Cell identity prediction from single-cell RNA-sequencing data

Marina Potsi

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2020

Abstract

Single-cell RNA-sequencing (scRNA-seq) has made it possible to identify rare cell subpopulations in tissues by revealing the heterogeneity in gene expression between individual cells. In this dissertation, we analyse single-cell datasets from the developing mouse brain cortex for two continuous embryonic days, E13 and E14. Our aim is to identify novel subpopulations of cells that change their identity and function after a mutation of the Pax6 gene. We refer to these cells as *ectopic* and utilise machine learning techniques to identify them. Firstly, we explore methods for clustering ectopic cells that result in better separation of cell types in E14 than that obtained by the baseline single-cell analysis pipeline. We demonstrate that autoencoder neural network models that simultaneously perform dimensionality reduction and clustering achieve this goal. Secondly, we infer ectopic expression in E13 by utilising prior knowledge from E14. We identify differentially expressed genes that highly correlate with known ectopic genes and create a pipeline to automatically assign cells to specific cell types. We show that a Random Forest classifier trained on E14 can predict ectopic cells on E13, but the evaluation and interpretation of results is challenging. Finally, we discuss limitations of the current approaches and propose promising future directions for more accurate identification of ectopic cells.

Acknowledgements

I would like to thank my supervisors, Dr. Ian Simpson, for his valuable advice and for offering me the opportunity to work on such an interesting project, and Zrinko Kozic for his extensive feedback, support and patient explanation of the biological side of the project, as well as his suggestions and time spent on the project. Many additional thanks go to Prof. David Price and Kai Boon Tan for their constant feedback and help with interpreting the results, as well as Andreas C. Kapourani for inspiring me to dive into single-cell RNA-seq analysis and discussing directions.

I am grateful for the wonderful people I have met this year at the 7th floor of Appleton Tower and for the long walks and interesting ideas we have shared.

Finally, I would like to thank my parents and my sister Evi for their unconditional love and support, my friends in Athens and Edinburgh for making this year a lot more pleasant and Andreas Grivas for always being there, brightening my days and supporting me patiently throughout this intense year.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Marina Potsi*)

Table of Contents

Abbreviations

E13 Embryonic day 13

E14 Embryonic day 14

CDBS Center for Discovery Brain Sciences

DE Differential Expression

PCA Principal Component Analysis

PCR Polymerase Chain Reaction

scRNA-seq Single-cell RNA-sequencing

SVM Support Vector Machines

UMAP Uniform Manifold Approximation and Projection

UMI Unique Molecular Identifier

Chapter 1

Introduction

1.1 Single-cell RNA-sequencing analysis

Single-cell RNA-sequencing (scRNA-seq) is a recent sequencing technology used to extract the expression values of each gene across a population that consists of thousands or millions of cells. As opposed to bulk RNA-seq, which provides an average expression profile of all cells in the population, single-cell RNA-seq examines and captures the heterogeneity that is present in gene expression values within single cells, even for cells of the same cell type [Abdelaal et al., 2019]. It has revolutionised the understanding of the expression profiles of single cells at an exceptional resolution and provides better insights for the function of a single cell [Eberwine et al., 2014].

Since scRNA-seq emerged, it has been used to address multiple biological questions and various applications have become possible. Examples include inferring the developmental trajectory of cells [Saelens et al., 2019], studying intratumour heterogeneity in cancers [Lawson et al., 2018] and inference of gene regulatory networks [Pratapa et al., 2020] among others. One of the most significant applications of scRNA-seq is the identification of cell types that are present in complex tissues such as the brain [Regev et al., 2017], resulting in detection of rare or novel subpopulations that would otherwise remain obscure.

Given the vast increase in the size and availability of single-cell datasets, machine learning techniques become vital for identifying patterns automatically and extracting insights. Most single-cell datasets do not have any ground truth label annotations for the cells, with the exception of a few well-known curated datasets that are used in the literature. Depending on the presence or absence of cell type labels, machine learning is used to identify cell types in an unsupervised or a supervised manner. In cases where

cell type labels are present, either from manual or automatic annotation, cell type labels can be predicted with classifiers. In an unsupervised setting, cells can be clustered in a low dimensional space and cell types can be identified based on high expression of marker genes in each cluster, as in [Ntranos et al., 2016] and [Lin et al., 2017].

This dissertation primarily investigates an unsupervised approach for cell type identification from scRNA-seq data, due to the absence of true cell type labels in the given datasets. After creating a cell type annotation pipeline, a supervised approach is also examined based on prior biological knowledge for the available cell types. The single-cell datasets we are using were provided by the collaborating Center for Discovery Brain Sciences (CDBS) at the University of Edinburgh. These cells originate from the developing mouse brain cortex for two subsequent embryonic days (days 13 and 14, referred to as E13 and E14 respectively). The Pax6 gene, a major transcription factor in the brain, regulates the expression of other genes during brain development and determines the identity of cells in the developing cortex. Researchers at CDBS, after inactivating the function of Pax6 which causes a mutation in the embryonic brain, noticed the appearance of new cell subpopulations with abnormal gene expression in E14, referred to as *ectopic*.

1.2 Hypothesis and objectives

This research builds upon the hypothesis that there is an ectopic subpopulation of cells in E13, but its expression signal is not as strong as in E14. We investigate machine learning techniques to use the prior knowledge obtained by CDBS about ectopic cells in E14 in order to infer which cells are ectopic in E13. While there are multiple directions to approach this problem, this dissertation focuses on answering the following objectives:

1. Which genes are more likely to have ectopic expression? How can we best assign cell types to cells of E14?

2. Following the standard clustering approach of single-cell RNA-seq analysis, can we use deep neural network methods to obtain a better separation of cells in E14?

3. What insights can we derive from E14 to help us predict ectopic cells in E13?

1.3 Dissertation structure

The dissertation is structured in three main chapters. Chapter 2 gives an overview of single-cell RNA-seq analysis, its challenges regarding cell type identification and describes the methods that will be used throughout the dissertation. Chapter 3 provides a description of the mouse brain datasets that were used and the single-cell RNA-seq analysis pipeline in order to retain high quality cells and discard as much biological and technical noise as possible. The preprocessed datasets are then used in Chapter 4, which covers the sequence of experiments performed with the goal of identifying ectopic cell subpopulations.

Finally, Chapter 5 discusses the conclusions from the experiments and the results obtained from Chapter 4, as well as limitations encountered and potential directions for future work. Additional material can be found in the Appendix, including analysis plots and tables containing all the genes related to each cell type found from the experiments.

Chapter 2

Background

2.1 Single-cell RNA-sequencing analysis

scRNA-seq is a powerful technology that enables the measurement of the expression values for each gene across a population of thousands or millions of cells. It produces an estimate of abundance of mRNA molecules expressed in each cell and provides information about the function of a cell, and subsequently cell types and cell states, at a higher resolution. Given the highly heterogeneous nature of gene expression, scRNA-seq makes it possible to identify rare subpopulations of cells, which would otherwise remain unobserved [Buettner et al., 2015a].

The first mRNA sequencing at the single-cell level was introduced by Tang et al. [2009], but it has recently received more attention due to a sudden increase in the size and availability of single-cell datasets. This resulted in the development of better pipelines for single-cell analysis and more advanced computational methods. Prior to scRNA-seq, cell populations were sequenced in bulk, a process known as bulk RNA-seq, which would only give an average estimate of the expression profiles of cells, treating the cell population as homogeneous and subsequently masking the cellular heterogeneity. On the other hand, scRNA-seq can reveal the heterogeneity of a sample tissue by identifying distinct and rare subpopulations of cells, allowing to study gene expression variability between these subpopulations.

Following is a brief description of the single-cell data generation process from biological samples. Initially, the sample tissue from the organism of interest is digested with the usage of specific enzymes (*single-cell dissociation*) and a single-cell suspension is formed in order to capture individual cells. Single cells are then isolated (*single-cell isolation*) and specific chemicals cut the cell membranes to capture the

mRNA in each cell, reverse-transcribe to cDNA and amplify it (*library construction*). Usually, captured mRNA molecules are labeled with a Unique Molecular Identifier (*UMI*) [Liu and Trapnell, 2016], which helps to reduce the bias effects from the PCR[1] amplification process. This way, the mRNA molecules that have the same UMI are supposed to be extracted from the same input molecule. A more detailed description of the experimental design process can be found in Ziegenhain et al. [2017] and Mereu et al. [2020].

scRNA-seq analysis has seen an exponential growth in the availability of tools during the recent years. More and more computational methods and frameworks are constantly being developed and added to current pipelines, allowing for tailored methods for various single-cell datasets. Single-cell RNA-seq analysis is mainly supported by packages in R and Python, the most popular ones being the following: *Seurat* [Stuart et al., 2019], *Monocle* [Satija et al., 2015], *Scater* [McCarthy et al., 2017], *Scran* [Lun et al., 2016] (in R) and *Scanpy* [Wolf et al., 2018], [Buettner et al., 2015b] (in Python). These frameworks cover wide parts of the analysis pipeline, are well-documented and provide detailed tutorials with well-annotated datasets.

2.1.1 Challenges in single-cell RNA-sequencing

Despite the rapid progress in the single-cell genomics field, there are still fundamental biological and computational challenges that need to be addressed. Most of them result from the high-dimensional and sparse nature of single-cell data, stored as gene expression matrices. Recently, the number of cells assayed by the sequencing technologies has increased from thousands to millions, resulting in high-dimensional single-cell datasets that require more computational power for processing and analysis, hence more efficient computational methods are constantly in demand [Lähnemann et al., 2020].

To describe one of the major challenges in single-cell analysis, the high degree of *sparsity* in single-cell datasets, we briefly explain the morphology of single-cell gene expression matrices. A gene expression matrix X contains G genes and N cells, where each value X_{ij} indicates the expression value of a gene j in cell i. The expression matrix is also known as the count matrix, as each expression value represents the count of captured, reverse-transcribed, amplified and sequenced mRNA molecule. Count matrices suffer from a high number of *dropouts*, or high sparsity, where the read

[1] Polymerase chain reaction (PCR) is a method used to make millions or billions of copies from a specific DNA sample so it can be studied in more detail.

molecule counts are not captured and therefore many transcripts display zero counts in every cell. Dropouts can happen due to a low sequencing depth (that is, the number of unique transcripts detected in each cell), meaning the low-expressed transcripts might not be recorded even if they are present.

Besides the challenges that emerge from the biological nature of single-cell datasets, technical issues might also obscure gene expression patterns that we are interested in identifying. Different lab experiments and conditions in which single-cell datasets are created or different sequencing machines can have a significant impact on the reproducibility of the results. These differences, also known as batch effects, need to be resolved in order to gain confidence in the resulting statistical conclusions [Kiselev et al., 2019].

The aforementioned challenges can exacerbate the biological interpretation of the results, which is a difficult task on its own even without the presence of these issues. High unwanted cell-to-cell variability driven by biological and technical factors might limit the interpretability of generated clusters of cells and can hinder important gene expression signals that would otherwise reveal rare subpopulations of cells [Hicks et al., 2018, Vallejos et al., 2017].

One of the biggest challenges in scRNA-seq is the inability of interpreting results and defining whether a model performs well. The lack of ground truth labels regarding cell identities in scRNA-seq data makes it hard to determine whether computational methods perform well on a specific dataset.

Finally, even though some methods from bulk RNA-seq can successfully be applied to scRNA-seq, most of them have to be adapted to the specific properties of single-cell data. Considering that scRNA-seq is an emerging field, most computational methods were established after 2014 and many more are constantly under development, putting a burden in the effort of standardisation and usability of the methods [Luecken and Theis, 2019].

2.2 Differential Expression Analysis

Differential expression (DE) analysis is a series of statistical tests performed between two different cell populations. DE aims to find a subset of genes that are highly expressed between groups of interest, signifying whether the difference in gene expression profiles between the populations is statistically significant. Differentially expressed genes found by the analysis that are specific in each cluster are referred to

as *marker genes*, allowing us to characterise a cluster with a cell type identity and determine the factors of heterogeneity between clusters.

DE analysis algorithms identify differences in expression profiles between clusters and create a ranked list of the top n highly expressed genes for each cluster. Different methods can be used for ranking, such as pairwise t-test, Wilcoxon rank-sum test and logistic regression. For our analysis in Sections 4.1.2 and 4.3.2 we use the Wilcoxon rank-sum test [Hollander et al., 2013], which tests whether two samples come from the same distribution.

2.3 Dimensionality reduction

Single-cell RNA-seq datasets provide information about the expression profile of single cells across multiple genes. Each gene represents a dimension of the data, however not all genes based on their expression profiles are important for further analyses, due to sensitivity of the clustering algorithms on high dimensional data. Moreover, high dimensionality sets boundaries to many other statistical analyses due to computational limitations of the mathematical operations on large gene expression matrices.

Dimensionality reduction is a common technique employed in single-cell analysis, aiming to reduce the numbers of dimensions in the data while retaining the underlying biological structure of the data as much as possible. Inherently, the expression profiles of single cells are low-dimensional, since there are many correlations between genes, so they can be described by fewer dimensions. Another objective for reducing the high dimensionality of single-cell datasets is for visualisation purposes. By reducing the dimensions to only two or three, we can obtain a visual representation of the cellular space and its underlying topology.

Dimensions of single-cell datasets can be reduced by either linear or non-linear projections of gene expression vectors. Depending on the approach, we can obtain a different representation of the reduced dimensionality of the data. Non-linear methods are more flexible at capturing the structure of the data in a smaller number of dimensions compared to linear methods. However, the latter are more widely used in scRNA-seq analysis, occasionally as a preceding step prior to non-linear methods.

In the following subsections we describe three dimensionality reduction techniques based on different transformations: i) PCA, one of the most prevalent linear dimensionality reduction methods, ii) UMAP [McInnes et al., 2018], a non-linear method that has recently gained popularity in single-cell analysis, and iii) autoencoders, arti-

ficial neural networks which can be linear or non-linear depending on the activation function.

2.3.1 PCA

Principal Component Analysis (PCA) [Hotelling, 1933] is a linear dimensionality reduction technique that projects the points of a high dimensional space into a lower dimensional space, by finding directions in the data space that maximise the variance of each dimension, while retaining as much information as possible. The data is transformed into a new coordinate system, where the first coordinate (first principal component) retains the greatest variance of the data, the second coordinate retains the second greatest variance and so on. The principal components are ranked by variance and the top k are selected as a lower-dimensionality projection of the initial dimensions.

2.3.2 UMAP

UMAP (Uniform Manifold Approximation and Projection) is a non-linear, nearest neighbour based graph method for dimensionality reduction that was recently published by McInnes et al. [2018].

