Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Statistics and Machine Learning
2020 | LIU-IDA/STAT-A--20/005--SE

Utilizing unlabeled data in cell type identification

– A semi-supervised learning approach to classification

Thijs Quast (thiqu264)

Supervisor : Oleg Sysoev
Examiner : Krzysztof Bartoszek

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

Recent research in bioinformatics has presented multiple cell type identification methdologies using single cell RNA sequence data (scRNA-seq). However, a consensus on which cell typing methodology consistently demonstrates superior performance remains absent. Additionally, very few studies approach cell type identification through a semi-supervised learning study, whereby the information in unlabeled data is leveraged to train an enhanced classifier. This paper presents cell annotation methodologies through self-learning and graph-based semi-supervised learning, in both raw count scRNA-seq data as well as in a latent embedding. I find that a self-learning framework enhances performance compared to a solely supervised learning classifier. Additionally, modelling on the latent data representations consistently outperforms modelling on the original data. The results show an overall accuracy of 96.12%, whereas additional models achieve an average precision rate of 95.12% and an average recall rate of 94.40%. The semi-supervised learning approaches in this thesis compare favourable to scANVI in terms of accuracy, average precision rate, average recall rate and average f1-score. Moreover, results for alternative scenarios, in which cell types among training and test data do not perfectly overlap, are reported in this thesis.

Acknowledgments

Oleg Sysoev, thank you for your great supervision and close involvement during this thesis. Throughout the entire process you have often expressed your appreciation for the work that I was doing. As a student, such recognition is very much appreciated and it is a great motivational factor. Thank you very much for this.

Professor Mikael Benson, thank you for including me in your research group from the start of this thesis. As a student, being included in an academic research group while writing a thesis, gives one a feeling of recognition that the work conducted during a thesis is appreciated and relevant for the academic community. Thank you.

Krzysztof Bartoszek, thank you for your constructive and concise feedback on my work. Presenting a thesis to an examiner to some extent is, and will always be, accompanied by some sense of nervousness. During feedback sessions you have always expressed your appreciation for my thesis and I have perceived all your feedback as being very constructive. Many thanks.

Martin Smelik, thank you for your very detailed and thorough revision of my thesis. It is very much appreciated that a fellow student puts so much effort into a thesis revision as you did. I believe your efforts and comments with respect to the topic at hand, as well as your profound statistical and mathematical knowledge, have greatly improved this thesis. Thank you.

A special thanks to everyone at the CPMed research group. Thank you all for your help and involvement during my thesis project.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Among the greatest challenges faced in current health care systems is the failure of patients to respond to drug treatment [38]. According to previous studies [24], it is probable that the absence of desired drug effects is due to the complex interaction of thousands of genes among a wide range of cell types. Therefore, in order to improve drug treatment, it is of severe importance to determine which cells and which genes are essential to target during treatment. However, in order to assess the importance of different genes and cell types in drug treatment, one must first be able to accurately classify cells, which leads to the scope of this thesis.

As research shows that different cell types show differences in specific gene expression levels [12], the purpose of this thesis is to develop a semi-supervised classification model which is capable of accurately identifying cell types across distinct datasets based on gene expression levels. A wide range of studies that classify cell types is already present, which are discussed in detail in section 1.2.

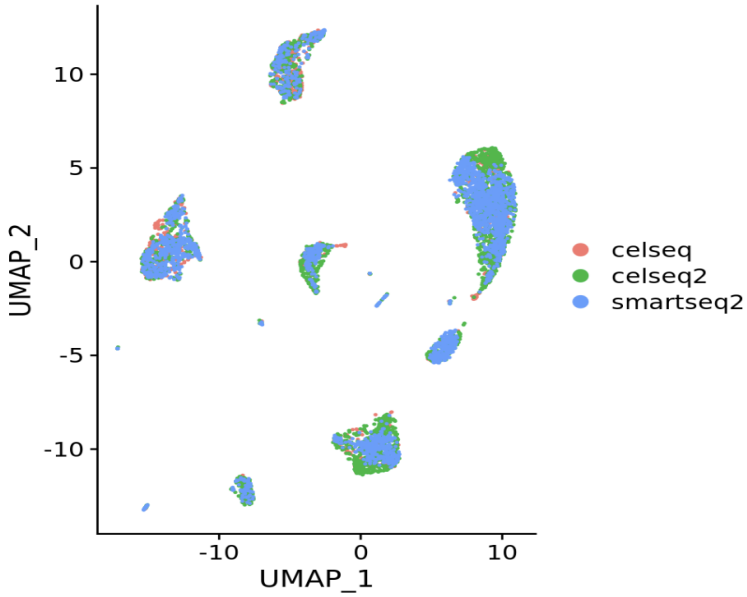
Nevertheless, there are very few studies that leverage the information in unlabeled data in order to develop a cell type classifier. Recently one of such methods was proposed by Xu et al., [39], who approach cell type identification through a semi-supervised learning framework. In their research, Xu et al., [39] transfer data into a latent space while referring to available labels. Subsequently, posterior probabilities are obtained for every cell belonging to each of the available labels.

