et al., 2014). In my project and for the purpose of the cosine similarity calculations, I treat the features as if they are the users. For highly sparse features (greater than 80% sparsity) I use the cosine rule to find the similarity between them.

In Table 3.5, I summarise the feature groups for each of the organisms, how many features each group contains, the sparsity of each group of features, the metric used to assess their similarity/correlation and the number of features with correlation bigger than 0.90.

For the majority of variables the shape of the distribution is not normal. In figure 3.14, I show a correlogram containing the distribution of each variable, a scatter plot for each pair of variables and the Spearman correlation of the variables coupled together for the length related features. As expected (for *Drosophila*), isoform mean length is highly correlated with both isoform min length (rho = 0.97) and isoform max length (rho = 0.99). I only show this type of visual for the length related features as the other groups contain too many features to be shown in such a visual.

To calculate the Pearson correlation coefficient I used the inbuilt correlation function in the Python package Pandas (McKinney, 2010). To calculate Spearman correlation I used the correlation function in the R package stats (*R a Language and Environment for Statistical Computing*, 2010) and to find the cosine similarity I used the Python package scikit-learn (Pedregosa et al., 2011).

After finding the correlation between the features within each group, I found the correlation between all the features. For the mouse dataset I found no additional correlation between features of different groups, that is bigger than 0.90. For the fruit fly dataset however I found that there is additional high (higher than 0.90) correlation between two pairs of features. These are the pairs MF_group_38.0-CC_group_18.0 and ATATG_5UTR-PF09138. MF_group_38.0 and CC_group_18.0 correspond to a MF and a CC group I created respectively. The first group is made of the term "catalytic activity" and the second of the terms "protein-DNA complex" and "nucleosome". The feature "ATATG_5UTR" corresponds to the frequency of occurrence of the 5mer ATATG in the 5' UTR and the PF09138 corresponds to the frequency of occurrence of the PF09138 domain. The long name of the PF09138 domain is the "Ubiquitin related modifier".

To calculate the correlation between all features I used the "corrcoef" function in the Python package Numpy (Harris et al., 2020) as it is the most computationally efficient for such large matrices. It calculates the Pearson correlation.

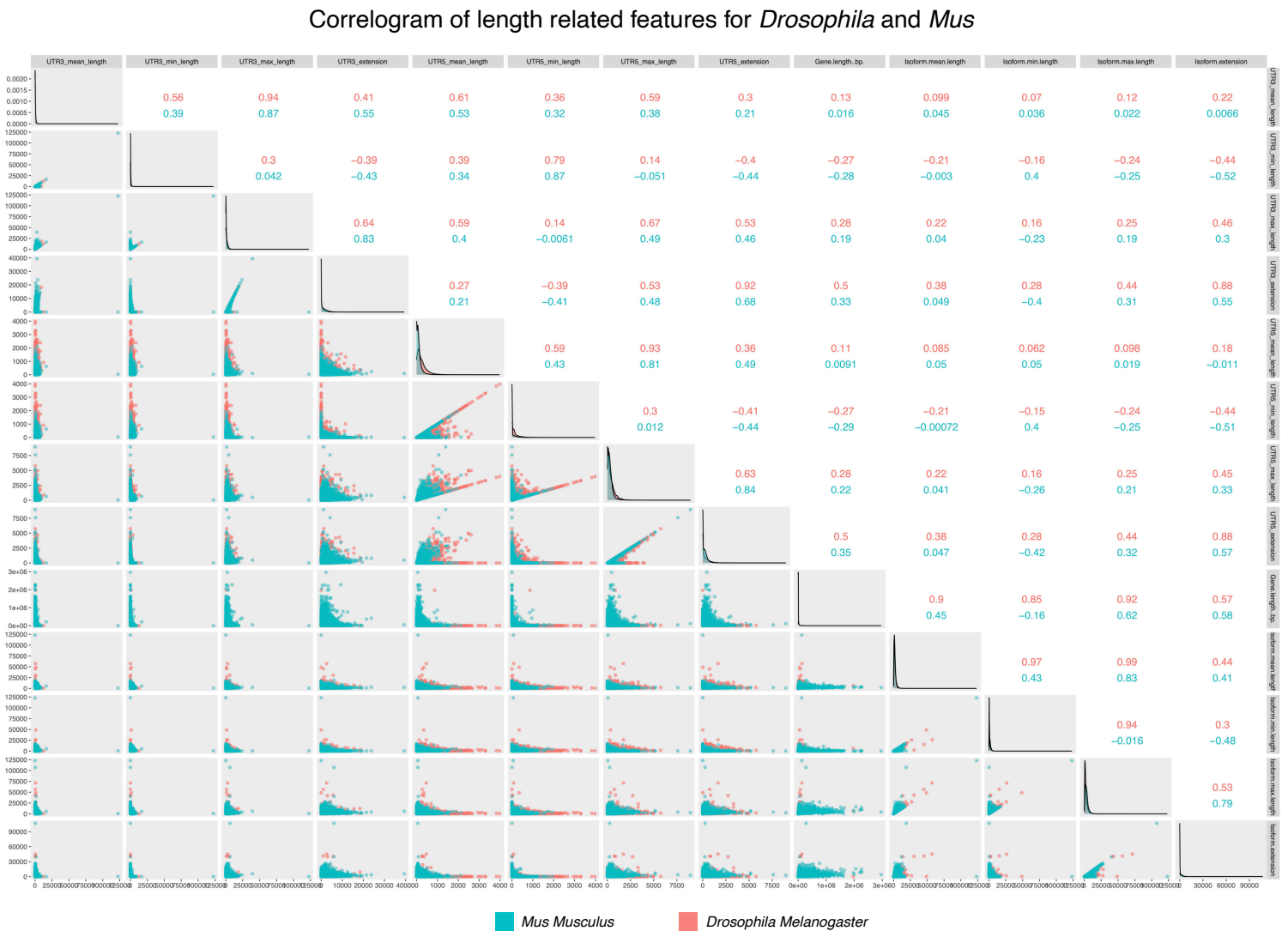


Figure 3.14: Correlogram of the all the length related features in my dataset for both *Drosophila* and *Mus*. *Drosophila* is represented with pink and *Mus* with cyan. The lower left triangle of the correlogram shows scatterplot distributions between pairs of variables. The upper right triangle shows the value of Spearman correlation coefficient for each pair of variable, the different colours correspond to the two organisms. The diagonal shows the distribution of each variable for each organism.

Highly correlated features are usually redundant and hence can decrease the accuracy of a classifying machine learning algorithm while increasing the computational time and cost needed to train a classifier (Danasingh et al., 2020). I plan to combine pairs of highly redundant features and make a new feature instead of dropping one of the features of the pair. With highly correlated features, we can resort to domain knowledge to decide which feature is most relevant to the model and drop the other one. However here, I choose to combine features because both features arise from the same group of features (so they provide similar domain knowledge) and it is difficult to decide which of the two features to drop, i.e. which of the two might be contributing more to the accuracy of the algorithm. For *Mus* this approach will be particularly helpful for the Pfam domain group features and RBP target sites group features as there are a lot of pairs that are correlated. This applies to *Drosophila* for these two groups. Combining highly correlated features can improve computational efficiency, as reducing the number of features means less computational cost during model training. Additionally, compared to dropping one of the features of the highly correlated pair, combining the features in a meaningful way preserves information which can be propagated into the model during training in a cost-effective way. Furthermore, highly correlated features might introduce noise to the algorithm which can reduce the accuracy of the algorithm.

A further observation is that none of pairs of features in the GO term groupings have a very high correlation. In fact when I examined them further the highest absolute correlation observed was between a pair within the CC grouping which was 81.4 % (mouse) and 85.8 % (fly). The rest of the pairs had absolute correlations below 70% with most of the pairs below 50%. This shows that the method I devised for the merging of the individual GO terms into bigger groups (described in section 3.2.4) was successful in grouping similar GO terms together as the groups of GO terms are not highly correlated to each other.

3.4 Summary and Conclusions

In this Chapter I identified sets of genes for which their transcripts localise in glial cells, from data generated in-house and data collated from literature, for both *Drosophila* and *Mus*. I explored functional, biological and structural aspects of genes that could contribute to glial cell localisation.

I presented how I retrieved this information for the *Drosophila* and *Mus* genomes from online public bioinformatics repositories, how I manipulated the heterogeneous datasets and created an integrated set of features describing each gene.

I utilised the high quality annotations produced in Chapter 2 describing transcript localisation in the *Drosophila* nervous system. I focused in localisation in glial protrusions in the PNS and considered the set of annotations as ground truth for mRNA localisation. I examined RNA-seq datasets found in literature regarding transcript presence in mammalian glial protrusions and created a set labels for presence or absence in protrusions for *Mus* genes. I found that there is a significant overlap between the high confidence fly orthologs of the localised mouse genes and the localised fly gene list described in Chapter 2.

For both *Drosophila* and *Mus*, I retrieved genome-wide information related to mRNA localisation mechanisms from widely accessible online databases. The datasets include GO terms annotations, Pfam domain information and RNA binding proteins target sites. Moreover sequence information for the coding sequence and for the 3' UTR and 5'UTR ends. From the sequence information I was able to infer secondary structure information in the form of mean proportion single-stranded content. I also retained physical gene information, examples include isoform number, isoform mean length, 3' UTR extension length and 5' UTR extension length. Additionally I included half-life information for the *Drosophila* genes.

I discussed the complications arising from retrieving data from different sources and the challenges involved in making the datasets interoperable. I highlighted how the tree structure of the GO annotations dataset introduces an additional level of complexity. I explained the method I devised to prune the GO tree such that the GO terms used are not too general (map to many genes) or too specific (map to only a one or two genes). I explained how I used the Lin's semantic similarity metric to group together similar GO terms and create categories of GO terms.

Next I discussed how I extracted features out of the raw datasets. For example, for the sequence data, I found the frequency counts of 3mers, 4mers and 5mers in specific parts of the sequence (3' UTR, 5' UTR and coding sequence) and created a frequency table per k-mer instance.

I followed a similar approach for the Pfam domains presence in each gene as well as for the RBP target sides. Numerical features such as 3' UTR extension length or the percentage of GC content in a gene were more straightforward and simple normalisation methods were used.

After this I explored my data to observe that there were a lot of missing values for non protein-coding genes. I justified why this was expected for some of the features and decided to use only protein-coding genes for training predictive algorithms, as the percentage of missing values was the lowest. Additionally, the ground truth localisation labels I use to train the model describe protein-coding genes only, therefore the genes I will try to predict as being localised will inevitably be protein-coding too.

Finally I investigated the correlation and collinearity of the features, using filter methods feature selection techniques. I grouped similar features together to perform the calculation and I found pairs of features that are highly correlated (correlation greater than 0.9). These pairs of features are potentially redundant and one of the two features could be dropped when training a machine learning algorithm. Highly correlated features were mostly found in the Pfam features group and the RBP target sites group for both *Drosophila* and *Mus*.

The work completed in this Chapter is a good illustration of the preparatory steps required to make data "machine learning ready". Collating big biological data and engineering features to be used to train predictive algorithms is complicated, time consuming and laborious. My work provides an exemplar of data identification, integration and homogenisation and the methods I described can be used to produce the exact datasets and features described in this thesis for both *Drosophila* and *Mus*. The approach can also be generalised to all other organisms with data available and accessible in ENSEMBL.

4

Using supervised and unsupervised learning to predict mRNA localisation in glia

In Chapter 2, I explored the distributions of mRNA and its encoded protein expression for 200 genes within the *Drosophila* nervous system. To do that, I devised a novel pipeline starting with managing the vast amount of raw images produced using microscopy for those 200 genes using online imaging repositories. Next in the pipeline is the creation of figures for those genes in a systematic way and scoring the figures, by a group of experts, for multiple mRNA and protein expression levels, patterns and their presence or absence. The scoring process was managed using bespoke software whose development I led in partnership with Dr. David Pinto. The final step of the pipeline is the novel visualisation of both the figures, the scoring and other associated metadata related to the figures for the 200 genes. Lastly, I generated sets of labels describing the presence or absence of mRNA and protein in twelve tissues of interest within the *Drosophila* nervous system shown in seven figures for each gene using the scoring outputted from the pipeline.

In Chapter 3 I focused on one tissue of interest within the *Drosophila* nervous system, the peripheral glial cells. I identified different publicly available bioinformatics repositories such as ENSEMBL, the GO knowledgebase, Pfam, REACTOME, oRNAment and Intermine that provide datasets potentially related to mRNA localisation for a number of model organisms. I retrieved these data, integrated them and made them interoperable, tackling multiple challenges involving

the heterogeneity of data, the different structure of the data (such as hierarchical versus tabular data), the nature of information (gene-centric, transcript-centric or protein-centric) and missing data. I did so for both *Drosophila* and *Mus*. Using RNA-seq analysis on multiple mammalian datasets for mRNA expression in mice provided by Dr Jeff Lee (from the raw sequencing data of twelve datasets), I produced labels classifying genes as having their transcripts localised in glial protrusions. Using high-confidence *Drosophila* homologs of the *Mus* genes I produced another set of labels classifying localisation for 1,700 *Drosophila* genes. Additionally, I used the mRNA localisation labels produced in the previous chapter for the 200 genes for *Drosophila* and produced three sets of localisation labels. I produced features from the integrated bioinformatics dataset and combined them with the labels. Thus I produced three different datasets that are machine-learning-compatible and ready to be probed by different machine learning algorithms.

In this chapter, I use various machine learning methods to explore the glial mRNA localisation related datasets with the aim of gaining insights into mRNA localisation in peripheral glial cells in both *Drosophila* and *Mus*. This chapter has a twofold objective; firstly to understand which subset of the data is more important in determining and predicting glial mRNA localisation and secondly to predict candidate genes that have high potential of being localised in peripheral glial cells. In this work, I describe how I achieve these objectives, by using both supervised and unsupervised machine learning methods. I also explain the reasoning behind the selection of each method and how each selection is the best choice given the structure of my datasets.

I start this chapter by re-examining feature correlation, but this time using machine learning techniques ("wrapper" type feature selection methods) and evaluating their output in comparison to the more traditional methods used in Chapter 3 ("filter" type feature selection methods). I use RF to understand the contribution of each feature in predicting other features in an attempt to filter out correlated and collinear features. I also use PCA, a dimensionality reduction technique to reduce the dimensionality of the data to fewer components capturing the variance of the dataset. I apply PCA to both the whole dataset and subsets of highly-dimensional groups of features. Additionally, I investigate the feature contribution to the principal components (PCs) using the

impurity importance metric of RF. I finish the exploration of the importance of the features described by investigating whether there is a natural partition of the dataset into localised and non localised genes using k-means clustering. Besides the whole dataset, I use k-means clustering with the subset of features that are expected to have a bigger weight in predicting mRNA localisation.

Inspired from the literature for candidate gene prediction using machine learning, I adapt a python pipeline for candidate sleep related gene prediction provided by Y. Y. Lee et al. (2023) to one of my one datasets. I explain the principles behind the model pipeline, and how its use of positive and unlabelled (PU) data learning is a good fit for one of my *Drosophila* datasets. I train and validate the different models suggested in the pipeline and use the feature selection approach of the pipeline to investigate the contribution of my dataset's predictor variables to the model. I compare the results with the previous approaches of feature selection I devised. Finally, I select the best performing machine learning model to infer genes (from a pool of genes that have not been previously experimentally tested by our lab for the presence of mRNA in glial protrusions) that are highly expected to be localised in glial protrusions. Finally, I perform GO enrichment analysis to investigate if the top predicted candidates for glial localisation are enriched in localisation or transport related processes, as well as in glial or neuronal protrusion cellular components.

4.1 Introduction

Machine learning is a sub-field of science that is expanding at an accelerating pace. At the intersection of computer science and statistics (Pugliese et al., 2021), machine learning has the power to revolutionise the analysis of data in any field that it is applied to, including biology. However, there are many impediments to the application of machine learning in biological discovery. In the application of machine learning models to biology, the idea of a "one method fits all" approach does not hold true. Biological questions are complex and each question requires very distinct and specialised datasets to answer it. No single machine learning model can address all biological questions. In fact, most machine learning models are developed to address questions regarding specific types of datasets, for example two deep learning methods, convolutional NN and recurrent NN (Xu & Jackson, 2019). These were designed for different types of data, the

former for image recognition and the latter for speech recognition. Therefore each biological question needs to be approached using a different machine learning model, and often it is not obvious which one will produce the best results. Researchers should evaluate distinct models and report the results. Through careful consideration of the type and amount of data available they should choose the best performing one for the specific dataset.

Here, a range of machine learning models are considered. Decision tree-based models (Decision Trees (Rokach & Maimon, 2005), RFs (Breiman, 2001), Adaptive Boosting (Freund & Schapire, 1995)), NNs (NNs (Gurney, 1997) and Ensemble NNs (Hansen & Salamon, 1990)), probabilistic models (Naive Bayes (Webb, 2016)) and Linear Regression (Logistic and Linear SVMs (Cortes & Vapnik, 1995)) are the models examined. The available data have several attributes that limit their applicability to training machine learning models such as being small in size, imbalanced or have only positive labels available. In terms of the data characteristics an advantage of using Decision tree-based models is that they can handle smaller datasets and ones that contain both numerical and categorical data. However, they are not as effective with imbalanced datasets as they can be biased towards the majority class. An advantage of using RF is the fact that they give access to feature importance metrics that can aid in the explainability of the model. Feature importance metrics in RF include the permutation feature importance and mean decrease in impurity (or Gini impurity) (Nembrini et al., 2018). When permuting a feature, a feature is selected randomly and its values are randomly shuffled. The decrease in a model's performance is the value of the metric. The mean decrease in impurity measure tells us the probability of misclassifying an observation, which provides additional insights in understanding the model predictions.

NNs on the other hand require large datasets and benefit significantly from more data, therefore not all of my datasets are suitable for training NNs. However, they can model complex and non-linear relationships between the data which is an advantage as the combination of different sources of data required to built my datasets have likely incorporated such complex relationships between the features. Probabilistic models such as Naive Bayes can perform well

with smaller datasets and are very fast, but as they assume the features are independent of each other they can also miss capturing complex relationships between the data. SVMs are effective in high dimensional spaces which makes them suitable for the datasets at hand as they have high dimensionality in terms of features.

In this chapter, I aimed to understand better which machine learning methods could be used for my data. I tried different approaches, including unsupervised learning (dimensionality reduction and clustering) and supervised methods (RF, NNs, Naive bayes and more) and selected the best performing model. I also tried to incorporate feature selection techniques that provide explainable results regarding the importance and choice of features to be included in the model training. My final goal was to predict a list of candidate genes that their mRNA is highly likely to be present in glial protrusions in the *Drosophila* nervous system and test them for GO terms enrichment.

4.1.1 Aims and Objectives

1. Explore the highly dimensional datasets using unsupervised techniques, try to understand their structure and reduce the dimensionality of the datasets using PCA.
2. Using feature selection methods and feature importance measures, understand which features would contribute most to the model's prediction and hence which biological variables are most important to glial mRNA localisation.
3. Use the most suitable features to train and evaluate machine learning models to predict which mRNAs are likely to be localised in glial protrusions. Produce a list of candidate genes for further computational validation using GO enrichment methods or experimental validation using smFISH.

4.2 Materials and Methods

4.2.1 Data and data preparation

In Chapter 3 I created datasets ready to be used for machine learning applications for both *D. melanogaster* and *M. musculus*. Each dataset consists of the labels of localisation and the features

that I believe can describe localisation. I obtained and curated the features after detailed literature review of mechanisms contributing to mRNA localisation.

I split the data into a training and a testing set. For *Drosophila* the data included labels for 200 genes so I chose to split the labelled data by assigning 80% of the data to the training set and 20% of the data to the testing set. For *Mus*, the labelled dataset accounted for most of the genome therefore the dataset available for modelling was much bigger than the *Drosophila* dataset available for modelling. Hence I decided to assign 50% of data to the training set and 50% to the testing set so that I can test the machine learning model with a bigger pool of data. I assigned the data using the scikit-learn (Pedregosa et al., 2011) function of sampling without replacement, sklearn.utils.random.sample_without_replacement(). Throughout the experiments in this section, I used the same splitting on the training and testing dataset so that my experimental results are consistent with each other.

4.2.2 Supervised machine learning

In general, the goal of a machine learning algorithm is to identify the best hypothesis for a selected dataset. For a classification model which is part of supervised learning, the hypothesis is the identification of the class for an unknown sample (Carracedo-Reboredo et al., 2021). There are many supervised classification algorithms, such as Decision Trees (Rokach & Maimon, 2005), RF (Breiman, 2001), Naive Bayes (Webb, 2016) and NNs (Gurney, 1997), all of them used in multiple biological applications.

In this thesis, I have explored the aforementioned models, but mainly use RFs. There are several reasons for that. As highlighted in the introduction, NNs are usually used when there are large amounts of data with their features highly structured (clear relations between the features) (Greener et al., 2022b). In this project, some of the features have clear relations between them such as all the k-mers between them or the pfam domains between them. The groups containing related features are not related between them, for example the k-mer features are not related to the pfam features. Additionally, for the *Drosophila* datasets, the number of labelled data is quite small. Therefore NNs are likely not a good choice. Advantages of using RF include

that they can work well with smaller datasets, they deal well with missing values and outliers and are easy to interpret. They also often require minimal hyper parameter tuning (Greener et al., 2022b). Most importantly, they provide many measures to identify feature importance aiding to the understanding of which features contribute the most to the model and therefore define the outcomes and the predictions (Greener et al., 2022b). In section 4.3.3, I show how RF outperforms the other models in the predictions.

Throughout this thesis, I used the scikit-learn (Pedregosa et al., 2011) implementation of RF (sklearn.ensemble.RandomForestClassifier()), with default parameters. The well established RF variable importance measures (that are also defined as functions in python within scikit-learn), have limitations. In general, they are reliable measures to use in many applications for feature selection but when there is large variability in the features in terms of their scale or number of categories for categorical features they are not as sensible to use (Strobl et al., 2007). Additionally, some of the measures, such as Gini impurity (sklearn.ensemble.RandomForestClassifier(), with the parameter "criterion" set to "gini"), do not work well when there is multicollinearity within features. Many research groups have tried to address the issue of multicollinearity so that correlated features are dropped and errors propagated into the variable importance measures are limited. One example is the Variance Inflation Factor (VIF). VIF is used in multiple regression predictions to measure the amount of multicollinearity between explanatory variables (Marcoulides & Raykov, 2019) and therefore help in feature selection. Another measurement and the one I use in this thesis is provided by the python package rfpimp (Parr, n.d.), by the rfpimp.feature_dependence_matrix() function. The principle behind the measure, is that to identify if a feature *i* is dependent on other features, is to train a model to predict the feature *i*. By treating feature *i* as the target variable and all other features as the variables used to predict that, one can identify which features *i* is more dependent on. The rfpimp implementation (Parr, n.d.) is based on RFs.

I used this function on both the *Drosophila* and *Mus* datasets to see which features are most dependent on each other. My largest dataset is about 50,000*11,000 points, so this adds a lot of computational complexity when calculating the importance of the whole dataset using rfpimp.

I tested the function to see how much time it takes to run it based on the dimensionality of the dataset. I did so by reducing the dataset to different numbers of dimensions and measuring the time it takes to finish the calculation. I fitted a line to the data points and found that the algorithm's time complexity increases exponentially with time. I therefore found an optimal number of features (50 features) that when the rfpimp.feature_dependence_matrix() function was used would produce results in a reasonable amount of time.

I also split the features in groups as I did in Chapter 3. For example the Pfam related features were grouped together and the feature_dependence_matrix matrix was constructed for groups of 50 Pfam features. I did that for all features. I then dropped features that were at least 90% depended on other features. I then merged the remaining features and found the feature_dependence_matrix matrix. I did that iteratively until I narrowed down the features as much as possible. I present the results in the Results and Discussion section.

4.2.3 Dimensionality reduction

When PCA is applied to a dataset, it reduces the dimensions of the dataset by geometrically projecting them onto lower dimensions called PCs (Lever et al., 2017). Each PC is orthogonal to each other and they build a new set of coordinate axes. The first PC is selected so that it captures the maximum variance of the projected data and the second captures the maximum remaining variance (Lever et al., 2017). The goal of PCA is to provide a summarised version of the data using a smaller number of dimensions. However, when using PCA and selecting a limited number of PCs some variance amongst the data is inevitably lost. Additionally when using the whole number of PCs required to capture the full variance there is some loss of interpretability. This is because it is not possible to find out which features/dimensions contribute to each PC and information is therefore lost.

Here, I attempted to regain some of this information. The approach I devised consists of using the PCs obtained from performing PCA on a dataset and it is shown in Figure 4.1. Next, I used the PCs as a target variable in a predictive machine learning model trained using the dataset. Finally, after I trained the model I used the model's inbuilt functionality to calculate feature importances to

understand which features of the original dataset contribute the most to the training of the model. Those features are the ones that contribute the most to the first PC and also variance. Here, I was able to use the Gini impurity as each PC is orthogonal to each other and therefore there is no correlation between the PCs (Jolliffe & Cadima, 2016). As these are used as features and they are uncorrelated there is no concern about multicollinearity along the features.

More specifically, I used this approach for the mouse dataset. I applied PCA (sklearn.decomposition.PCA()) to a scaled dataset and got a reduced number of dimensions (Figure 4.1, Step 1). I then trained a RF Regressor using a scikit-learn (Pedregosa et al., 2011) implementation. The first PC was used as a target variable for the regressor (Figure 4.1, Step 2). The scaled dataset was used as the input to train the regressor. The feature importance function of the RF Regressor (sklearn.ensemble.RandomForestRegressor.feature_importances_) was used to assign an importance score to the features/dimensions contributing the most to the predictive model. The value calculated is based on impurity and the higher the value the more important the contribution of the feature. Next, I selected the features with the highest importance score and trained a RF Classifier using only this subset of features from the original dataset (Figure 4.1, Step 3). To calculate the importance of each feature, I used the Gini impurity metric (Figure 4.1, Step 2). When using the Gini impurity, each feature gets a percentage of importance value with all the importances summing up to 100 %. If the importance was evenly distributed over the ~10,000 features that would have been approximately 0.01 %. Therefore features with importance bigger than 1 % are considered features of high importance. I used the original localisation labels as the predictor variables and found the accuracy of the model (Figure 4.1, Step 3). Finally, I counted the number of features N with importance bigger than 1 %. I selected N random features and trained a model to see the accuracy when using randomly chosen features with no prior knowledge. I compared the accuracy of these two models. In results I report later, I found that for the mouse dataset, around a 100 PCs, capture the biggest increase in variance of the dataset. Therefore, I repeated the above procedure for a selected number of PCs, starting from the first PC going up to a 100 PCs (Figure 4.1, Step 4). I report the results of this analysis in the Results and Discussion section.