Given the original input data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, UMAP constructs a k-nearest-neighbours graph with weighted edges. For each datapoint x_i, its k nearest neighbours are calculated using a dissimilarity metric d resulting in a set of points $\{x_{i1}, \ldots, x_{ik}\}$. Then for each vertex, after the graph is constructed, weights are added to each edge:

$$w_i^{(\mathbf{X})}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\max(0, d(\mathbf{x}_i, \mathbf{x}_j)) - \rho_i}{\sigma_i}\right), \tag{2.1}$$

where ρ is the distance to the nearest neighbour and σ is the diameter of the neighbourhood. Similarly, the graph and weights matrix for a lower dimensional representation $\mathbf{Y}$ is also constructed and the distance between the two weight matrices is computed using cross-entropy. Finally, the lower dimensional representation of the data $\mathbf{X}$ is given by minimising this distance using gradient descent.

UMAP and t-SNE (t-distributed Stochastic Neighbor Embedding), also a non-linear manifold-learning method published by Maaten and Hinton [2008], have been used extensively for dimensionality reduction and visualisation of scRNA-seq data. While t-SNE focuses more on the local structure of the data, resulting in being over-confident about cell similarities and differences between cell populations, UMAP pre-

serves a more global structure and can efficiently be computed on large datasets [Becht et al., 2018].

2.3.3 Autoencoder

An autoencoder is a neural network that learns a lower-dimensional representation (embedding) of an input vector $\mathbf{x}$ in an unsupervised manner, usually for dimensionality reduction purposes [Goodfellow et al., 2016]. It is defined mathematically as a non-linear mapping $\mathbf{h}$ of the original input $\mathbf{x}$ to a lower-dimensional feature representation, such that $\mathbf{h} = f(\mathbf{x})$.

Autoencoders consist of two main parts: an *encoder*, that compresses the original input to a latent space of lower dimensions, and a *decoder* that reconstructs $\mathbf{x}$ from the lower-dimensional embedding, so that the reconstructed output $\hat{\mathbf{x}}$ is as similar as possible to $\mathbf{x}$. The network is trained to minimise the *reconstruction error* $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$, in most cases being the squared loss, in order to reconstruct the original input and at the same time to avoid replicating it.

2.4 Unsupervised representation learning

Unsupervised representation learning refers to learning low-dimensional feature representations from unlabelled high-dimensional data. Here we refer to models that perform unsupervised representation learning and clustering on the reduced latent space at the same time.

DEC Xie et al. [2016] developed *Deep Embedded Clustering* (DEC) to learn a lower-dimensional representation of the initial features with deep learning and at the same time to perform clustering on the lower-dimensional embedding. Let X be the initial high-dimensional space and Z the lower-dimensional latent feature space resulting from X through a non-linear mapping $f_\theta : X \to Z$, parametrizing the learnable parameters θ with deep neural networks. DEC initialises the parameters θ and the k initial cluster centroids $\{\mu_j\}_{j=1}^{k}$ with a deep autoencoder and iteratively clusters the data points of the latent feature space Z.

Tian et al. [2019] extended DEC to specifically model single-cell RNA-seq data by utilising the Zero Inflated Negative Binomial (ZINB) distribution, in a model referred to as *scDeepCluster*, which we experiment with in Section 4.2.

scDeepCluster scDeepCluster simultaneously projects the high-dimensional feature (gene) space into a lower dimensional latent space while optimising clustering. It leverages the count nature of the gene expression matrix with a negative binomial distribution, which is widely used for single-cell data since it models discrete overdispersed data (mRNA molecule counts). Therefore, it models directly the count data through a zero-inflated negative binomial (ZINB) loss function, which replaces the typical mean square error (MSE) in autoencoders.

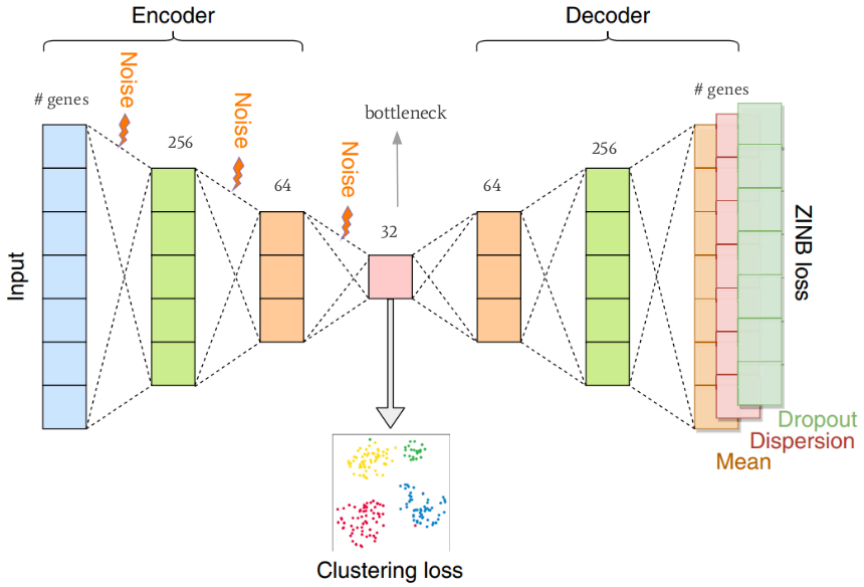


Figure 2.1: The architecture of scDeepCluster [Tian et al., 2019], which comprises of the encoder and decoder neural networks. # genes is the initial dimensionality of the single-cell datasets and the number on top of each layer indicates the number of units in each layer. The bottleneck size is 32.

Let X be the initial gene expression matrix of raw counts. A denoising technique is introduced to slightly corrupt the input with random Gaussian noise ε, such that $X_{\text{corrupt}} = X + \varepsilon$, in order to prevent the autoencoder from memorising X and also to enable generalisability. The autoencoder has integrated noise in every layer, as can be seen from Figure 2.1, and is trained to minimise the loss function $L(X, g_{W'}(f_W(X_{\text{corrupt}})))$, where $f_W(X_{\text{corrupt}})$ is the encoder function and $g_{W'}(f_W(X_{\text{corrupt}}))$ the decoder function with learned weights W and W' respectively.

The lower-dimensional representation of the genes is learned with a ZINB model-based autoencoder, its loss being the ZINB distribution likelihood, which is defined as:

$$\begin{aligned} \text{ZINB}\left(X \mid \pi, \mu, \theta\right) &= \pi\delta_0\left(X\right) + \left(1-\pi\right)\text{NB}\left(X \mid \mu, \theta\right), \text{where} \\ \text{NB}\left(X \mid \mu, \theta\right) &= \frac{\Gamma(X+\theta)}{X!\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{X} \end{aligned} \tag{2.2}$$

and μ, θ and π are the parameters of the ZINB model-based autoencoder representing the mean and dispersion of NB distribution and the probability of dropouts. The fully connected layers Mean, Dispersion and Dropout are appended to the last layer of the decoder, which represent estimations of the parameters μ, θ and π. Therefore, the loss function of the autoencoder is the negative log likelihood of the ZINB model: $L_{\text{ZINB}} = -\log\left(\text{ZINB}\left(X \mid \pi, \mu, \theta\right)\right)$

After learning a lower-dimensional representation of the genes with the ZINB model, cells are clustered in this latent space using the *Kullback-Leibler* (KL) divergence clustering loss as in *DEC*, which measures how two distributions P and Q differ. The clustering loss is then defined as:

$$L_{\text{c}} = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.3}$$

where q_i defines the similarity between the embedded point and the cluster center and p_i is the target distribution. Finally, the loss function of the ZINB model-based autoencoder comprises of the ZINB loss and the clustering loss with a positive coefficient γ accounting for the relative weights of both losses:

$$L = L_{ZINB} + \gamma L_c \tag{2.4}$$

2.5 Clustering

Clustering is an unsupervised machine learning technique that is extensively used in single-cell analysis after preprocessing the datasets, in order to group cells based on the similarity of their gene expression profiles which can be computed with distance metrics, including Euclidean, Hamming or Mahalanobis distance, cosine similarity or correlation metrics.

Clustering is utilised to extract biological insights about the cell types that represent each group of cells, hence inferring the identity of each single cell. However, due to the high dimensionality of the data, that is, the large number of genes in a gene expression matrix, distance metrics between cells do not perform well since cells have

shorter distances between them, resulting in a poor identification of distinct clusters. For this reason, clustering is usually performed on a reduced set of features (genes) using feature selection and dimensionality reduction techniques to help towards noise reduction.

Kiselev et al. [2019] provide an extensive overview of the most prominent clustering methods for single-cell RNA-seq analysis, along with the challenges and limitations involved that make the biological characterisation of the identified clusters a difficult problem. Here we describe the two main clustering algorithms, K-Means [Lloyd, 1982] and Leiden [Traag et al., 2019].

K-Means K-Means is initialised by manually selecting k centroids for k clusters, where each cell is assigned to its nearest centroid. Then new centroids are assigned by calculating the mean of all cells that were assigned to previous centroids. K-Means computes the difference of the old and new centroids until it becomes too small.

Leiden Leiden is a community detection algorithm that has been proposed for single-cell clustering by Levine et al. [2015]. It initially starts from a partition of single cell communities and then moves the cell nodes from one community to the other by refining the partitions. A network is then created from the refined partition and the non-refined partition is used as an initial partition for the network. Leiden moves the nodes in the network and stops when there are no improvements in the clusters.

2.6 Evaluation metrics

This section describes the evaluation metrics used to assess the obtained clusters of the bottleneck layer of the autoencoder in Section 4.2: *Purity*, *Normalized Mutual Information* (NMI) [Strehl and Ghosh, 2002] and *Adjusted Rand Index* (ARI) [Hubert and Arabie, 1985]. These metrics are calculated between the true labels (cell type annotations from Section 4.1.3) and the obtained clusters. Moreover, the evaluation of the classifiers' performance in Section 4.3.1 is performed by calculating the *Accuracy* and *Macro F1-Score* on the entire single-cell dataset using 3-fold cross-validation.

Purity Purity assigns the most frequent class to each cluster and evaluates the assignment by counting the points that are assigned correctly. Let $A = \{a_1, ...a_k\}$ the set containing the clusters and $B = b_1, ...b_i$ the set of different classes. Purity is then defined as:

$$\text{Purity}(A,B) = \frac{1}{N}\sum_{k}\max_{j}\left|a_k \cap b_j\right| \tag{2.5}$$

It takes values between 0 and 1, with 1 representing perfect clustering and values close to 0 poor clustering purity.

Normalized Mutual Information (NMI) Given the obtained clustering labels and the true labels, as mentioned above, the Mutual Information (MI) score measures the degree of similarity between two labels. NMI is a normalised measure of MI that scales the values between 0 and 1, where 0 represents no similarity and 1 represents perfect similarity.

Adjusted Rand Index (ARI) Let A be the set of true class labels and B the set of the cluster labels. The Random Index (RI) metric sums the number of pairs of points that are in the same set in A and B and in different sets in A and B, divided by the number of all pairs. However, RI does not always have a value close to zero for any assignments that are made randomly. ARI corrects for this by the following formula:

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}, \tag{2.6}$$

where $E[RI]$ is the expected value of RI.

ARI is an adjusted measure of RI, which calculates the pairs of points between two clusters A and B that belong to the same set in A and in the same set in B, and it can take negative values.

Macro F1-Score Given a binary classification with two classes *Positive* and *Negative*, we define:
TP (*True Positives*): # of *Positive* examples correctly classified as *Positive*
FP (*False Positives*): # of *Negative* examples incorrectly classified as *Positive*
TN (*True Negatives*): # of *Negative* examples correctly classified as *Negative*
FN (*False Negatives*): # of *Positive* examples incorrectly classified as *Negative*

Then Precision $= \frac{TP}{TP+FP}$, Recall $= \frac{TP}{TP+FN}$ and $F_1 = 2\,\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$. The F1-scores are computed for each class and averaged with the arithmetic mean.

Accuracy Accuracy computes the fraction of the predicted labels that are correct and is defined as:

$$Accuracy = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{2.7}$$

Chapter 3

Datasets and Data Preprocessing

This chapter describes the single-cell datasets from the developing mouse brain cortex originated from lab experiments at CBDS, and analyses the pipeline developed for preprocessing of the data. The pipeline consists of the following steps: quality control, normalisation, feature selection, dimensionality reduction and removal of unwanted biological variation, such as the cell cycle phase.

3.1 Single-cell datasets from the mouse brain cortex

As mentioned in Section 1.1, Pax6 plays a crucial role in the development of tissues in the early development of the brain and acts as a transcription factor by controlling the expression of other genes. In order to study the regulation and function of Pax6 in the development of the brain cortex, researchers used Pax6loxP [Simpson et al., 2009] to conditionally inactivate Pax6 from the cerebral cortex of a group of mouse embryos on embryonic day 9 (E9). On embryonic days 13 (E13) and 14 (E14), the effects of Pax6 inactivation start to become apparent. The Pax6 genotype of such embryos is *homozygous mutant*, since both alleles of Pax6 are absent and the function of the gene is inactive, and mouse embryos with heterozygous Pax6 deletion are used as a *control* group.

The single-cell libraries were constructed using the 10X Genomics™ Chromium Controller. The expression values of each gene in a cell (raw counts of RNA molecules) were stored in csv files, resulting in the following 4 datasets: *E13_mutant, E13_control, E14_mutant, E14_control*. The dimensions of these datasets can be seen in Table 3.1.

Dataset - Mouse Model	Name	Cells (*Samples*)	Genes (*Features*)
E13 Homozygous Mutant	E13_mutant	6,333	30,213
E14 Homozygous Mutant	E14_mutant	4,446	30,213
E13 Heterozygous Control	E13_control	3,797	30,213
E14 Heterozygous Control	E14_control	4,380	30,213

Table 3.1: The four datasets of control and mutant mouse models from two different embryonic days (E13 and E14) with their corresponding gene expression matrix sizes.

3.2 Single-cell RNA-sequencing analysis pipeline

Analysis of single-cell RNA-seq datasets can be performed with any of the packages that were mentioned in Section 2.1, although a careful selection should be made depending on the datasets and the problem being tackled. In this project, we use Python's Scanpy framework to build a pipeline for analysis and preprocessing of the datasets and follow the best practices techniques from [Luecken and Theis, 2019] and [Amezquita et al., 2019]. It should be noted that we initially performed the scRNA-seq analysis in Seurat, a framework written in R, and there were not any major differences in the functionality or runtime of the methods compared to Scanpy. However, in contrast to R-based frameworks, Scanpy's implementation in Python enables an easier integration with Machine Learning methods and frameworks, such as Tensorflow [Abadi et al., 2016].

Single-cell RNA-seq datasets are matrices of cells and genes that contain the expression value of each gene in each cell. We refer to these datasets as *count matrices* because the values represent the counts of mRNA molecules captured for each gene. In the R-based pipelines, the dimension of the dataset is defined by (number of genes $\times$ number of cells), however we will follow the notation (number of cells $\times$ number of genes), which is used in Python and generally in Machine Learning.

In the subsections below, we outline the preprocessing steps that were followed in order to reduce any noise resulting from unwanted biological or technical factors, namely quality control, normalisation, feature selection and elimination of unwanted variation.