This thesis differs from the work by Xu et al. [39], as more conventional semi-supervised learning approaches such as self-learning [6] and Graph-Based methods [6] are considered. Assessing whether conventional semi-supervised learning models outperform the methodology proposed by Xu et al. [39], is therefore an important contribution in cell type identification research as well as in semi-supervised learning [6].

Semi-supervised learning implementations are ideal when one has access to a large amount of unlabeled data in combination with the availability of fewer labeled data points [6]. Specifically, one aims to leverage information in the unlabeled data [6] in order to train an enhanced classifier, which would ideally perform superior to a solely supervised classifier. As accurately labeled data is scarce in bioinformatics, and the costs of manually labelling data are high [40], it is self-evident that a semi-supervised learning approach to cell typing is

desired. The aim of this thesis is to leverage semi-supervised learning in order to improve decision boundaries between clusters of cell types that would have been deficient when based on few labeled instances only. The presumption that semi-supervised learning is a reasonable alternative in cell typing is based on previous work by Stuart et al. [29], who show that when integrating multiple scRNA-seq datasets in a coherent latent space, observations from multiple datasets complement each other in determining decision boundaries among cell types, see figure 1.1 [29].

Figure 1.1: Expansion of cluster boundaries



In figure 1.1, Uniform Manifold Approximation and Projection (UMAP) is a learning technique that enables the visual representation of highly dimensional data [18], celseq, celseq2 and smartseq2 are scRNA-seq datasets [29]. Although semi-supervised learning methodologies on scRNA-seq data have not extensively been implemented, this specific type of learning has proven to be valuable in other domains. Nigam et al. [21] for example, find that leveraging unlabelled data improves classification accuracy in a text document classification setting. Also, Erman et al. [8] show an increase in precision rates when using semi-supervised methodologies in a network traffic classification problem.

As the raw scRNA-seq count data is subject to high dimensionality and a large number of zeros, it is expected that modelling the original representation of the data will not return optimal results. Therefore, besides modelling on original data, the gene expressions are transferred into a latent space by using recently published autoencoder implementations [15]. This autoencoder reduces dimensionality and removes noise. It is therefore expected that this representation of the data will lead to superior cell type identification results. This presumption is supported by findings by Eraslan, et al. [7] who show an improvement on scRNA-seq analyses data subsequent to noise removal by using an autoencoder.

This thesis is structured as follows: the remainder of section 1 describes present literature in the field of studies, followed by the specific aim of this research paper. Section 2 thoroughly describes which data is studied and how the datasets are obtained. Section 3 discusses the methodologies applied. Results are discussed in section 4 and, a general discussion of the research paper is presented in section 5. The thesis work is concluded in section 6.

1.2 Literature review

Recent research on cell typing using scRNA-seq data has led to many different approaches in cell type classification. Abdelaal et al., [1] provide an extensive comparison study across 22 different cell typing classifiers and find substantial differences in performance among classifiers, leaving numerous research opportunities in cell classification performance.

In this section, I discuss recently published papers, which approach the cell typing problem from different angles. Stuart et al., [29] classify cells by identifying and leveraging similarity among individual cells across different datasets. Correspondence between pairs of cells is referred to as "anchors" and is used to integrate datasets into a shared latent space. Similarities, based on the distance between cells in the original data representation of distinct datasets, can be used to accurately identify and classify cell types due to this similarity extraction and dimensionality reduction. Contrary to Stuart et al. [29], Kiselev et al., [13] focus on dimensionality reduction by extracting particular genes that are most important in underlying biological differences among cell types. Classification predictions are then made based on the relevant expressed genes only.