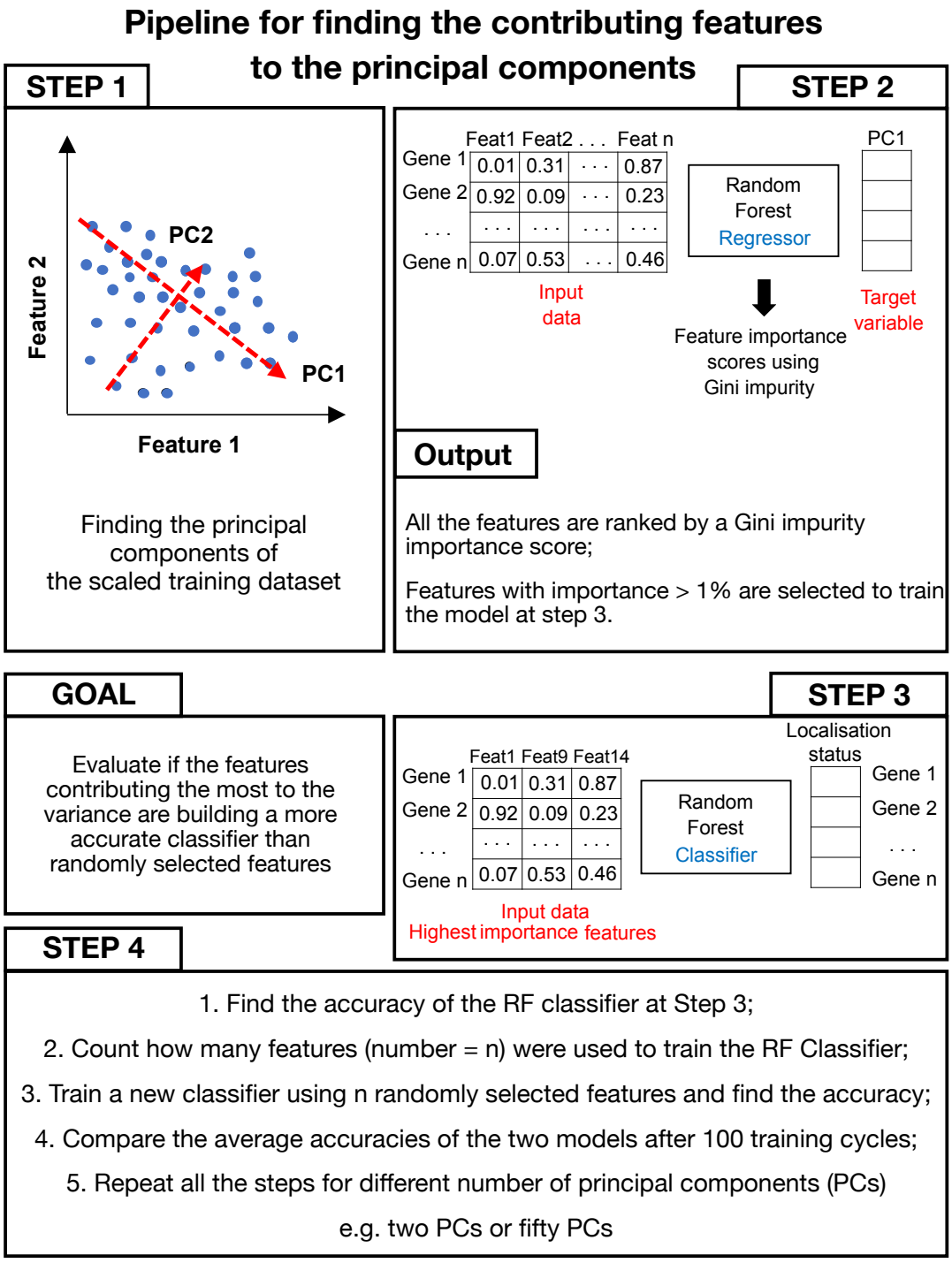


Figure 4.1: Pipeline for finding the features that contribute the most to the variance of a dataset. (Step1) Find the principal components of the training dataset. (Step2) Select the first *n* principal components and set them as a target variable in a RF regressor. Use the training dataset to train the regressor and find using the Gini impurity metric the most contributing features. (Step3) Select the features with impurity bigger than 1% to train a RF classifier to predict transcript localisation. (Step4) Find the accuracy of the RF classifier and then compare it with a RF classifier that has been trained with randomly selected features. Repeat the whole procedure, selecting different numbers of first *n* principal components.

4.2.4 Clustering techniques

Contrary to supervised learning, unsupervised learning does not require the data to be labelled. Clustering, an unsupervised machine learning method, seeks to uncover groups in data (Everitt, 2011) and also it finds patterns in data; whether they are there or not (Altman & Krzywinski, 2017).

The general approach used in clustering is to firstly find the similarity between the data and secondly to use the similarity to group the samples together into clusters. In this case the samples are genes. When samples cluster into well defined clusters it is very useful as it shows there is a natural partition between the data. However if these clusters correspond to samples' biological attributes that were not used in grouping the data, it is even more valuable (Altman & Krzywinski, 2017). For example, if my data were clustered together into two clusters and one of the clusters corresponded to localised genes and the other into not localised genes that would increase my trust in the validity of the clusters (Altman & Krzywinski, 2017) and I could use the same clustering approach to group data for which I do not know their localisation state.

Well defined clusters are those for which the inter-cluster distance is maximised (clusters further apart) and the intra-cluster distance minimised (more compact clusters) (Ikotun et al., 2023). The two major categories of clustering are agglomerative and partitional clustering. Examples of agglomerative clustering include linkage clustering (the outcome is visualised as a dendrogram) and examples of partitional clustering k-means (Altman & Krzywinski, 2017). Agglomerative clustering starts by treating each sample as one cluster and based on similarity merges different samples together. This iterative approach continues until one big cluster is formed. In contrast to agglomerative clustering, partitional clustering requires that the number of clusters k is known in advance and that the data are split into k clusters based on similarity (Altman & Krzywinski, 2017). Agglomerative clustering is usually computationally expensive and can be slow in contrast to partitional clustering which is faster and more scalable.

K-means, a partitional clustering algorithm, works iteratively. After the user specifies the number of clusters, the coordinates of the clusters (or centroids) are randomly selected from points from the dataset. Then a distance metric (Euclidean or Manhattan) is used to calculate

the distance of the dataset points from the centres of the clusters and then the point is assigned to the closest cluster. After the clusters are created the centre of the clusters is calculated again and the above procedure repeated until the centres of the clusters do not move - therefore reaching convergence at a local minimum.

In this work, I used k-means clustering (Lloyd, 1982) to investigate whether clustering can unveil the natural structure within my data. I used the scikit-learn implementation of k-means (sklearn.cluster) (sklearn.cluster.KMeans()) (Pedregosa et al., 2011). I firstly used the WCSS metric also known as the Elbow method to find the optimal number of clusters. It is known as the Elbow method because when plotting the WCSS value for different clusters the shape of the line is like an elbow. The optimal value of the number of clusters is found at the point where the two lines resembling an elbow meet.

This is necessary as the number of clusters in k-means needs to be defined in the beginning of the clustering. After finding the optimal number of clusters I applied the k-means clustering to the training set of both the *Mus* and *Drosophila* datasets. I applied this both to the full dataset (after scaling the data), to the full dataset after applying PCA to the dataset and the subset of the dataset that corresponds to the "important features" found in a previously described approach.

Unfortunately the clustering did not give any meaningful results nor provide any insights. There was no obvious separation of the localised versus not localised genes and the clusters were not well defined or separated from each other. This could be because k-means assumes that the clusters are of similar size and that they should be spherical. Spherical clusters assume that all the points have the same variance. Indeed from the results, I have seen that when clustering the datasets after applying PCA to the datasets the clusters are more well defined and separated. As there were no significant results and conclusions to be drawn, I present part of the results in Appendix E.

4.2.5 Adapting the approach for identifying candidate sleep regulators to identifying genes with potential to localise

After evaluating the existing literature, I concluded that very few groups have explored the concept of predicting localisation in the same manner that I have. For instance, while other groups have

focused on using a single data source or datatype (for example sequence k-mers only), I have integrated a wide range of datasets from different sources to predict localisation. Interestingly, during my literature search, I came across an approach which has similarities to my own method, however it is used to predict candidate genes that are sleep regulators (Y. Y. Lee et al., 2023).

Y. Y. Lee et al. (2023) curated a list of "sleep genes" by detailed literature review and used them as labels to train a machine learning model. They identified 109 genes as positive for being sleep regulators. As features, they used sleep-gene-associated functional information consisting of annotated gene sets, genome-wide profiling data and more. These data were sourced amongst others from the GO annotation database (The Gene Ontology Consortium et al., 2021), the Molecular signatures database (MSigDB) (Liberzon et al., 2011) and the Gene Expression Omnibus (GEO) (Edgar, 2002). These were processed to produce around 7,000 features which is a very similar number to the number of features I used in my project as well. Additionally, the type of information used is quite similar as GO annotations were used (likewise, I used them too) and TF target groups (similarly, I used RNA binding protein target sites).

Another advantage I spotted in this approach was the fact that they only curated genes related to sleep regulation and therefore positive labels but they had no information on genes that do not regulate sleep and therefore no negative labels. On that account they needed to create a machine learning model that can take this into consideration when making predictions. In Chapter 3 section 3.3.1, I described three lists of genes that were labelled for localisation in glia. List D2 (Table 3.3) describes gene localisation in *Drosophila* for 1,700 genes but has no information for genes not related to localisation. Based on these facts I decided to use Lee's et al. method (Y. Y. Lee et al., 2023) to predict localisation for the rest of the *Drosophila* genome based on training by the 1,700 localised genes. Additionally the model is written in Python and it is freely available on GitHub, here (`https://github.com/yyenglee/ml-sleep/tree/python`).

Learning from only positive data is known in literature as "learning from Positive and Unlabelled data" (Bekker & Davis, 2020; X.-L. Li & Liu, 2005). The non labelled data are treated as negative labels. PU learning is a special case of binary classification (Bekker & Davis,

2020). Y. Y. Lee et al. (2023) trained ensembles of multiple weak classifiers. The classifiers are considered weak as they are trained using PU data. The unlabelled data are treated as negative labels while in reality could have been positive as well. The negative labels are expected to contain a mixture of positive and negative labels and therefore the classifiers are not as strong. The very many weakly trained classifiers are averaged and the predictions are made using the average performance of the classifiers (Y. Y. Lee et al., 2023).

I downloaded their implementation of the model and the example input and output data available. I tested the pipeline and verified that all the separate modules and the different components of the pipeline were working as anticipated (integration testing). I also tested that when running the pipeline the output data matched the expected output data (functional testing). After testing the pipeline, I modified the structure of my data to meet the requirements of the python code, named "ml-sleep-python". "ml-sleep-python" was designed to accept mouse and human genes so I adapted it to be able to process fly genes. A uniqueness of the "ml-sleep-python" model is that it uses the "Evidence Factors" (EFs) (Y. Y. Lee et al., 2023) to assess the usefulness of a feature in prediction. EFs are based on "Bayes Factors" (Kass & Raftery, 1995). A Bayes Factor is the ratio of the likelihood of one hypothesis to the likelihood of another. Here, EFs assess the degree of genes of interest over-representation in each of these features. Only features with EFs higher than a threshold are used to train the model. I decided to use the same threshold as Y. Y. Lee et al. (2023) which is 3. The EFs of highly pairwise correlated features are compared and the feature with the higher value is kept while the other features are dropped from the training. I defined highly pairwise correlated features as features with correlation higher than 90%, for consistency. This is the value I used in earlier work presented in this Thesis to assess high correlation between pairs of features.

I also adjusted the weight of the positive labels (the genes that represent the localisation at the glial protrusions), from 100 to 4. The number of "sleep genes" is 109 and there are roughly 20,000 genes used to train the "ml-sleep-python" model. This number of positive versus unlabelled data is highly skewed and Y. Y. Lee et al. (2023) increased the weight of the positive labels by

100 to make the positive versus unlabelled data roughly equivalent (109 times 100 ~ 10,000), with each class to have the size of half the full dataset. In my localisation dataset there 1,700 genes that are considered localised and an approximate total of 1,4000 genes. As 1,700 times 4 is approximately 7,000, this equates to half the size of the complete gene set.

Eight machine learning models were evaluated to find which one is the best predictor. These included Naive Bayes (Webb, 2016), Logistics SVMs, Linear SVMs (Cortes & Vapnik, 1995), Decision trees (Rokach & Maimon, 2005), RFs (Breiman, 2001), Adaptive boosting (Freund & Schapire, 1995), NNs (Gurney, 1997) and Ensemble NNs (Hansen & Salamon, 1990).

Sensitivity and PU adjusted F1 score (Bekker & Davis, 2020; Y. Y. Lee et al., 2023; Luo et al., 2023) were used to assess the models performances. According to the literature (Bekker & Davis, 2020), sensitivity can be estimated using the established way for positive and unlabelled data but the F1 score cannot. F1 score for positive and unlabelled data can be estimated by tweaking the original F1 score (Luo et al., 2023) to be equal to the square of sensitivity divided by the portion of positive prediction (Luo et al., 2023) compared to all the predictions, as seen below:

$$\text{F1-score}^{PU} = \frac{\text{sensitivity}}{\text{percentage of positive prediction}} \tag{4.1}$$

Sensitivity or true positive rate, is equal to the number of samples correctly predicted as positive divided by the total number of positive data.

$$\text{sensitivity} = \frac{TP}{TP+FN} \tag{4.2}$$

These measures were evaluated in three ways: 1. Using the training set of labels and sub-setting smaller proportions of labels to train the models and to predict; 2. Using the validation set of labels to predict in an already trained model; 3. Using a randomly shuffled set of labels, i.e. shuffling the labels describing positive and negative samples so that the labels were no longer always representative.

The finalised models were also assessed using the receiver operating characteristic (ROC) curve and the area under that curve (AUC). The ROC curve is a measure of the true positive

rate against the false positive rate and additionally the AUC can indicate the performance of the model. The true positive rate was defined earlier (same as sensitivity) and the false positive rate is calculated using the following equation:

$$\text{false positive rate} = \frac{FP}{FP+TN} \tag{4.3}$$

and it is equal to the number of samples falsely predicted as positive divided by the total number of negative data.

For the "ml-sleep-python" model and based on the above measurements, RF was deemed the best model to use. I also evaluated the same models based on my data using the above metrics and the results of the comparison are presented in section 4.3.3 together with the predicted candidate localised genes. The adapted model with example input and output data are available in my GitHub repository in the form of an archive file (`https://github.com/mkiourlappou/thesis-project/blob/main/ml-sleep-python-drosophila.zip`).

4.2.6 Pathway enrichment analysis using REACTOME pathways data

As mentioned in Chapter 3 section 3.2.2, the REACTOME repository (Jassal et al., 2019; Sidiropoulos et al., 2017) contains a knowledge base of biomolecular pathways and aids our understanding of biological processes. I used the pipeline described below to assess the pathway enrichment of the predicted localised genes.

Firstly, to access the REACTOME pathways for both species, the gene to pathway query from FlyMine (`https://www.flymine.org/flymine/templates/Gene_Pathway`, last accessed on 20/04/2023) and MouseMine (`https://www.mousemine.org/mousemine/templates.do`, last accessed on 20/04/2023) was used to extract the REACTOME ID annotations for each gene.

The REACTOME pathway hierarchical relationship between terms was accessed through here[1]. It was used to find the groups of pathways that were related hierarchically and reduce the levels of hierarchy to a higher level, as it was observed that some pathways only had very

[1] `https://reactome.org/download/current/ReactomePathwaysRelation.txt`, last accessed on 20/04/2023

few genes assigned to them. Like in the case of GO annotations, some children pathway terms were of too high a level, providing very little extra information compared to their parent term. Accordingly, all the pathways and the genes corresponding to them were grouped to the second hierarchical level, called the "sub-pathways" level. Information about the "top-level pathway" (first hierarchical) level was retained. The "sub-pathways" level was used in the enrichment analysis to calculate the enrichment and the "top-level pathway" to group the "sub-pathways" together for highlighting the similarity of the pathways.

The enrichment analysis was performed using R code I adapted, written by Dr. Jeff Lee. This is available at the GitHub repository here (`https://github.com/jefflee1103/Gala2023_glia-localised-RNA/blob/main/Rscript/reactome-pathway-association_enrichment.R`) and it is published in Gala et al. (2022). The hypergeometric test (R:phyper) was utilised to assess the enrichment of the REACTOME pathway terms for the transcripts localised followed by Bonferroni's method of multiple hypothesis correction (R: p.adjust(pvalue, method = "bonferroni")).

The results of the enrichment analysis of the candidate genes compared to the rest of the genome are presented in the Results and Discussion section.

4.3 Results and Discussion

4.3.1 Exploring the correlation between a dataset's features using machine learning approaches

As discussed in this Chapter and in Chapter 3, correlation and collinearity between features might introduce noise to the model and redundant information. Here I tried to explore correlation and collinearity between features with machine learning approaches. These types of correlation methods are types of "wrapper" feature selection techniques as they are based on machine learning algorithms, as described in Chapter 3 in section 3.2.4. I used the rfpimp package and explored the dependence of groups of features on each other. As expected there was a high correlation between features within the same group. I used the package on the training set of both *Mus* and *Drosophila* datasets. As the *Drosophila* training set is quite small (<200 genes), I found many

spurious correlations between the features as for this part of the dataset some features (specifically the Pfam andd RBP frequency features) only had 0 values. If a bigger part of the dataset was used perhaps there would be more truthful correlation but I used the training set only. This was because otherwise there might be data leakage to the testing set and also introduction of bias from the correlation analysis before training the model, to the trained model.

I therefore present the correlation analysis results for the *Mus* dataset for which there were no obvious issues regarding spurious correlation as it is much bigger compared to the *Drosophila* dataset. The rfpimp package (Parr, n.d.) provides a dependence value for each feature. I used 90% as cutoff for feature dependence and only kept features with lower dependence than that. In Table 4.1 I show how many features were selected as the most important for each group of features and found to be relatively independent to each other.

Table 4.1: Table comparing the number of features before and after the dependence analysis between features of the same group

Organism	Features grouping	Size of grouping	Independent Features
Mus	Length related	13	5
Mus	5' UTR k-mers	1344	1328
Mus	3' UTR k-mers	1344	1342
Mus	Coding sequence k-mers	1344	1290
Mus	Pfam domains	6598	3
Mus	RBP target sites	453	328
Mus	MF GO terms	78	4
Mus	BP GO terms	131	3
Mus	CC GO terms	101	6

Comparing the results shown in Table 4.1 to the correlation results shown in Chapter 3 we can see that finding correlation between the features or dependence for this specific set of features is more systematic using machine learning approaches although more expensive computationally. I arrived at this conclusion as in the results shown in Table 4.1 more features and more groups of features are found as dependent and correlated compared to the results in Table 3.5. If there was enough time and computational power it would have been better to be able to apply the rfpimp feature dependence matrix to the whole set of features to examine dependence more completely rather than per group.

4.3.2 Dimensionality reduction using PCA

I performed PCA on the training set for both *Drosophila* and *Mus* datasets to examine the dimensionality of the datasets. Both datasets were standardised using a scikit-learn (Pedregosa et al., 2011) function to scale to zero mean and unit variance before transforming them using PCA (sklearn.preprocessing.StandardScaler()).

When using PCA, the number of PCs produced can be at most the minimum of the number of observations and the number of variables in a dataset (Nguyen & Holmes, 2019). The observations in my datasets correspond to genes (rows) and the variables to the features (columns). In this case the number of genes in each dataset should not be less than the number of features. Both my datasets have more than 11,000 features. The *Drosophila* training set has 160 genes therefore applying PCA on the whole *Drosophila* training set does not produce useful results. This is evident in Figures 4.2 A, B, C and D. Figure 4.2 A shows the results of PCA analysis performed on the *Drosophila* training set and Figure B on the *Mus* training set. Both panels of the figure show the analysis in the form of explained variance captured per PC. The results are shown per component explained variance captured and cumulative explained variance. In Figure A, the variance is shown to be fully captured by 160 components which is the same number as the number of genes included in this training set. In Figure B, we can see approximately 11,000 PCs, which is a similar number to the number of features in the mouse training dataset. Additionally, full variance is captured when around 7,000 PCs are used as we can see the red line plateauing then. Thus we can conclude that the number of dimensions needed to be included for the mouse dataset can be reduced from ~11,000 to ~7,000. Figures 4.2 C and D show the PCA analysis on the whole *Drosophila* training set (C) and the whole *Mouse* training set (D). Each dot represents one gene, the genes that are localised in glial protrusions visualised with magenta and the ones that are not localised with cyan. In both Figures (C) and (D) it is evident that the first two principal components do not separate each dataset into distinct groups as the data points are clustered together.

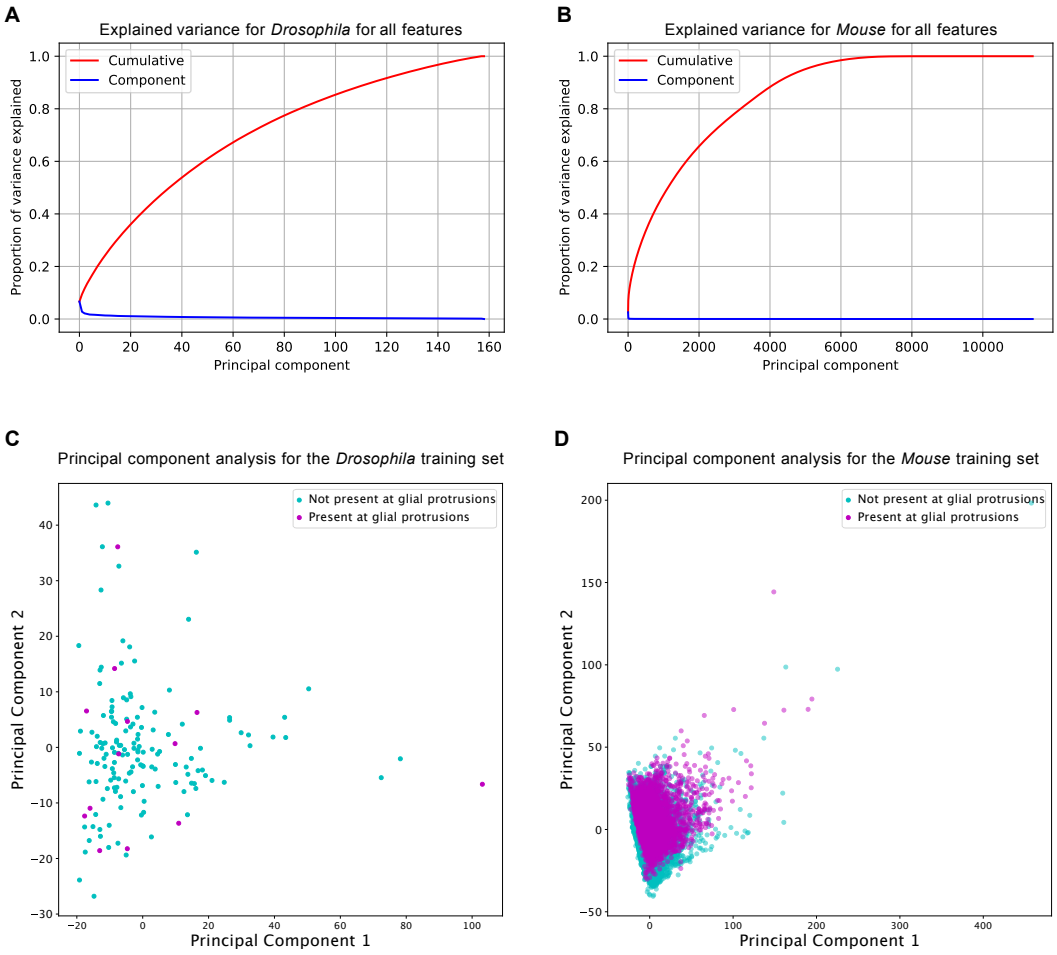


Figure 4.2: PCA explained variance and PCA analysis for all features for *Drosophila* and *Mus* training data. (A-B) The PCA explained variance by each component is shown by the blue line. The cumulative explained variance explained by the first n components is shown by the red line. (A) Variance explained for *Drosophila* (B) Variance explained for *Mus*. PCA analysis for all features for the (C) *Drosophila* and (D) *Mus* training data. The genes that are known to be present in protrusions are shown with magenta and the ones that are not present in protrusions with cyan.

In Chapter 3, I investigated the collinearity between pairs of features that originated from the same source of data. I found that there were highly correlated features in the Pfam and RBP target site feature families. For the k-mers feature family, there were highly correlated features only in the mouse coding sequence k-mers. More specifically the AAA 3mer was highly correlated (more

than 90%) with the AAAA 4mer. I decided to perform PCA on the k-mers features families to reduce their dimensionality (highly dimensional features with 1,344 dimensions for each part of the sequence). I already performed analysis to find redundant features using correlation and I did find highly correlated features within the Pfam and RBP target site feature families, but not within the k-mer families. Therefore I thought it was more necessary to explore the dimensionality of k-mers.

Figure 4.3 shows the results of the PCA application on the portion of *Mus* dataset referring to k-mers in the 5' UTR (Figure 4.3 A-C) and 3' UTR (Figure 4.3 D-F). Figure 4.4 shows the explained variance of the PCA application to the portion of the dataset referring to k-mers in the coding sequence of *Mus*.

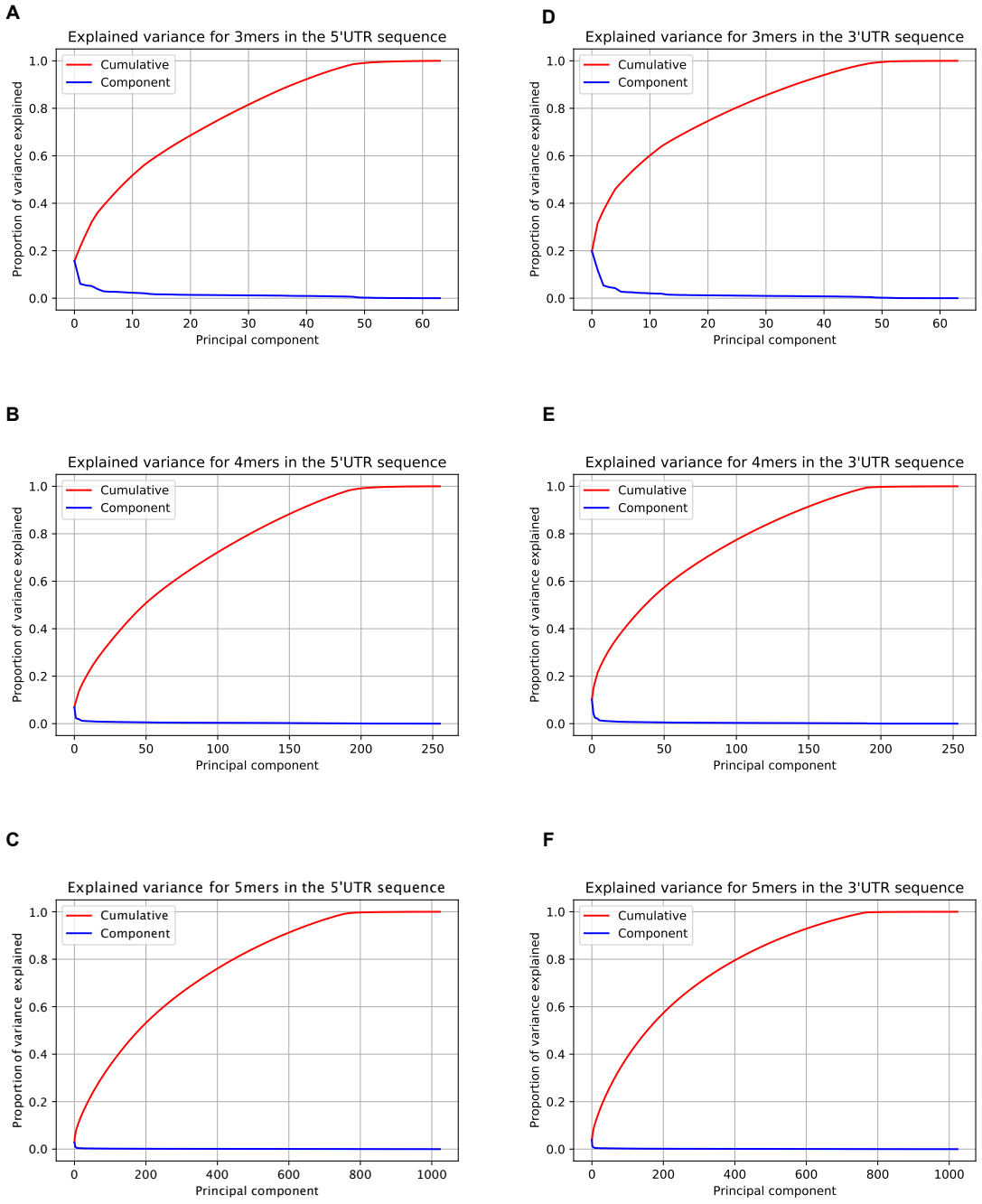


Figure 4.3: PCA explained variance for k-mers in the 5' UTR and 3' UTR sequence for *Mus* training data. The explained variance by each component is shown by the blue line. The cumulative explained variance explained by the first *n* components is shown by the red line. (A) Explained variance for 3mers in the 5' UTR sequence. (B) Explained variance for 4mers in the 5' UTR sequence. (C) Explained variance for 5mers in the 5' UTR sequence. (D) Explained variance for 3mers in the 3' UTR sequence. (E) Explained variance for 4mers in the 3' UTR sequence. (F) Explained variance for 5mers in the 3' UTR sequence.