3.2.1 Quality Control

The assignment of UMIs to the mRNA molecules helps the detection of contaminated cells as well as the elimination of the PCR amplification bias, as described in section 2.1, however noise is still present in the datasets. Quality control is used to remove cells of low quality, which might have either been damaged during the experimental design process or have failed to be completely captured by the sequencing workflow. More specifically, there are three main aspects that quality control is focusing on in single-cell datasets: the total number of genes that are expressed, the total counts per cell and the proportion of mitochondrial gene expression [Griffiths et al., 2018].

Cells are considered to have low quality if the library size[1] is small, or else if the total count of molecules across all features for a cell is small, as well as if they contain only a few expressed genes. A possible explanation for this is that during library preparation the RNA that is expressed might have been lost, therefore it is not present in the final counts matrix. Similarly, a high percentage of mitochondrial genes in a cell is an indication of a low-quality, damaged cell, where mitochondrial mRNA might have leaked out into the cytoplasm through the mitochondrial membrane.

Usually, low quality cells should be removed from the dataset at the beginning of the preprocessing pipeline so that the downstream analysis is not affected by them. However, the three quality control aspects that were mentioned above should be considered simultaneously before considering a threshold to remove these cells. Initially, we filter out cells for which there are less than 200 expressed genes and we filter out genes that are detected in less than 3 cells. Knowing that count matrices suffer from high sparsity, namely that many genes are not expressed at all in any cells, we expect that the latter step will reduce a high percentage of the initial number of genes present in the data. Indeed, the dimensionality of the datasets was reduced approximately 50%, meaning that about 13,000 genes in each dataset were filtered out during this step.

In Figures 3.1 and 3.2 we use violin plots to depict the distribution of the quality control measures for each mutant dataset. We remove cells that we believe are outliers, such as those that have more than 10 mitochondrial counts in E14_mutant. The distribution of the cells in mutants is similar for the control datasets as well.

[1] Library size refers to the total sum of count molecules for each cell across all genes.

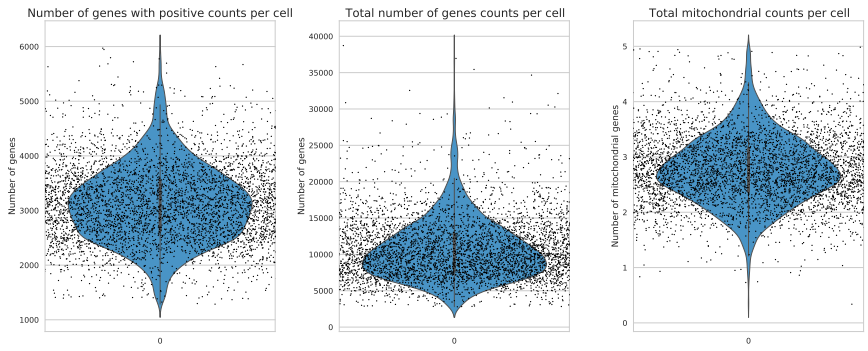


Figure 3.1: Quality control measures for E13_mutant

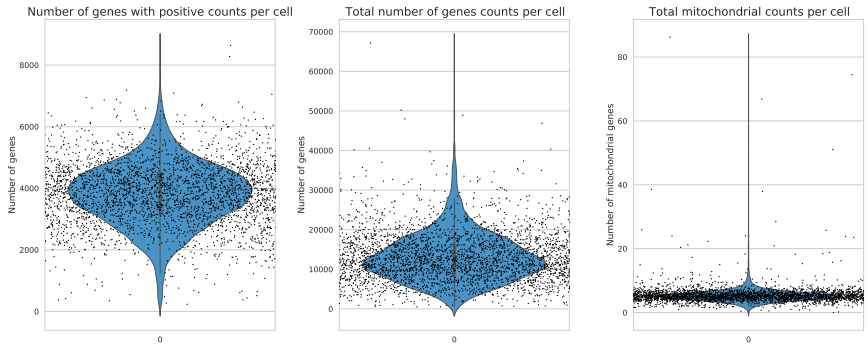


Figure 3.2: Quality control measures for E14_mutant

3.2.2 Normalisation

After removing low-quality cells, we normalise the counts per cell to enable the comparison of counts between cells. After normalisation, we log transform the data matrix X, such that $X = log(X + 1)$. This is a particularly important step since it reduces the mean–variance relationship in single-cell data [Brennecke et al., 2013] and allows to direct compare cells.

3.2.3 Feature Selection

The single-cell datasets initially contained expression values for 30,213 genes, as reported in Table 3.1. However, a vast majority of these genes contain either very low amount of mRNA molecules or no counts at all, thus providing no further information about the expression profile of the cells. During the Quality Control process, almost half of these genes (~16,000) were removed as a result of being expressed

in less than 3 cells. Despite the lower dimensional representation of the initial gene expression matrix that results from elimination of sparse genes, the datasets are still high-dimensional.

Therefore, to further reduce the dimensionality of the data, we need to select a subset of genes containing useful biological information without retaining random technical noise which can suppress any heterogeneity signals. Brennecke et al. [2013] state that further analysis performed only on this subset of genes, strongly highlights biological signals in scRNA-seq datasets.

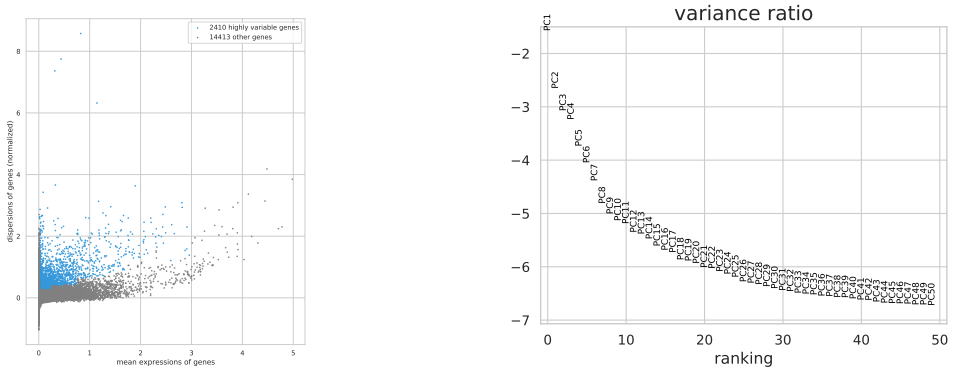
The feature selection process identifies genes that display high cell-to-cell variation due to biological differences. This process assumes that large differences in the expression of genes across single cells can be attributed to important biological differences between cells, rather than technical noise. The subset of highly variables genes is computed as following: the mean and a dispersion measure (variance divided by the mean) is calculated for each gene in each cell, using the default cutoff values. The genes are then divided into 20 bins and the normalised dispersion of all genes in each bin is compared to the average expression values. Approximately 2,000 genes were identified as highly variable for each single-cell dataset. Figure 3.3a shows the mean and dispersion of each gene in the *E14_mutant* dataset, where we observe that approximately 2,500 genes out of almost 16,000 genes are identified as highly variable.

Finally, we apply scaling, a linear transformation technique that transforms the data to have zero mean and unit variance. More specifically, the expression of each gene is shifted and scaled, so that the mean expression across cells becomes 0 and the variance across cells becomes 1. This step is important for further statistical analysis since it gives equal weights to all genes, thus prevents highly expressed genes from affecting the distribution.

3.2.4 Dimensionality Reduction

Next, we reduce the dimensionality of the datasets by applying PCA on the scaled data, retaining only the highly variable genes. Considering that different genes in the dataset are correlated given a specific biological process, we can project the genes into fewer uncorrelated dimensions. A heuristic way to investigate how much each principal component (PC) contributes to the total variance of the data and subsequently in order to consider the number of PCs to be selected for clustering, we plot the log estimate of the variance that is explained by the PCs and identify the point where the

variance is not that significant, as can be seen from the elbow plots of E14_mutant in Figure 3.3b. The point where the curve flattens is around 30 PCs, but we select 40 for further analysis to ensure that lower biological variation information that might be important for identification of rare cell types is not excluded.



(a) (b)

Figure 3.3: (a) Plot of dispersion versus mean for all genes in E14_mutant. The highly variable genes are highlighted in blue and are used in further analyses. (b) Log variance ratio across the first 50 principal components for E14_mutant.

3.2.5 Elimination of unwanted biological variation

In addition to technical variation that is present in the data because of high dimensionality, batch effects and high rate of dropouts, biological variation can also obscure the underlying biological processes of interest. Examples of such biological variation include the *percentage of mitochondrial counts* and the *cell cycle phase*, which can conceal the identity of cell populations if cells are clustered by this kind of variation.

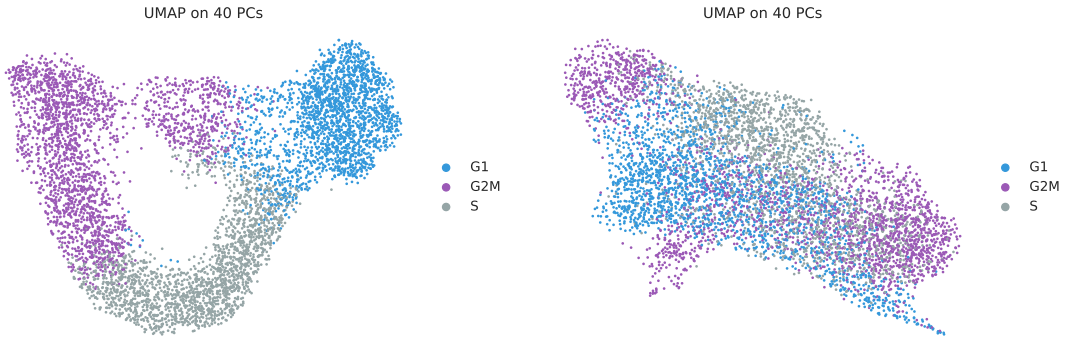
We briefly describe what a cell cycle phase is. A eukaryotic cell undergoes through a series of events that causes its reproduction and division into daughter cells. These events are called cell cycle phases and can be defined as [Morgan, 2007]: i) **G1** (*Growth*): the cell increases in size and its components are duplicated, ii) **S** (*DNA Synthesis*): each chromosome is replicated iii) **G2** (*Growth and preparation for mitosis*: the cell develops further and prepares for mitosis iv) **M** (*Mitosis*): the cell is divided into two daughter cells.

To determine whether the effect of cell cycle phase is strong enough to separate the cells according to this source of variation, it is useful to cluster the cells and explore whether they are separated by the different cell cycle phases. To do so, we first obtain a list of known marker genes from the literature for phases S and G2M and use a scoring

algorithm that assigns a score for each phase at every cell by calculating the difference between the mean expression of the given list and the marker genes. The cell is then assigned to the phase with the highest score. After the assignment, we cluster the cells and annotate by the cell cycle phases.

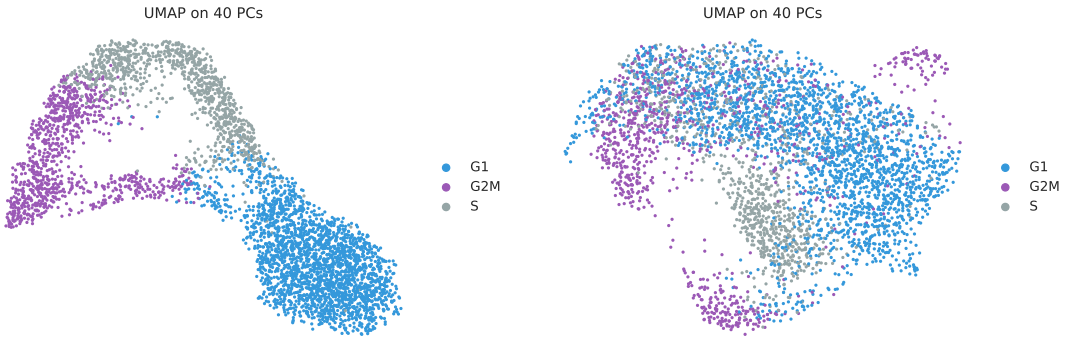
Figures 3.4a and 3.5a show the cell cycle annotations on E13_mutant and E14_mutant. We observe that the cell cycle variation is strongly present in the dataset, as cells are finely clustered by their phase annotations. It is clear that they follow a developmental trajectory, as inspected from the circular flow of the cells. The effect of cell cycle heterogeneity is reduced with a linear regression model and the corrected data projection of E13_mutant and E14_mutant is illustrated in Figures 3.4b and 3.5b, where we observe that cell cycle effect has been reduced significantly but not entirely. We argue that even though the presence of cell cycle variation might cluster cells based on cell cycle instead of cell types, controlling for cell cycle might have major drawbacks if cell types are strongly related to the cell cycle phase. Therefore, we must be cautious when completely regressing out the cell cycle phase, as we might unintentionally remove biological signal that contributes to better clustering of cell types. This is further discussed in Section 5.2 after reviewing the classification results.

Finally, after completion of the necessary preprocessing steps, Table 3.2 reports the final dimensions of all 4 datasets that we use in further analyses, along with the initial sizes to enable for comparison between dimensions.



(a) E13_mutant *before* cell cycle phase regression (b) E13_mutant *after* cell cycle phase regression

Figure 3.4: Cells in E13_mutant dataset plotted by the cell cycle phase assignments.



(a) E14_mutant *before* cell cycle phase regression (b) E14_mutant *after* cell cycle phase regression

Figure 3.5: Cells in E14_mutant dataset plotted by the cell cycle phase assignments.

Dataset	**Cells**	**Genes**	**Cells (preproc.)**	**Genes (preproc.)**
E13_mutant	6,333	30,213	6,285	1,722
E14_mutant	4,446	30,213	4,296	2,410
E13_control	3,797	30,213	3,749	2,057
E14_control	4,380	30,213	4,175	2,135

Table 3.2: Control and mutant datasets from embryonic days E13 and E14 with their corresponding gene expression matrix sizes *before* and *after* preprocessing.

Chapter 4

Experiments and Results

The previous chapter outlined the preprocessing steps that were applied to the single-cell datasets in order to eliminate technical and biological variation. In this chapter, we employ an experimental analysis aiming to identify and characterise ectopic cells that appear in the mutant datasets for days E13 and E14 after inactivation of Pax6.