Alquicira-Hernandez et al. [2] classify cells according to a specific form of feature selection in latent space. Specifically, the methodology starts by extracting principal components from the training data. Subsequently, only principal components that explain a certain variation in the data remain, and a classification model is trained on this selection of principal components. In order to make predictions, the test data is projected into the same principal component space and predictions are made by the model developed on the training data.

The approach of this thesis differs from previously mentioned methodologies as well as the work conducted by Xu et al., [39] in the following ways: (1) I approach semi-supervised learning through self-learning as well as Graph-Based methods. (2) Self-learning is conducted on the latent representation as well as the original data. (3) I compare semi-supervised learning to supervised learning both for the original data as well as in latent space.

Previous work on dimensionality reduction includes Risso et al. [25], who focus on modelling over representation of zeros in gene expressions through so called zero-inflated negative binomial distributions (ZINB) [20] of the genes, which is elaborated upon in section 2. Comparably, Pierson et al., [23] model the high presence of zeros through a zero-inflated factor analysis. The method is related to Principal Component Analysis dimensionality reduction, however it focuses on correlation rather than covariance. And recently, Amodio et al., [3] presented the usefulness of deep learning and autoencoders developing latent representations of scRNA-seq data.

1.3 Aim

Naturally, further research on the implementation of machine learning methodologies in bioinformatics is welcomed. Although research opportunities in both scRNA-seq data as well as in semi-supervised learning methods are endless, this thesis aims to answer the following four research questions:

1. Which semi-supervised learning methods are suitable in cell typing?
2. Do the semi-supervised learning methodologies implemented in this thesis perform favourably compared to existing cell typing methodologies?
3. Does semi-supervised learning outperform supervised learning in cell typing?
4. Is noise removal and dimensionality reduction beneficial in cell typing?

The datasets used in this thesis express raw count data on gene expression levels across a variation of cell types.

2 Data

Section 2 discusses the data used in this thesis. An overview and summary statistics on the training and test datasets is provided in section 2.1. Additionally, preprocessing techniques are elaborated upon in section 2.2. Gene expression levels follow a Zero Inflated Negative Binomial distribution (ZINB), which is described in section 2.3. ScRNA-seq data is subject to several very specific characteristics, which are discussed in subsection 2.4. As common practice in statistics, models are trained on the training data and evaluated on the test data. As the scope of this thesis is biological, one can refer to the training data as reference data and test data as query data respectively.

2.1 Acquiring datasets

From *Hemberg Group's* GitHub page[1], two datasets from distinct mouse retina tissues are obtained. Specifically, the training dataset comes from the research conducted by Shekhar et al. [28] whereas the test dataset comes from the work by Macosko et al. [17].

Subsequently, using the *SingleCellExperiment* [16] package through the *BiocManager* [19] installation in *R* [30], the raw count data of gene expressions per individual cell are extracted from the *.rds* files. The raw count matrices are afterwards merged with the matching cell labels, which are downloaded from the same source. Resulting from this, two datasets of raw scRNA-seq count data with matching labels are created and ready to be analyzed. In order to assess performance of classification models, labels from the Macosko dataset are removed.

2.2 Data description

The training dataset consists of a total 6,950 cells and 13,166 genes, whereas the test dataset consists of 44,808 cells and 23,288 genes. In order to build a classification model, overlapping cell types and genes from both datasets are extracted.

Previously mentioned training and test datasets overlap in 12,333 genes, dispersed over 5 common cell types: Amacrine, Bipolar, Cones, Muller and Rods. In order to speed up computation times and avoid dealing with a highly imbalanced training dataset, Bipolar cells are downsampled to 2945 observations. Resulting in a final training dataset consisting of

[1] https://hemberg-lab.github.io/scRNA.seq.datasets/mouse/retina/

Table 2.1: Training data

Cell type	# Observations
Amacrine	252
Bipolar	2945
Cones	48
Muller	2945
Rods	91
Total	6281

Table 2.2: Test data

Cell type	# Observations
Amacrine	4426
Bipolar	6285
Cones	1868
Muller	1624
Rods	29400
Total	43603

6,281 cells and 12,333 genes, shown in table 2.1. Also, the modified test dataset consists of a total of 43,603 cells and the same 12,333 genes as the training dataset. A numerical overview of the test dataset is shown in table 2.2.