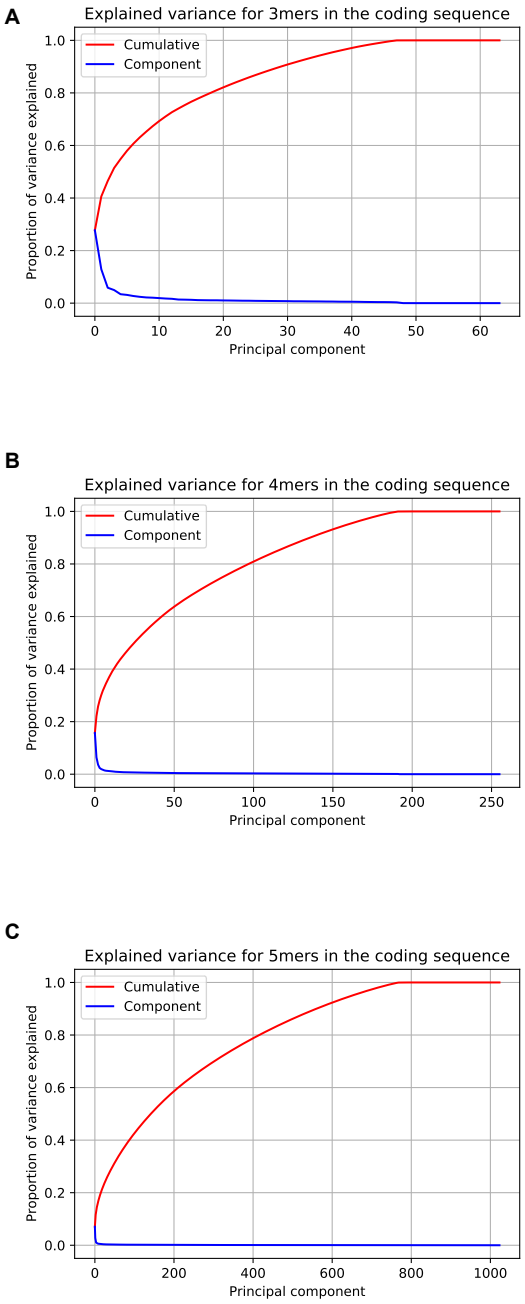


Figure 4.4: PCA explained variance for k-mers in the coding sequence for *Mus* training data. The explained variance by each component is shown by the blue line. The cumulative explained variance explained by the first n components is shown by the red line. (A) Explained variance for 3mers in the coding sequence. (B) Explained variance for 4mers in the coding sequence. (C) Explained variance for 5mers in the coding sequence.

As we can observe in Figures 4.3 and 4.4, most of the variance can be captured by the first 80% of the PCs in all cases therefore we can reduce the dimensionality of the k-mer features by 20%. Likewise in Figures 4.5 we can see that for *Drosophila* too the full amount of explained variance is captured by the first 48 components (~ 80%). For *Drosophila*, for reasons explained earlier, I could only use PCA to reduce the dimensions of the 3mers for all regions of the sequence as the dimensions of the 3mers features is 64 which is smaller than the number of observations (genes) as required for reliable PCA results.

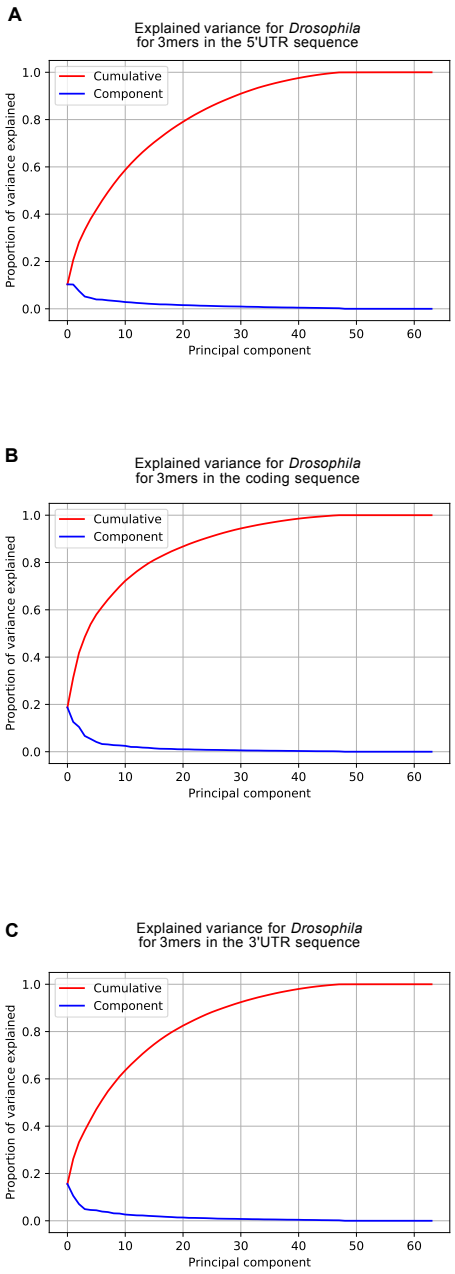


Figure 4.5: PCA explained variance for 3mers in the sequence for *Drosophila* training data. The explained variance by each component is shown by the blue line. The cumulative explained variance explained by the first n components is shown by the red line. (A) Explained variance for 3mers in the 5' UTR sequence. (B) Explained variance for 3mers in the coding sequence. (C) Explained variance for 3mers in the 3' UTR sequence.

The fact that we can reduce the dimensionality to about 80% for the k-mer features is useful to know. Minimising the number of features contributes to the reduction of the computation time as training the algorithm for prediction will be faster when fewer features are used. If a small dataset is used (less than a hundred features), reducing the number of features might result in a significant loss of information. However, as the total number of k-mer features is in the scale of thousands, a 20% reduction in the number of features still leaves a significant and informative number of features. On the other hand, if we use the PCs instead of the actual features, we sacrifice some of the explainability we could get out of our model regarding our features.

As discussed in the Materials and Methods section, I used the RF's Gini impurity metric to understand which features contribute the most to the PCs (also shown in Figure 4.1). When using the Gini impurity metric, each feature gets a percentage of that value. As there are more than 10,000 features, if that percentage is distributed equally it would be very small. Therefore I define the highest contributing features as the ones that have Gini impurity value bigger than 1%. In Figure 4.6, I present the highest contributing features, when examining the first n PCs.

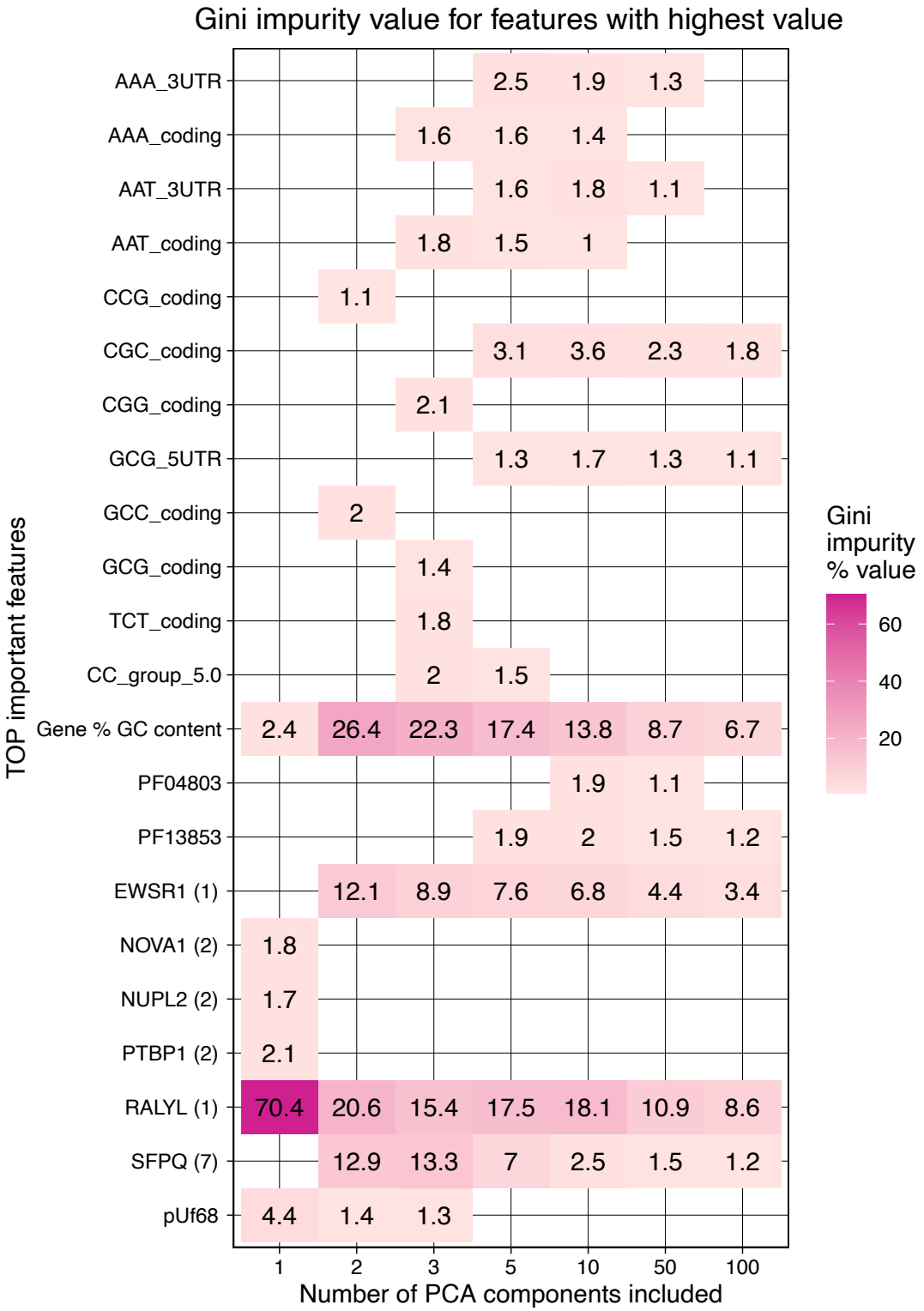


Figure 4.6: A heatmap showing the Gini impurity of top contributing features per cumulative number of PCs considered. Each row represents a feature, and the values on each tile show the percentage contribution to the variance captured by the first *n* PCs considered. Each column is focused on a different selection of PCs and the tiles along each column represent the values of Gini impurity of the corresponding feature. Only features that have a non zero value contribute significantly to the variance of the PCs examined.

In Figure 4.6, we can see that one of the top contributing features is the CC_group_5.0. This corresponds to a CC GO terms grouping I created. The group consists of the following two terms: 1. postsynaptic specialisation 2. postsynaptic density. This feature is one of the features that contributes the most to the variance of the whole dataset. The two terms, "postsynaptic specialisation" and "postsynaptic density" show clear support of the fact that the dataset I built is related to mRNA localisation in glial processes. Postsynaptic density is related to peripheral glia as glia modulate the processes of the synapses (synapses are comprised of presynaptic and postsynaptic specialisations in this concept) (Eroglu & Barres, 2010). Glia express receptors for neurotransmitters, allowing the neurotransmitters to communicate with synaptic functions (Eroglu & Barres, 2010). Therefore there is an extensive association of peripheral glial with the postsynaptic density and specialisation. Additionally, RALYL, the RBP target site found to contribute the most to the first PC is known from literature to participate in distal mechanisms of polarised cells, such as neurons (Rossi et al., 2017), therefore it is also a good indication that is obviously relevant to localisation features that capture the variance of the dataset.

This means that if I were to train a machine learning model with only these features, they would likely be very informative as some of the features have obvious relations to mRNA localisation in glial cells. I examined that by training a model with only the highlighted features and I report the results of this analysis in Figure 4.7. As I described in the Methods section, section 4.2.3, I selected the features to train each new RF model based on their contribution (Gini impurity value) when training an RF regressor (described in section 4.2.3 paragraph 3 and in Figure 4.1). Each combination of features shown in the columns of the heatmap in Figure 4.6 that has Gini impurity larger than one was used to train a RF classifier and the accuracy of the models is shown in Figure 4.7 (A) and their sensitivity in Figure 4.7 (C). For each combination of features I trained the RF model one hundred times. In panel (A) I show the mean accuracy (out-of-bag score) calculated when using the testing and training datasets in the form of boxplots to capture the distribution of the accuracy for the one hundred models trained. In panel (C) I show the mean sensitivity calculated when using the trained algorithm to predict the classes of the test dataset in the form of boxplots (sensitivity can only be found for the testing set). For each set of feature

combinations shown in Figure 4.6 the corresponding distribution of accuracy for one hundred RF classifiers trained using these features are shown in Figure 4.7 (A). The x-axes in both Figure 4.6 and Figure 4.7 refer to the number n of the first PCs the specific combinations of features capture the highest amount variance of. As explained earlier the variance here is captured by the features that have Gini impurity value bigger than one. Furthermore, I wanted to compare the accuracy and the sensitivity of a model trained with the features that capture the highest impurity versus a model that is trained with randomly selected features. To do this, I kept the number of features used in each experiment constant (as shown in Figure 4.1). For example, based on Figure 4.6, for one PC there are six features contributing and for two PCs there are seven. Therefore, for one PC I used six randomly selected features to train an RF model and for two PCs I used seven randomly selected features. The accuracy is shown in Figure 4.7 (B) for one hundred RF classifiers trained for each cumulative PC. If we compare the accuracy shown in each boxplot for each PC in Panel (A) with Panel (B) we can see that the accuracy is consistently higher when examining the sets of features with Gini impurity larger than one compared to randomly selected features. Additionally, if we compare the mean sensitivity shown in each boxplot in Panels (C) and (D) the sensitivity is also considerably higher for each PC when comparing the high Gini impurity selected features versus the randomly selected features.

The mean accuracy (out-of-bag score) of a RF classifier trained using all the features one hundred times is 0.77 for the training set and 0.77 for the testing set. The mean sensitivity for the testing set is 0.50. The accuracy using all the features is slightly higher both for the training and the testing sets when compared with the mean accuracy when using the most important features as shown in Figure 4.7 (A). The sensitivity however (Figure 4.7 (C)) is higher for the features selected based on variance calculations related to the 5, 10, 50 and 100 first PCs. This shows that this feature selection method provides improvement in the sensitivity and it is faster computationally.

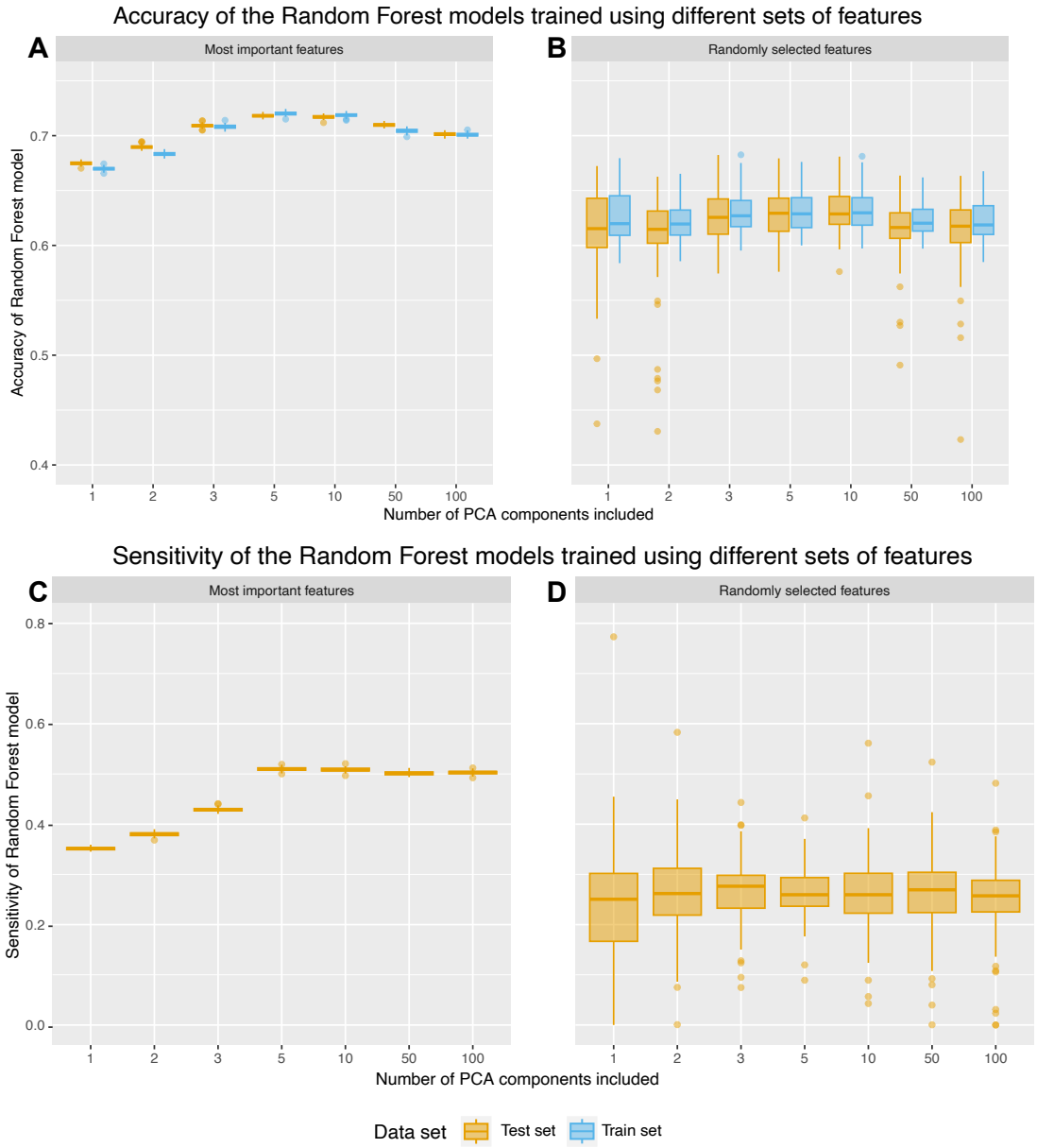


Figure 4.7: A comparison of the accuracy and sensitivity of the RF classifier, trained using importance features versus randomly selected features. Both Panels (A) and (B) show the RF classifier mean accuracy, for each set of data. The boxplots show the mean accuracy values over 100 iterations of training. Boxplots in blue show the accuracy calculated using out-of-bag score for the training data and boxplots in yellow the accuracy when the trained model was tested using the testing set. Panel (A) shows the accuracy when trained with the highest contributing features to the variance of n number of PCs. Panel (B) shows the accuracy when randomly selected features were used. For each n number of PCs the number of randomly selected features is the same as the number of important features used in Panel (A). Similarly Panels (C) and (D) show the mean sensitivity of the RF classifier for the testing dataset, over 100 iterations of prediction. Panel (C) shows the mean sensitivity when the model is trained with the highest contributing features to the variance of n number of PCs. Panel (D) shows the sensitivity when randomly selected features were used. For each n number of PCs the number of randomly selected features is the same as the number of important features used in Panel (D).

When I applied PCA to the subset of features that were the most contributing, the first few PCs did indeed capture most of the variance. This was expected because since we found them contributing most to the variance we would expect the features to be transformable using PCA to a dataset with fewer dimensions.

4.3.3 Candidate gene selection by adapting a sleep-gene-candidates identification approach

As described in the corresponding methods section, I used a model found in the literature regarding predicting candidate sleep genes (Y. Y. Lee et al., 2023) and adapted it to my *Drosophila* dataset. The model is written in python and is a supervised predictive machine learning model. As labels, I used the 1,700 genes found by Dr Jeff Lee, expected to be localised as they are high confidence homologs of localised genes in glia mammalian transcripts. Below, I discuss the results found from adapting the "ml-sleep-python" pipeline (Y. Y. Lee et al., 2023).

I used the ml-sleep-python pipeline implementation of EFs to find if any of the features I constructed is enriched in the 1,700 localised genes compared to the unlabelled data (the rest of the protein-coding genome). Features with EFs bigger than 3 were considered as positive evidence. Pairs of features with correlation bigger than 0.9 were further examined, and the feature within the pair with the lowest EFs was dropped. In the table below, Table 4.2, I provide the features with EFs value bigger than 3 and not dropped. These features are used to train and evaluate the machine learning models as well as making predictions.

Table 4.2: Table with feature names that we used to train the machine learning model and the Evidence Factor value for each feature. All the features presented here are RBP target sites. When there are multiple motifs per RBP this is indicated with the number in the parenthesis. How the Evidence Factor values are calculated is explained in the Materials and Methods.

Feature name	Evidence Factors
ELAVL1 (4)	3.14
ENSXETG00000012802	3.06
HNRNPD (1)	3.54
IGF2BP2 (5)	3.54
ILF2 (3)	3.08
KHSRP (1)	3.48
Khdrbs1 (1)	3.19
MBNL1 (1)	3.26

Feature name	Evidence Factors
PRR3 (3)	3.06
RBM47 (1)	3.13
RBM4B (1)	3.11
RBM4B (4)	3.01
RBM5 (2)	3.72
SF1 (3)	3.19
SNRNP70	3.03
TRA2A (1)	3.70
TRNAU1AP (1)	3.79
ZC3H10	3.01
ZCRB1 (1)	3.27
orb2	3.04
qkr58E1	3.4
sup26CONSTRUCT	3.57

Interestingly, all the features retained and used to train this model are all RNA binding protein (RBP) target sites. Most of the RBP target sites have high EFs value and about a hundred have an EFs value larger than 3. I decided to follow the same approach Y. Y. Lee et al. (2023) have used, which is only selecting features with EFs value higher than 3. This leads to many of the original features that I mined to not be used at all in the prediction.

The next step in the pipeline was to use these features to train and evaluate all the models considered to be used for the prediction. The results of the evaluation are shown in Figure 4.8. Eight models were evaluated and the metrics used to evaluate them was Sensitivity and F1 score adjusted for PU data. Sensitivity measurements are shown in panel A of Figure 4.8 and the adjusted F1 score in panel B of Figure 4.8. Panel C of Figure 4.8 shows the ROC curves for all models. To calculate the false positive rate, the unlabelled data were treated as negatively labelled data, i.e. that the genes that are unlabelled are not localised. The AUC is also shown in the legend of the graph and it is an indicator of the performance of the machine learning model, with larger area under the curve meaning a better performing model.

Sensitivity is a measure of how well a machine learning model does in predicting positive cases correctly. The F1 score is a metric that measures the model's accuracy. Therefore the selection criteria for the machine learning model should be that both sensitivity and the adjusted

F1 score are as high as possible. To evaluate the models, I used both the training set of genes and the validation set of genes. When training a model there are inbuilt functions to separate the training set of genes into parts of randomly selected genes. The first part is used to train the model and the second part to evaluate the predictions of the model. These are split using different ratios, starting with 80% training and 20% predicting, and going to 20% training and 80% predicting, in increments of 10%s. The ratios are shown with the different shapes of points in Figure 4.8A and B, and the different models are shown with different colours. On top of each graph, we can see the mean value of the metric evaluated for each model for all the ratios of the training to prediction sets for all iterations. The random-shuffle labels are labels that were shuffled and assigned randomly to genes irrespective of the actual trait of the gene, if it is localised or not. This is so that we check how the model performs when it is fed with noise. Therefore the selected model should have both very low sensitivity and adjusted F1-score.

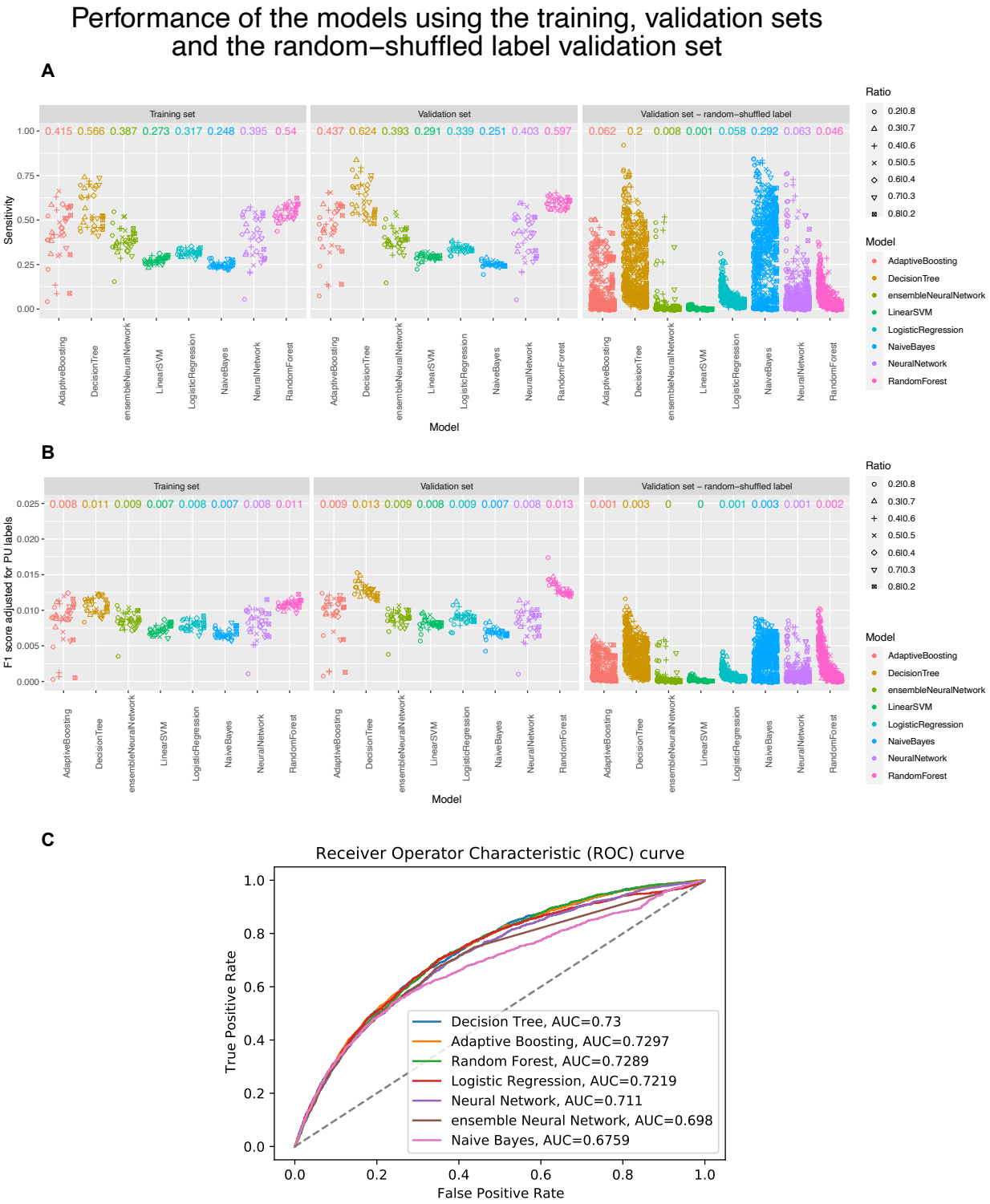


Figure 4.8: Evaluation of models performance. Sensitivity (A) and adjusted F1 score (B) were used to evaluate the performance of 8 machine learning models: Adaptive Boosting, Decision Tree, Ensemble NN, Linear Support Vector Machines, Logistic Regression, Naive Bayes, NN and RF. Sensitivity and adjusted F1 score were evaluated for 7 ratios of training to validation/prediction sets and each training cycle for the models was repeated 10 times for each ratio. On top of the graphs, the mean performance of each model over all the ratios and iterations is shown. Each model was evaluated using three different sets of labels, the training set, the validation set and a validation set where the labels for the genes were randomly shuffled so that part of them did not represent the true label of a gene. (C) Receiver operating characteristic (ROC) curves for all tested models. Areas under the curve (AUC) for each model are listed in the legend.