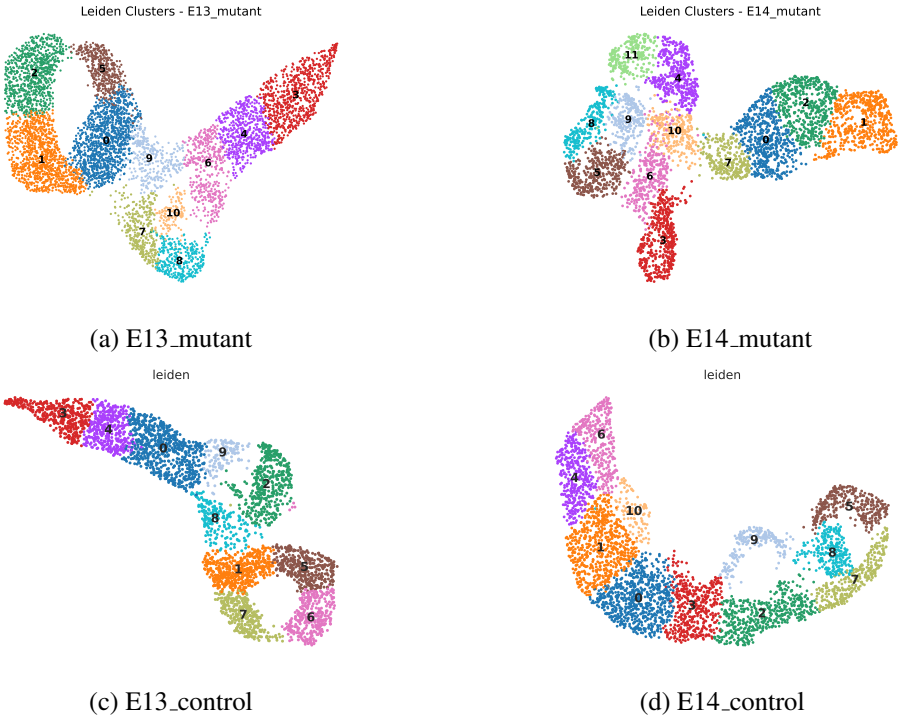
In Section 4.1 we start by demonstrating the hypothesis of the collaborating researchers at CDBS that ectopic cell populations are more distinctly visible at the E14_mutant dataset than the E13_mutant and we investigate the expression of more ectopic genes in E14_mutant to help us assign cell types to our data. Section 4.2 analyses a more robust clustering of cells in E14_mutant by using a deep denoising autoencoder, which simultaneously performs dimensionality reduction and clustering at the bottleneck layer. Finally, in Section 4.3, we follow a supervised approach to predict ectopic cells in E13_mutant given the prior knowledge we have about ectopic cells and cell type assignments from E14_mutant.

4.1 Identification of cell types

The identification of cell types from single-cell gene expression matrices is a particularly challenging problem, given that there are no ground truth labels available in the datasets, apart from the count of mRNA molecules from genes expressed in every cell. Unsupervised clustering is therefore widely applied in scRNA-seq data analysis in order to discover patterns with a biological meaning and possible groupings of cells in specific clusters that share similar expression profiles.

We initially explore our mutant datasets by applying the Leiden clustering algorithm, introduced in Section 2.3.2, to compute a neighbourhood graph of the cells and

subsequently embed it with UMAP to project the points on a two-dimensional plane for visualisation. For both datasets E13_mutant and E14_mutant, the features used for clustering are the highly variable genes projected on the first 40 principal components as identified from the preliminary steps in Sections 3.2.3 and 3.2.4. We use the default parameters for the size of the local neighbourhood (neighbouring cell points) calculated for manifold approximation, by using the euclidean distance to measure the distance between the points. The total number of clusters in the data can not be determined by the user and it changes for different values of the local neighbourhood size, resulting in preserving either a local or a more global view of the manifold. Figure 4.1 illustrates the clusters obtained from Leiden for the mutant and control datasets for both embryonic days E13 and E14.



(a) E13_mutant

(b) E14_mutant

(c) E13_control

(d) E14_control

Figure 4.1: Clusters identified by the Leiden algorithm on the first 40 PCs. Each cluster is represented with a distinct number and colour.[1]

4.1.1 Known marker genes

In Section 2.2 we highlighted the importance of utilising prior biological knowledge for the expression profiles of known cell types, based on the genes that are highly active

[1]Clusters of a specific number and/or colour in E13_mutant do not correspond to the same cluster or cells in E14_mutant and vice versa.

in each type. Cells in this dataset belong to one of the three major cell types:

- *Neural Progenitors*: cells that divide a limited number of times to produce daughters that can differentiate into specialized cell types.
- *Intermediate Progenitors*: cells produced by Neural Progenitors that are in a higher state of differentiation and will divide once more to produce neurons.
- *Post-mitotic neurons*: cells that have fully differentiated and do not divide anymore.

These cell types were identified based on prior biological knowledge of the brain cortex cell identities, previous lab experiments and from the bulk RNA-seq analysis performed before scRNA-seq. Moreover, a curated list of marker genes was identified for each cell type, where genes exhibit higher expression in a specific cell type compared to the rest. All three cell types are expected to be present both in the *control* groups (represent normal conditions) and the *mutant* groups (after Pax6 inactivation).

However, after inactivation of Pax6, some cells do not behave as expected, hence their identity is unknown and expression is abnormal, or else *ectopic*. Ectopic cells appear only in the mutant datasets, where the function of Pax6 has been diminished. Moreover, we identify some genes in E13_mutant and E14_mutant that serve as ectopic markers as they are highly expressed in some cells in the mutant datasets but not expressed in the control datasets. The identified marker genes for each cell type are listed in Table 4.1.

Cell Type	**Marker Genes**
Neural Progenitors	Pax6, Vim, Sox2
Intermediate Progenitors	Eomes, Btg2
Post-mitotic Neurons	Tbr1, Sox5
Ectopic	Gsx2, Prdm13, Dlx1, Dlx2, Dlx5, Gad1, Gad2, Ptf1a, Msx3, Helt, Olig3

Table 4.1: Marker genes identified for the 3 main cell types and for ectopic cells that are present in the developing brain cortex.

In order to examine the ectopic expression in the mutant datasets, we plot the normalised expression of each gene in Figure 4.2. We observe that in E14_mutant (Figure 4.2a), most ectopic marker genes such as *Dlx1*, *Dlx2*, *Dlx5* and *Gad1*, have a very

high expression value in clusters located at the lower part of the two-dimensional plot. A similar pattern has also been observed for the expression of marker genes for the 3 main cell types for both datasets. However, the expression of the ectopic marker genes has a weaker signal in E13_mutant, as can be seen in Figure 4.2b and we can not specify a distinct cluster of cells expressing these genes, but rather the cells are low in number and are sparsely spread out through the dataset.

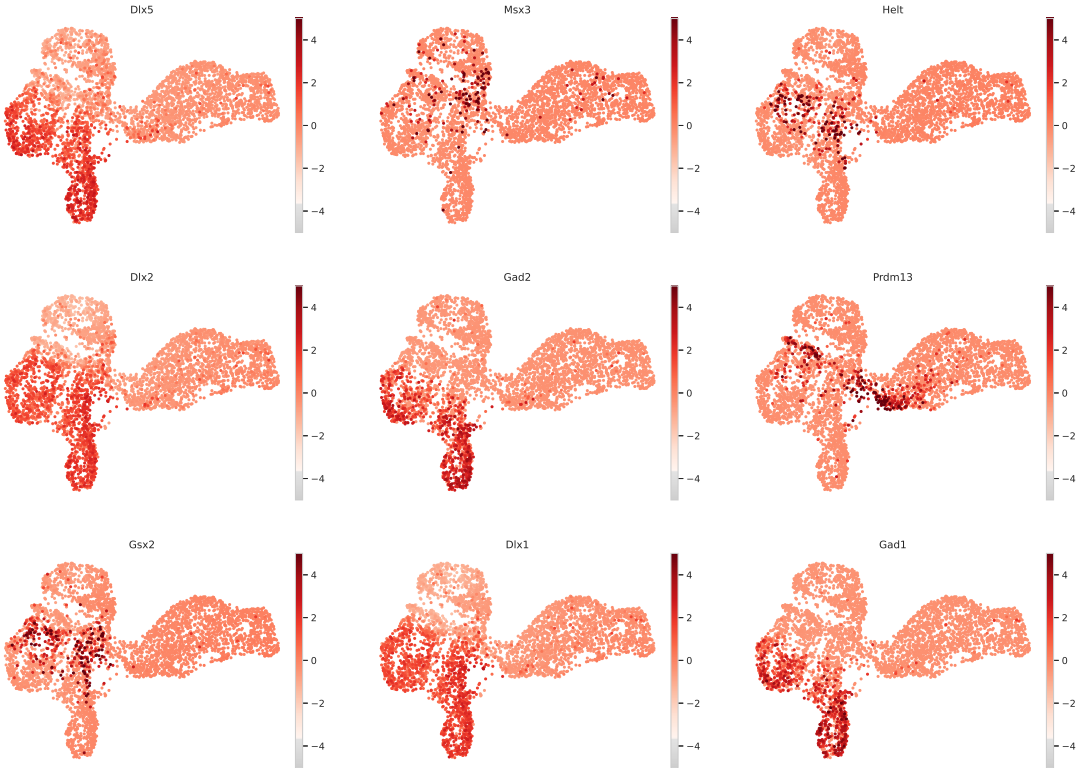
Furthermore, the heatmaps in Figure 4.3 enable a better comparison of the normalised marker gene expressions between the two datasets. Each row represents a cell grouped by the Leiden clusters and each column is a marker gene grouped by the cell type where it is highly expressed. In E14_mutant (Figure 4.3b), we observe that the ectopic genes *Dlx1, Dlx2, Dlx5* are highly expressed in clusters *3, 5, 6* and *8* when at the same time *Eomes* and *Tbr1* have almost no expression in the same clusters. Also marker genes *Gad1* and *Gad2* follow the same pattern but indicate a lower expression signal. Based on this visualisation, we confirm the biological evidence of the CBDS researchers, supporting that Pax6 inactivation causes some progenitor cells to develop into a new cell type instead of following their normal developmental trajectory.

More interestingly, the expression pattern of ectopic marker genes in E13_mutant (Figure 4.3a) is not similar to E14_mutant, even though the development of the cells is one day apart, from day E13 to day E14. We would expect that ectopic genes *Dlx1, Dlx2* and *Dlx5* would be highly expressed in cells belonging to clusters *0*, *1*, *2*, *5*, where *Eomes* and *Tbr1* are turned off, but their expression is nearly zero, except for a few cells. *Msx3* however, shows higher expression in nearly all clusters, thus is not that informative to help us characterise ectopic cells.

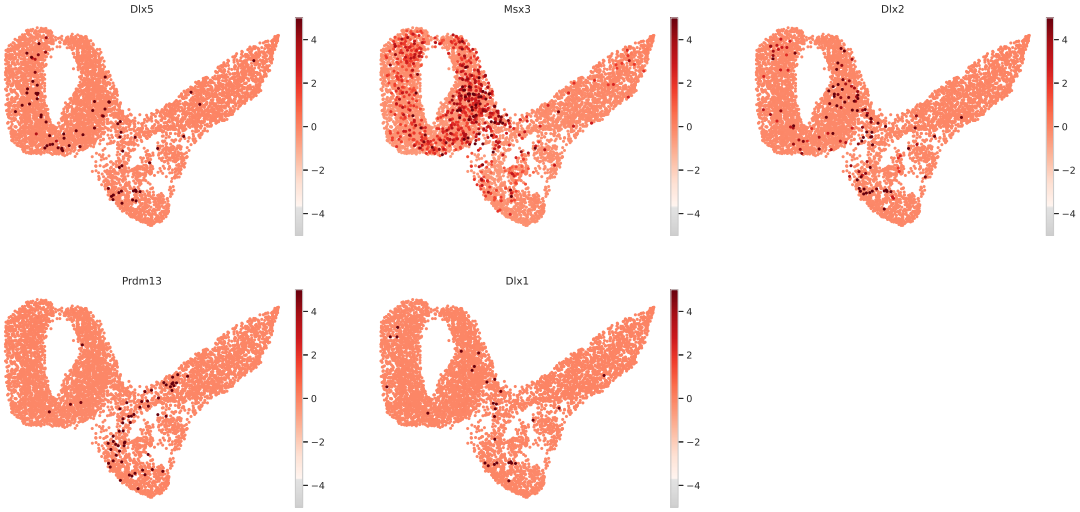
4.1.2 Differential Expression Analysis

Differential Expression Analysis, introduced in Section 2.2, is a useful process that enables the identification of marker genes that are upregulated in each cluster compared to the remaining clusters. By having a larger subset of genes highly expressed in each cluster, we are able to acquire higher confidence for the cell type identity of each cluster, without relying exclusively on the previously defined marker genes based on prior knowledge (listed in Table 4.1).

We apply differential expression analysis on E14_mutant, since the expression signal of ectopic marker genes is stronger and the grouping of ectopic cells is more consistent. To rank the genes in each cluster depicted in Figure 4.1b, we use the Wilcoxon



(a) E14_mutant



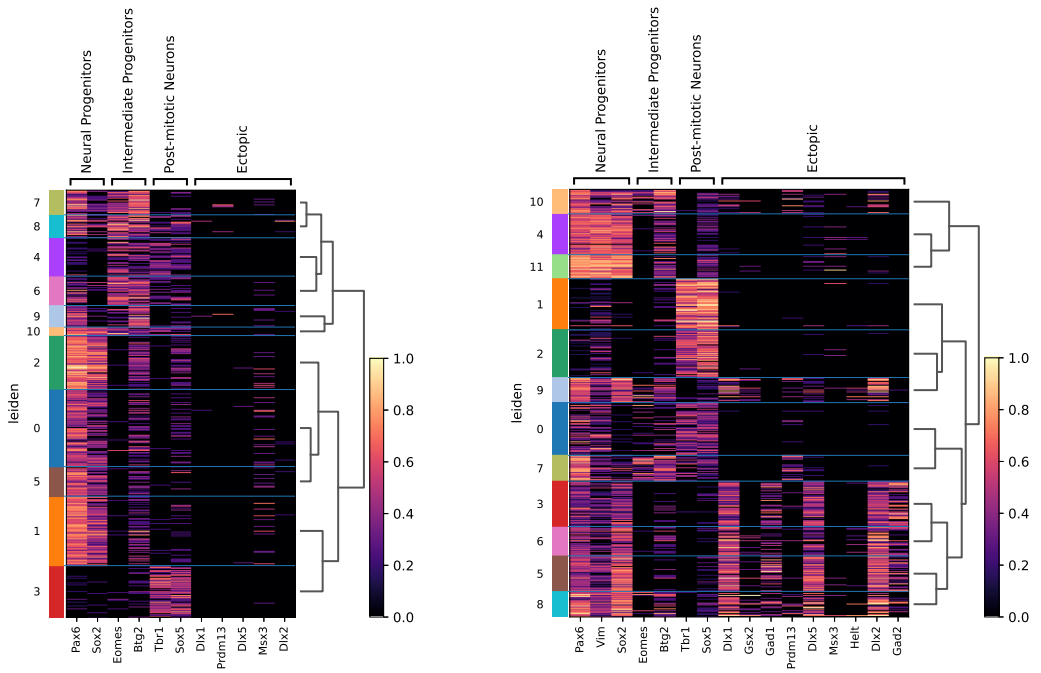
(b) E13_mutant

Figure 4.2: Marker genes found to have ectopic expression.

rank-sum method, described in Section 2.2, and we report the top 10 genes for each cluster[2] in Table 4.2.

An interesting observation from the heatmap in Figure 4.4a is the co-expression of

[2]The top 100 differentially expressed genes for each cluster can be found in the Appendix.