A representation of a few sampled cells from the training data is shown in table 2.3. As can be seen in table 2.3, zeros are over represented in the data. Specifically, 93.4% of values in the dataset are zero reads. A zero, however, does not necessarily mean that the gene is not expressed in the cell, it could have been missed during measurements. Considering over

Table 2.3: Sample of cells and gene expression from training data

Observation	Gene #1	Gene ...	Gene # 12333	Cell type
1	0	...	0	Bipolar
2	0	...	2	Amacrine
3	1	...	0	Rods
4	0		0	Rods
...	...	...	...	...
...	...	...	...	...
6281	0	...	1	Rods

representation of zeros, gene expressions are considered to follow a so called Zero Inflated Negative Binomial (ZINB) distribution [15]. Specific applications on modelling the ZINB distribution are elaborately discussed in section 3. Because of over represented zeros in the data, many genes are expressed in only very few cells. Meaning that many expression levels of individual genes contain very little information to be used in the classifier. The distribution of gene expressions among number of cells in the training data is shown in figure 2.1 below.

Figure 2.1: Genes expressed over cells

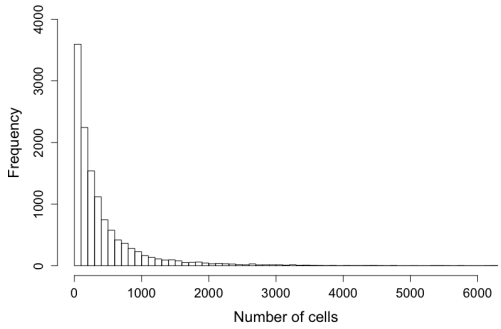


Figure 2.2: Genes expressed per cell

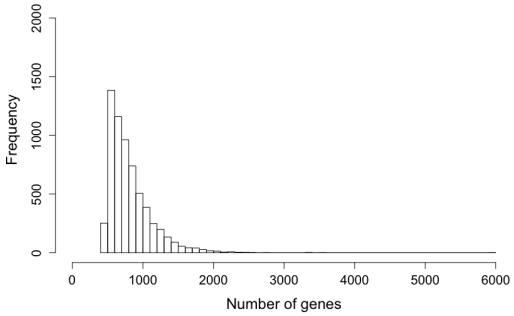


Figure 2.1 shows that the vast majority of genes express itself in less than 1,000 out of a total of 6,281 cells. Resulting from this, by computing the arithmetic mean, on average a gene expresses itself in only 414 out of 6,281 cells. Related to this are gene expressions within the

individual cells. Figure 2.2 shows that a cell on average, based on the arithmetic mean, shows gene expressions from 813 out of 12,333 genes. As models in this thesis are computed on the original data as well as in a latent space, the large number of zero reads in the data is solely challenging when working with the original data representation, which is only part of the methodological framework of this thesis. When modelling in latent space, the zero reads are resolved and no longer problematic.

2.3 Zero Inflated Negative Binomial Distribution

In bioinformatics, through transcriptome sequencing one can analyze cells and represent gene expressions levels within such cells numerically. One of the aims of transcriptome sequencing is to find so called differentially expressed (DE) genes. These are genes which show different levels of expressions under distinct conditions [33], such as a patient having a certain disease or not. Despite great improvements in transcriptome sequencing technologies, current obtained scRNA-seq data is still highly subject to dropout events. Meaning the scRNA-seq data contains a substantial amount of zero reads [33].

Zero reads in scRNA-seq data can occur because of two reasons. (1) Biological zeros, meaning that a specific gene is simply not expressed in a particular cells. Which means the zero read is as accurate representation of the gene expression level. (2) Technical zeros, meaning that a gene is expressed in the cell but not detected by the sequencing hardware [33].