As seen in Figure 4.8, the model with the highest sensitivity and adjusted F1 score for the training and validation set is the RF (sensitivity = 0.54, 0.597 and F1 score = 0.011, 0.013 respectively) and the Decision Tree (sensitivity = 0.566, 0.5624 and F1 score = 0.011, 0.013 respectively). However, if I take into account the evaluation of the metric for the random-shuffled labels, the models with the lowest values are RF, Ensemble NNs and Linear SVMs. As a result, I decided to use the RF model to make predictions for the rest of the *Drosophila* genome. RF was also the model found to be the best performing in the Y. Y. Lee et al. (2023) pipeline. In the Y. Y. Lee et al. (2023) pipeline when using the training set of labels, for RF the sensitivity was 0.964 and the F1 score was 98.21. When using the validation set, the sensitivity was 0.433 and the F1 score was 24.85. Finally when using the randomly shuffled labels the sensitivity was 0 and the F1 score was 0.007. For the training set the sensitivity and the F1 score are higher indicating better performing models. However the sensitivity for the validation set is lower in the Y. Y. Lee et al. (2023) pipeline compared to my pipeline which shows that my model is performing better for never-before-seen data. In both pipelines the F1 score is very low for the randomly shuffled labels. The reason for these discrepancies could be perhaps the fact that default parameters were used for each classifier algorithm and although this has worked better for their dataset did not work as well for mine. As a result, fine-tuning could have been used to improve the performance of the models if time permitted.

I also trained the models using all features, i.e. not using the EF step to filter and select features. I did this to compare the results and investigate whether the feature selection step improves the results in addition to limiting the computational complexity. I present the results in Figure 4.9. As seen in Figure 4.9 panel A, the sensitivity for the training set of labels using all features is on average the same as for the sensitivity for the models trained using fewer features (Figure 4.8). However, for the models trained using all features the sensitivity for the validation set and the random-shuffled label set are very high, for all models excluding Ensemble NN, NN and Naive Bayes is one. Firstly, ideally we would like the sensitivity of the random-shuffled label set to be very low as these are randomly assigned labels, but instead they are one. This in addition to the fact that the validation set exhibits very high sensitivity while the training set does not

display a similarly high level suggests that the models may be overfitted. Sensitivity as high as one can be achieved when the classifier classifies all the instances as positive as sensitivity measures the true positive rate. This shows that the model is less robust to the addition of large amounts of noise, if we treat the thousands of added features as noise (compared to the fewer used when the EF step was used to select features).

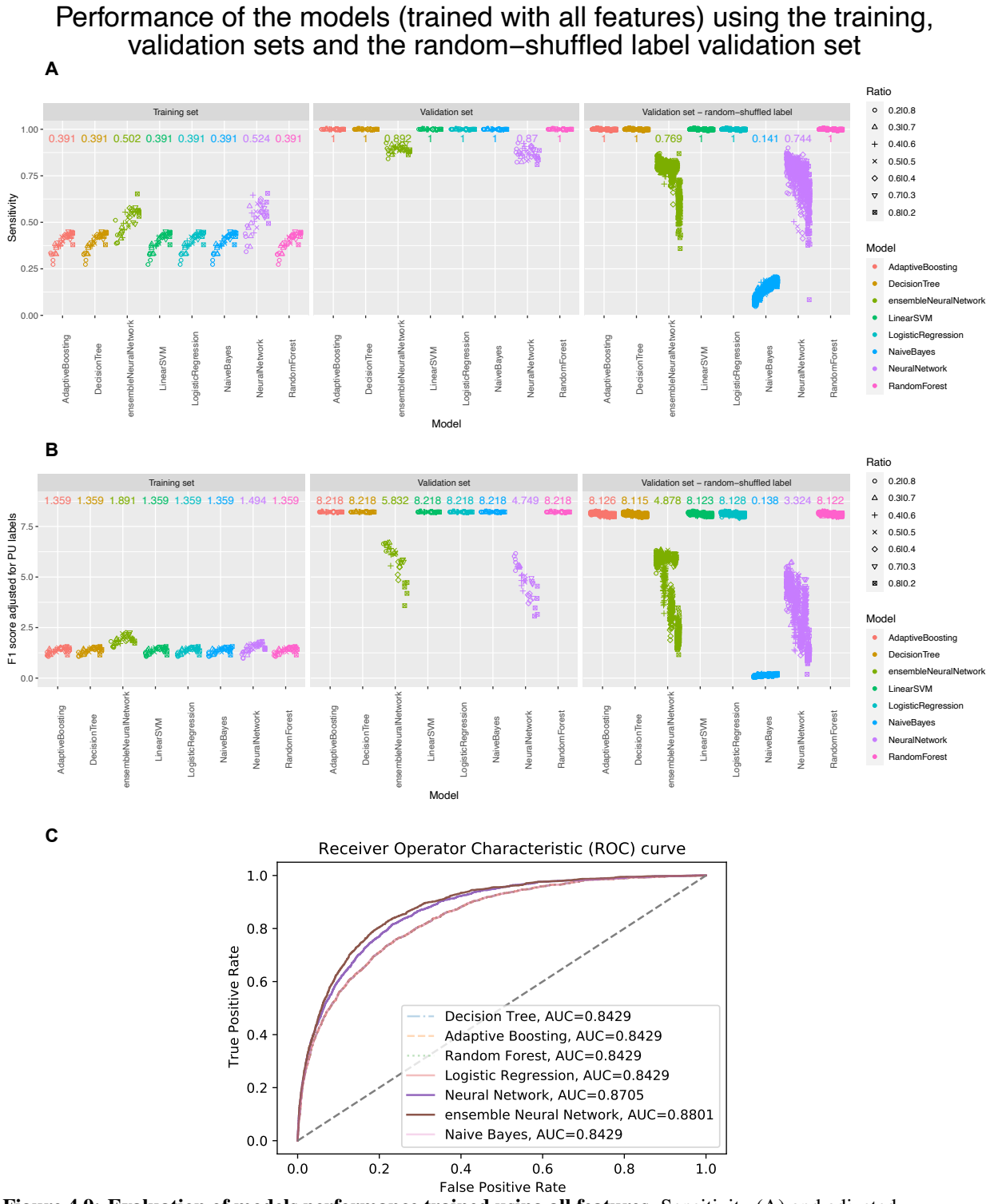


Figure 4.9: Evaluation of models performance trained using all features. Sensitivity (A) and adjusted F1 score (B) were used to evaluate the performance of 8 machine learning models. The EFs were not used to filter features and all the features were used to train the models. Sensitivity and adjusted F1 score were evaluated for 7 ratios of training to validation/prediction sets and each training cycle for the models was repeated 10 times for each ratio. On top of the graphs, the mean performance of each model over all the ratios and iterations is shown. Each model was evaluated using three different sets of labels, the training set, the validation set and a validation set where the labels for the genes were randomly shuffled so that part of them didn't represent the true label of a gene. (C) Receiver operating characteristic (ROC) curves for all tested models. Areas under the curve (AUC) for each model are listed in the legend.

Based on the results presented in Figure 4.9 compared to the results presented in Figure 4.8, I conclude that selecting a subset of features improves the results. The feature selection step using EFs prevents the models from overfitting, gives the expected low sensitivity to randomly shuffling the labels and improves the computational time the model needs to be trained by 10 times. It also allows the models to train on the relevant features that and learn useful data relations.

Next, I used the RF trained model to make predictions on the rest of the dataset. To assess the class of the genes, I used the sklearn.ensemble.RandomForestClassifier.predict_proba() function of the RF classifier, implemented with scikit-learn (Pedregosa et al., 2011). This function gives the probability of the gene to be classified as the positive class (present at glial protrusions) or the negative class (absent from glial protrusions). The two probabilities sum up to 1. As one of the approaches used to tackle learning from PU data is to train many classifiers and average their predictions, I averaged the predict_proba score for all the different proportions of training data (seven proportions, ranging from 20% of the data up to 80%, in increments of 10%). I considered genes with a probability of the positive class greater than 50%, to be localised. The higher the probability the more confident the localisation prediction is. This resulted in 4,117 genes predicted to be localised. Within those I retained 978 genes of the original 1,700 used to train the model, approximately 58%. I don't think this is surprising since the sensitivity of the RF model was found to be 59.7%. I intersected the list of 4,117 genes predicted to be localised with the list of 18 genes obtained from the smFISH experiments which are considered ground truth. 13/18 of these genes were retained within the list of predictions, which is 72% correct positive prediction. I also intersected the 182 genes we have examined with smFISH and I found 125 out of 182 predicted to be localised. As a result we can conclude that the percentage of false positive prediction is quite high for this model.

There are many limitations when using this approach and therefore it is expected that the results are not perfect. Limitations include the fact that the labels used to train this model are not ground truth and experimentally obtained. They are labels for *Drosophila* inferred from mammalian data although the class assigned to each gene is inferred from high confidence

homologs of mammalian genes to fly genes. Additionally, the fact that I don't have definite negative labels, i.e. genes that are not localised and I am using an approach for positive and unlabelled data is introducing another level of uncertainty which I tried to deal with as best as possible following approaches from literature.

I assessed the top 66 predictions of localised genes, excluding the ones that are already tested using smFISH, as these could become potential candidate genes for further smFISH experiments to test the localisation of mRNA in glial protrusions. These 66 predicted genes had a probability score of being localised between 0.745 and 0.751. Since none of the GO terms related features were used for training the RF model or in the predictions, I decided to perform a GO enrichment analysis to see if the predicted top genes were enriched in any localisation related GO terms. The results are shown in Figure 4.10. The analysis was performed using the enrichGO function from the clusterProfiler package (Wu et al., 2021; Yu et al., 2012) in R. High level categories for each domain were plotted using the enrichplot package to see the broad categories of enrichment for each domain. Panel (A) in Figure 4.10 depicts the enrichment for the CC, (B) for the BP and (C) for the MF domain.

I also looked specifically for enrichment of any localisation or transport related GO term and there were many terms that were enriched for the top predicted genes. I included all the GO terms here irrespective of hierarchy and level. Results are shown in Figure 4.11.

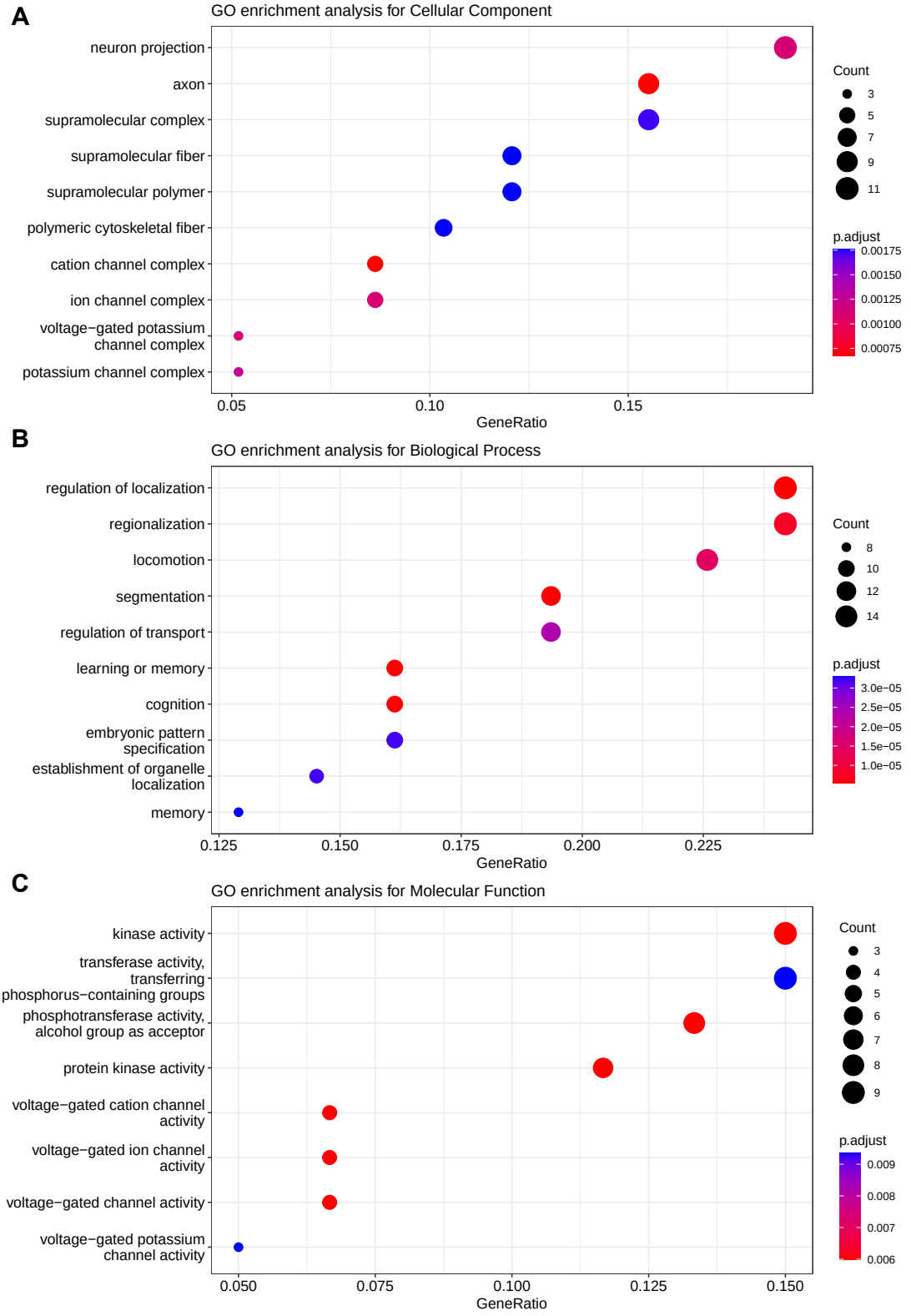


Figure 4.10: GO enrichment analyses for the top candidate genes. (A) GO enrichment analysis for the CC domain (B) GO enrichment analysis for the BP domain (C) GO enrichment analysis for the MF domain. The number of genes that is enriched in each term is shown by the bubble size and the adjusted p value for significance for each term is shown using the colour map. The GeneRatio in the x-axis shows the amount of the enriched genes compared to the total number of genes examined for enrichment. The y-axis in every panel shows the GO terms enriched.

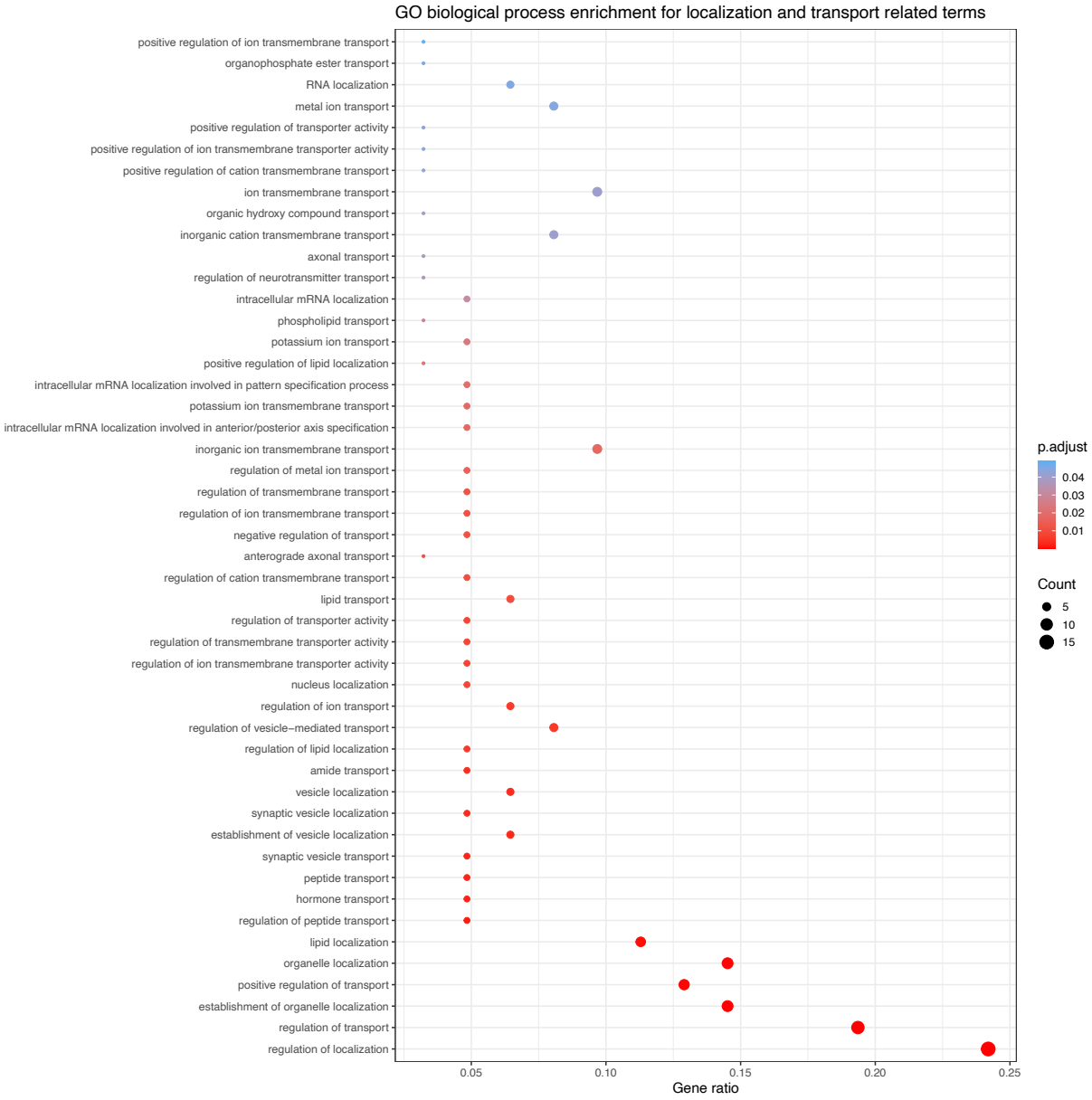


Figure 4.11: GO enrichment analysis for the BP domain, for localisation and transport related terms, for the top candidate genes. GO enrichment analysis for the BP domain, on the y-axis the GO localisation and transport related terms are shown and on the x-axis the ratio of the genes enriched for each term compared to all the genes examined is shown. The size of the bubbles shows the number of genes enriched and the different colours the adjusted p value.

As we can see in both Figures 4.10 B and 4.11, when looking at the BP domain, the genes are highly enriched in mRNA localisation terms. As seen in Figure 4.10 A when examining the CC domain, the genes are highly enriched in axon and neuron projection. Both findings are very promising indicating the potential for mRNA localisation in axons, glia and neuron projections.

I have also performed REACTOME enrichment analysis using the approach described earlier; I did not find any pathways enriched perhaps due to a small gene set and additionally because the REACTOME annotations seem to be more sparse than the GO annotations and not all genes were annotated with a pathway.

Additionally, I compared the 66 localisation candidate genes with genes documented in the literature as being present in specific subcellular locations. Specifically, I juxtaposed them with a dataset generated by the Lécuyer lab (Lefebvre et al., 2017) which provides transcriptome subcellular and extracelullar distributions using CeFra-seq (Lefebvre et al., 2017). CeFra-seq is an RNA-seq technology applied to subcellular components after their biochemical separation, yielded transcript expression counts for the nucleus, cytosol, endomembranes and insoluble fraction in human cells. This dataset was employed to train a deep recurrent NN, RNATracker (Yan et al., 2019), for predicting mRNA localisation in human cells in predefined cellular locations. The normalised dataset they used in RNATracker describes quantified transcript expression in FPKM (Fragments Per Kilobase of exon per million fragments Mapped) for the human genome (63,677 genes) for the four compartments, with FPKM values ranging from 0 to hundreds of thousands. Since localisation in the cytosol compartment is the closest equivalent to localisation at the tips of glia and away from the nucleus, I filtered out genes with FPKM higher than 10 from this compartment. I chose this cut-off after analysing the transcript distribution (59,617 genes with FPKM smaller than 10 and 4,060 with FPKM larger than 10) and to maintain consistency with the analysis conducted in section 3.2.1, where a cut-off of 10 TPM was set to select genes whose transcripts localise in glial protrusions. Next, I identified high confidence *Drosophila* orthologs for the human genes that their mRNAs localise in cytosol, using the DIOPT Ortholog Finder version 9 (Hu et al., 2021, 2011). To ensure a high degree of confidence in orthologs, only

those with a DIOPT score of 8 or higher were considered. This comparison yielded 13 genes as high-confidence orthologs after evaluating against the initial pool of 66 top candidates. These genes are: *unc-104*, *Glut1*, *Gug*, *par-1*, *app*, *Fmr1*, *msn*, *GramD1B*, *CtBP*, *Hrb27C*, *Nedd4*, *Dp1* and *Acsl*. The comparative findings enhance the robustness of my predictions that the top 66 candidates are highly likely to be localised.

4.4 Summary and Conclusions

In this Chapter, I examined the potential of various machine learning methods to understand and predict mRNA localisation in glia. I started by exploring the dependency of features on other features using machine learning to find which features are correlated with other features and therefore provide redundant information. Subsequently, I analysed the variance of the *Drosophila* and *Mus* datasets using PCA analysis and understood how the dimensionality of the datasets can be reduced. In the case of *Mus* the dataset's feature dimension can be reduced by approximately 35%. I then devised a novel approach to understand which features capture the most variance in the dataset and contribute the most to the PCs. I trained a RF regressor using the features and used the PCs as the target variable that needs to be predicted. I then found which features contribute the most to the prediction using Gini impurity for the *Mus* dataset. The highlight of these results is the contribution of RBP target sites to the variance. I will highlight a similar finding in the next paragraph as RBP target sites have been found to be a major contributor to mRNA localisation in glial cells in fruit flies as well. This is a major result, as I used two different approaches for two different organisms to reach a similar conclusion. I also attempted to use k-means clustering to understand if there is a natural grouping given by the features between the genes labelled as localised and the genes labelled as non localised, but the results did not point to a successful partition.

Next, I used a candidate gene approach found in literature (Y. Y. Lee et al., 2023) to predict genes related to sleep regulation. I adapted it for my *Drosophila* dataset and the main reason I chose this approach was its suitability for PU data. One of my two *Drosophila* datasets is constructed from information about localised genes only, and therefore only has positive labels.

The "ml-sleep-python" approach uses the EFs, based on "Bayes Factors" (Kass & Raftery, 1995), as a feature selection technique. Features with an EFs value larger than 3 that are not correlated with other features are used to train a range of machine learning models. After models were trained and the accuracy of the models was assessed, I found that RF is the most suitable model to be used to make predictions. I used the trained RF model to predict genes that are highly likely to be localised. I then performed GO and REACTOME enrichment analysis to explore the functional and biomolecular pathway attributes of the predicted genes.

Key findings in this Chapter include the prominence of RBP target sites in determining transcript localisation in glial protrusions. Additionally, there is the fact that the candidate genes for localisation are highly enriched in localisation and transport, with approximately fifty GO terms containing either of those terms to be highly enriched in the subset of 66 genes identified. When I examine the individual RBP target sites, I spot many interesting findings. For example, ELAVL1 is a very well known conserved RBP, playing a diverse role in mRNA regulation. It contributes to the regulation by mediating the elongation of 3' UTR isoforms and this has been observed in neurites (S. Lee et al., 2021) and ELAV factors in *Drosophila* are known to play a significant part in splicing (S. Lee et al., 2021). Additionally the IGF2BP2 RBP, comes from a family of RBPs that are very well known to be involved in RNA regulation at several levels as well (Degrauwe et al., 2016).

From the genes that were predicted to be localised, I isolated the 66 genes that had the highest probability to be localised but also they are genes that we have not already performed any smFISH experiments on to visualise the transcripts. A future goal is to verify the localisation of those genes in glial cells in the *Drosophila* brain using experimental methods.

5
Discussion and future perspectives

RNA localisation is a very important protagonist in the nervous system's biological functions. RNA localisation is one of the most significant post-transcriptional regulation mechanisms. This holds true, especially in long cells with many protrusions and polarised such as neurons and glia, where proteins could be required far from the cell body (Holt et al., 2019). In cases where proteins are required for prompt use at distances away from the soma, regulated on-site translation of mRNAs is extremely useful (Donlin-Asp et al., 2021; Holt & Schuman, 2013). One localised mRNA can produce multiple protein molecules for immediate consumption. Additionally, mRNAs are not just transported intracellularly but also intercellularly (Holt & Schuman, 2013), an observation we also made in the Davis lab (Titlow et al., 2023).

Both *Drosophila* and *Mus* model organisms have been used to study mRNA localisation in neurons and glial protrusions in the nervous system; however, *Drosophila* has not been used as extensively. Therefore, contributing to the knowledge of mRNA localisation and transport in glial cells in *Drosophila* is immensely valuable. Additionally, it has been shown that in a healthy CNS there is proof of localised translation and mRNA localisation in glia (Blanco-Urrejola et al., 2021) and it is also understood that dysregulation of these processes can lead to neurological and neurodegenerative diseases (Blanco-Urrejola et al., 2021).

In Chapter 3, I presented our microscopy YFP trap screen for a collection of 200 genes where we visualised simultaneously single molecules of mRNA and protein using smFISH, taking advantage of the YFP reporter inserted at the endogenous gene locus to do that. We produced approximately 1,400 figures, as we examined seven compartments in the *Drosophila* larval brain. In each figure describing the compartment, we included one or two microscopy images describing different tissues within the compartment. Then we carefully and systematically examined the mRNA and protein distribution levels and the spatial map of the single molecules. This enabled us to make important conclusions regarding mRNA localisation and post-transcriptional regulation, which we published (Titlow et al., 2023). Firstly, we observed that there was a lack of correlation in the distribution of transcripts and proteins encoded by the same gene. This lack of correlation or discordance is one of the manifestations of post-transcriptional regulation. We observed discordance in at least one tissue of interest for 195 out of the 200 genes examined. Additionally, the results show mRNA localisation in glia and the NMJ for 18 genes out of the 200, which, to our knowledge, is the first report in literature for mRNA localisation in *Drosophila* glial peripheral processes. This is of immense importance, as these results could imply that regulation of mRNA localisation in glial cells is highly probable (Titlow et al., 2023).

The YFP gene trap microscopy screen we produced for seven compartments in the *Drosophila* nervous system, encompassing 200 CPTI lines, is a highly valuable addition to the existing mRNA localisation literature. This significance arises not only from the intermediate-throughput imaging screen that allows us to study mRNA and protein distribution but also from the systematic approach used to create the screen, enabling us to score and annotate the transcriptome and proteome visualised in the microscopy images. Our polished and high-quality sets of annotations for 200 genes were generated by three experts. Annotation by experts, compared to "citizen science" annotation, is of great importance, as experts possess extensive knowledge in the domain of interest, and this significance is amplified when multiple experts annotate the same images. In the work presented in Chapter 2, as a primary quality check for our experts' annotations, we employed a majority vote to select the "correct" annotation in cases of disagreement among the experts. A secondary layer of quality checks included assessing the inter-annotator agreement

and Cohen's kappa (Cohen, 1960), which accounts for chance agreement. Consequently, we have a high level of confidence in the accuracy of our annotations, approaching as close to ground truth as possible. As a result, our annotations serve as a benchmark and a gold standard against which automated annotation systems can be compared.