(a) E13_mutant (b) E14_mutant

Figure 4.3: Heatmap of the expression value of the marker genes for the 3 main brain cell types and the ectopic for each cell in a cluster. Rows are cells grouped by the clusters shown in Figure 4.1 and columns are the known marker genes, grouped by the cell type they are mostly upregulated in.

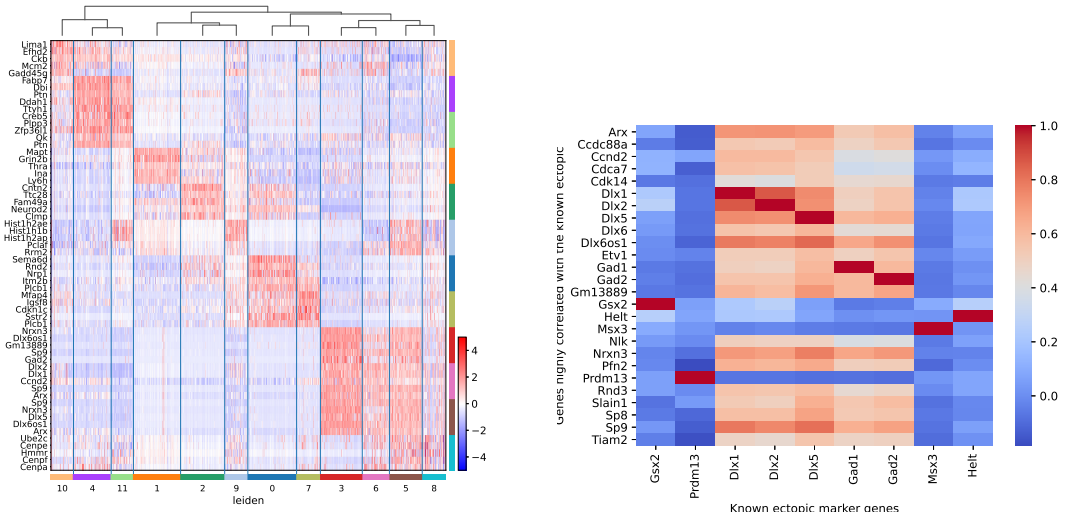
marker genes in clusters, which can also be verified by the dendrogram at the top. We recognise the known ectopic marker genes *Dlx1, Dlx2, Dlx5, Gad1, Gad2* and we can see from the plot that they are upregulated mostly in clusters *3*, *6* and *5*, indicating that these clusters have high ectopic expression.

Following this observation, we can now investigate other markers that are highly expressed in these clusters, which we have no prior information about. Examples of potential ectopic marker genes can be identified by calculating the correlation between the known marker genes and the top ranked marker genes we obtained from DE analysis. For this we used Pearson correlation, which calculates the linear correlation between two variables and takes values between -1 and 1, with -1 representing negative correlation, 0 no correlation and 1 positive correlation. A subset of genes with a positive correlation of greater than 0.5 is illustrated in Figure 4.4b. The genes in x-axis are the known ectopic marker genes and in y-axis the differentially expressed marker genes that were found to be mostly correlated with the ectopic ones. We notice that genes *Arx*, *Nrxn3*, *Pfn2*, *Sp8* and *Sp9* have a high positive correlation (values closer to 1) with the known ectopic genes *Dlx1*, *Dlx2*, *Dl5*, *Gad1 Gad2*, which we provide to

Rank	0	1	2	3	4	5	6	7	8	9	10	11
#1	Sema6d	Mapt	Cntn2	Nrxn3	Fabp7	Sp9	**Dlx2**	Mfap4	Ube2c	Hist1h2ae	Lima1	Creb5
#2	Rnd2	Grin2b	Ttc28	Dlx6os1	Dbi	Nrxn3	**Dlx1**	Igsf8	Cenpe	Hist1h1b	Efhd2	Plpp3
#3	Nrp1	Thra	Fam49a	Gm13889	Ptn	**Dlx5**	Ccnd2	Cdkn1c	Hmmr	Hist1h2ap	Ckb	Zfp36l1
#4	Itm2b	Ina	Neurod2	Sp9	Ddah1	Dlx6os1	Sp9	Sstr2	Cenpf	Pclaf	Mcm2	Qk
#5	Plcb1	Ly6h	Clmp	**Gad2**	Ttyh1	Arx	Arx	Plcb1	Cenpa	Rrm2	Gadd45g	Ptn
#6	Igfbpl1	Cnih2	Neurod6	Slain1	Zfp36l1	Pclaf	Cdca7	Shf	Ccnb1	Slbp	**Btg2**	Sox9
#7	Sstr2	Islr2	Gpm6a	**Dlx5**	Mt1	**Dlx1**	Dlx6os1	Fam110a	Tpx2	Dek	Hes6	Gas1
#8	Cttnbp2	Tubb2a	Zbtb18	Etv1	**Vim**	Pfn2	Mcm2	Nhlh1	Cks2	Dut	Ung	Fabp7
#9	Sorbs2	Mef2c	Znrf2	Sp8	Phgdh	Rrm2	Mcm6	Neurog2	Cdc20	Insm1	Zeb1	Dbi
#10	Rbfox3	Lrfn5	Dpy19l1	**Gad1**	Mfge8	Top2a	**Dlx5**	Igfbpl1	Sgol2a	Atad2	Chd7	Nes

Table 4.2: Top 10 differentially expressed genes for each cluster {0-11} in E14_mutant. Known marker genes (see Table 4.1) are highlighted in bold.

the collaborating researchers at CBDS for further experimentation in the lab in order to check for ectopic expression.



(a) (b)

Figure 4.4: (a) Heatmap of the expression of the top 5 differentially expressed genes for each cluster in E14_mutant. The numbers on x-axis represent clusters and the genes on the y-axis are the top 5 marker genes identified in each cluster. (b) Normalised correlation values of the differentially expressed genes with the known ectopic marker genes. The x-axis denotes the known ectopic marker genes and the y-axis the differentially expressed genes in the ectopic cluster.

4.1.3 Cell type annotation

Most scRNA-seq analyses depend on experts' knowledge to manually assign the cells to a list of expected cell types, based on biological assumptions which are almost always dataset-specific. The cell type annotation process can be tedious and inefficient,

resulting in big variations between annotations which prevent the reproducibility of the results across different datasets and research labs. This issue is exacerbated as the dimensionality of single-cell datasets increases exponentially, thus requiring more time and human resources to complete the annotation of a single dataset [Abdelaal et al., 2019].

Here we implement an automatic pipeline to assign cells into cell types by combining two sets of information: the known marker genes and the top 100 differentially expressed marker genes. For each cluster, we calculate the overlap counts of the known marker genes with the top differentially expressed genes resulted from the analysis in Section 4.1.2 and assign the cell type that maximises this overlap. Based on this process, we annotate the single cells in E14_mutant following the assumption that if a known ectopic marker gene is identified in the top 100 differentially expressed genes in a cluster, especially high in ranking, then we are more confident that this cluster expresses ectopic genes.

Mathematically, we define the overlap score with the following Equation 4.1:

$$\text{overlap}(G,R) = \frac{N(G \cap R)}{N(G)} \tag{4.1}$$

where G is the set of known marker genes, R is the set of the 100 ranked differentially expressed genes and N is the cardinality operator of a set, which works as a normalisation factor to account for the difference in the number of known marker genes identified for each cell type. For example, the total marker genes for Ectopic cells are 11, while for Neural or Intermediate Progenitors the number is 2 and 3 respectively (Table 4.1).

The annotation process is illustrated in Figure 4.5. From the overlap gene matrix on the left, we assign each cell of E14_mutant into one of the 5 following cell types: *Neural Progenitors, Intermediate Progenitors, Post-mitotic Neurons, Ectopic* and *Unknown*[3]. After the annotation process is complete, we proceed with finding the top marker genes for each cell type. The heatmap at the bottom left corner of 4.5 shows that the expression of marker genes is high for each cell type and very low for all other cell types, indicating that the identified genes can be considered as markers for a specific cell type with high confidence. The cell type annotations on the two-dimensional UMAP plot of E14_mutant can be seen more distinctly in Figure 4.6.

[3]A cluster is annotated as *Unknown* if no known marker genes were found in the top 100 ranked list.

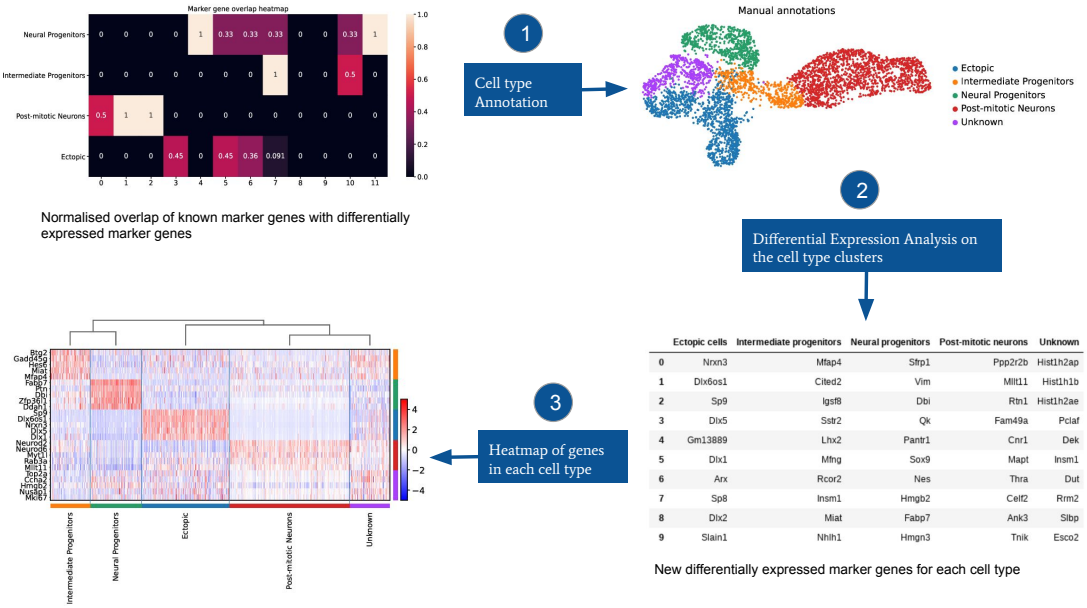


Figure 4.5: Cell type annotation process based on the argmax of the overlap counts of the known marrker genes with the top k differentially expressed genes.

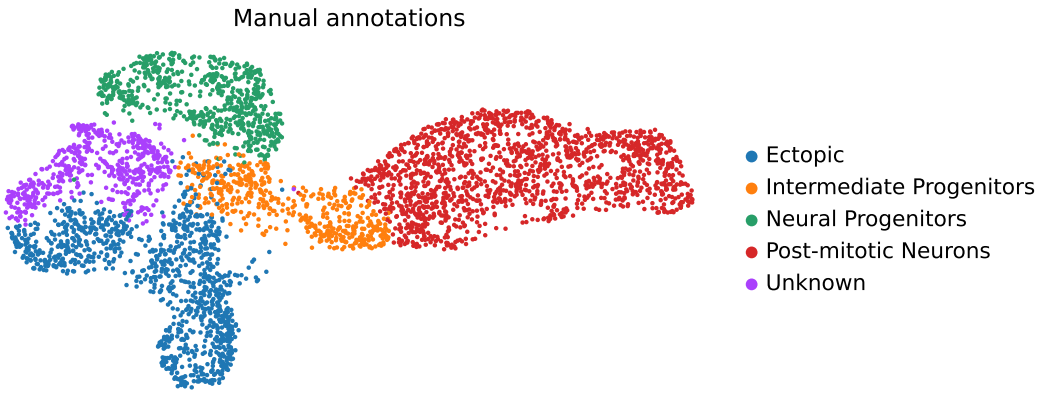


Figure 4.6: Cell type annotations on the E14_mutant dataset, visualised on a two-dimensional UMAP plot.

4.2 Unsupervised feature representation for clustering

The initial scRNA-seq analysis conducted in Chapter 3 and Section 4.1 follows best practices suggested by Seurat and Scanpy for scRNA-seq preprocessing and analysis. This analysis, including normalisation, log-transformation, linear dimensionality reduction with PCA followed by graph-based clustering and projection in two dimensional plots with UMAP, might not be optimal for cases where we want to detect novel cell populations, such as the ectopic ones, in an unsupervised manner.

Deep learning approaches that employ non-linear dimensionality reduction and subsequent clustering of cells, potentially have greater power in performing cell type

identification and do not assume strict linear relationships present in the data. In this section, we experiment with *scDeepCluster* [Tian et al., 2019], a recent denoising autoencoder model specifically tailored for scRNA-seq data, described in detail in subsection 2.4. We explain the experiments using the bottleneck layer of the autoencoder model and interpret the results.

4.2.1 Bottleneck layer

We are mostly interested in the latent feature space representation of scDeepCluster, also known as the *bottleneck layer* of the autoencoder which has the lowest dimensionality of all layers.

The encoder and decoder each consist of two hidden fully connected layers. The number of units in the first layer of the encoder is 256 with a following layer of 64 units, and the decoder has the same layers in reverse order. The size of the bottleneck layer is 32 and the numbers of the clusters in K-Means is set to 10. Initially, we replace the existing preprocessing pipeline of the model with the one analysed in Section 3.2, to ensure reproducibility between the different single-cell datasets and to be able to compare the autoencoder with the dimensionality reduction and clustering approach that was performed in Section 4.1 on the same dataset. Then, we change the number of expected clusters from 10 to 5, as we expect to see 5 different cell types in E14_mutant, based on the annotations in 4.1.3. However, we acknowledge that we cannot know the true number of clusters so we have to rely on assumptions based on the expected number of cell types for the specific dataset.