Due to the high presence of zeros in the gene expression data, one can model gene expression levels according to a Zero Inflated Negative Binomial (ZINB) distribution, which is an expansion of a regular Negative Binomial count distribution, that accounts for the excess of zeros being present. The density function for the ZINB distribution is provided below.

$$f_{ZINB}(y|\mu,\theta,\pi) = \pi\delta_0(y) + (1-\pi)f_{NB}(y|\mu,\theta), \tag{2.1}$$

π is the mixture component of the distribution, which refers to the probability of zero inflation, and takes on a value between 0 and 1. $\delta_0(y)$ is equal to ∞ when y=0 and 0 otherwise. f_{NB} is the density function of a negative binomial distribution with mean paramter μ and disperion parameter θ.

Figure 2.3: Individual gene expression level

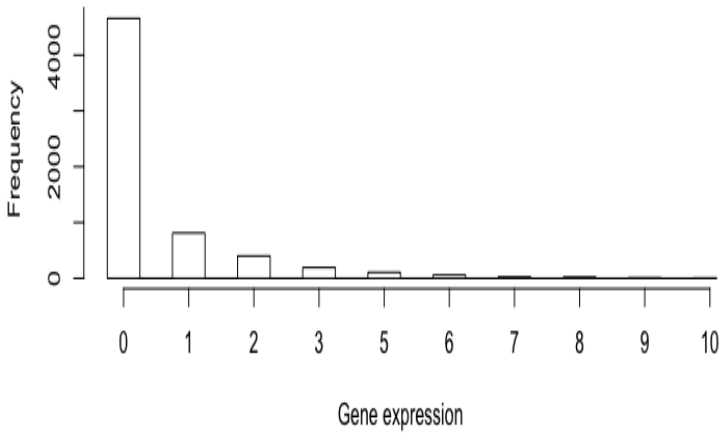


Figure 2.3 shows the raw gene expression levels from a sampled gene from the training data. Clearly, zeros are highly represented in the data, whereas the frequency of counts seem to decrease as the expression level increases, meaning that modelling raw gene expression levels according to the ZINB distribution seems reasonable.

Whether developing models and predicting on on raw gene expression data is reasonable shall be concluded after implementing the proposed methods in section 3 on the raw count data. Where one might argue that little explanatory power is present in such noisy raw count data, on the other hand, still several gene expression levels are observed which can result in reasonable models. Also, having a benchmark on original data makes it possible to determine the enhanced performance of the proposed scVI autoencoder presented in section 3.

2.4 Single Cell RNA sequence data

Besides high dimensionality and the large number of dropouts, represented by zero reads, interpreting raw scRNA-seq data is troublesome as it is subjected to the biological batch effect of experiments.

A consensus on the exact definition of the biological batch effect remains absent in current literature. However, one can attribute this phenomenon to the fact that different experiments inevitably lead to different measurement results due to experimental variations, specifically because of minor differences in time and the setting of the location. Thus, one can impossibly perfectly replicate experimental environments and this will inevitably lead to some uncontrollable measurement errors which are irrelevant of biological differences [14].

In order to model scRNA-seq data properly, some form of dimensionality reduction is desired. However, as the datasets are subject to high dimensionality, but also to over representation of zero reads and the batch effect, regular dimensionality reduction tools such as Principal Component Analysis and Linear Discriminant Analysis do not suffice.

Alternatively, the gene expression data is transformed into latent space by means of an autoencoder. Lopez et al., [15] recently published an open source *Python* [35] library, called scVI [15], that removes noise and reduces dimensionality for gene expression data. Additionally, this tool accounts for the batch effect by successfully harmonizing datasets obtained from distinct experiments and thereby taking previously mentioned difficulties with biological data from different experiments into account. A detailed explanation on the implemented dimensionality reduction tool is presented in section 3. Results from dimensionality reduction- and noise removal steps are presented in section 4.

3 Method

This section describes the methodologies implemented in this thesis. 3.1 discusses a supervised learning approach through a grouped LASSO regression. Cross-validation is briefly discussed upon in 3.2. Additionally, 3.3 describes how a grouped LASSO regression is used in a semi-supervised self-learning framework. Transferring the data in a latent representation is discussed in section 3.4, followed by the discussion of Graph-Based semi-supervised learning methods in 3.5. The benchmark methodology scANVI is discussed in section 3.6, whereas additional scenarios are considered in 3.7. In order to handle an imbalanced classification problem, extensive evaluation criteria are required, which are discussed in detail in section 3.8.