As our collection of microscopy images consists of approximately 1,400 different images, it represents a reasonably large and meticulously curated set of high-quality images suitable for training machine learning models dedicated to image analysis. Tools such as FISH-quant (Imbert et al., 2022) would greatly benefit from using our dataset, which includes both images and annotations, for benchmarking and evaluating their algorithms or for training future algorithms. FISH-quant offers users the capability to segment nuclei and cells, detect single mRNA molecules, and further decompose overlapping clusters of mRNA into their individual forms. Additionally, it can quantify mRNA localisation patterns. Therefore, a dataset like the one we produced, which visualises mRNA using smFISH, is immensely valuable to the developers of FISH-quant. RS-FISH (Bahry et al., 2022) and deepBlink (Eichenberger et al., 2021) are also similar software tools to FISH-quant, specialising in fluorescent spot detection. They would likewise derive significant benefits from our dataset.

In Chapter3, I explored publicly available, well-established biological repositories. I then identified datasets with potential relevance to mRNA localisation and devised a method to retrieve and seamlessly integrate these datasets. Subsequently, I encoded the datasets into features, providing a machine-learning-compatible and information-rich pool of data. The approach I used for constructing this dataset is largely organism-agnostic, meaning that the same pool of data can be created for many species as long as their genomic, transcriptomic, and proteomic information is curated by ENSEMBL (Kinsella et al., 2011). Therefore, my approach can be utilised by others to construct similar datasets and analyse them for other organisms, such as humans. Species-agnostic approaches in bioinformatics are becoming increasingly indispensable as scientists grapple with questions that involve multiple organisms. Cross-species research is already being employed in drug discovery, particularly for diseases with limited access to patient

data, such as osteosarcomas and other cancers (Somarelli et al., 2020). Hence, cross-species drug discovery pipelines offer valuable insights in such cases.

Additionally, as described in my introduction, there are several models that specifically aim to predict mRNA localisation (Garg et al., 2020; Meher et al., 2021; Yuan et al., 2023; Zhang et al., 2021). These models primarily rely on features derived from gene sequences. The datasets I have produced for both *Drosophila* and *Mus* represent a resource that can be easily replicated for different organisms and can be utilised for the study and prediction of mRNA localisation. Furthermore, it can be effortlessly modified, and specific subsets of the data can be adapted for other purposes, such as studying localisation in neurons or other tissues.

In Chapter 4, I successfully adapted an approach to predict candidate sleep genes (Y. Y. Lee et al., 2023) for the prediction of genes localised in glial protrusions. This illustrates the versatility, value, and adaptability of our datasets, which can be employed, with appropriate modifications, in candidate gene identification techniques. The field of candidate gene selection using machine learning is still in its nascent stages, with examples in the literature focusing on prioritising candidate genes for conditions such as autism spectrum disorder (ASD) (Y. Lin et al., 2020) and various diseases (Azadifar & Ahmadi, 2022; J. Chen et al., 2021). The latter approaches are based on training data associated with GO terms, encoded within graph-based representation of the GO ontology, and deep graph convolutional networks. However, relying solely on GO terms may have limitations in terms of achieving optimal results.

A common theme among the prioritisation tools found in the literature is their application to identify candidate disease genes. This is expected, as combating disease ranks among the most significant and pressing challenges that humanity faces. To the best of my knowledge, my work is the first to employ candidate prioritisation tools for studying mRNA localisation. Understanding mRNA localisation can also enhance our knowledge of diseases, as dysregulation of mRNA localisation and post-transcriptional control has significant implications in neurodegenerative diseases (Blanco-Urrejola et al., 2021).

A recurring discovery when employing feature selection techniques to identify essential features contributing to the model's predictions in Chapter 4 is the recognition of RBP target sites as crucial elements in mRNA localisation. Specific target sites that are predicted to be the top contributors include RALYL, EWSR1, SFPQ for *Mus* and for *Drosophila* ELAVL1, ENSXETG00000012802, HNRNPD, IGF2BP2 among others.

From among these, several RBP target sites are documented in the literature as being involved in various aspects of mRNA metabolism. For instance, RALYL, as reported by Rossi et al. (2017), is implicated in distal processes of polarised cells, including neurons in human cell lines MCF7 and HeLa. Additionally, the IGF2 mRNA-binding protein family is prominently featured in the literature for its role in controlling essential cellular functions and regulating the metabolism of key mRNAs, as highlighted in studies like (Wächter et al., 2013). Specifically, in mammals, Wächter et al. (2013) notes that IGF2BP paralogues contain two RNA recognition motifs, which may elucidate the function of the IGF2 family. These findings further underscore the significance of employing machine learning models in research, as predictive algorithms can, as demonstrated, assist in narrowing down relevant data for addressing biological problems and offering valuable insights into potential solutions.

5.1 Future perspectives

In this section, I describe potential directions for future research that can enhance and complement the work presented in this thesis.

5.1.1 Predicting mRNA localisation patterns based on the microscopy images and the localisation labels for the 200 genes

In Chapter 2, I generated annotations that describe the mRNA and protein distributions of 200 genes in the *Drosophila* nervous system across twelve tissues of interest, following the scoring of microscopy images. These images represent a transcriptomic and proteomic map of single mRNA and protein molecules, created using an approach that enables the simultaneous visualisation of mRNA and proteins in the nervous system. I utilised these annotations to train machine

learning algorithms designed to predict mRNA localisation within a specific tissue. The models were fed input features derived from localisation-related datasets obtained from bioinformatics libraries or the literature. The data types composing these features included both numerical and text data, with no images included.

A future perspective to consider for this part of my work is to utilise the microscopy images generated from our dual mRNA-protein expression screen for 200 genes as input for image-based machine learning algorithms. There are two possible directions for this approach. Firstly, one could use a subset of the images and a portion of the generated labels from the annotations to train and fine-tune a predictive machine learning algorithm. Then, employ the trained model to predict localisation labels for the remaining images. Since we examined mRNA and protein distributions in six compartments across twelve tissues, we have a wide array of annotations to train on for some tissues and predict for others. Alternatively, we could train for all tissues for some genes and predict for the remaining genes, provided they have been imaged. Secondly, one could employ pre-trained supervised algorithms, trained to detect single molecules of mRNA or protein, and use them to analyse the microscopy images we produced. The outputs of these algorithms would yield localisation labels or labels describing the distributions of the transcripts/proteins. Based on this output, I could compare the human expert-made annotations with the machine-generated annotations and evaluate the accuracy of the algorithm through comparison. Additionally, there are unsupervised techniques based on clustering that could group mRNAs or proteins based on the tissues in which they are concentrated and provide appropriate labels.

Chouaib et al. (2020) developed a strategy that combines both supervised and unsupervised techniques for co-detecting mRNA and protein localisation patterns. They applied this strategy to human cell lines expressing GFP-tagged genes, allowing them to simultaneously visualise the GFP-expressing proteins and the mRNA (using smFISH). This process was conducted for 523 different cell lines, with subsequent manual annotation of the images. To segment nuclei, they employed pre-trained NNs, and for mRNA single molecule detection, they utilized FISH-quant (Mueller et al., 2013). They applied t-SNE (a clustering method) as an unsupervised approach (Imbert

et al., 2022), and RF as a supervised approach to assign each localization pattern. I am keenly interested in applying their algorithmic strategy to the Davis lab's smFISH/YFP trap collection of microscopy images and evaluating the results. I am curious to know whether the machine-generated annotations will align with the human-made annotations and, if so, to what extent?

5.1.2 Impact of RNA secondary structure on mRNA localisation

Alphafold 2 (Jumper et al., 2021) has the ability to predict the 3D structure of proteins very accurately. With the development of this groundbreaking technology, the path is now open to the possibility of accurately predicting secondary and tertiary RNA structures as well.

It is established that the secondary structure of RNA molecules can function as a signal for localisation and as a motif for recognition of RBPs (Hamilton & Davis, 2007). Additionally, it can influence post-transcriptional regulation through specific shaping of the secondary structures such as stem-loops and hairpins (Wan et al., 2011).

While current computational methods for predicting RNA localisation have tried to incorporate into their training datasets RNA secondary structure, they have not been successful in producing the desired results. Models such as RNATracker (Yan et al., 2019) and RNAlight (Yuan et al., 2023) have integrated secondary structure features but when assessing the performance of the models with and without the secondary structure derived features, there was no significant improvement in the accuracy of the prediction. The RNAlight creators (Yuan et al., 2023) have suggested that this could be because the secondary structure was predicted/obtained from the RNA primary sequences and thus the secondary structure did not enhance the prediction, as the information from the primary sequences was already included. Likewise the RNATracker (Yan et al., 2019) authors argue that the source of data they used for secondary structure perhaps is not as accurate as it is based only on relatively short-range interactions. I have also incorporated secondary structure features in my dataset, produced using the same package as the one used by Yan et al. (2019) and Yuan et al. (2023). The feature selection step I used in my pipeline filtered out the secondary structure features therefore I agree with Yan et al. (2019) and Yuan et al. (2023) in their conclusions of the usefulness of these features.

Perhaps integrating experimentally derived results for RNA secondary structure and/or more accurate computational predictions could give improved predictions of RNA localisation. I am interested in examining what would be the importance of features generated from secondary structure datasets and if their contribution to machine learning predictive algorithms is more significant compared to the secondary structure datasets I have already tried. One idea of how to do that, is to incorporate the experimental results on secondary structure found from Berkowitz et al. (2016). Berkowitz et al. (2016) created a resource that provides access to RNA secondary structure probing data available for mammalian transcriptomes. Using experimental results, amongst other the icSHAPE dataset (Flynn et al., 2016), they added scores to the nucleotides reflecting reactivity of unpaired nucleotides. The reactivity is higher when unpaired nucleotides were examined versus when they were involved in a structure. The scoring therefore was inverted to present the degree of structure. I thought of incorporating the scores in the form of average score per k-mer per gene or in any other reasonable way.

5.1.3 Improving of the machine learning algorithm prediction

In Chapter 4, I explored a range of feature selection approaches aimed at identifying the most effective predictors for mRNA localisation. Subsequently, I trained an RF model to predict mRNA localisation using these selected features. Additionally, I implemented an established pipeline that evaluated the performance of various machine learning algorithms, ultimately selecting the best-performing one for predicting candidate genes. In future work, I aim to further fine-tune the models, experiment with different parameters, and explore additional feature selection techniques. Furthermore, I intend to parallelise my pipeline to enhance its performance and efficiency.

5.2 Conclusion

In conclusion, mRNA localisation in distal processes in glial cells in both *Drosophila* and *Mus* is a very important mechanism for post-transcriptional regulation. Using machine learning approaches, I constructed a list of candidate genes that have high probability of localising in glial cells in *Drosophila*. Additionally, using machine learning models I built, I found that one of the most

crucial contributors to mRNA localisation is the RNA binding protein target sites and specifically one of the RBP target sites I isolated is IGF2BP2 which is known to control the regulation of key mRNAs (Wächter et al., 2013). This is a very important finding that demonstrates that my machine learning algorithm is working successfully as it predicts results supported by the existing literature.

In the process of creating and training the machine learning models, I produced a set of labels describing mRNA localisation and a set of features with data that could explain mRNA localisation, both integral components to the training of machine learning models. I produced the labels and the features in a systematic way for both *Drosophila* and *Mus*, after extensive literature search. Therefore the methods used to produced them are available as pipelines described in this work for the recreation of the datasets and adaptation to other organisms. In particular, for the creation of the labels describing mRNA localisation in *Drosophila*, I produced a novel pipeline for the systematic collection and management of the microscopy images visualising the mRNA and protein distribution. Following these two steps, I created an application, named Annotate.OMERO.Fig, for expert annotators to score the images in a standardised way. After the collection and analysis of the annotations, I used a cutting edge visualisation software to display all of the microscopy images and the annotations together with a subset of the features describing mRNA localisation. The integration of the multimodal data only emphasises the significance of interoperability and enables the user to make discoveries through observing patterns amongst the now integrated heterogeneous data.

Appendices

A

List of CPTI lines screened

Table A.1: CPTI lines screened using smFISH

CPTI line ID	Flybase Gene ID	Gene name
CPTI003733	FBgn0000053	ade3
CPTI000876	FBgn0000097	aop
CPTI001198	FBgn0000117	arm
CPTI001409	FBgn0000392	cup
CPTI001709	FBgn0001098	Gdh
CPTI002761	FBgn0002921	Atpalpha
CPTI002828	FBgn0003435	sm
CPTI002487	FBgn0003716	tkv
CPTI002927	FBgn0004924	Top1
CPTI001194	FBgn0005777	PpD3
CPTI002342	FBgn0010215	alpha-Cat
CPTI002408	FBgn0010238	Lac
CPTI001457	FBgn0010905	Spn
CPTI001769	FBgn0011763	dp
CPTI001977	FBgn0013997	Nrx-IV
CPTI000915	FBgn0020653	Trxr-1
CPTI002103	FBgn0022986	qkr58E-1
CPTI001879	FBgn0023215	Mnt
CPTI100035	FBgn0023407	B4
CPTI001997	FBgn0024989	CG3777
CPTI001685	FBgn0026721	fat-spondin
CPTI003426	FBgn0027087	Aats-his
CPTI000633	FBgn0028327	l(1)G0320
CPTI002750	FBgn0028341	Ptpmeg2
CPTI000326	FBgn0028343	Ankle2
CPTI002793	FBgn0030478	CG1640
CPTI002807	FBgn0030521	CtsB
CPTI001277	FBgn0031268	cold
CPTI001775	FBgn0032949	Lamp
CPTI001828	FBgn0034075	Asph
CPTI002818	FBgn0036239	Pop2
CPTI002443	FBgn0039274	CG11920
CPTI001771	FBgn0051694	CG31694
CPTI002796	FBgn0052423	shep
CPTI001308	FBgn0052479	Usp10
CPTI001081	FBgn0052626	AMPdeam
CPTI002905	FBgn0063485	Lasp
CPTI002889	FBgn0082582	tmod
CPTI001586	FBgn0259735	CG31738
CPTI003598	FBgn0261238	Alh
CPTI001261	FBgn0261552	ps
CPTI002315	FBgn0262559	mdh2
CPTI001735	FBgn0262743	Fs(2)Ket
CPTI001056	FBgn0263997	CG15307
CPTI004479	FBgn0264442	ab
CPTI001714	FBgn0264975	Nrg
CPTI001259	FBgn0265140	Neu3
CPTI001328	FBgn0265523	Smr
CPTI001234	FBgn0266521	stai
CPTI004117	FBgn0285926	Imp

Table A.2: CPTI lines screened using smFISH, continued

CPTI line ID	Flybase Gene ID	Gene name
CPTI003745	FBgn0001235	hth
CPTI004152	FBgn0001250	if
CPTI003339	FBgn0002645	Map205
CPTI002102	FBgn0002772	Mlc1
CPTI004179	FBgn0003261	Rm62
CPTI000037	FBgn0003676	T-cp1
CPTI003684	FBgn0004177	mts
CPTI003091	FBgn0004227	nonA
CPTI000433	FBgn0004647	N
CPTI003588	FBgn0004872	piwi
CPTI002645	FBgn0005671	Vha55
CPTI004519	FBgn0011206	bol
CPTI001962	FBgn0013733	shot
CPTI100064	FBgn0015221	Fer2LCH
CPTI000038	FBgn0015222	Fer1HCH
CPTI000155	FBgn0015229	glec
CPTI000925	FBgn0015776	nrv1
CPTI003971	FBgn0016059	Sema-1b
CPTI002187	FBgn0020660	eIF-4B
CPTI002285	FBgn0022960	vimar
CPTI003151	FBgn0023509	mip130
CPTI002201	FBgn0026418	Hsc70Cb
CPTI003806	FBgn0030870	CG6398
CPTI002305	FBgn0031307	CG4726
CPTI000774	FBgn0031645	CG3036
CPTI002939	FBgn0031883	Caper
CPTI000379	FBgn0032298	CG6724
CPTI000286	FBgn0032409	Ced-12
CPTI003020	FBgn0034282	Mapmodulin
CPTI000318	FBgn0036165	chrb
CPTI003957	FBgn0036213	RpL10Ab
CPTI003049	FBgn0036816	Indy
CPTI001180	FBgn0037007	BNIP3
CPTI000765	FBgn0037073	tsr1
CPTI002636	FBgn0037110	ORMDL
CPTI004189	FBgn0037707	RnpS1
CPTI000077	FBgn0041210	HDAC4
CPTI001718	FBgn0045823	vsg
CPTI004292	FBgn0052685	Zap3
CPTI001046	FBgn0053087	LRP1
CPTI003173	FBgn0053293	CG12163
CPTI002023	FBgn0250823	Gish
CPTI001995	FBgn0259214	PMCA
CPTI000033	FBgn0259244	CG42342
CPTI003818	FBgn0260003	Dys
CPTI003250	FBgn0261090	Sytbeta
CPTI000736	FBgn0261722	CG6151
CPTI100062	FBgn0261822	Bsg
CPTI003089	FBgn0261885	osa
CPTI002437	FBgn0263594	growl

Table A.3: CPTI lines screened using smFISH, continued

CPTI line ID	Flybase Gene ID	Gene name
CPTI000658	FBgn0000017	Abl
CPTI003522	FBgn0000183	BicD
CPTI000616	FBgn0000547	ed
CPTI000091	FBgn0000721	for
CPTI000303	FBgn0001124	Got1
CPTI000207	FBgn0001624	dlg1
CPTI002853	FBgn0003165	pum
CPTI000263	FBgn0003459	stwl
CPTI001175	FBgn0004449	Ten-m
CPTI000252	FBgn0004839	otk
CPTI000877	FBgn0010217	ATPsyn-beta
CPTI001658	FBgn0010316	dap
CPTI000371	FBgn0011725	twin
CPTI003785	FBgn0011817	nmo
CPTI000332	FBgn0015513	mbc
CPTI002152	FBgn0015773	NetA
CPTI001455	FBgn0015777	nrv2
CPTI003490	FBgn0016797	fz2
CPTI100045	FBgn0020238	14-3-3epsilon
CPTI000609	FBgn0020443	Elf
CPTI000337	FBgn0021764	sdk
CPTI100043	FBgn0023167	SmD3
CPTI003394	FBgn0024238	Fim
CPTI003501	FBgn0026259	eIf5B
CPTI000365	FBgn0026533	Dek
CPTI002789	FBgn0027356	Amph
CPTI000298	FBgn0027529	tapas
CPTI002324	FBgn0027598	cindr
CPTI000472	FBgn0036449	bmm
CPTI003911	FBgn0037010	CG4825
CPTI002772	FBgn0039348	CG4673
CPTI003100	FBgn0052062	CG32062
CPTI000739	FBgn0085451	htk
CPTI003923	FBgn0259143	CG42258
CPTI000590	FBgn0259212	cno
CPTI000031	FBgn0259984	kuz
CPTI000640	FBgn0261278	grp
CPTI003605	FBgn0261618	larp
CPTI003472	FBgn0261836	Msp300
CPTI002085	FBgn0261988	Gprk2
CPTI000567	FBgn0262733	Src64B
CPTI003732	FBgn0263037	CG43332
CPTI001589	FBgn0263391	hts
CPTI000944	FBgn0264607	CaMKII
CPTI002907	FBgn0265434	zip
CPTI003323	FBgn0266098	rg
CPTI000194	FBgn0266579	tau
CPTI000190	FBgn0283521	lola
CPTI002049	FBgn0284408	trol
CPTI000688	FBgn0286818	Pdi

Table A.4: CPTI lines screened using smFISH, continued

CPTI line ID	Flybase Gene ID	Gene name
CPTI001284	FBgn0000253	Cam
CPTI002786	FBgn0000261	Cat
CPTI001279	FBgn0000635	Fas2
CPTI000833	FBgn0000636	Fas3
CPTI002264	FBgn0000711	flw
CPTI001918	FBgn0001145	Gs2
CPTI003669	FBgn0001215	Hrb98DE
CPTI002805	FBgn0001987	Gli
CPTI000850	FBgn0003041	pbl
CPTI000023	FBgn0003371	Sgg
CPTI000760	FBgn0003721	Tm1
CPTI001728	FBgn0004167	kst
CPTI001678	FBgn0004456	Mew
CPTI001693	FBgn0004868	Gdi
CPTI000199	FBgn0004878	Cas
CPTI002578	FBgn0010415	sdc
CPTI002406	FBgn0011224	heph
CPTI001309	FBgn0011592	fra
CPTI001880	FBgn0014007	Ptp69D
CPTI000748	FBgn0015774	NetB
CPTI100051	FBgn0015790	Rab11
CPTI003532	FBgn0016685	Nlp
CPTI001595	FBgn0016977	spen
CPTI001580	FBgn0022382	Pka-R2
CPTI003151	FBgn0023509	mip130
CPTI002562	FBgn0026250	eIF-1A
CPTI001414	FBgn0028622	Qsm
CPTI002292	FBgn0029687	Vap-33-1
CPTI001740	FBgn0029939	CG9650
CPTI001659	FBgn0030400	Pits
CPTI100013	FBgn0031294	ia2
CPTI001993	FBgn0031632	CG15628
CPTI100044	FBgn0031990	PAPLA1
CPTI100012	FBgn0033446	CG1648
CPTI000836	FBgn0035397	CG11486
CPTI003231	FBgn0035533	Cip4
CPTI004071	FBgn0036337	Adenok
CPTI002646	FBgn0039882	CG11576
CPTI001430	FBgn0040207	kat80
CPTI001473	FBgn0041605	cpx
CPTI000161	FBgn0041789	Pax
CPTI003917	FBgn0051363	Jupiter
CPTI004166	FBgn0259176	bun
CPTI000727	FBgn0261617	nej
CPTI000239	FBgn0263396	sqd
CPTI001039	FBgn0263930	dally
CPTI001427	FBgn0264078	Flo-2
CPTI004362	FBgn0264307	orb2
CPTI001015	FBgn0267376	SelR
CPTI004394	FBgn0285937	Rab1

Table A.5: CPTI lines screened twice using smFISH

CPTI line ID	Flybase Gene ID	Gene name
CPTI000033	FBgn0259244	CG42342
CPTI000303	FBgn0001124	Got1
CPTI002772	FBgn0039348	CG4673
CPTI003732	FBgn0263037	CG43332
CPTI001046	FBgn0053087	LRP1
CPTI002187	FBgn0020660	eIF-4B
CPTI003020	FBgn0034282	Mapmodulin
CPTI003091	FBgn0004227	nonA
CPTI004152	FBgn0001250	if
CPTI001828	FBgn0034075	Asph
CPTI002315	FBgn0262559	mdh2
CPTI002818	FBgn0036239	Pop2
CPTI003733	FBgn0000053	ade3
CPTI100035	FBgn0023407	B4
CPTI001993	FBgn0031632	CG15628
CPTI003669	FBgn0001215	Hrb98DE
CPTI004362	FBgn0264307	orb2
CPTI100064	FBgn0015221	Fer2LCH

B

Sensitivity analysis of perturbations to the annotations

The table in Appendix B shows a summary of the agreement level between the annotators for each question in the form of total number of annotations in each category. I conducted a sensitivity analysis to study how changing the vote of one annotator each time affects the consensus for each annotation. Various scenarios were explored; for example in cases of complete agreement, changing the vote of one annotator does not influence the consensus or the annotation's classification. Conversely, in situations of partial agreement, the alteration of one annotator's vote holds greater significance because depending on the change, it can lead to a shift in the annotation class or result in the loss of consensus. As there are three annotators for each question, it was not difficult to explore all possible scenarios of perturbations and find the probability of each. I summarised the conclusions of the exploration in the last two rows of the table, highlighted in orange.

Table B.1: Summary of question-specific annotation agreement and probability of class change caused by perturbations

Question	Complete agreement	Partial agreement		No agreement
VNC:Is RNA present in soma?	145	54	54	1
VNC:Is RNA present in neuropil?	90	105	105	5
VNC:Is RNA present in midline glia?	36	117	117	47
VNC:Is protein present in soma?	137	63	63	0
VNC:Is protein present in neuropil?	136	64	64	0
VNC:Is protein present in midline glia?	29	125	125	46
OL:Is RNA present in soma?	147	46	46	7
OL:Is RNA present in neuropil?	69	125	125	6
OL:Is protein present in soma?	144	56	56	0
OL:Is protein present in neuropil?	129	71	71	0
NMJ:Is RNA present in post synaptic density?	105	84	85	11
NMJ:Is RNA present in neuron?	85	89	90	26
NMJ:Is RNA present in muscle?	136	64	65	0
NMJ:Is RNA present in axon terminal?	83	109	109	8
NMJ:Is protein present in post synaptic density?	145	55	56	0
NMJ:Is protein present in neuron?	139	61	61	0
NMJ:Is protein present in muscle?	108	92	92	0
NMJ:Is protein present in axon terminal?	134	66	66	0
nerve:Is RNA present in nerve?	105	51	51	4
nerve:Is RNA present in glia?	124	32	32	4
nerve:Is protein present in nerve?	99	61	61	0
nerve:Is protein present in glia?	109	51	51	0
MB:Is RNA present in MB lobes?	120	79	79	1
MB:Is RNA present in MB calyx?	139	61	61	0
MB:Is RNA present in Kenyon cell somata?	91	107	107	2
MB:Is protein present in MB lobes?	126	74	74	0
MB:Is protein present in MB calyx?	119	81	81	0
MB:Is protein present in Kenyon cell somata?	101	99	99	0
CB:Is RNA present in other cells?	16	184	184	0
CB:Is RNA present in NBs?	133	65	65	2
CB:Is RNA present in anlagen?	120	79	79	1
CB:Is protein present in other cells?	24	176	176	0
CB:Is protein present in NBs?	94	105	105	1
CB:Is protein present in anlagen?	68	130	130	2
Sensitivity to perturbations (%)	**0**	**0**	**22.5**	**22.5**	**100**
How perturbations can affect the class	**No class change**	**No class change**	**Class change**	**No class consensus**	**Class consensus**

C

Multi-Dimensional Viewer (MDV): RNA/protein distribution of 200 YFP trap lines

Figures showing how the YFP trap collection can be interrogated using the MDV. Figure 1 in Appendix C shows the landing view of our collection in MDV. A user can select an image from the collection of smaller images to enlarge it in the central panel. On the filter box on the left a user can filter out based on the questions shown images and genes with the quality selected. Additionally, the GO Annotations barplot changes to show the annotations of the selection of images and genes. Figure 2 in Appendix C, shows how the a user can filter out genes based on the expression and their UTR length. The last Figure in Appendix C, shows how the GO annotations per domain (BP, MF and CC) can be interrogated in a more detailed level.