As with Section 4.1.1, the aim of this section is to use the E14_mutant dataset that has a stronger ectopic signal in order to extract information about the cell types, but using a more robust dimensionality reduction and clustering method. We initially experimented with a different size for the bottleneck in the range (2, 4, 8, 16, 32), in order to test which dimensionality provides the best clustering for E14_mutant. We use the evaluation metrics *Purity*, *Normalized Mutual Information* (NMI) and *Adjusted Rand Index* (ARI), defined in Section 2.6, to compare the purity of the clusters on the bottleneck to our custom cell type annotations from Section 4.1.3. The results for the different bottleneck sizes can be seen in Table 4.3 and the clusters in each bottleneck layer are depicted in Figure 4.7. We observe that all metrics are higher for the bottleneck layer with dimensionality 16 (Purity = 0.6936, NMI = 0.7135, ARI = 0.6157), indicating that the generated clusters are more pure and have fewer mixed

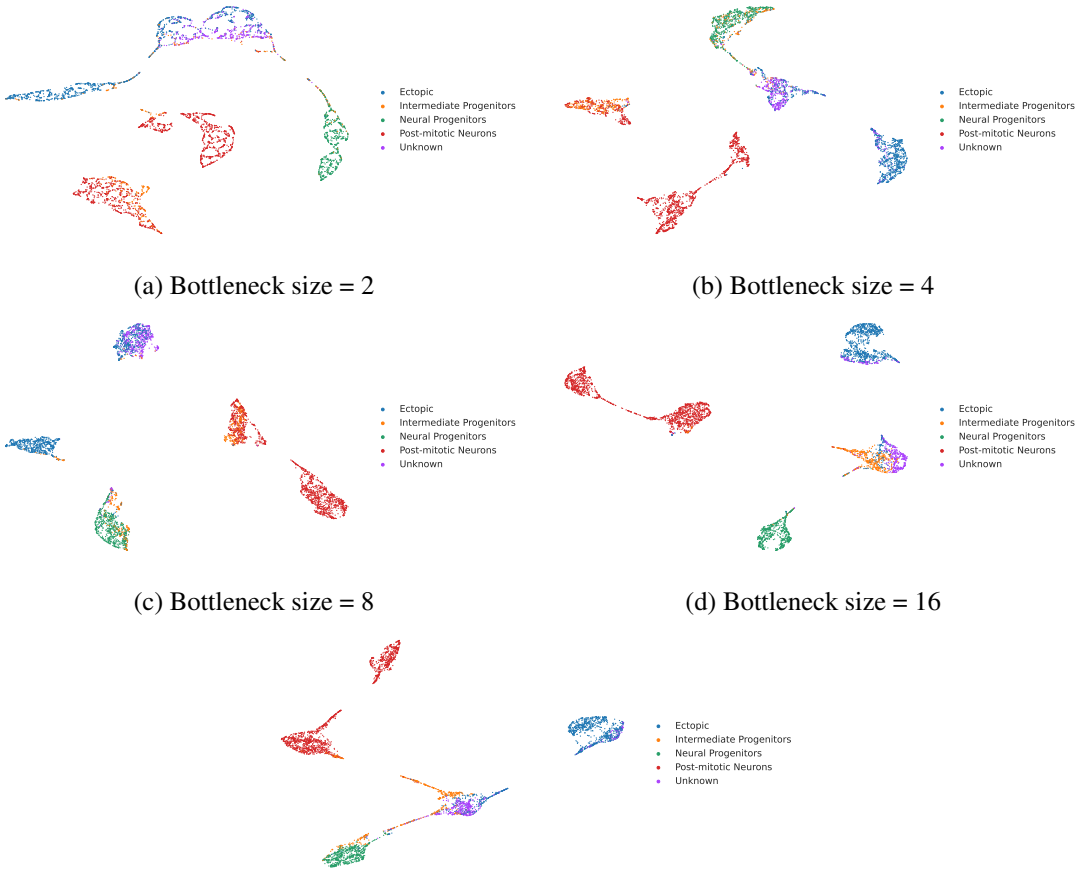
cell types in a single cluster. This can also be confirmed with a visually inspection of the clusters for the varying bottleneck size. The ectopic cells, in which we are mostly interested, can be found mainly in the one cluster with a few cells that are of "Unknown" type. The cluster containing the Neural Progenitors and Post-mitotic Neurons respectively seems to be very pure, apart from a couple of cells belonging to Ectopic and Intermediate Progenitors. The remaining cluster contains a mixture of all cell types and is not informative regarding cell types.

Bottleneck size	**Purity**	**NMI**	**ARI**
32	0.6761	0.6714	0.5783
16	**0.6936**	**0.7135**	**0.6157**
8	0.6814	0.6419	0.5109
4	0.6676	0.5377	0.4142
2	0.6793	0.6192	0.5033

Table 4.3: Purity, Normalised Mutual Information (NMI) and Adjusted Rand Index (ARI) metrics for different values of the bottleneck size.

The next experiment involves changing the number of layers and the bottleneck size simultaneously in order to evaluate clustering on the bottleneck layer. First we add an extra layer of size 128 between the layers of size 256 and 64, and we gradually reduce the bottleneck size, by adding an extra layer of a smaller size until we reach a bottleneck of size 2. Using the same evaluation metrics as before, we report the scores obtained from each experiment and the layers with the corresponding sizes that were used in each in Table 4.4. However, we observe that increasing the number of layers and further decreasing the bottleneck size, especially at the lowest dimensionality possible, the autoencoder does not succeed in reconstructing the initial dataset and does not result in meaningful clusters that optimally separate the cell types. Therefore, we proceed with the latent feature representation with the highest score obtained from the previous experiment, where there are two layers of size 256 and 64 respectively and a bottleneck layer of size 16 (Table 4.3).

As a final step, we validate the clustering of cells in the bottleneck layer by plotting the expression of the known ectopic marker genes to examine whether the expression lies on the identified ectopic cluster on the lower dimensional space. Figure 4.8 confirms that the lower dimensional bottleneck layer of size 16 correctly preserves the information about the ectopic cell type from the initial dataset. Especially the expres-



(a) Bottleneck size = 2

(b) Bottleneck size = 4

(c) Bottleneck size = 8

(d) Bottleneck size = 16

(e) Bottleneck size = 32

Figure 4.7: Clusters created for different values of the bottleneck layer of the scDeepCluster autoencoder model

Encoder layer sizes	Num. layers	Purity	NMI	ARI
(# genes, 256, 128, 64, 32)	4	**0.6889**	0.5761	0.4787
(# genes, 256, 128, 64, 32, 16)	5	0.6746	**0.6648**	**0.6648**
(# genes, 256, 128, 64, 32, 16 8)	6	0.4642	0.4944	0.3281
(# genes, 256, 128, 64, 32, 16 8, 2)	7	0.5217	0.3538	0.3015

Table 4.4: Purity, Normalised Mutual Information (NMI) and Adjusted Rand Index (ARI) metrics for a different number of layers and a varying bottleneck size, which is reduced in each experiment.

sion of ectopic marker genes *Dlx1, Dlx2, Dlx5* and *Gad1, Gad2* is high in the ectopic cluster on the right hand side of the plot and absent from the clusters on the left, which have been assigned as the Post-mitotic Neurons in Figure 4.7.

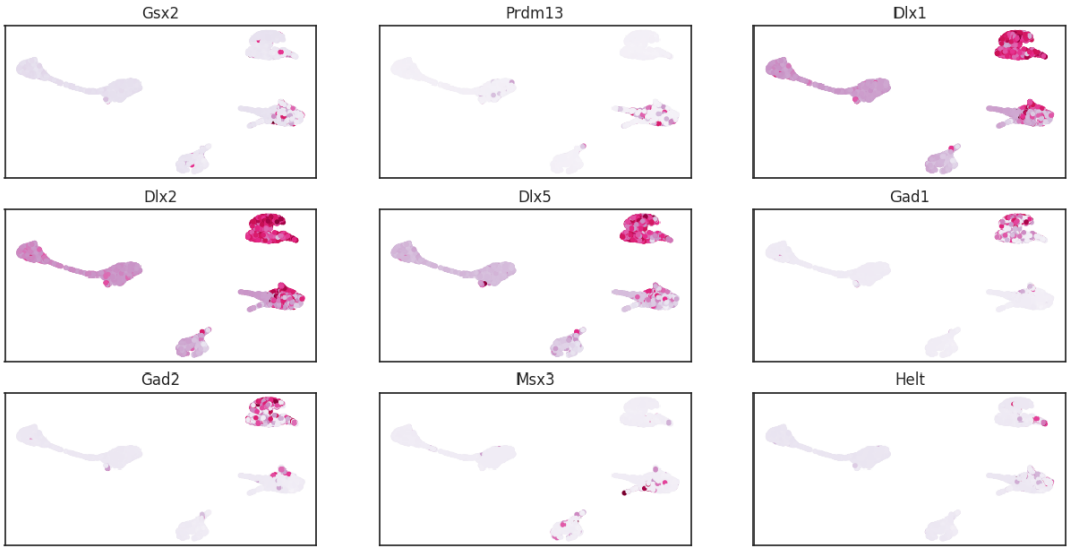


Figure 4.8: Expression of each gene in the predefined ectopic marker genes list at the bottleneck layer of size 16. The ectopic marker genes are mostly expressed in the ectopic cluster of Figure 4.7d.

4.3 Supervised cell type prediction

While in the previous two sections we examined the identification of cell types with a linear dimensionality reduction approach and a deep learning approach on the E14_mutant, in this section we investigate cell type identification in E13_mutant, where the signal of ectopic cells is very low and their characterisation can not be easily accomplished. We follow a supervised approach, where we use E14_mutant to train and validate classifiers and E13_mutant as the final test set.

4.3.1 Model Selection

Following the above rationale, we select four classifiers: *Support Vector Machines (SVM)* [Cortes and Vapnik, 1995], *Random Forest* [Breiman, 2001], *Decision Tree* [Breiman et al., 1984] and *Logistic Regression* to perform multiclass classification using the following cell types as class labels: i) *Neural Progenitors*, ii) *Intermediate Progenitors*, iii) *Post-mitotic Neurons*, iv) *Ectopic* and v) *Unknown*. Using the annotations on E14_mutant (visible in the UMAP plot in Figure 4.6), we perform grid search on the hyperparameters of each method using 3-fold cross-validation on the dataset. This way we avoid overfitting on E14_mutant, aiming for a model that generalises well on other datasets of the same cell types, such as E13_mutant.

Classifier	**Val. Accuracy** (%)	**Val. Macro F1-score** (%)
SVM	88.99 ± 1.01	82.60 ± 2.18
Random Forest	$\mathbf{93.15 \pm 0.51}$	$\mathbf{91.08 \pm 0.28}$
Decision Tree	89.21 ± 0.21	86.44 ± 1.28
Logistic Regression	91.06 ± 0.18	88.28 ± 0.71

Table 4.5: Accuracy and macro F1-score using 3-fold cross-validation on the E14_mutant dataset.

Table 4.5 shows that the Random Forest classifier has the highest accuracy and macro F1-score on the 3 validation splits (Accuracy = 93.15 ± 1.01, Macro F1-score = 91.08 ± 0.28), with a small difference from the Logistic Regression classifier. We evaluate the Random Forest classifier on E13_mutant and we obtain predicted cell types for each cell, which can be seen on the two-dimensional UMAP plot in Figure 4.9. We emphasise that even though there are no true labels for E13_mutant, we expect the cells of a specific cell type to have a small distance between them and a large distance with cells of other cell types.

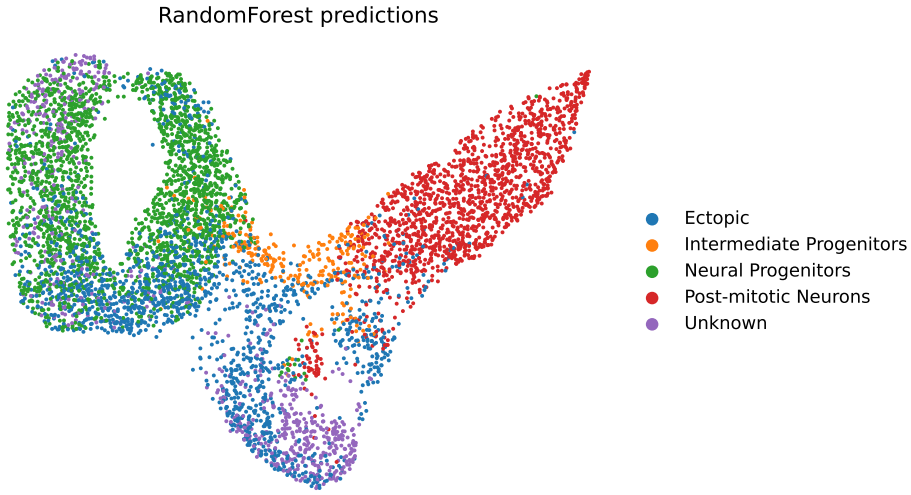


Figure 4.9: Projection of cells in E13_mutant, annotated with the predicted cell types by Random Forest.

As can been seen in Figure 4.9, most cell points on the left have been predicted as *Neural Progenitors*, cells in the middle as *Intermediate Progenitors* and the ones on the right as *Post-mitotic Neurons*. The prediction of these 3 cell types aligns with the expression of the known marker genes, which implies that the prediction of Random Forest on E13_mutant correlates with the biological assumptions with regards to the

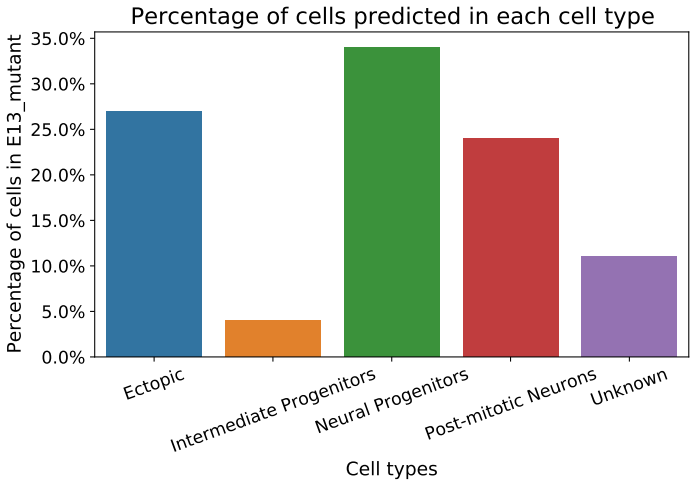


Figure 4.10: Percentage of predicted cells for the 5 cell types.

cell types. Interestingly though, the barplot in Figure 4.10 shows that the number of cells that the classifier predicted as ectopic is higher than expected, given that the expression of ectopic marker genes in E13_mutant is very weak. However, the classifier could not separate them well enough from the Unknown cells, which can be seen mostly blended with Neural Progenitor and Ectopic cells.

4.3.2 Differential Expression Analysis on predictions

To identify the genes that Random Forest used to separate the cells into the 5 distinct cell types, we perform Differential Expression Analysis on the predicted cells, aiming to find a set of genes that are highly expressed in the Ectopic and possibly the Unknown cell type compared to the rest. This will give us an indication of possible additional marker genes for ectopic cells that were previously masked by the expression of more active genes.

The top 20 ranked differentially expressed genes for each predicted cell type category can be seen in Table 4.6. We notice that the differentially expressed genes found for ectopic cells do not include any of the known ectopic marker genes and the only known ones are *Pax6* and *Eomes*, which are low in the rank. After showing the obtained marker genes for *Ectopic* and *Unknown* to the CDBS researchers, the conclusion is that many highly ranked genes are related to cancer and to cell cycle phase, particularly phase *S*, indicating that cell cycle effect is still prevalent in the data even though it was regressed out, as described in Section 3.2.5. These findings indicate that specific phases of the cell cycle are possibly associated with ectopic cell types, confirming that

early decisions about cell cycle heavily affect the results.