When working on a classification problem, it is essential to compare performance of different models. Especially, to assess whether more sophisticated models outperform a rather simplistic model, the presence of a benchmark model is important. In this thesis the benchmark model is a supervised grouped LASSO regression. As scRNA-seq data consists of high dimensionality and zeros are over represented, implementing a grouped LASSO regression allows for both regularization as well as variable selection [31]. Starting with the results of this model, several more enhanced methods are implemented, aiming to outperform benchmark classification performance.

Additionally, in order to consider the developed methodologies in an academic context, results are compared to the single-cell ANnotation Variational Inference (scANVI), which is a recently published semi-supervised implementation to cell type identification [39].

3.1 Grouped LASSO Regression

When handling high dimensionality of data, an Ordinary Least Squares (OLS) framework may not always produce desirable results, nor may it be applicable when dimensionality exceeds the number of observations [37]. Firstly, the common bias-variance tradeoff seems to be present, in which OLS models produce low bias, but high variance in the predictions. Reducing model complexity by reducing the impact of (or setting equal to zero) of several coefficients decreases variance in the predictions on previously unseen data.

Secondly, when dealing with excessive dimensionality, as is the case with scRNA-seq data, determining coefficients for each of the explanatory variables creates a model that is hardly

interpretable. Instead, determining a subset of variables that show to have most impact on the variable to be predicted seems feasible [31].

As this thesis aims at optimizing a classification problem, the proposed methods are written in the logistic regression format, as this enables the model to produce probabilities for each of the potential classes. Specifically, as this classification problem consists of five classes, a multinomial logistic model, which is regularized through a grouped LASSO regression, is desirable.

Considering a multinomial model, one can write the response variable as G in equation 3.1, where M denotes the number of distinct classes. The goal of the grouped LASSO regression is to provide a mapping from the matrix of explanatory variables x to the dependent variable y, which in this case is represented by the different classes.

$$G = \{1, 2, ..., M\}, \tag{3.1}$$

where in the scope of this thesis, except for the setting described in section 4.9, the data consists of five cell types and thus M is equal to five. Resulting from this, class probabilities can be computed accordingly:

$$Pr(G = m|X = x) = \frac{e^{\beta_{0m}+\beta_m^T x}}{\sum_{l=1}^{M} e^{\beta_{0l}+\beta_l^T x}} \tag{3.2}$$

As a regularization component is desired, in the optimization, the model optimizes over the penalized negative log-likelihood function defined as:

$$L = -\left[\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{m=1}^{M} y_{il}(\beta_{0m} + x_i^T\beta_m) - log(\sum_{m=1}^{M} e^{\beta_{0m}+x_i^T\beta_m})\right)\right] + \lambda\left[\alpha\sum_{j=1}^{p}||\beta_j||_2\right], \tag{3.3}$$

β is a matrix of coefficients of dimensionality $p \times M$, where p is the number of groups in the model and M the range of classes in the dependent variable. β_m denotes to which outcome category m is referred to, and β_j refers to row j which is a vector of length M with coefficients for variable j. Y is a matrix of dimensions $N \times M$, in which N is the total number of observations and Y indicates to which class the observation belongs [11].

$\lambda > 0$ defines the regularization parameter of the LASSO regression, whereby higher λ values indicate stronger regularization of the model. For every group of explanatory variables in the model, the λ parameter penalizes by shrinking the coefficient. Considering equation 3.3, the lambda parameter shrinks positive and negative β coefficients to zero. Therefore, a group of variables is only beneficial and included in the model if additional performance of this variable outweighs the penalty factor lambda [10]. Subsequently, λ is estimated using a cross-validation framework, which is elaborated upon in section 3.2.