C. Multi-Dimensional Viewer (MDV): RNA/protein distribution of 200 YFP trap lines

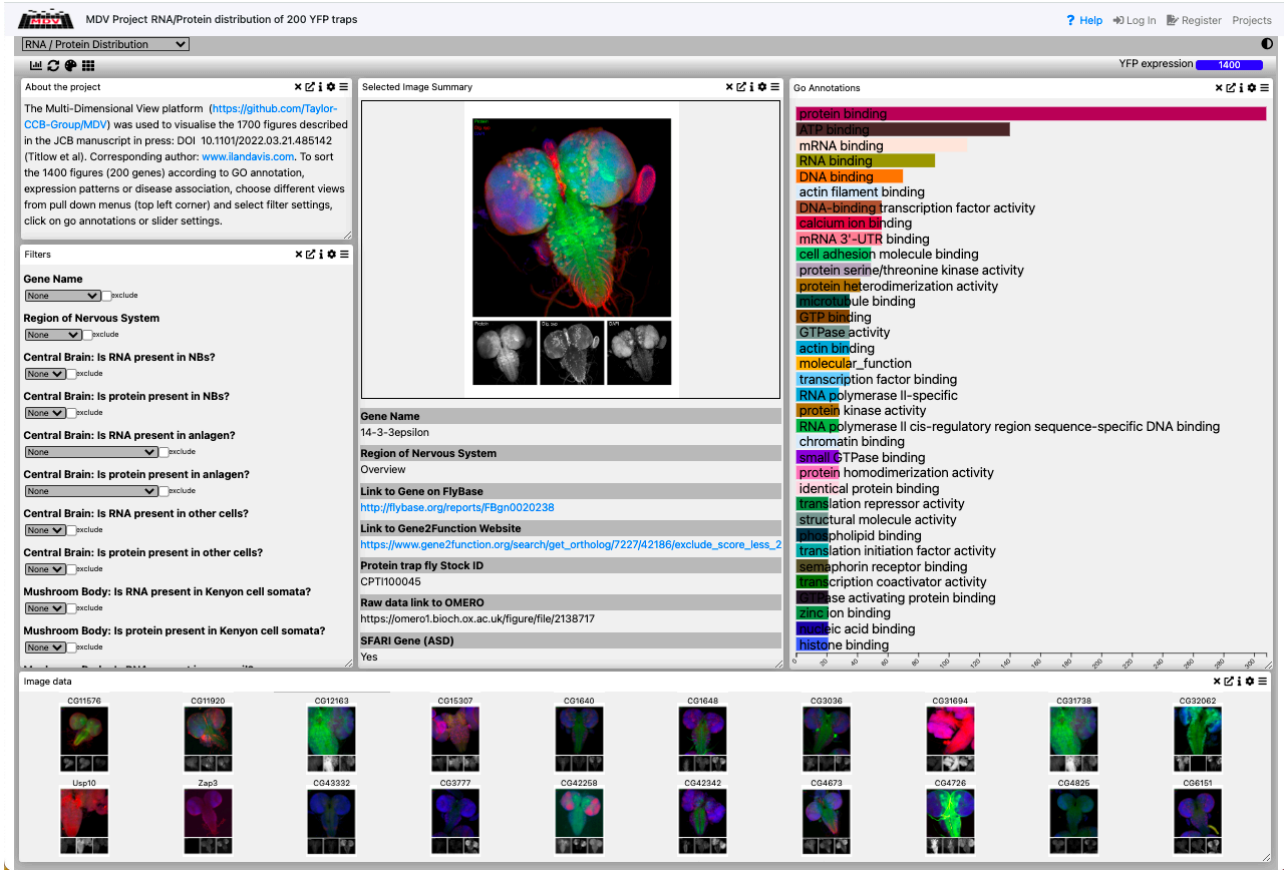


Figure B.1: MDV first view of the YFP collection.

C. Multi-Dimensional Viewer (MDV): RNA/protein distribution of 200 YFP trap lines

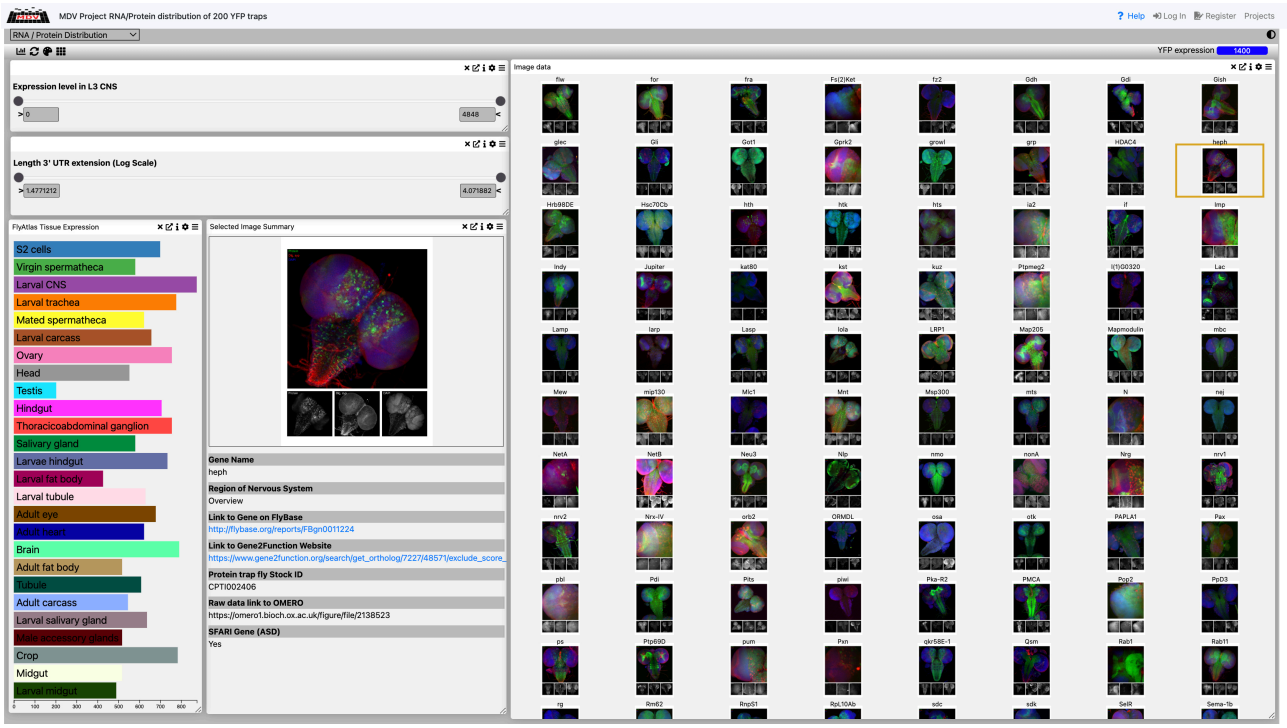


Figure B.2:. Exploring the expression and UTR length of genes within the YFP collection.

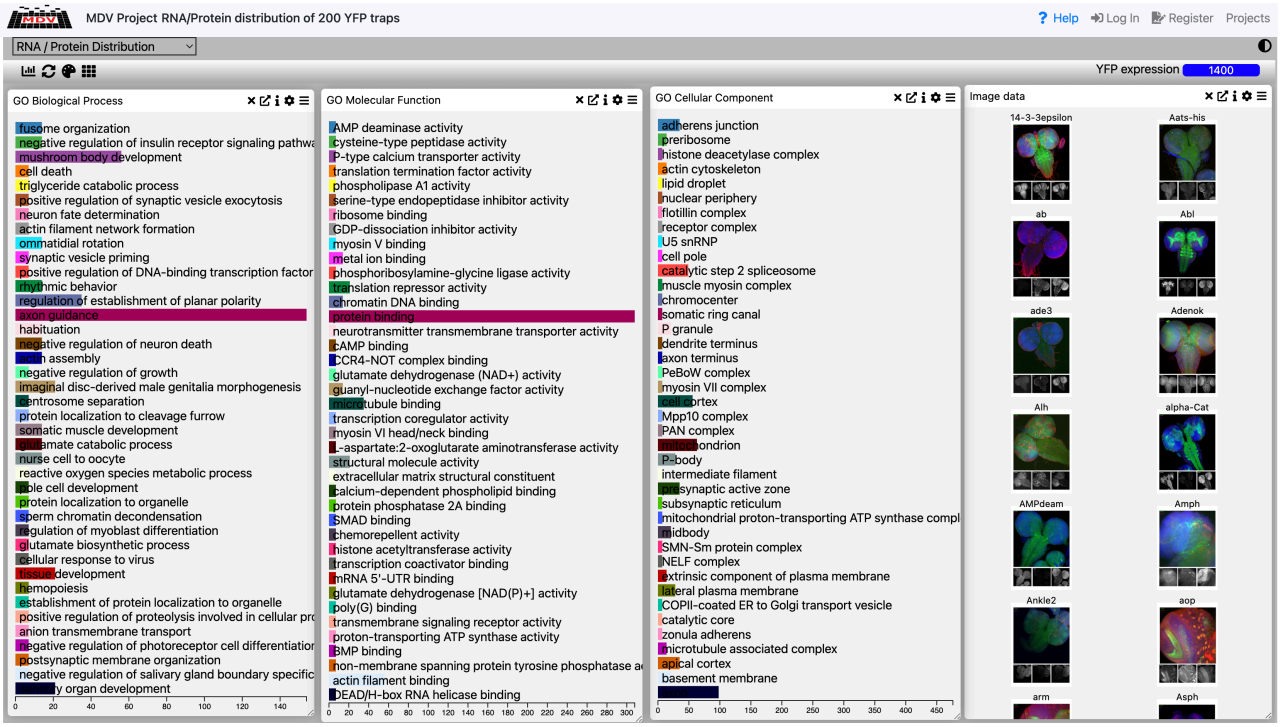


Figure B.3: Interrogating the GO annotations in more detail, focusing on each domain, for the genes in the collection.

D

Datasets used and analysed to get localisation labels

Table D.1: Datasets used to get localisation labels for mouse genes.

Dataset ID	Reference	Technique	Organism	Cell type
1	Thomsen et al, 2013	Helicos-seq	Mouse	astrocytes
2	Sakers et al, 2017	RNA-seq	Mouse	astrocytes
3	Boulay et al, 2017	TRAP-seq	Mouse	astrocytes
4	Mazare et al, 2020	TRAP-seq	Mouse	astrocytes
5	Thomsen et al, 2013	Helicos-seq	Mouse	astrocytes
6	Vasek et al, 2021	TRAP-seq	Mouse	perisynaptic schwann cells
7	Thakurela et al, 2016	RNA-seq	Mouse	astrocytes
8	Thakurela et al, 2017	RNA-seq	Mouse	astrocytes
9	Thakurela et al, 2018	RNA-seq	Mouse	astrocytes
10	Thakurela et al, 2019	RNA-seq	Mouse	astrocytes
11	Azevedo et al, 2018	mRNA-seq	Rat	oligodendrocyte
12	Castro et al, 2020	RNA-seq	Mouse	perisynaptic schwann cells

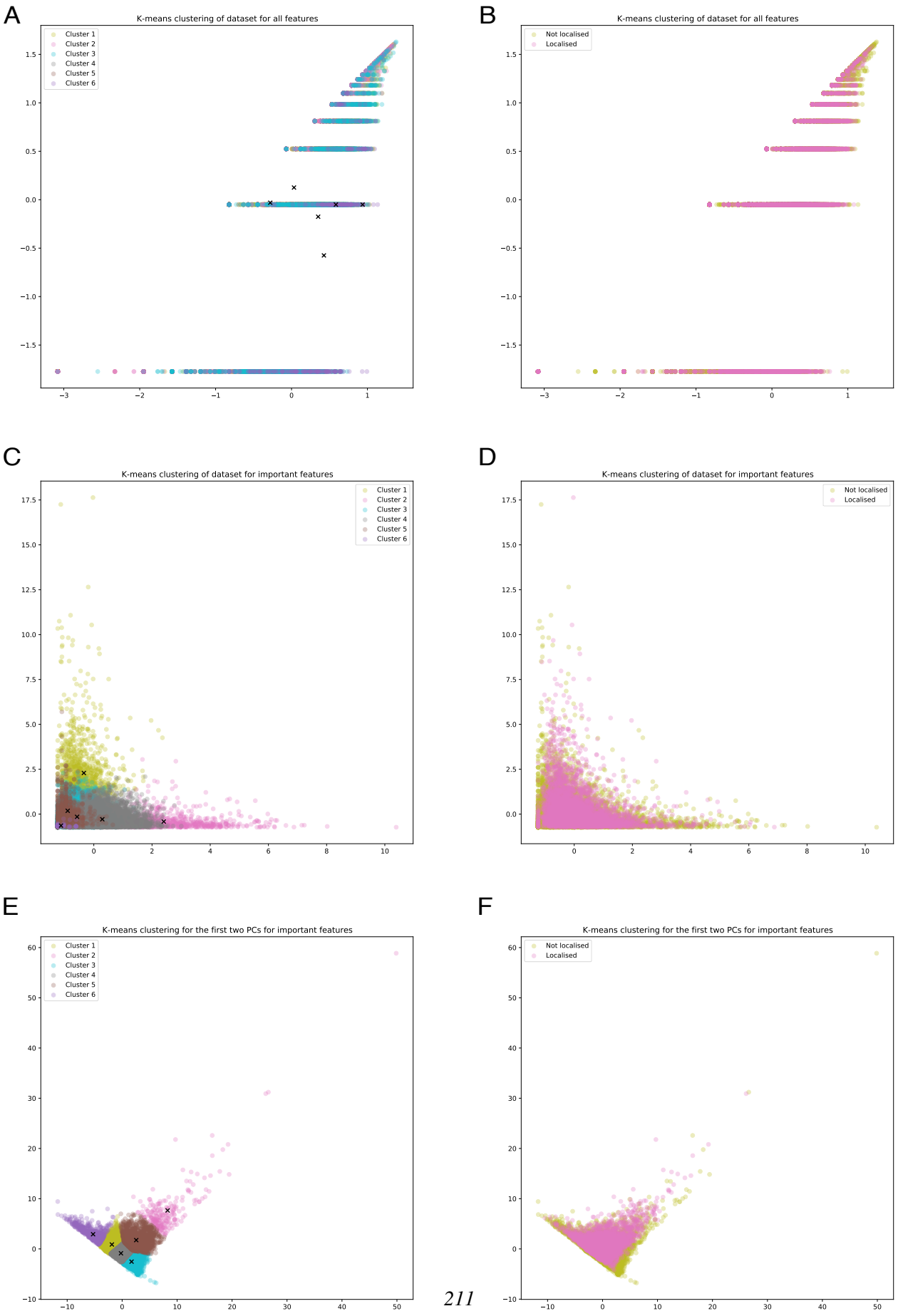
E

Clustering results using k-means

Figure showing the k-means clustering results when applied to the mouse dataset. The figure in Appendix E shows the clustering results when I used the k-means algorithm to cluster the mouse dataset. Panels A and B, show the results of the k-means clustering when we apply it to a normalised and scaled mouse dataset. The colours in Panel A show the predicted cluster that each point is expected to belong to and the colours in Panel B shows the localisation status of each points. The textual representation of each colour is shown in the legends of each panel. Panel C and D show the results of the k-means clustering when we apply it to a filtered subset of the normalised and scaled mouse dataset. I filtered the subset based on the important features result shown in Figure 4.5 of chapter 4. As for Panels A and B, Panels C and D show the same points but Panel C shows their distributions in clusters and D their localisation properties. The cross signs show the centers of the predicted clusters. Panel E and F show the results of the k-means clustering when we apply it to a filtered subset of the mouse dataset that has been transformed using PCA.

We can see that we get the optimal separation of clusters (no overlapping points) when we use the dataset that has been transformed using PCA. Even then the predicted clusters do not match the localisation properties of the samples, so there are not any clear conclusions we can make.

E. Clustering results using k-means



References

Abdi, H., & Williams, L. J. (2010). Principal component analysis: Principal component analysis. *WIREs Comp Stat*, *2*(4), 433–459. https://doi.org/10.1002/wics.101

Adadi, A. (2021). A survey on data-efficient algorithms in big data era. *J Big Data*, *8*(1), 24. https://doi.org/10.1186/s40537-021-00419-9

Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022). Machine-Learning-Based Disease Diagnosis: A Comprehensive Review. *Healthcare*, *10*(3), 541. https://doi.org/10.3390/healthcare10030541

Alberts, B. (Ed.). (2002). *Molecular biology of the cell* (4th ed). Garland Science.

Allan, C., Burel, J.-M., Moore, J., Blackburn, C., Linkert, M., Loynton, S., MacDonald, D., Moore, W. J., Neves, C., Patterson, A., Porter, M., Tarkowska, A., Loranger, B., Avondo, J., Lagerstedt, I., Lianas, L., Leo, S., Hands, K., Hay, R. T., … Swedlow, J. R. (2012). OMERO: Flexible, model-driven data management for experimental biology. *Nat Methods*, *9*(3), 245–253. https://doi.org/10.1038/nmeth.1896

AlQuraishi, M. (2021). Machine learning in protein structure prediction. *Current Opinion in Chemical Biology*, *65*, 1–8. https://doi.org/10.1016/j.cbpa.2021.04.005

Altman, N., & Krzywinski, M. (2017). Clustering. *Nat Methods*, *14*(6), 545–546. https://doi.org/10.1038/nmeth.4299

Alwosheel, A., Van Cranenburgh, S., & Chorus, C. G. (2018). Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of Choice Modelling*, *28*, 167–182. https://doi.org/10.1016/j.jocm.2018.07.002

An, J. J., Gharami, K., Liao, G.-Y., Woo, N. H., Lau, A. G., Vanevski, F., Torre, E. R., Jones, K. R., Feng, Y., Lu, B., & Xu, B. (2008). Distinct Role of Long 3' UTR BDNF mRNA in Spine Morphology and Synaptic Plasticity in Hippocampal Neurons. *Cell*, *134*(1), 175–187. https://doi.org/10.1016/j.cell.2008.05.045

Apweiler, R. (2004). UniProt: The Universal Protein knowledgebase. *Nucleic Acids Research*, *32*(90001), 115D–119. https://doi.org/10.1093/nar/gkh131

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). Gene Ontology: Tool for the unification of biology. *Nat Genet*, *25*(1), 25–29.

https://doi.org/10.1038/75556

Azadifar, S., & Ahmadi, A. (2022). A novel candidate disease gene prioritization method using deep graph convolutional networks and semi-supervised learning. *BMC Bioinformatics*, *23*(1), 422. https://doi.org/10.1186/s12859-022-04954-x

Azevedo, M. M., Domingues, H. S., Cordelières, F. P., Sampaio, P., Seixas, A. I., & Relvas, J. B. (2018). Jmy regulates oligodendrocyte differentiation via modulation of actin cytoskeleton dynamics. *Glia*, *66*(9), 1826–1844. https://doi.org/10.1002/glia.23342

Bahry, E., Breimann, L., Zouinkhi, M., Epstein, L., Kolyvanov, K., Mamrak, N., King, B., Long, X., Harrington, K. I. S., Lionnet, T., & Preibisch, S. (2022). RS-FISH: Precise, interactive, fast, and scalable FISH spot detection. *Nat Methods*, *19*(12), 1563–1567. https://doi.org/10.1038/s41592-022-01669-y

Barbosa, C., Peixeiro, I., & Romão, L. (2013). Gene Expression Regulation by Upstream Open Reading Frames and Human Disease. *PLoS Genet*, *9*(8), e1003529. https://doi.org/10.1371/journal.pgen.1003529

Bekker, J., & Davis, J. (2020). Learning from positive and unlabeled data: A survey. *Mach Learn*, *109*(4), 719–760. https://doi.org/10.1007/s10994-020-05877-5

Benoit Bouvrette, L. P., Bovaird, S., Blanchette, M., & Lécuyer, E. (2019). oRNAment: A database of putative RNA binding protein target sites in the transcriptomes of model species. *Nucleic Acids Research*, gkz986. https://doi.org/10.1093/nar/gkz986

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*, 281–305. http://dblp.uni-trier.de/db/journals/jmlr/jmlr13.html#BergstraB12

Berkowitz, N. D., Silverman, I. M., Childress, D. M., Kazan, H., Wang, L.-S., & Gregory, B. D. (2016). A comprehensive database of high-throughput sequencing-based RNA secondary structure probing data (Structure Surfer). *BMC Bioinformatics*, *17*(1), 215. https://doi.org/10.1186/s12859-016-1071-0

Bittern, J., Pogodalla, N., Ohm, H., Brüser, L., Kottmeier, R., Schirmeier, S., & Klämbt, C. (2021). Neuron–glia interaction in the Drosophila nervous system. *Develop. Neurobiol.*, *81*(5), 438–452. https://doi.org/10.1002/dneu.22737

Blake, J. A., Baldarelli, R., Kadin, J. A., Richardson, J. E., Smith, C. L., Bult, C. J., the Mouse Genome Database Group, Anagnostopoulos, A. V., Beal, J. S., Bello, S. M., Blodgett, O., Butler, N. E., Campbell, J., Christie, K. R., Corbani, L. E., Dolan, M. E., Drabkin, H. J., Flores, M., Giannatto, S. L., … Zhu, Y. 'Sophia'. (2021). Mouse Genome Database (MGD): Knowledgebase for mouse–human comparative biology. *Nucleic Acids Research*, *49*(D1), D981–D987. https://doi.org/10.1093/nar/gkaa1083

Blanco-Urrejola, M., Gaminde-Blasco, A., Gamarra, M., Cruz, A. de la, Vecino, E., Alberdi, E., & Baleriola, J. (2021). RNA Localization and Local Translation in Glia in Neurological

and Neurodegenerative Diseases: Lessons from Neurons. *Cells*, *10*(3), 632. https://doi.org/10.3390/cells10030632

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-Based Systems*, *86*, 33–45. https://doi.org/10.1016/j.knosys.2015.05.014

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Buccitelli, C., & Selbach, M. (2020). mRNAs, proteins and the emerging principles of gene expression control. *Nat Rev Genet*, *21*(10), 630–644. https://doi.org/10.1038/s41576-020-0258-4

Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Naumann, F., & Harmouch, H. (2022). *The Effects of Data Quality on Machine Learning Performance*. arXiv. http://arxiv.org/abs/2207.14529

Burel, J.-M., Besson, S., Blackburn, C., Carroll, M., Ferguson, R. K., Flynn, H., Gillen, K., Leigh, R., Li, S., Lindner, D., Linkert, M., Moore, W. J., Ramalingam, B., Rozbicki, E., Tarkowska, A., Walczysko, P., Allan, C., Moore, J., & Swedlow, J. R. (2015). Publishing and sharing multi-dimensional image data with OMERO. *Mamm Genome*, *26*(9-10), 441–447. https://doi.org/10.1007/s00335-015-9587-6

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Caleydo. (n.d.). http://www.caleydo.org/

Carbon, S., & Mungall, C. (2023). *Gene Ontology Data Archive*. Zenodo. https://doi.org/10.5281/ZENODO.7709866

Carracedo-Reboredo, P., Liñares-Blanco, J., Rodríguez-Fernández, N., Cedrón, F., Novoa, F. J., Carballal, A., Maojo, V., Pazos, A., & Fernandez-Lozano, C. (2021). A review on machine learning approaches and trends in drug discovery. *Computational and Structural Biotechnology Journal*, *19*, 4538–4558. https://doi.org/10.1016/j.csbj.2021.08.011

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Chekulaeva, M., & Landthaler, M. (2016). Eyes on Translation. *Molecular Cell*, *63*(6), 918–925. https://doi.org/10.1016/j.molcel.2016.08.031

Chen, J., Althagafi, A., & Hoehndorf, R. (2021). Predicting candidate genes from phenotypes, functions and anatomical site of expression. *Bioinformatics*, *37*(6), 853–860. https://doi.org/10.1093/bioinformatics/btaa879

Chen, Y., Li, Y., Narayan, R., Subramanian, A., & Xie, X. (2016). Gene expression inference

with deep learning. *Bioinformatics*, *32*(12), 1832–1839. https://doi.org/10.1093/bioinformatics/btw074

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., Xie, W., Rosen, G. L., Lengerich, B. J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A. E., Shrikumar, A., Xu, J., … Greene, C. S. (2018). Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface.*, *15*(141), 20170387. https://doi.org/10.1098/rsif.2017.0387

Chollet, F. (2018). *Deep learning with Python*. Manning Publications Co.

Chouaib, R., Safieddine, A., Pichon, X., Imbert, A., Kwon, O. S., Samacoits, A., Traboulsi, A.-M., Robert, M.-C., Tsanov, N., Coleno, E., Poser, I., Zimmer, C., Hyman, A., Le Hir, H., Zibara, K., Peter, M., Mueller, F., Walter, T., & Bertrand, E. (2020). A Dual Protein-mRNA Localization Screen Reveals Compartmentalized Translation and Widespread Co-translational RNA Targeting. *Developmental Cell*, *54*(6), 773–791.e5. https://doi.org/10.1016/j.devcel.2020.07.010

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Mach Learn*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

COSMIC – COVID-19 Lung Single Cell Mass Cytometry Imaging Consortium STUDY, Weeratunga, P., Denney, L., Bull, J. A., Repapi, E., Sergeant, M., Etherington, R., Vuppussetty, C., Turner, G. D. H., Clelland, C., Cross, A., Issa, F., Andrea, C. E. de, Bermejo, I. M., Sims, D., McGowan, S., Zurke, Y.-X., Ahern, D. J., Gamez, E. C., … Ho, L.-P. (2022). *Unbiased single cell spatial analysis localises inflammatory clusters of immature neutrophils-CD8 T cells to alveolar progenitor cells in fatal COVID-19 lungs* [Preprint]. Respiratory Medicine. https://doi.org/10.1101/2022.12.21.22283654

Cothren, S., Meyer, J., & Hartman, J. (2018). Blinded Visual Scoring of Images Using the Freely-available Software Blinder. *BIO-PROTOCOL*, *8*(23). https://doi.org/10.21769/BioProtoc.3103

Court, R., Namiki, S., Armstrong, J. D., Börner, J., Card, G., Costa, M., Dickinson, M., Duch, C., Korff, W., Mann, R., Merritt, D., Murphey, R. K., Seeds, A. M., Shirangi, T., Simpson, J. H., Truman, J. W., Tuthill, J. C., Williams, D. W., & Shepherd, D. (2020). A Systematic Nomenclature for the Drosophila Ventral Nerve Cord. *Neuron*, *107*(6), 1071–1079.e2. https://doi.org/10.1016/j.neuron.2020.08.005

Cox, J., Oh, E. Y., Simmons, B., Lintott, C., Masters, K., Greenhill, A., Graham, G., & Holmes, K. (2015). Defining and Measuring Success in Online Citizen Science: A Case Study of Zooniverse Projects. *Comput. Sci. Eng.*, *17*(4), 28–41. https://doi.org/10.1109/MCSE.2015.65

Cruz, A., Arrais, J. P., & Machado, P. (2019). Interactive and coordinated visualization approaches for biological data analysis. *Briefings in Bioinformatics*, *20*(4), 1513–1523. https://doi.org/10.1093/bib/bby019

Cui, T., Dou, Y., Tan, P., Ni, Z., Liu, T., Wang, D., Huang, Y., Cai, K., Zhao, X., Xu, D., Lin, H., & Wang, D. (2022). RNALocate v2.0: An updated resource for RNA subcellular localization with increased coverage and annotation. *Nucleic Acids Research*, *50*(D1), D333–D339. https://doi.org/10.1093/nar/gkab825

Cunningham, F., Allen, J. E., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., Austine-Orimoloye, O., Azov, A. G., Barnes, I., Bennett, R., Berry, A., Bhai, J., Bignell, A., Billis, K., Boddu, S., Brooks, L., Charkhchi, M., Cummins, C., Da Rin Fioretto, L., … Flicek, P. (2022). Ensembl 2022. *Nucleic Acids Research*, *50*(D1), D988–D995. https://doi.org/10.1093/nar/gkab1049

Danasingh, A. A. G. S., Subramanian, A. alias B., & Epiphany, J. L. (2020). Identifying redundant features using unsupervised learning for high-dimensional data. *SN Appl. Sci.*, *2*(8), 1367. https://doi.org/10.1007/s42452-020-3157-6

Dara, S., Dhamercherla, S., Jadav, S. S., Babu, C. M., & Ahsan, M. J. (2022). Machine Learning in Drug Discovery: A Review. *Artif Intell Rev*, *55*(3), 1947–1999. https://doi.org/10.1007/s10462-021-10058-4

Data Mining. (2012). [Computer software]. Elsevier. https://doi.org/10.1016/C2009-0-61819-5

Davani, A. M., Díaz, M., & Prabhakaran, V. (2022). Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations. *Transactions of the Association for Computational Linguistics*, *10*, 92–110. https://doi.org/10.1162/tacl_a_00449

Degrauwe, N., Suvà, M.-L., Janiszewska, M., Riggi, N., & Stamenkovic, I. (2016). IMPs: An RNA-binding protein family that provides a link between stem cell maintenance in normal development and cancer. *Genes Dev.*, *30*(22), 2459–2474. https://doi.org/10.1101/gad.287540.116

Ding, H., Wang, C., Huang, K., & Machiraju, R. (2014). iGPSe: A visual analytic system for integrative genomic based cancer patient stratification. *BMC Bioinformatics*, *15*(1), 203. https://doi.org/10.1186/1471-2105-15-203

Donlin-Asp, P. G., Polisseni, C., Klimek, R., Heckel, A., & Schuman, E. M. (2021). Differential regulation of local mRNA dynamics and translation following long-term potentiation and depression. *Proc. Natl. Acad. Sci. U.S.A.*, *118*(13), e2017578118. https://doi.org/10.1073/pnas.2017578118

Dunn, W., Burgun, A., Krebs, M.-O., & Rance, B. (2016). Exploring and visualizing multidimensional data in translational research platforms. *Brief Bioinform*, bbw080. https://doi.org/10.1093/bib/bbw080

Edgar, R. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization ar-

ray data repository. *Nucleic Acids Research*, *30*(1), 207–210. https://doi.org/10.1093/nar/30.1.207

Eichenberger, B. T., Zhan, Y., Rempfler, M., Giorgetti, L., & Chao, J. A. (2021). deepBlink: Threshold-independent detection and localization of diffraction-limited spots. *Nucleic Acids Research*, *49*(13), 7292–7297. https://doi.org/10.1093/nar/gkab546

Eliceiri, K. W., Berthold, M. R., Goldberg, I. G., Ibáñez, L., Manjunath, B. S., Martone, M. E., Murphy, R. F., Peng, H., Plant, A. L., Roysam, B., Stuurman, N., Swedlow, J. R., Tomancak, P., & Carpenter, A. E. (2012). Biological imaging software tools. *Nat Methods*, *9*(7), 697–710. https://doi.org/10.1038/nmeth.2084

Emrich, S. J., Barbazuk, W. B., Li, L., & Schnable, P. S. (2007). Gene discovery and annotation using LCM-454 transcriptome sequencing. *Genome Res.*, *17*(1), 69–73. https://doi.org/10.1101/gr.5145806

Eroglu, C., & Barres, B. A. (2010). Regulation of synaptic connectivity by glia. *Nature*, *468*(7321), 223–231. https://doi.org/10.1038/nature09612

Everitt, B. (Ed.). (2011). *Cluster analysis* (5th ed). Wiley.