Rank	Ectopic	Interm. Prog.	Neural Prog.	Post-mitotic Neur.	Unknown
0	Rrm2	Neurog2	Gas1	Rtn1	Top2a
1	Pclaf	Hes6	Fabp7	Mllt11	Cdk1
2	Pcna	Gadd45g	Id4	Neurod6	Nusap1
3	Lig1	**Btg2**	Hes1	Neurod2	Spc25
4	Slbp	Mfng	Sfrp1	Tubb3	Ube2s
5	Mcm6	Sox4	Ptn	Rit1.1	Cks2
6	Tyms	Meg3	Ttyh1	Dcx	Smc2
7	Rpa2	Rprm	Dbi	Elavl3	Ccnb1
8	Dut	Btbd17	Zfp36l1	Rab3a	Smc4
9	Nasp	Chd7	Hmgn3	Rnd2	Birc5
10	Gmnn	Cited2	Pantr1	Stmn3	Ccna2
11	Mcm5	Neurog1	Aldoc	Tagln3	H2afv
12	Tipin	Dll1	Psat1	Sox11	Cdca8
13	Tk1	Plk3	Sox9	Cd24a	Cks1b
14	Dhfr	Pak3	**Sox2**	Igfbpl1	Prc1
15	Fen1	Elavl4	Mif	Cnr1	Cenpf
16	Clspn	**Eomes**	Mpped2	Vat1	Hmgn2
17	Hells	Fam110a	Phgdh	Podxl2	Bub3
18	Chaf1b	Rgs16	**Pax6**	**Tbr1**	Arl6ip1
19	Uhrf1	Igsf8	Ndrg2	Crmp1	Ube2c

Table 4.6: Top 10 differentially expressed genes for each predicted cell type in E13_mutant. Known marker genes (see Table 4.1) are highlighted in bold.

Chapter 5

Conclusions

The main goal of this dissertation was to identify rare, ectopic subpopulations in single-cell RNA-seq datasets from the developing mouse brain cortex. Building upon the biological hypothesis of the collaborating researchers at CBDS, we have made considerable progress towards the objectives outlined in Section 1.2:

1. In Section 4.1.2, we investigated the E14 dataset to identify a list of genes that are highly correlated with the predefined ectopic marker genes. These genes can be used by the collaborating researchers at CDBS for further investigation of ectopic cells by performing lab experiments. Then we proceeded with a cell type annotation process in E14, based on the assumptions we had for cell types and what constitutes ectopic expression.

2. We demonstrated in Section 4.2.1 that the use of an autoencoder neural network model, that simultaneously performs dimensionality reduction and clustering, projects cells from E14 into a latent feature space where there is better separation of cell clusters with regards to their cell types.

3. We investigated a supervised approach in Section 4.3, where we trained classifiers on E14 and predicted cell types on E13 based on representations learned from E14. We demonstrated that the Random Forest classifier can effectively predict cell types in E14 (approximately 92% F1-score) and identifies a larger number of ectopic cells in E13. However, the interpretation of the highly expressed genes in the predicted ectopic cells is complicated and is discussed in the following section.

The contributions of this project consist of the outcomes of the experiments listed above and the preprocessing pipeline of the single-cell RNA-seq datasets in Python, as

previously it was only available in R. Preprocessing of single-cell datasets in particular is a big part of the analysis and had to be evaluated multiple times before and during the experiments.

Section 5.1 below discusses limitations we faced with regards to the single-cell datasets, the machine learning methods and the results we obtained. Finally, Section 5.2 presents possible extensions and directions for future work.

5.1 Limitations

As Clevers et al. [2017] have pointed out, the concept of a cell type is not strictly defined. The single-cell datasets we used do not contain any ground truth cell type labels, thus it is difficult to define what each cluster represents biologically. Also, the results can only be evaluated by visual inspection of the clusters and the expression of each marker gene. This way of evaluation is not feasible for computational methods and it is usually performed by biology experts, since prior biological knowledge is required.

Following the same reasoning, the cell type annotation process performed on the E14_mutant dataset, (described in Section 4.1.3), is based on assumptions which take into account the presence of the three specific main cell types and the nature of ectopic marker genes. This implies that any assumptions the classifier makes for E14_mutant will be applied to E13_mutant, which might not be biologically correct. Moreover, the fact that the assumptions are dataset-specific prevents the reusability of the methods and the reproducibility of results in other datasets, although the preprocessing pipeline should not differ a lot between datasets.

In addition, single-cell classification is challenging due to substantial differences between the two datasets. A well-known variation of single-cell datasets is the batch effect, described thoroughly in Section 2.1.1. Nevertheless, the specific datasets suffer from another source of variation that occurs from the sequencing technologies. A different version of the 10X Genomics sequencing was used to sequence the mouse cells for days E13 and E14, resulted in a difference of approximately 2,000 additional cells in E13_mutant that in the other 3 datasets (E14_mutant, E13_control, E14_control). This variation breaks the assumption of machine learning algorithms that the examples in the training set (in this case E14_mutant) and test set (E13_mutant) follow the same distribution.

5.2 Future Work

Section 3.2.5 emphasised the importance of eliminating unwanted biological variation from the specific single-cell datasets, such as the cell cycle phase, since it might mask the heterogeneity of ectopic cells. However, the effect of cell cycle phase has been removed with a linear regression model, thus assuming that the expression of the phases of the cell are linear. One should treat the cell cycle phase as periodic and could experiment with non-linear methods to remove its effect, such as the *Cyclum* autoencoder method proposed by [Liang et al., 2020]. This step could also investigate if the genes for the predicted ectopic cells in E13 (Table 4.6) are highly expressed due to the cell cycle phase effect still present in the data or driven by another variation.

Following the experiment in Section 4.2, one possible extension is to take into account the probabilistic nature of the count data and investigate the latent space of variational autoencoders, such as *scVI* [Lopez et al., 2018] to also account for uncertainty.

Finally, another approach to examine the identification of single-cells is to integrate the two datasets E13_mutant and E14_mutant and eliminate technical variations such as batch effects between the two samples. This way the two integrated datasets could be compared against the control groups, which were not used in this project. An initial integration experiment has been conducted and can be found in the Appendix, however no results about ectopic cells are available. We conclude that the mutant and control single-cell datasets are a great resource to further investigate ectopic gene expression.

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

T. Abdelaal, L. Michielsen, D. Cats, D. Hoogduin, H. Mei, M. J. Reinders, and A. Mahfouz. A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome biology*, 20(1):194, 2019.

R. A. Amezquita, A. T. Lun, E. Becht, V. J. Carey, L. N. Carpp, L. Geistlinger, F. Martini, K. Rue-Albrecht, D. Risso, C. Soneson, et al. Orchestrating single-cell analysis with bioconductor. *Nature Methods*, pages 1–9, 2019.

E. Becht, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Evaluation of umap as an alternative to t-sne for single-cell data. *BioRxiv*, page 298430, 2018.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

P. Brennecke, S. Anders, J. K. Kim, A. A. Kołodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. A. Teichmann, J. C. Marioni, et al. Accounting for technical noise in single-cell rna-seq experiments. *Nature methods*, 10(11):1093–1095, 2013.

F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155–160, 2015a.

F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155–160, 2015b.

H. Clevers, S. Rafelski, M. Elowitz, A. Klein, J. Shendure, C. Trapnell, E. Lein, E. Lundberg, M. Uhlen, A. Martinez-Arias, et al. What is your conceptual definition of "cell type" in the context of a mature organism? *Cell Systems*, 4(3): 255–259, 2017. doi: 10.1016/j.cels.2017.03.006.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

J. Eberwine, J.-Y. Sul, T. Bartfai, and J. Kim. The promise of single-cell sequencing. *Nature methods*, 11(1):25–27, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

J. A. Griffiths, A. Scialdone, and J. C. Marioni. Using single-cell genomics to understand developmental processes and cell fate decisions. *Molecular systems biology*, 14(4):e8046, 2018.

S. C. Hicks, F. W. Townes, M. Teng, and R. A. Irizarry. Missing data and technical variability in single-cell rna-sequencing experiments. *Biostatistics*, 19(4):562–578, 2018.

M. Hollander, D. A. Wolfe, and E. Chicken. *Nonparametric statistical methods*, volume 751. John Wiley & Sons, 2013.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

V. Y. Kiselev, T. S. Andrews, and M. Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020.

D. A. Lawson, K. Kessenbrock, R. T. Davis, N. Pervolarakis, and Z. Werb. Tumour heterogeneity and metastasis at single-cell resolution. *Nature cell biology*, 20(12): 1349–1360, 2018.

J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, D. A. El-ad, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, et al. Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197, 2015.

S. Liang, F. Wang, J. Han, and K. Chen. Latent periodic process inference from single-cell rna-seq data. *Nature communications*, 11(1):1–8, 2020.

P. Lin, M. Troup, and J. W. Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology*, 18(1):59, 2017.

S. Liu and C. Trapnell. Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research*, 5, 2016.

S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15:1053–1058, Dec 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0229-2.

M. D. Luecken and F. J. Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019.

A. T. Lun, D. J. McCarthy, and J. C. Marioni. A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Research*, 5, 2016.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

D. J. McCarthy, K. R. Campbell, A. T. Lun, and Q. F. Wills. Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics*, 33(8):1179–1186, 2017.

L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

E. Mereu, A. Lafzi, C. Moutinho, C. Ziegenhain, D. J. McCarthy, A. Álvarez-Varela, E. Batlle, D. Grün, J. K. Lau, S. C. Boutet, et al. Benchmarking single-cell rna-sequencing protocols for cell atlas projects. *Nature Biotechnology*, pages 1–9, 2020.

D. O. Morgan. *The cell cycle: principles of control*. New science press, 2007.

V. Ntranos, G. M. Kamath, J. M. Zhang, L. Pachter, and N. T. David. Fast and accurate single-cell rna-seq analysis by clustering of transcript-compatibility counts. *Genome biology*, 17(1):1–14, 2016.

A. Pratapa, A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods*, 17(2):147–154, 2020.

A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, et al. Science forum: the human cell atlas. *Elife*, 6:e27041, 2017.

W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys. A comparison of single-cell trajectory inference methods. *Nature biotechnology*, 37(5):547–554, 2019.

R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

T. I. Simpson, T. Pratt, J. O. Mason, and D. J. Price. Normal ventral telencephalic expression of pax6 is required for normal development of thalamocortical axons in embryonic mice. *Neural development*, 4(1):19, 2009.

A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. M. III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of single-cell data. *Cell*, 177:1888–1902, 2019. doi: 10.1016/j.cell.2019.05.031. URL https://doi.org/10.1016/j.cell.2019.05.031.

F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, et al. mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377–382, 2009.

T. Tian, J. Wan, Q. Song, and Z. Wei. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4):191–198, 2019.

V. A. Traag, L. Waltman, and N. J. van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

C. A. Vallejos, D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni. Normalizing single-cell rna sequencing data: challenges and opportunities. *Nature methods*, 14 (6):565, 2017.

F. Wolf, P. Angerer, and F. Theis. Scanpy: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19, 12 2018. doi: 10.1186/s13059-017-1382-0.

J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

C. Ziegenhain, B. Vieth, S. Parekh, B. Reinius, A. Guillaumet-Adkins, M. Smets, H. Leonhardt, H. Heyn, I. Hellmann, and W. Enard. Comparative analysis of single-cell rna sequencing methods. *Molecular cell*, 65(4):631–643, 2017.

Appendix A

Additional plots

A.1 Additional plots of mutant datasets

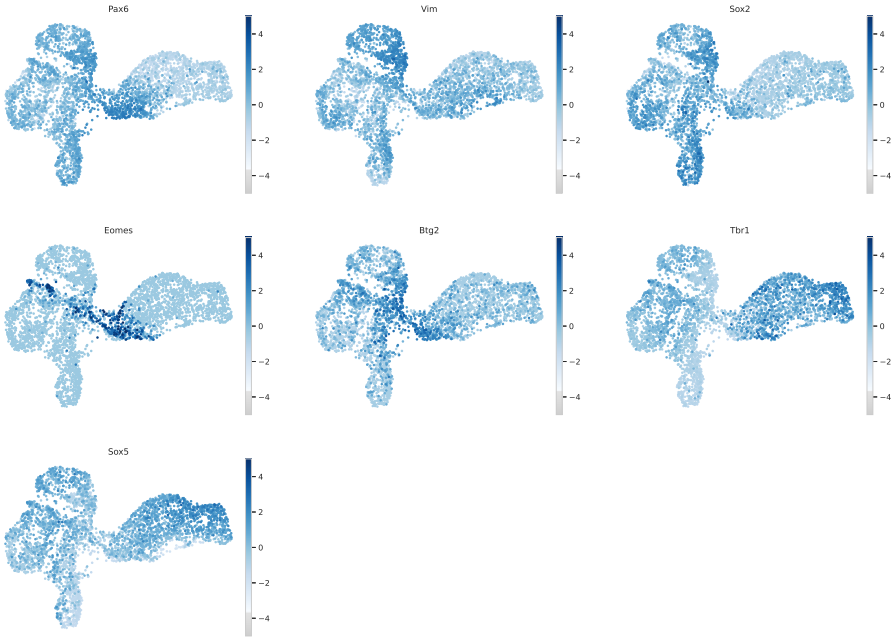


Figure A.1: Marker gene expression of the 3 main cell types for E14_mutant.

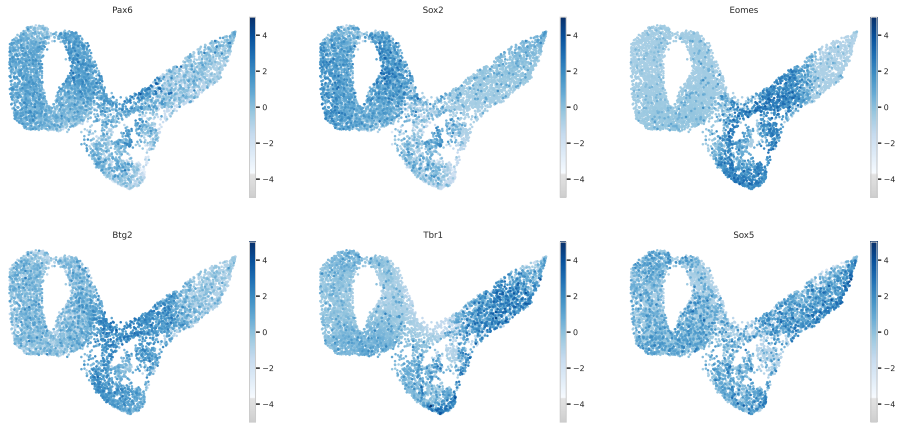


Figure A.2: Marker gene expression of the 3 main cell types for E13_mutant.