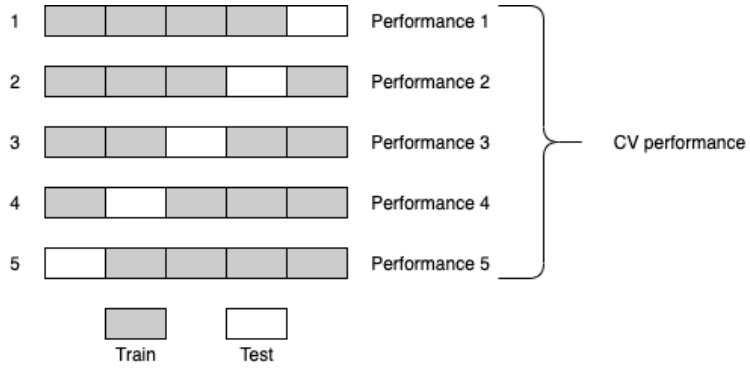
Although at first, implementing a regular LASSO regression seems obvious, the grouped LASSO implementation has several advantages over the conventional LASSO regression model. Firstly, grouped LASSO implementations are useful in the bioinformatics domain, as algorithms as such have successfully identified relevant genes [41]. Also, given the biological context, several genes might have similar biological characteristics, it is therefore more realistic to model such genes as a group of variables [10].

Besides advantages on the modelling part of the algorithms, the grouped LASSO has a great advantage over the regular LASSO regression in terms of computational complexity. Where the cross-validation on the training data of this thesis for a regular LASSO regression takes approximately 32 hours, the same optimization requires only 1.5 hours for the grouped LASSO algorithm. Although running times for the regular LASSO regression might first still seem feasible, when dealing with biological datasets which vastly exceed the number of observations in the current training data, an optimization procedure as such is simply no longer realistic. Grouped LASSO regressions are performed using the *glmnet* [9] package in *R* [30].

3.2 Cross-validation

In order to estimate performance of a developed model and to find optimal parameter values, one can implement cross-validation, which is especially useful when availability of data is scarce. Specifically, k-fold cross-validation divides the (training) dataset into k folds of approximately equal size. [10].

Figure 3.1: Cross-validation, when k=5



Afterwards the model is trained on k-1 folds and tested on fold k. This is done k times in which the test fold is different every time. Finally, the average cross-validation is computed according to averaging out over the performance of all k models. Naturally, optimal model parameters are found through minimizing the cross-validation error term [10]. A graphical representation of cross-validation is presented in figure 3.1.

3.3 Self-learning

In conventional machine learning settings, it is common to solve applied problems by *supervised* or *unsupervised* methods. Where in the first case one has access to labelled data and aims to find a relation between the explanatory and dependent variables in the model, and in the latter case one has the goal of finding relevant, and previously unknown, structures in unlabeled data [6].

Table 3.1: Predicted probabilities per observation in the test data

Observation	c_1	c_2	c_3	c_4	c_5
1	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
...	...	...	...	...	...
...	...	...	...	...	...
n	p_{n1}	p_{n2}	p_{n3}	p_{n4}	p_{n5}

Unsurprisingly, the *semi-supervised learning* methodology is a compromise between both *supervised-* and *unsupervised learning*, in which part of the available data is labeled whereas for the remaining part of the data, labels are absent [6]. In a cell typing scenario, previous research [29] has shown that when different datasets of cell types are mapped in the same space, decision boundaries of cell clusters complement each other, as shown in figure 1.1. Therefore, when one has a labelled training set and an unlabelled test set, in this case leveraging information from the unlabeled test set is reasonable.

Perhaps the most straightforward implementation of a semi-supervised learning approach is through so called *self-learning*. In this methodology a classifier is trained on the labeled data, and as in any supervised approach the classifier predicts on the unlabeled data according to the trained model. The model returns a n x c matrix of probabilities where n

is the number of instances in the test data and c is the number of classes an observation can belong to, see table 3.1.

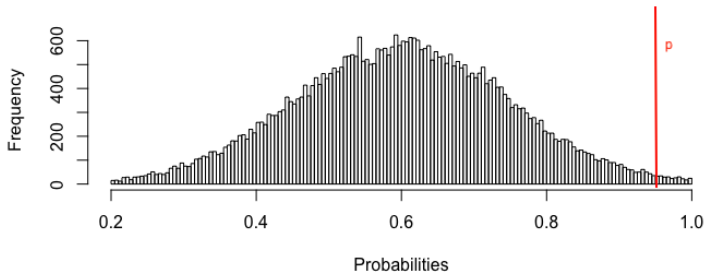
Naturally row wise summations of table 3.1 must return 1 for each row, meaning that for each prediction, the sum of the distribution of predicted probabilities over all classes is equal to 1.