Femino, A. M., Fay, F. S., Fogarty, K., & Singer, R. H. (1998). Visualization of Single RNA Transcripts in Situ. *Science*, *280*(5363), 585–590. https://doi.org/10.1126/science.280.5363.585

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378–382. https://doi.org/10.1037/h0031619

Flynn, R. A., Zhang, Q. C., Spitale, R. C., Lee, B., Mumbach, M. R., & Chang, H. Y. (2016). Transcriptome-wide interrogation of RNA secondary structure in living cells with icSHAPE. *Nat Protoc*, *11*(2), 273–290. https://doi.org/10.1038/nprot.2016.011

Fowler, P. W., Wright, C., Spiers, H., Zhu, T., Baeten, E. M., Hoosdally, S. W., Gibertoni Cruz, A. L., Roohi, A., Kouchaki, S., Walker, T. M., Peto, T. E., Miller, G., Lintott, C., Clifton, D., Crook, D. W., Walker, A. S., The Zooniverse Volunteer Community, & The CRyPTIC Consortium. (2022). A crowd of BashTheBug volunteers reproducibly and accurately measure the minimum inhibitory concentrations of 13 antitubercular drugs from photographs of 96-well broth microdilution plates. *eLife*, *11*, e75046. https://doi.org/10.7554/eLife.75046

Freeman, M. R., Delrow, J., Kim, J., Johnson, E., & Doe, C. Q. (2003). Unwrapping Glial Biology. *Neuron*, *38*(4), 567–580. https://doi.org/10.1016/S0896-6273(03)00289-7

Freund, Y., & Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In G. Goos, J. Hartmanis, J. Leeuwen, J. G. Carbonell, J. Siekmann, & P. Vitányi (Eds.), *Computational Learning Theory* (Vol. 904, pp. 23–37). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-59119-2_166

Friederich, P., Krenn, M., Tamblyn, I., & Aspuru-Guzik, A. (2021). Scientific intuition inspired by machine learning-generated hypotheses. *Mach. Learn.: Sci. Technol.*, *2*(2), 025027.

https://doi.org/10.1088/2632-2153/abda08

Gala, D. S., Lee, J. Y., Kiourlappou, M., Titlow, J. S., Teodoro, R. O., & Davis, I. (2022). *Mammalian glial protrusion transcriptomes predict localization of* drosophila *glial transcripts required for synaptic plasticity* [Preprint]. Cell Biology. https://doi.org/10.1101/2022.11.30.518536

Garg, A., Singhal, N., Kumar, R., & Kumar, M. (2020). mRNALoc: A novel machine-learning based in-silico tool to predict mRNA subcellular localization. *Nucleic Acids Research*, *48*(W1), W239–W243. https://doi.org/10.1093/nar/gkaa385

Geisler, C., & Swarts, J. (2019). *Coding Streams of Language: Techniques for the Systematic Coding of Text, Talk, and Other Verbal Data*. The WAC Clearinghouse; University Press of Colorado. https://doi.org/10.37514/PRA-B.2019.0230

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (First edition). O'Reilly Media.

Ghosh, A., Nashaat, M., Miller, J., Quader, S., & Marston, C. (2018). A comprehensive review of tools for exploratory analysis of tabular industrial datasets. *Visual Informatics*, *2*(4), 235–253. https://doi.org/10.1016/j.visinf.2018.12.004

Giandomenico, S. L., Alvarez-Castelao, B., & Schuman, E. M. (2022). Proteostatic regulation in neuronal compartments. *Trends in Neurosciences*, *45*(1), 41–52. https://doi.org/10.1016/j.tins.2021.08.002

Gligorijević, V., & Pržulj, N. (2015). Methods for biological data integration: Perspectives and challenges. *J. R. Soc. Interface.*, *12*(112), 20150571. https://doi.org/10.1098/rsif.2015.0571

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Gramates, L. S., Agapite, J., Attrill, H., Calvi, B. R., Crosby, M. A., Santos, G. dos, Goodman, J. L., Goutte-Gattat, D., Jenkins, V. K., Kaufman, T., Larkin, A., Matthews, B. B., Millburn, G., Strelets, V. B., the FlyBase Consortium, Perrimon, N., Gelbart, S. R., Agapite, J., Broll, K., … Lovato, T. (2022). FlyBase: A guided tour of highlighted features. *Genetics*, *220*(4), iyac035. https://doi.org/10.1093/genetics/iyac035

Gratzl, S., Gehlenborg, N., Lex, A., Pfister, H., & Streit, M. (2014). Domino: Extracting, Comparing, and Manipulating Subsets Across Multiple Tabular Datasets. *IEEE Trans. Visual. Comput. Graphics*, *20*(12), 2023–2032. https://doi.org/10.1109/TVCG.2014.2346260

Greener, J. G., Kandathil, S. M., Moffat, L., & Jones, D. T. (2022a). A guide to machine learning for biologists. *Nat Rev Mol Cell Biol*, *23*(1), 40–55. https://doi.org/10.1038/s41580-021-00407-0

Greener, J. G., Kandathil, S. M., Moffat, L., & Jones, D. T. (2022b). A guide to machine learning for biologists. *Nat Rev Mol Cell Biol*, *23*(1), 40–55. https://doi.org/10.1038/s41580-021-00407-0

Guide to go subsets. (n.d.). http://geneontology.org/docs/go-subset-guide/

Gurney, K. (1997). *An introduction to neural networks*. UCL Press.

Häcker, U., Nystedt, S., Barmchi, M. P., Horn, C., & Wimmer, E. A. (2003). *piggyBac* -based insertional mutagenesis in the presence of stably integrated *p* elements in *drosophila*. *Proc. Natl. Acad. Sci. U.S.A.*, *100*(13), 7720–7725. https://doi.org/10.1073/pnas.1230526100

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th Python in Science Conference* (pp. 11–15).

Hamilton, R. S., & Davis, I. (2007). RNA localization signals: Deciphering the message with bioinformatics. *Seminars in Cell & Developmental Biology*, *18*(2), 178–185. https://doi.org/10.1016/j.semcdb.2007.02.001

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Trans. Pattern Anal. Machine Intell.*, *12*(10), 993–1001. https://doi.org/10.1109/34.58871

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Harrow, J., Frankish, A., Gonzalez, J. M., Tapanari, E., Diekhans, M., Kokocinski, F., Aken, B. L., Barrell, D., Zadissa, A., Searle, S., Barnes, I., Bignell, A., Boychenko, V., Hunt, T., Kay, M., Mukherjee, G., Rajan, J., Despacio-Reyes, G., Saunders, G., … Hubbard, T. J. (2012). GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Res.*, *22*(9), 1760–1774. https://doi.org/10.1101/gr.135350.111

Hartman, J. H., Smith, L. L., Gordon, K. L., Laranjeiro, R., Driscoll, M., Sherwood, D. R., & Meyer, J. N. (2018). Swimming Exercise and Transient Food Deprivation in Caenorhabditis elegans Promote Mitochondrial Maintenance and Protect Against Chemical-Induced Mitotoxicity. *Sci Rep*, *8*(1), 8359. https://doi.org/10.1038/s41598-018-26552-9

Hentze, M. W., Castello, A., Schwarzl, T., & Preiss, T. (2018). A brave new world of RNA-binding proteins. *Nat Rev Mol Cell Biol*, *19*(5), 327–341. https://doi.org/10.1038/nrm.2017.130

Hilgers, V., Lemke, S. B., & Levine, M. (2012). ELAV mediates 3' UTR extension in the *drosophila* nervous system. *Genes Dev.*, *26*(20), 2259–2264. https://doi.org/10.1101/gad.199653.112

Hilgers, V., Perry, M. W., Hendrix, D., Stark, A., Levine, M., & Haley, B. (2011). Neural-specific elongation of 3' UTRs during *drosophila* development. *Proc. Natl. Acad. Sci. U.S.A.*, *108*(38), 15864–15869. https://doi.org/10.1073/pnas.1112672108

Hill, D. P., Blake, J. A., Richardson, J. E., & Ringwald, M. (2002). Extension and Integration of

the Gene Ontology (GO): Combining GO Vocabularies With External Vocabularies. *Genome Res.*, *12*(12), 1982–1991. https://doi.org/10.1101/gr.580102

Holt, C. E., Martin, K. C., & Schuman, E. M. (2019). Local translation in neurons: Visualization and function. *Nat Struct Mol Biol*, *26*(7), 557–566. https://doi.org/10.1038/s41594-019-0263-5

Holt, C. E., & Schuman, E. M. (2013). The Central Dogma Decentralized: New Perspectives on RNA Function and Local Translation in Neurons. *Neuron*, *80*(3), 648–657. https://doi.org/10.1016/j.neuron.2013.10.036

Hu, Y., Comjean, A., Rodiger, J., Liu, Y., Gao, Y., Chung, V., Zirin, J., Perrimon, N., & Mohr, S. E. (2021). FlyRNAi.org—the database of the Drosophila RNAi screening center and transgenic RNAi project: 2021 update. *Nucleic Acids Research*, *49*(D1), D908–D915. https://doi.org/10.1093/nar/gkaa936

Hu, Y., Flockhart, I., Vinayagam, A., Bergwitz, C., Berger, B., Perrimon, N., & Mohr, S. E. (2011). An integrative approach to ortholog prediction for disease-focused and other functional studies. *BMC Bioinformatics*, *12*(1), 357. https://doi.org/10.1186/1471-2105-12-357

Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. R. (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, *21*(8), 1509–1515. https://doi.org/10.1093/bioinformatics/bti171

Hubé, F., & Francastel, C. (2015). Mammalian Introns: When the Junk Generates Molecular Diversity. *IJMS*, *16*(3), 4429–4452. https://doi.org/10.3390/ijms16034429

Huntley, R. P., Sawford, T., Martin, M. J., & O'Donovan, C. (2014). Understanding how and why the Gene Ontology and its annotations evolve: The GO within UniProt. *GigaSci*, *3*(1), 4. https://doi.org/10.1186/2047-217X-3-4

Idrissi, N., & Zellou, A. (2020). A systematic literature review of sparsity issues in recommender systems. *Soc. Netw. Anal. Min.*, *10*(1), 15. https://doi.org/10.1007/s13278-020-0626-2

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, *622*, 178–210. https://doi.org/10.1016/j.ins.2022.11.139

Imaging AI Software Company. (n.d.). In *Zegami*. Retrieved December 12, 2022, from https://zegami.com/

Imbert, A., Ouyang, W., Safieddine, A., Coleno, E., Zimmer, C., Bertrand, E., Walter, T., & Mueller, F. (2022). FISH-quant v2: A scalable and modular tool for smFISH image analysis. *RNA*, *28*(6), 786–795. https://doi.org/10.1261/rna.079073.121

Ismaila, L. E., Rasti, P., Bernard, F., Labriffe, M., Menei, P., Minassian, A. T., Rousseau, D., & Lemée, J.-M. (2022). Transfer Learning from Healthy to Unhealthy Patients for the Automated Classification of Functional Brain Networks in fMRI. *Applied Sciences*, *12*(14),

6925. https://doi.org/10.3390/app12146925

Jain, S., & Bader, G. D. (2010). An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. *BMC Bioinformatics*, *11*(1), 562. https://doi.org/10.1186/1471-2105-11-562

Janitza, S., & Hornung, R. (2018). On the overestimation of random forest's out-of-bag error. *PLoS ONE*, *13*(8), e0201904. https://doi.org/10.1371/journal.pone.0201904

Jassal, B., Matthews, L., Viteri, G., Gong, C., Lorente, P., Fabregat, A., Sidiropoulos, K., Cook, J., Gillespie, M., Haw, R., Loney, F., May, B., Milacic, M., Rothfels, K., Sevilla, C., Shamovsky, V., Shorser, S., Varusai, T., Weiser, J., ... D'Eustachio, P. (2019). The reactome pathway knowledgebase. *Nucleic Acids Research*, gkz1031. https://doi.org/10.1093/nar/gkz1031

John, M. (2021). *Statistical Analysis Handbook*. The Winchelsea Press.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Phil. Trans. R. Soc. A.*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kass, R. E., & Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, *90*(430), 773–795. https://doi.org/10.1080/01621459.1995.10476572

Khatri, P., & Draghici, S. (2005). Ontological analysis of gene expression data: Current tools, limitations, and open problems. *Bioinformatics*, *21*(18), 3587–3595. https://doi.org/10.1093/bioinformatics/bti565

Kim, T., Song, B., & Lee, I.-S. (2020). Drosophila Glia: Models for Human Neurodevelopmental and Neurodegenerative Disorders. *IJMS*, *21*(14), 4859. https://doi.org/10.3390/ijms21144859

Kinsella, R. J., Kahari, A., Haider, S., Zamora, J., Proctor, G., Spudich, G., Almeida-King, J., Staines, D., Derwent, P., Kerhornou, A., Kersey, P., & Flicek, P. (2011). Ensembl BioMarts: A hub for data retrieval across taxonomic space. *Database*, *2011*(0), bar030–bar030. https://doi.org/10.1093/database/bar030

Kiourlappou, M., Sergeant, M., Titlow, J. S., Lee, J. Y., Ennis, D., Taylor, S., & Davis, I. (2023). *Multi-Dimensional Data Viewer (MDV) user manual for data exploration: "Systematic analysis of YFP traps reveals common discordance between mRNA and protein across the nervous system"*. Zenodo. https://doi.org/10.5281/ZENODO.6374011

Kiourlappou, M., Taylor, S., & Davis, I. (2022). *Zegami user manual for data exploration: "Sys-*

tematic analysis of YFP gene traps reveals common discordance between mRNA and protein across the nervous system". Zenodo. https://doi.org/10.5281/ZENODO.7308444

Kiruba Nesamalar, E., Satheesh Kumar, J., & Amudha, T. (2021). Influence of an efficient Hierarchical Clustering Algorithm in analyzing Cancer affected DNA Dataset. *J. Phys.: Conf. Ser.*, *1998*(1), 012030. https://doi.org/10.1088/1742-6596/1998/1/012030

Klopfenstein, D. V., Zhang, L., Pedersen, B. S., Ramírez, F., Warwick Vesztrocy, A., Naldi, A., Mungall, C. J., Yunes, J. M., Botvinnik, O., Weigel, M., Dampier, W., Dessimoz, C., Flick, P., & Tang, H. (2018). GOATOOLS: A Python library for Gene Ontology analyses. *Sci Rep*, *8*(1), 10872. https://doi.org/10.1038/s41598-018-28948-z

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, 1137–1143.

Konopka, B. M., Lwow, F., Owczarz, M., & Łaczmański, Ł. (2018). Exploratory data analysis of a clinical study group: Development of a procedure for exploring multidimensional data. *PLoS ONE*, *13*(8), e0201950. https://doi.org/10.1371/journal.pone.0201950

Kügelgen, N., & Chekulaeva, M. (2020). Conservation of a core neurite transcriptome across neuronal types and species. *WIREs RNA*, *11*(4). https://doi.org/10.1002/wrna.1590

Kuhn Cuellar, L., Friedrich, A., Gabernet, G., Garza, L. de la, Fillinger, S., Seyboldt, A., Koch, T., Oven-Krockhaus, S. zur, Wanke, F., Richter, S., Thaiss, W. M., Horger, M., Malek, N., Harter, K., Bitzer, M., & Nahnsen, S. (2022). A data management infrastructure for the integration of imaging and omics data in life sciences. *BMC Bioinformatics*, *23*(1), 61. https://doi.org/10.1186/s12859-022-04584-3

Kumar, P., Bhatnagar, R., Gaur, K., & Bhatnagar, A. (2021). Classification of Imbalanced Data:Review of Methods and Applications. *IOP Conf. Ser.: Mater. Sci. Eng.*, *1099*(1), 012077. https://doi.org/10.1088/1757-899X/1099/1/012077

Kvilekval, K., Fedorov, D., Obara, B., Singh, A., & Manjunath, B. S. (2010). Bisque: A platform for bioimage analysis and management. *Bioinformatics*, *26*(4), 544–552. https://doi.org/10.1093/bioinformatics/btp699

Lambert, N., Robertson, A., Jangi, M., McGeary, S., Sharp, P. A., & Burge, C. B. (2014). RNA Bind-n-Seq: Quantitative Assessment of the Sequence and Structural Binding Specificity of RNA Binding Proteins. *Molecular Cell*, *54*(5), 887–900. https://doi.org/10.1016/j.molcel.2014.04.016

Lee, J. Y., Wing, P. A., Gala, D. S., Noerenberg, M., Järvelin, A. I., Titlow, J., Zhuang, X., Palmalux, N., Iselin, L., Thompson, M. K., Parton, R. M., Prange-Barczynska, M., Wainman, A., Salguero, F. J., Bishop, T., Agranoff, D., James, W., Castello, A., McKeating, J. A., & Davis, I. (2022). Absolute quantitation of individual SARS-CoV-2 RNA molecules provides a new paradigm for infection dynamics and variant differences. *eLife*, *11*, e74153. https:

//doi.org/10.7554/eLife.74153

Lee, S., Wei, L., Zhang, B., Goering, R., Majumdar, S., Wen, J., Taliaferro, J. M., & Lai, E. C. (2021). ELAV/Hu RNA binding proteins determine multiple programs of neural alternative splicing. *PLoS Genet*, *17*(4), e1009439. https://doi.org/10.1371/journal.pgen.1009439

Lee, W.-N., Shah, N., Sundlass, K., & Musen, M. (2008). Comparison of ontology-based semantic-similarity measures. *AMIA Annu Symp Proc*, *2008*, 384–388.

Lee, Y. Y., Endale, M., Wu, G., Ruben, M. D., Francey, L. J., Morris, A. R., Choo, N. Y., Anafi, R. C., Smith, D. F., Liu, A. C., & Hogenesch, J. B. (2023). Integration of genome-scale data identifies candidate sleep regulators. *Sleep*, *46*(2), zsac279. https://doi.org/10.1093/sleep/zsac279

Lefebvre, F. A., Cody, N. A. L., Bouvrette, L. P. B., Bergalet, J., Wang, X., & Lécuyer, E. (2017). CeFra-seq: Systematic mapping of RNA subcellular distribution properties through cell fractionation coupled to deep-sequencing. *Methods*, *126*, 138–148. https://doi.org/10.1016/j.ymeth.2017.05.017

Lei, Z., & Dai, Y. (2006). Assessing protein similarity with Gene Ontology and its use in subnuclear localization prediction. *BMC Bioinformatics*, *7*(1), 491. https://doi.org/10.1186/1471-2105-7-491

Lever, J., Krzywinski, M., & Altman, N. (2017). Principal component analysis. *Nat Methods*, *14*(7), 641–642. https://doi.org/10.1038/nmeth.4346

Lex, A., Streit, M., Schulz, H.-J., Partl, C., Schmalstieg, D., Park, P. J., & Gehlenborg, N. (2012). StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization. *Computer Graphics Forum*, *31*(3pt3), 1175–1184. https://doi.org/10.1111/j.1467-8659.2012.03110.x

Li, H., Janssens, J., De Waegeneer, M., Kolluru, S. S., Davie, K., Gardeux, V., Saelens, W., David, F. P. A., Brbić, M., Spanier, K., Leskovec, J., McLaughlin, C. N., Xie, Q., Jones, R. C., Brueckner, K., Shim, J., Tattikota, S. G., Schnorrer, F., Rust, K., … Zinzen, R. P. (2022). Fly Cell Atlas: A single-nucleus transcriptomic atlas of the adult fruit fly. *Science*, *375*(6584), eabk2432. https://doi.org/10.1126/science.abk2432

Li, S., Besson, S., Blackburn, C., Carroll, M., Ferguson, R. K., Flynn, H., Gillen, K., Leigh, R., Lindner, D., Linkert, M., Moore, W. J., Ramalingam, B., Rozbicki, E., Rustici, G., Tarkowska, A., Walczysko, P., Williams, E., Allan, C., Burel, J.-M., … Swedlow, J. R. (2016). Metadata management for high content screening in OMERO. *Methods*, *96*, 27–32. https://doi.org/10.1016/j.ymeth.2015.10.006

Li, X.-L., & Liu, B. (2005). Learning from Positive and Unlabeled Examples with Different Data Distributions. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos,

D. Tygar, M. Y. Vardi, G. Weikum, J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, & L. Torgo (Eds.), *Machine Learning: ECML 2005* (Vol. 3720, pp. 218–229). Springer Berlin Heidelberg. https://doi.org/10.1007/11564096_24

Li, X., & Wang, C.-Y. (2021). From bulk, single-cell to spatial RNA sequencing. *Int J Oral Sci*, *13*(1), 36. https://doi.org/10.1038/s41368-021-00146-0

Li, Z., Wang, Y., Ganan-Gomez, I., Colla, S., & Do, K.-A. (2022). A machine learning-based method for automatically identifying novel cells in annotating single-cell RNA-seq data. *Bioinformatics*, *38*(21), 4885–4892. https://doi.org/10.1093/bioinformatics/btac617

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., & Mesirov, J. P. (2011). Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, *27*(12), 1739–1740. https://doi.org/10.1093/bioinformatics/btr260

Light, R. J. (1971). Measures of response agreement for qualitative data: Some generalizations and alternatives. *Psychological Bulletin*, *76*(5), 365–377. https://doi.org/10.1037/h0031643

Lim, S. J., Son, M., Ki, S. J., Suh, S.-I., & Chung, J. (2023). Opportunities and challenges of machine learning in bioprocesses: Categorization from different perspectives and future direction. *Bioresource Technology*, *370*, 128518. https://doi.org/10.1016/j.biortech.2022.128518

Lin, D. (1998). An Information-Theoretic Definition of Similarity. In J. W. Shavlik (Ed.), *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconson, USA, July 24-27, 1998* (pp. 296–304). Morgan Kaufmann. http://dblp.uni-trier.de/db/conf/icml/icml1998.html#Lin98

Lin, W.-Y., Liu, C.-H., Cheng, J., & Liu, H.-P. (2022). Alterations of RNA-binding protein found in neurons in Drosophila neurons and glia influence synaptic transmission and lifespan. *Front. Mol. Neurosci.*, *15*, 1006455. https://doi.org/10.3389/fnmol.2022.1006455

Lin, Y., Afshar, S., Rajadhyaksha, A. M., Potash, J. B., & Han, S. (2020). A Machine Learning Approach to Predicting Autism Risk Genes: Validation of Known Genes and Discovery of New Candidates. *Front. Genet.*, *11*, 500064. https://doi.org/10.3389/fgene.2020.500064

Lister, R., O'Malley, R. C., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H., & Ecker, J. R. (2008). Highly Integrated Single-Base Resolution Maps of the Epigenome in Arabidopsis. *Cell*, *133*(3), 523–536. https://doi.org/10.1016/j.cell.2008.03.029

Liu, F., & Demosthenes, P. (2022). Real-world data: A brief review of the methods, applications, challenges and opportunities. *BMC Med Res Methodol*, *22*(1), 287. https://doi.org/10.1186/s12874-022-01768-6

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, *28*(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

Loizides, F., & Schmidt, B. (Eds.). (2016). *Positioning and power in academic publishing: Players, agents and agendas: Proceedings of the 20th International Conference on Electronic Publishing*. Ios Press, Inc.