A.2 Preprocessing of control datasets

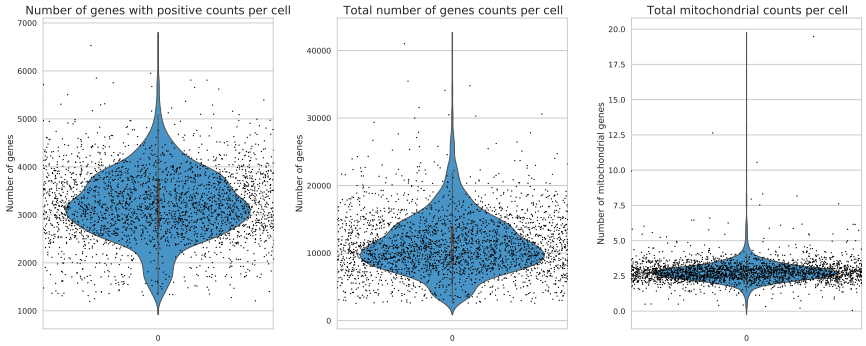


Figure A.3: Quality control measures for E13_control

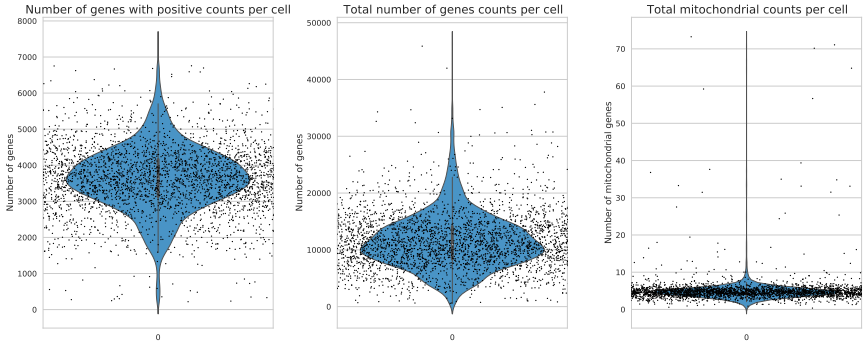


Figure A.4: Quality control measures for E14_control

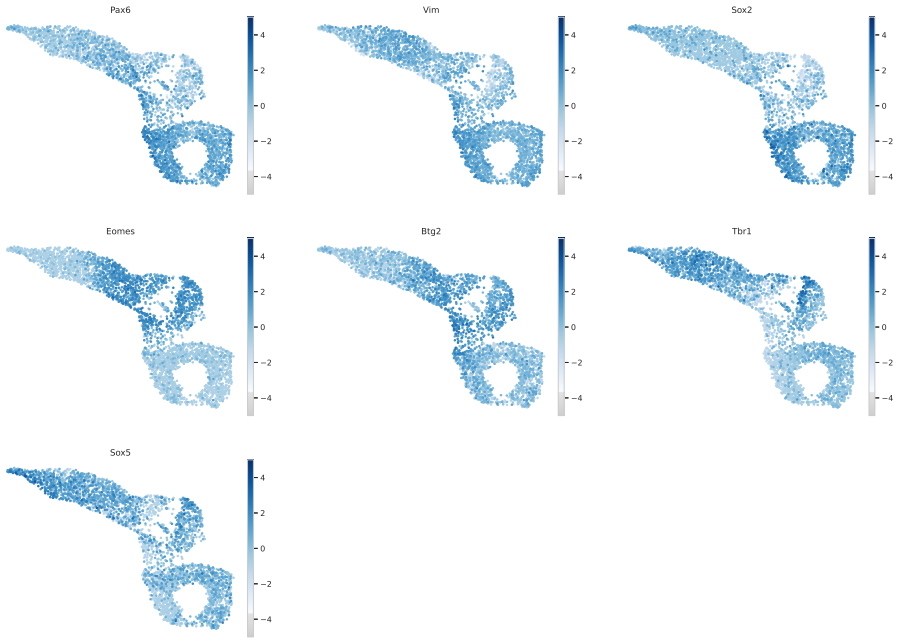


Figure A.5: Marker gene expression of the 3 main cell types for E13_control.

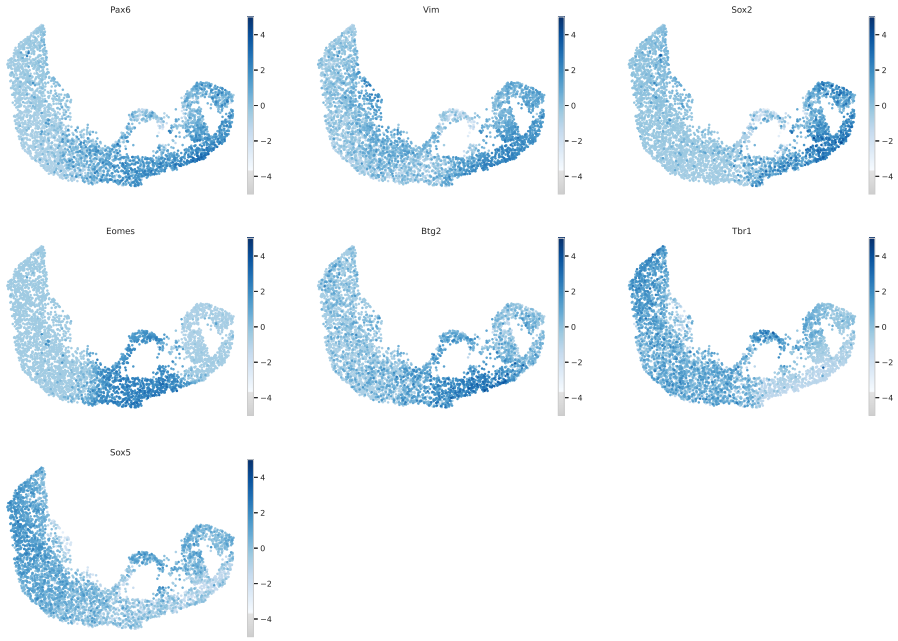
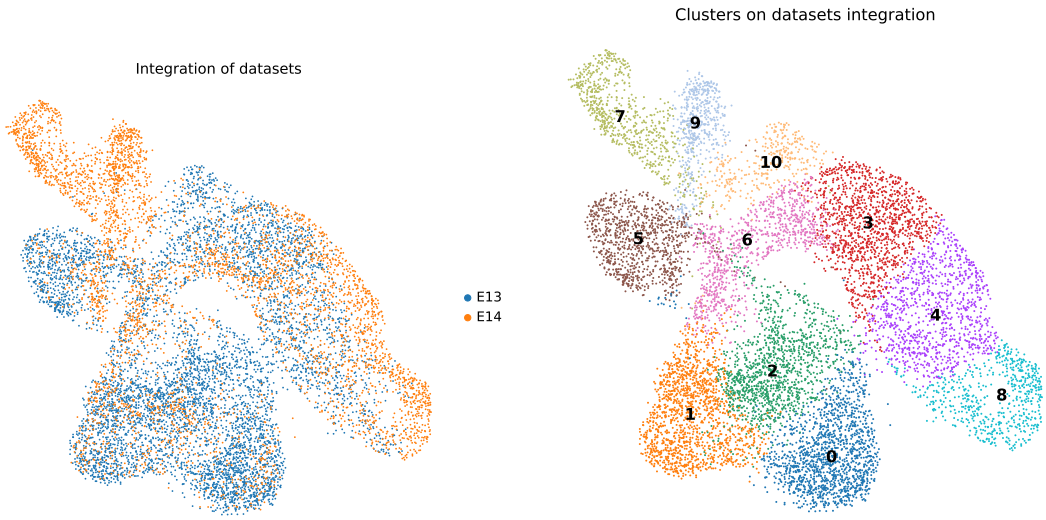


Figure A.6: Marker gene expression of the 3 main cell types for E14_control.

A.3 Datasets Integration



(a) Dataset integration by removing batch effects. (b) Clusters identified on the integrated dataset.

Figure A.7: Integration of the E13_mutant and E14_mutant datasets.

Appendix B

Differentially Expressed Genes (DEGs)

B.1 100 top DE genes on predictions of E13_mutant

	Ectopic	Interm. Progen.	Neural Progen.	Post-mitotic N.	Unknown
0	Rrm2	Neurog2	Gas1	Rtn1	Top2a
1	Pclaf	Hes6	Fabp7	Mllt11	Cdk1
2	Pcna	Gadd45g	Id4	Neurod6	Nusap1
3	Lig1	Btg2	Hes1	Neurod2	Spc25
4	Slbp	Mfng	Sfrp1	Tubb3	Ube2s
5	Mcm6	Sox4	Ptn	Rit1.1	Cks2
6	Tyms	Meg3	Ttyh1	Dcx	Smc2
7	Rpa2	Rprm	Dbi	Elavl3	Ccnb1
8	Dut	Btbd17	Zfp36l1	Rab3a	Smc4
9	Nasp	Chd7	Hmgn3	Rnd2	Birc5
10	Gmnn	Cited2	Pantr1	Stmn3	Ccna2
11	Mcm5	Neurog1	Aldoc	Tagln3	H2afv
12	Tipin	Dll1	Psat1	Sox11	Cdca8
13	Tk1	Plk3	Sox9	Cd24a	Cks1b
14	Dhfr	Pak3	Sox2	Igfbpl1	Prc1
15	Fen1	Elavl4	Mif	Cnr1	Cenpf
16	Clspn	Eomes	Mpped2	Vat1	Hmgn2
17	Hells	Fam110a	Phgdh	Podxl2	Bub3
18	Chaf1b	Rgs16	Pax6	Tbr1	Arl6ip1
19	Uhrf1	Igsf8	Ndrg2	Crmp1	Ube2c

20	Ung	Miat	Nes	Rbfox2	Tuba1b
21	Prim1	Spsb4	Pea15a	Dpysl3	Tpx2
22	Mcm2	Rcor2	Creb5	Nrep	Cdca3
23	Tcf19	Dll3	Ddah1	Nfib	Plk1
24	Dtl	Trp53i11	Mt3	Hist3h2ba	Aurka
25	Ccne1	Lzts1	Qk	Pcp4	Cenpe
26	Mcm7	Prag1	Ccnd2	Ppp1r14a	Pbk
27	Dek	Igfbp2	Siva1	Cotl1	Tacc3
28	E2f1	Mtss1	Mt1	Cdk5r1	Aurkb
29	Rfc3	Dlk1	Plpp3	6330403K07Rik	H2afx
30	Gins2	Mapkapk2	Anp32b	Stmn2	Mis18bp1
31	Rad51	Rnf165	Ppp1r1a	Cdkn1c	Tubb4b
32	Ranbp1	Ppp1r14b	Prdx6	Basp1	Kif22
33	Atad2	Ubxn2a	Jun	Sorbs2	Calm2
34	Mcm3	Atp6v0e	Plagl1	Itm2b	Racgap1
35	Rrm1	Gadd45a	Nr2e1	Aplp1	H1f0
36	Dnajc9	Sstr2	Lfng	Nsg1	Ubald2
37	Ccne2	Insm1	Hes5	Klf7	H1fx
38	Cdt1	Cald1	Serpinh1	Thra	Incenp
39	Hat1	Lrp8	Rcn1	Dbn1	Cdc20
40	Mybl2	Prmt8	Acot1	Celf4	Mki67
41	Rpa1	F2r	Nrarp	Nsg2	Lockd
42	Cenph	Traf4	Cyr61	Celf3	Cdca2
43	Hes6	Tubb3	Mt2	Chd3	Pttg1
44	Cenpk	Srrm4	Ldha	Gdi1	Dbf4
45	Cdc45	Mycl	Pcgf5	Rundc3a	Ckap2l
46	Mms22l	Myo10	Fos	Parp6	Hn1
47	Chaf1a	Numbl	Ccnd1	Bcl11a	Hmmr
48	Meg3	Rhbdl3	Btg1	Map1b	Pimreg
49	Rad54l	Tmem2	Sparc	Nnat	Fbxo5
50	Siva1	Chn2	Dek	Gap43	Cenpa
51	Rad51ap1	Cbfa2t2	Meis2	Islr2	Kif2c
52	Mcm4	Mfap4	Gsta4	Snrpn	Kifc1
53	Fbxo5	Klhl7	Cdca7	Khdrbs2	Rrm2
54	Hist1h1b	Eya2	Hmga2	Mfap4	Spc24

55	Hist1h2ap	Cbfa2t3	H2afv	Map2	Rangap1
56	Anp32b	Htr3a	Cenpa	Atat1	Kif23
57	Fxyd6	Dmrta2	Cdca3	Rufy3	Sapcd2
58	Cdc6	Ebf2	Mest	Bhlhe22	Ccnb2
59	Smc2	Hes5	Tuba1b	Olfm1	Kpna2
60	Cdca7	Tox3	Ranbp1	Gpm6a	Sgo1
61	Mfng	Rai14	Scrn1	Cacna2d1	Pclaf
62	Rnaseh2b	Stat3	Lockd	Nhlh1	Kif20a
63	Pdlim1	Fezf2	Emx2	Stx7	Psrc1
64	Wdr76	Dhrs4	Tagln2	Ttc28	Nde1
65	Shmt1	Chmp1b	Mir670hg	Uchl1	Cenpl
66	Mthfd2	Hpcal1	Erf	Nrn1	Tuba1c
67	Hmgn5	Dll4	Ckb	Kif21b	Mad2l1
68	Gadd45g	Ddit4	Rgma	Myt1l	Knstrn
69	Mdk	Optc	Rest	Nxph4	Cdc25c
70	Csrp2	Kcnq1ot1	B3gat2	Epha5	Nasp
71	Syce2	D030055H07Rik	Rgcc	Ppp2r2b	Ska1
72	Car14	Afap1	Gli3	Ttc9b	Rrm1
73	Spc24	Tmem178	Sox21	Plcb1	Hmgn5
74	Esco2	Uncx	Ccnb2	Id2	Ckap2
75	Neurog2	Fbrsl1	Fam181b	Epha3	Nuf2
76	Tuba1b	Elavl2	Hmgn2	Tmem176b	Mxd3
77	Pbk	Kdm5b	Fgfbp3	Ctxn1	Fzr1
78	Insm1	Igfbpl1	Mycn	Celf2	Knl1
79	Hist1h1e	Rem2	Nde1	Srrm4	Kif20b
80	Btg2	E2f1	Cdca8	Sox4	Kif11
81	E2f7	St18	Cenpm	Apc2	Melk
82	Donson	Phldb2	Kbtbd11	Zbtb18	Ncapg
83	Carhsp1	Abcb9	Cks2	Nfia	Ccnf
84	Ehbp1	Dpysl4	Cdk6	Nav1	Sgol2a
85	Fam111a	Rhcg	Dct	Mien1	Dlgap5
86	Top2a	Abracl	H2afx	Elavl4	Hjurp
87	Ccnd2	Ly6e	Cks1b	Zeb2	Cdc25b
88	Neurog1	Prox1	Mdk	Dlgap4	Ndc80
89	Dlk1	Serping1	Ccna2	Scg3	Aspm

90	Arx	Myt1	Birc5	Wnt7b	Bora
91	Cenpm	Fgd4	Knstrn	Kif5c	Rnf26.1
92	Wfdc2	Map1a	Cdc20	L1cam	Dmrta2
93	Phgdh	RF01962	Cenpf	Lmo4	Kif15
94	Mad2l1	Nrp1	Dnajc9	Npdc1	Dnajc9
95	Hist1h2ae	Ckb	Smc4	Sox5	Cdca5
96	Zfp367	Cdc25b	Tubb4b	Rbfox3	Pak3
97	Jun	Tecpr1	Shisa2	Ppp3ca	Cenph
98	Rmi2	Fbxl20	Pou3f3	Neurod1	Hes6
99	Lpar1	Pgap1	Rnf26.1	Gng3	Troap