Lones, M. A. (2023). *How to avoid machine learning pitfalls: A guide for academic researchers*. arXiv. http://arxiv.org/abs/2108.02497

Lord, P. W., Stevens, R. D., Brass, A., & Goble, C. A. (2002). SEMANTIC SIMILARITY MEASURES AS TOOLS FOR EXPLORING THE GENE ONTOLOGY. *Biocomputing 2003*, 601–612. https://doi.org/10.1142/9789812776303_0056

Lorenz, R., Bernhart, S. H., Höner Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., & Hofacker, I. L. (2011). ViennaRNA Package 2.0. *Algorithms Mol Biol*, *6*(1), 26. https://doi.org/10.1186/1748-7188-6-26

Lowe, N., Rees, J. S., Roote, J., Ryder, E., Armean, I. M., Johnson, G., Drummond, E., Spriggs, H., Drummond, J., Magbanua, J. P., Naylor, H., Sanson, B., Bastock, R., Huelsmann, S., Trovisco, V., Landgraf, M., Knowles-Barley, S., Armstrong, J. D., White-Cooper, H., … St Johnston, D. (2014). Analysis of the expression patterns, subcellular localisations and interaction partners of *drosophila* proteins using a *pigP* protein trap library. *Development*, *141*(20), 3994–4005. https://doi.org/10.1242/dev.111054

Lubeck, E., Coskun, A. F., Zhiyentayev, T., Ahmad, M., & Cai, L. (2014). Single-cell in situ RNA profiling by sequential hybridization. *Nat Methods*, *11*(4), 360–361. https://doi.org/10.1038/nmeth.2892

Luebbert, L., & Pachter, L. (2023). Efficient querying of genomic reference databases with *gget*. *Bioinformatics*, *39*(1), btac836. https://doi.org/10.1093/bioinformatics/btac836

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nat Mach Intell*, *2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Luo, J., Poursafaei, F., & Liu, X. (2023). *Towards Improved Illicit Node Detection with Positive-Unlabelled Learning*. arXiv. http://arxiv.org/abs/2303.02462

Lyne, R., Smith, R., Rutherford, K., Wakeling, M., Varley, A., Guillier, F., Janssens, H., Ji, W., Mclaren, P., North, P., Rana, D., Riley, T., Sullivan, J., Watkins, X., Woodbridge, M., Lilley, K., Russell, S., Ashburner, M., Mizuguchi, K., & Micklem, G. (2007). FlyMine: An integrated database for Drosophila and Anopheles genomics. *Genome Biol*, *8*(7), R129. https://doi.org/10.1186/gb-2007-8-7-r129

MacDonald, J. M., Beach, M. G., Porpiglia, E., Sheehan, A. E., Watts, R. J., & Freeman, M. R. (2006). The Drosophila Cell Corpse Engulfment Receptor Draper Mediates Glial Clearance of Severed Axons. *Neuron*, *50*(6), 869–881. https://doi.org/10.1016/j.neuron.2006.04.028

Maglott, D., Ostell, J., Pruitt, K. D., & Tatusova, T. (2011). Entrez Gene: Gene-centered information at NCBI. *Nucleic Acids Research*, *39*(Database), D52–D57. https://doi.org/10.1093/nar/gkq1237

Marcoulides, K. M., & Raykov, T. (2019). Evaluation of Variance Inflation Factors in Regression Models Using Latent Variable Modeling Methods. *Educational and Psychological Measurement*, *79*(5), 874–882. https://doi.org/10.1177/0013164418817803

Markus, A. F., Kors, J. A., & Rijnbeek, P. R. (2021). The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, *113*, 103655. https://doi.org/10.1016/j.jbi.2020.103655

Martin, K. C., & Ephrussi, A. (2009). mRNA Localization: Gene Expression in the Spatial Dimension. *Cell*, *136*(4), 719–730. https://doi.org/10.1016/j.cell.2009.01.044

Marx, V. (2021). Method of the Year: Spatially resolved transcriptomics. *Nat Methods*, *18*(1), 9–14. https://doi.org/10.1038/s41592-020-01033-y

Matthews, K. A., Kaufman, T. C., & Gelbart, W. M. (2005). Research resources for Drosophila: The expanding universe. *Nat Rev Genet*, *6*(3), 179–193. https://doi.org/10.1038/nrg1554

Mazandu, G. K., Chimusa, E. R., & Mulder, N. J. (2016). Gene Ontology semantic similarity tools: Survey on features and challenges for biological knowledge discovery. *Brief Bioinform*, bbw067. https://doi.org/10.1093/bib/bbw067

McInnes, B. T., & Pedersen, T. (2013). Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *J Biomed Inform*, *46*(6), 1116–1124. https://doi.org/10.1016/j.jbi.2013.08.008

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Medioni, C., Mowry, K., & Besse, F. (2012). Principles and roles of mRNA localization in animal development. *Development*, *139*(18), 3263–3276. https://doi.org/10.1242/dev.078626

Meher, P. K., Rai, A., & Rao, A. R. (2021). mLoc-mRNA: Predicting multiple sub-cellular localization of mRNAs using random forest algorithm coupled with feature selection via elastic net. *BMC Bioinformatics*, *22*(1), 342. https://doi.org/10.1186/s12859-021-04264-8

Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L. L., Tosatto, S. C. E., Paladin, L., Raj, S., Richardson, L. J., Finn, R. D., & Bateman, A. (2021). Pfam: The protein families database in 2021. *Nucleic Acids Research*, *49*(D1), D412–D419. https://doi.org/10.1093/nar/gkaa913

Monarch, R., & Manning, C. D. (2021). *Human in the Loop Machine Learning: Active Learning and Annotation for Human-Centered AI*.

Motenko, H., Neuhauser, S. B., O'Keefe, M., & Richardson, J. E. (2015). MouseMine: A new data warehouse for MGI. *Mamm Genome*, *26*(7-8), 325–330. https://doi.org/10.1007/s00335-015-9573-z

Mueller, F., Senecal, A., Tantale, K., Marie-Nelly, H., Ly, N., Collin, O., Basyuk, E., Bertrand, E., Darzacq, X., & Zimmer, C. (2013). FISH-quant: Automatic counting of transcripts in 3D FISH images. *Nat Methods*, *10*(4), 277–278. https://doi.org/10.1038/nmeth.2406

Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., & Snyder, M. (2008). The Transcriptional Landscape of the Yeast Genome Defined by RNA Sequencing. *Science*, *320*(5881), 1344–1349. https://doi.org/10.1126/science.1158441

Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance? *Bioinformatics*, *34*(21), 3711–3718. https://doi.org/10.1093/bioinformatics/bty373

Nguyen, L. H., & Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLoS Comput Biol*, *15*(6), e1006907. https://doi.org/10.1371/journal.pcbi.1006907

Nijman, S., Leeuwenberg, A., Beekers, I., Verkouter, I., Jacobs, J., Bots, M., Asselbergs, F., Moons, K., & Debray, T. (2022). Missing data is poorly handled and reported in prediction model studies using machine learning: A literature review. *Journal of Clinical Epidemiology*, *142*, 218–229. https://doi.org/10.1016/j.jclinepi.2021.11.023

Norris, A. D., Gao, S., Norris, M. L., Ray, D., Ramani, A. K., Fraser, A. G., Morris, Q., Hughes, T. R., Zhen, M., & Calarco, J. A. (2014). A Pair of RNA-Binding Proteins Controls Networks of Splicing Events Contributing to Specialization of Neural Cell Types. *Molecular Cell*, *54*(6), 946–959. https://doi.org/10.1016/j.molcel.2014.05.004

Ou, J., He, Y., Xiao, X., Yu, T.-M., Chen, C., Gao, Z., & Ho, M. S. (2014). Glial cells in neuronal development: Recent advances and insights from Drosophila melanogaster. *Neurosci. Bull.*, *30*(4), 584–594. https://doi.org/10.1007/s12264-014-1448-2

Pal, S., Mondal, S., Das, G., Khatua, S., & Ghosh, Z. (2020). Big data in biology: The hope and present-day challenges in it. *Gene Reports*, *21*, 100869. https://doi.org/10.1016/j.genrep.2020.100869

Parr, T. (n.d.). *Rfpimp*. https://pypi.org/project/rfpimp/

Pavlopoulos, G. A., Malliarakis, D., Papanikolaou, N., Theodosiou, T., Enright, A. J., & Iliopoulos, I. (2015). Visualizing genome and systems biology: Technologies, tools, implementation techniques and trends, past, present and future. *GigaSci*, *4*(1), 38. https://doi.org/10.1186/s13742-015-0077-2

Payne, J. L., Khalid, F., & Wagner, A. (2018). RNA-mediated gene regulation is less evolvable than transcriptional regulation. *Proc. Natl. Acad. Sci. U.S.A.*, *115*(15). https://doi.org/10.1073/pnas.1719138115

Paysan-Lafosse, T., Blum, M., Chuguransky, S., Grego, T., Pinto, B. L., Salazar, G. A., Bileschi, M. L., Bork, P., Bridge, A., Colwell, L., Gough, J., Haft, D. H., Letunić, I., Marchler-Bauer,

A., Mi, H., Natale, D. A., Orengo, C. A., Pandurangan, A. P., Rivoire, C., … Bateman, A. (2023). InterPro in 2022. *Nucleic Acids Research*, *51*(D1), D418–D427. https://doi.org/10.1093/nar/gkac993

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Petch, J., Di, S., & Nelson, W. (2022). Opening the Black Box: The Promise and Limitations of Explainable Machine Learning in Cardiology. *Canadian Journal of Cardiology*, *38*(2), 204–213. https://doi.org/10.1016/j.cjca.2021.09.004

Pilaz, L.-J., & Silver, D. L. (2015). Post-transcriptional regulation in corticogenesis: How RNA-binding proteins help build the brain: Post-transcriptional regulation in corticogenesis. *WIREs RNA*, *6*(5), 501–515. https://doi.org/10.1002/wrna.1289

Prasanth, K. V., Prasanth, S. G., Xuan, Z., Hearn, S., Freier, S. M., Bennett, C. F., Zhang, M. Q., & Spector, D. L. (2005). Regulating Gene Expression through RNA Nuclear Retention. *Cell*, *123*(2), 249–263. https://doi.org/10.1016/j.cell.2005.08.033

Pugliese, R., Regondi, S., & Marini, R. (2021). Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Science and Management*, *4*, 19–29. https://doi.org/10.1016/j.dsm.2021.12.002

R a language and environment for statistical computing: Reference index. (2010). [Computer software]. R Foundation for Statistical Computing.

Raj, A., Bogaard, P. van den, Rifkin, S. A., Oudenaarden, A. van, & Tyagi, S. (2008). Imaging individual mRNA molecules using multiple singly labeled probes. *Nat Methods*, *5*(10), 877–879. https://doi.org/10.1038/nmeth.1253

Rajgor, D., Welle, T. M., & Smith, K. R. (2021). The Coordination of Local Translation, Membranous Organelle Trafficking, and Synaptic Plasticity in Neurons. *Front. Cell Dev. Biol.*, *9*, 711446. https://doi.org/10.3389/fcell.2021.711446

Ray, D., Kazan, H., Cook, K. B., Weirauch, M. T., Najafabadi, H. S., Li, X., Gueroussov, S., Albu, M., Zheng, H., Yang, A., Na, H., Irimia, M., Matzat, L. H., Dale, R. K., Smith, S. A., Yarosh, C. A., Kelly, S. M., Nabet, B., Mecenas, D., … Hughes, T. R. (2013). A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, *499*(7457), 172–177. https://doi.org/10.1038/nature12311

Reiter, L. T., Potocki, L., Chien, S., Gribskov, M., & Bier, E. (2001). A Systematic Analysis of Human Disease-Associated Gene Sequences In *Drosophila melanogaster*. *Genome Res.*, *11*(6), 1114–1125. https://doi.org/10.1101/gr.169101

Richardson, S. (2022). Exposing the many biases in machine learning. *Business Information*

Review, *39*(3), 82–89. https://doi.org/10.1177/02663821221121024

Roberts, J. C. (2007). State of the Art: Coordinated & Multiple Views in Exploratory Visualization. *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, 61–71. https://doi.org/10.1109/CMV.2007.20

Rokach, L., & Maimon, O. (2005). Decision Trees. In O. Maimon & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 165–192). Springer-Verlag. https://doi.org/10.1007/0-387-25465-X_9

Rossi, A., Moro, A., Tebaldi, T., Cornella, N., Gasperini, L., Lunelli, L., Quattrone, A., Viero, G., & Macchi, P. (2017). Identification and dynamic changes of RNAs isolated from RALY-containing ribonucleoprotein complexes. *Nucleic Acids Research*, *45*(11), 6775–6792. https://doi.org/10.1093/nar/gkx235

Rossum, G. van, & Drake, F. L. (2010). *The Python language reference* (Release 3.0.1 [Repr.]). Python Software Foundation.

Rueter, S. M., Dawson, T. R., & Emeson, R. B. (1999). Regulation of alternative splicing by RNA editing. *Nature*, *399*(6731), 75–80. https://doi.org/10.1038/19992

Ruiter, J. de. (2016). *PyBiomart 0.2.0*. https://github.com/jrderuiter/pybiomart

Sabou, M., Bontcheva, K., Derczynski, L., & Scharl, A. (2014). Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. *Proceedings of the Ninth International Conference on Language Resources and Evaluation ({LREC}'14)*, 859–866. http://www.lrec-conf.org/proceedings/lrec2014/pdf/497_Paper.pdf

Sadeghyan, S. (2018). *A new robust feature selection method using variance-based sensitivity analysis*. https://doi.org/10.48550/ARXIV.1804.05092

Sailem, H. Z., Rittscher, J., & Pelkmans, L. (2020). <Span style="font-variant:small-caps;">KCML : A machine-learning framework for inference of multi-scale gene functions from genetic perturbation screens. *Molecular Systems Biology*, *16*(3). https://doi.org/10.15252/msb.20199083

Samuels, T. J., Arava, Y., Järvelin, A. I., Robertson, F., Lee, J. Y., Yang, L., Yang, C.-P., Lee, T., Ish-Horowicz, D., & Davis, I. (2020). Neuronal upregulation of Prospero protein is driven by alternative mRNA polyadenylation and Syncrip-mediated mRNA stabilisation. *Biology Open*, bio.049684. https://doi.org/10.1242/bio.049684

Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation Coefficients: Appropriate Use and Interpretation. *Anesthesia & Analgesia*, *126*(5), 1763–1768. https://doi.org/10.1213/ANE.0000000000002864

Sergeant, M. J., Hughes, J. R., Hentges, L., Lunter, G., Downes, D. J., & Taylor, S. (2021). Multi Locus View: An extensible web-based tool for the analysis of genomic data. *Commun Biol*, *4*(1), 623. https://doi.org/10.1038/s42003-021-02097-y

Sidiropoulos, K., Viteri, G., Sevilla, C., Jupe, S., Webber, M., Orlic-Milacic, M., Jassal, B., May, B., Shamovsky, V., Duenas, C., Rothfels, K., Matthews, L., Song, H., Stein, L., Haw, R., D'Eustachio, P., Ping, P., Hermjakob, H., & Fabregat, A. (2017). Reactome enhanced pathway visualization. *Bioinformatics*, *33*(21), 3461–3467. https://doi.org/10.1093/bioinformatics/btx441

Smedley, D., Haider, S., Ballester, B., Holland, R., London, D., Thorisson, G., & Kasprzyk, A. (2009). BioMart – biological queries made easy. *BMC Genomics*, *10*(1), 22. https://doi.org/10.1186/1471-2164-10-22

Smith, P., King, O. N. F., Pennington, A., Tun, W., Basham, M., Jones, M. L., Collinson, L. M., Darrow, M. C., & Spiers, H. (2023). Online citizen science with the Zooniverse for analysis of biological volumetric data. *Histochem Cell Biol*, *160*(3), 253–276. https://doi.org/10.1007/s00418-023-02204-6

Smith, R. N., Aleksic, J., Butano, D., Carr, A., Contrino, S., Hu, F., Lyne, M., Lyne, R., Kalderimis, A., Rutherford, K., Stepan, R., Sullivan, J., Wakeling, M., Watkins, X., & Micklem, G. (2012). InterMine: A flexible data warehouse system for the integration and analysis of heterogeneous biological data. *Bioinformatics*, *28*(23), 3163–3165. https://doi.org/10.1093/bioinformatics/bts577

Somarelli, J. A., Rupprecht, G., Altunel, E., Flamant, E. M., Rao, S., Sivaraj, D., Lazarides, A. L., Hoskinson, S. M., Sheth, M. U., Cheng, S., Kim, S. Y., Ware, K. E., Agarwal, A., Cullen, M. M., Selmic, L. E., Everitt, J. I., McCall, S. J., Eward, C., Eward, W. C., & Hsu, D. S. (2020). A Comparative Oncology Drug Discovery Pipeline to Identify and Validate New Treatments for Osteosarcoma. *Cancers*, *12*(11), 3335. https://doi.org/10.3390/cancers12113335

Staab, S., & Studer, R. (Eds.). (2004). *Handbook on ontologies*. Springer.

Ståhl, P. L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J. F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J. O., Huss, M., Mollbrink, A., Linnarsson, S., Codeluppi, S., Borg, Å., Pontén, F., Costea, P. I., Sahlén, P., Mulder, J., Bergmann, O., ... Frisén, J. (2016). Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, *353*(6294), 78–82. https://doi.org/10.1126/science.aaf2403

Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: A generalist algorithm for cellular segmentation. *Nat Methods*, *18*(1), 100–106. https://doi.org/10.1038/s41592-020-01018-x

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25. https://doi.org/10.1186/1471-2105-8-25

Svoboda, P., & Cara, A. Di. (2006). Hairpin RNA: A secondary structure of primary importance. *Cell. Mol. Life Sci.*, *63*(7-8), 901–908. https://doi.org/10.1007/s00018-005-5558-5

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B.

B., Siddiqui, A., Lao, K., & Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nat Methods*, *6*(5), 377–382. https://doi.org/10.1038/nmeth.1315

Taylor, S., & Noble, R. (2014). HTML5 PivotViewer: High-throughput visualization and querying of image data on the web. *Bioinformatics*, *30*(18), 2691–2692. https://doi.org/10.1093/bioinformatics/btu349

Tennessen, J. M., & Thummel, C. S. (2011). Coordinating Growth and Maturation — Insights from Drosophila. *Current Biology*, *21*(18), R750–R757. https://doi.org/10.1016/j.cub.2011.06.033

The Gene Ontology Consortium, Carbon, S., Douglass, E., Good, B. M., Unni, D. R., Harris, N. L., Mungall, C. J., Basu, S., Chisholm, R. L., Dodson, R. J., Hartline, E., Fey, P., Thomas, P. D., Albou, L.-P., Ebert, D., Kesling, M. J., Mi, H., Muruganujan, A., Huang, X., … Elser, J. (2021). The Gene Ontology resource: Enriching a GOld mine. *Nucleic Acids Research*, *49*(D1), D325–D334. https://doi.org/10.1093/nar/gkaa1113

Theil, K., Herzog, M., & Rajewsky, N. (2018). Post-transcriptional Regulation by 3' UTRs Can Be Masked by Regulatory Elements in 5' UTRs. *Cell Reports*, *22*(12), 3217–3226. https://doi.org/10.1016/j.celrep.2018.02.094

Thibault, S. T., Singer, M. A., Miyazaki, W. Y., Milash, B., Dompe, N. A., Singh, C. M., Buchholz, R., Demsky, M., Fawcett, R., Francis-Lang, H. L., Ryner, L., Cheung, L. M., Chong, A., Erickson, C., Fisher, W. W., Greer, K., Hartouni, S. R., Howie, E., Jakkula, L., … Margolis, J. (2004). A complementary transposon tool kit for Drosophila melanogaster using P and piggyBac. *Nat Genet*, *36*(3), 283–287. https://doi.org/10.1038/ng1314

Thomas, M. (2009). Manual annotation of the mouse genome: HAVANA and EUCOMM. *Nat Prec*. https://doi.org/10.1038/npre.2009.3181.1

Thompson, M. K., Ceccarelli, A., Ish-Horowicz, D., & Davis, I. (2023). Dynamically regulated transcription factors are encoded by highly unstable mRNAs in the *drosophila* larval brain. *RNA*, *29*(7), 1020–1032. https://doi.org/10.1261/rna.079552.122

Thomsen, R., & Lade Nielsen, A. (2011). A Boyden chamber-based method for characterization of astrocyte protrusion localized RNA and protein. *Glia*, *59*(11), 1782–1792. https://doi.org/10.1002/glia.21223

Thomsen, R., Pallesen, J., Daugaard, T. F., Børglum, A. D., & Nielsen, A. L. (2013). Genome wide assessment of mRNA in astrocyte protrusions by direct RNA sequencing reveals mRNA localization for the intermediate filament protein nestin. *Glia*, *61*(11), 1922–1937. https://doi.org/10.1002/glia.22569

Thulasiraman, K., & Swamy, M. N. S. (1992). Directed Graphs. In *Graphs: Theory and Algorithms* (pp. 97–125). John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781118033104.ch5

Tinsley, H. E. A., & Weiss, D. J. (2000). Interrater Reliability and Agreement. In *Handbook of*

Applied Multivariate Statistics and Mathematical Modeling (pp. 95–124). Elsevier. https://doi.org/10.1016/B978-012691360-6/50005-7

Titlow, J. S., Kiourlappou, M., Palanca, A., Lee, J. Y., Gala, D. S., Ennis, D., Yu, J. J. S., Young, F. L., Susano Pinto, D. M., Garforth, S., Francis, H. S., Strivens, F., Mulvey, H., Dallman-Porter, A., Thornton, S., Arman, D., Millard, M. J., Järvelin, A. I., Thompson, M. K., … Davis, I. (2023). Systematic analysis of YFP traps reveals common mRNA/protein discordance in neural tissues. *Journal of Cell Biology*, *222*(6), e202205129. https://doi.org/10.1083/jcb.202205129

Titlow, J. S., Yang, L., Parton, R. M., Palanca, A., & Davis, I. (2018). Super-Resolution Single Molecule FISH at the Drosophila Neuromuscular Junction. In I. Gaspar (Ed.), *RNA Detection* (Vol. 1649, pp. 163–175). Springer New York. https://doi.org/10.1007/978-1-4939-7213-5_10

Tsanas, A., Little, M. A., & McSharry, P. E. (2010). *A Simple Filter Benchmark for Feature Selection*.

Tsubouchi, A., Yano, T., Yokoyama, T. K., Murtin, C., Otsuna, H., & Ito, K. (2017). Topological and modality-specific representation of somatosensory information in the fly brain. *Science*, *358*(6363), 615–623. https://doi.org/10.1126/science.aan4428

Tushev, G., Glock, C., Heumüller, M., Biever, A., Jovanovic, M., & Schuman, E. M. (2018). Alternative 3' UTRs Modify the Localization, Regulatory Potential, Stability, and Plasticity of mRNAs in Neuronal Compartments. *Neuron*, *98*(3), 495–511.e6. https://doi.org/10.1016/j.neuron.2018.03.030

Ung, K., Huang, T.-W., Lozzi, B., Woo, J., Hanson, E., Pekarek, B., Tepe, B., Sardar, D., Cheng, Y.-T., Liu, G., Deneen, B., & Arenkiel, B. R. (2021). Olfactory bulb astrocytes mediate sensory circuit processing through Sox9 in the mouse brain. *Nat Commun*, *12*(1), 5230. https://doi.org/10.1038/s41467-021-25444-3

Wächter, K., Köhn, M., Stöhr, N., & Hüttelmaier, S. (2013). Subcellular localization and RNP formation of IGF2BPs (IGF2 mRNA-binding proteins) is modulated by distinct RNA-binding domains. *Biological Chemistry*, *394*(8), 1077–1090. https://doi.org/10.1515/hsz-2013-0111

Wan, Y., Kertesz, M., Spitale, R. C., Segal, E., & Chang, H. Y. (2011). Understanding the transcriptome through RNA structure. *Nat Rev Genet*, *12*(9), 641–655. https://doi.org/10.1038/nrg3049

Wang, G., Oh, D.-H., & Dassanayake, M. (2020). GOMCL: A toolkit to cluster, evaluate, and extract non-redundant associations of Gene Ontology-based functions. *BMC Bioinformatics*, *21*(1), 139. https://doi.org/10.1186/s12859-020-3447-4

Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. *Proceedings of the NAACL Student Research Workshop*,

88–93. https://doi.org/10.18653/v1/N16-2013

Webb, G. I. (2016). Naïve Bayes. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning and Data Mining* (pp. 1–2). Springer US. https://doi.org/10.1007/978-1-4899-7502-7_581-1

Wen, H., Ding, G., Liu, C., & Wang, J. (2014). Matrix Factorization Meets Cosine Similarity: Addressing Sparsity Problem in Collaborative Filtering Recommender System. In L. Chen, Y. Jia, T. Sellis, & G. Liu (Eds.), *Web Technologies and Applications* (Vol. 8709, pp. 306–317). Springer International Publishing. https://doi.org/10.1007/978-3-319-11116-2_27

Williams, C. G., Lee, H. J., Asatsuma, T., Vento-Tormo, R., & Haque, A. (2022). An introduction to spatial transcriptomics for biomedical research. *Genome Med*, *14*(1), 68. https://doi.org/10.1186/s13073-022-01075-1

Williams, E., Moore, J., Li, S. W., Rustici, G., Tarkowska, A., Chessel, A., Leo, S., Antal, B., Ferguson, R. K., Sarkans, U., Brazma, A., Carazo Salas, R. E., & Swedlow, J. R. (2017). Image Data Resource: A bioimage data integration and publication platform. *Nat Methods*, *14*(8), 775–781. https://doi.org/10.1038/nmeth.4326

Wu, T., Hu, E., Xu, S., Chen, M., Guo, P., Dai, Z., Feng, T., Zhou, L., Tang, W., Zhan, L., Fu, X., Liu, S., Bo, X., & Yu, G. (2021). clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *The Innovation*, *2*(3), 100141. https://doi.org/10.1016/j.xinn.2021.100141

Xu, C., & Jackson, S. A. (2019). Machine learning and complex biological data. *Genome Biol*, *20*(1), 76, s13059-019-1689-0. https://doi.org/10.1186/s13059-019-1689-0

Yadav, S., Younger, S. H., Zhang, L., Thompson-Peer, K. L., Li, T., Jan, L. Y., & Jan, Y. N. (2019). Glial ensheathment of the somatodendritic compartment regulates sensory neuron structure and activity. *Proc. Natl. Acad. Sci. U.S.A.*, *116*(11), 5126–5134. https://doi.org/10.1073/pnas.1814456116

Yan, Z., Lécuyer, E., & Blanchette, M. (2019). Prediction of mRNA subcellular localization using deep recurrent neural networks. *Bioinformatics*, *35*(14), i333–i342. https://doi.org/10.1093/bioinformatics/btz337

Yang, Y., Xu, Z., & Song, D. (2016). Missing value imputation for microRNA expression data by using a GO-based similarity measure. *BMC Bioinformatics*, *17*(S1), S10. https://doi.org/10.1186/s12859-015-0853-0

Ye, P., Peyser, B. D., Pan, X., Boeke, J. D., Spencer, F. A., & Bader, J. S. (2005). Gene function prediction from congruent synthetic lethal interactions in yeast. *Mol Syst Biol*, *1*(1). https://doi.org/10.1038/msb4100034

Yeo, G., Holste, D., Kreiman, G., & Burge, C. B. (2004). Variation in alternative splicing across human tissues. *Genome Biol*, *5*(10), R74. https://doi.org/10.1186/gb-2004-5-10-r74

Yildirim, K., Petri, J., Kottmeier, R., & Klämbt, C. (2019). Drosophila glia: Few cell types and

many conserved functions. *Glia*, *67*(1), 5–26. https://doi.org/10.1002/glia.23459

Yu, G., Wang, L.-G., Han, Y., & He, Q.-Y. (2012). clusterProfiler: An R Package for Comparing Biological Themes Among Gene Clusters. *OMICS: A Journal of Integrative Biology*, *16*(5), 284–287. https://doi.org/10.1089/omi.2011.0118

Yuan, G.-H., Wang, Y., Wang, G.-Z., & Yang, L. (2023). RNAlight: A machine learning model to identify nucleotide features determining RNA subcellular localization. *Briefings in Bioinformatics*, *24*(1), bbac509. https://doi.org/10.1093/bib/bbac509

Zappulo, A., Bruck, D. van den, Ciolli Mattioli, C., Franke, V., Imami, K., McShane, E., Moreno-Estelles, M., Calviello, L., Filipchyk, A., Peguero-Sanchez, E., Müller, T., Woehler, A., Birchmeier, C., Merino, E., Rajewsky, N., Ohler, U., Mazzoni, E. O., Selbach, M., Akalin, A., & Chekulaeva, M. (2017). RNA localization is a key determinant of neurite-enriched proteome. *Nat Commun*, *8*(1), 583. https://doi.org/10.1038/s41467-017-00690-6

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *JASTT*, *1*(2), 56–70. https://doi.org/10.38094/jastt1224

Zhang, Z.-Y., Yang, Y.-H., Ding, H., Wang, D., Chen, W., & Lin, H. (2021). Design powerful predictor for mRNA subcellular location prediction in *Homo sapiens*. *Briefings in Bioinformatics*, *22*(1), 526–535. https://doi.org/10.1093/bib/bbz177

Zhao, C., & Wang, Z. (2018). GOGO: An improved algorithm to measure the semantic similarity between gene ontology terms. *Sci Rep*, *8*(1), 15107. https://doi.org/10.1038/s41598-018-33219-y

Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: Principles and techniques for data scientists* (First edition). O'Reilly.

Zuchero, J. B., & Barres, B. A. (2015). Glia in mammalian development and disease. *Development*, *142*(22), 3805–3809. https://doi.org/10.1242/dev.129304