Algorithm 1 Extract maximum predicted probability per prediction

1. m = [ ]
2. **for** j in 1:n **do**
3. $p_j = [p_{j1}, p_{j2}, p_{j3}, p_{j4}, p_{j5}]$
4. m[j] = max(p_j)
5. **end for**
6. Return m

Subsequently, in *self-learning*, the most confident predictions are added to the training data sequentially. By doing this, one thus assumes that the predicted classes for these predictions can be considered to be the true classes. Afterwards, the model is retrained on the appended training data, which means that, the classification model uses its own predictions to learn from [43]. An algorithmic representation of a self-learning framework is presented in algorithm 2. Extracting maximum probabilities is done according to algorithm 1. Additionally, figure 3.2 shows an example of how a distribution of maximum predicted probabilities, for a scenario of 5 classes in the dependent variable, can look. In this example a confidence threshold of 0.95 is chosen.

Figure 3.2: Predicted probabilities



One can iterate self-learning until certain conditions specified beforehand are met, such as a number of maximal iterations, or iterate until no more predictions are above the confidence threshold to be added to the training data.

Self-learning has proven to be effective, as for example, Roli et al. [26], find that using a self-learning framework significantly improves performance in an image recognition problem [26]. Also, the authors justify the use of this methodology because of its straightforward implementation and interpretation. In addition to image recognition, self-learning is widely used in natural language processing. Wang et al. [36], for example, use self-learning methods in extracting subjective information from sentences.

As it is fairly straightforward to obtain a model's confidence in predictions, it is much more challenging to determine which confidence threshold to select. Unfortunately, current literature provides no consensus on which level of probability should be chosen as a

Algorithm 2 Self-learning framework

- labeled: ($X_{1:l}$, $Y_{1:l}$), unlabeled: ($X_{l+1:n}$), max iterations: I, confidence threshold:P
- Supervised learning model g : X –> Y
- Assume for each prediction j: $max(p_j)$ > P is correct prediction

1. **while** i < I & | $max(p_j) > P$ for any j **do**
2. Train g on ($X_{1:l}$, $Y_{1:l}$)
3. Compute p_j for each j
4. Predict classes for $x_j \in (X_{l+1:n})$, according to $max(p_j)$
5. Append (x, $g(x)$) to ($X_{1:l}$, $Y_{1:l}$) where $max(p_j)$ > P
6. Return g
7. **end while**
8. Compute final predictions using g

confidence threshold. The choice of confidence threshold is a tradeoff between a high false-positive rate (fp), which means the model predicts an instance to belong to a certain class but is wrong and a low true-positive rate, which means the model is correct in its predictions. Naturally, a low confidence threshold results in an increase in the false-positive rate, whereas a high threshold results in too few predictions are appended to the training data to make a difference [4].

3.4 Latent representation

As the dimensionality of scRNA-seq data is high, and zeros are over represented in gene expressions, significantly reducing dimensionality of the raw count data seems reasonable. It is desired that the reduced representation of the data contains most, if not all, of the essential information in the original dataset and can be used to develop models [34]. One of the most common dimensionality reduction tools as such is the Principal Component Analysis (PCA). When working with correlated explanatory variables, PCA aims to extract essential information from the data and map this information to a new collection of statistically independent variables commonly defined as principal components [34]. When dealing with non-linear and sparse data such as scRNA-seq data, conventional dimensionality reduction tools such as PCA do however not suffice [34].

Alternatively, Lopez et al., propose *single-cell variational inference* (*scVI*), a deep autoencoder framework that maps scRNA-seq data into a latent representation [15]. The model works as following.

Each gene expression g per cell n is represented as x_{ng}, and is preferably accompanied with a matching batch indicator. In the encoding step, by using four different Neural Networks the raw count data is mapped to a latent representation z_n based on posterior distribution q.

$$q(z_n log(l_n)|x_n, s_n) \tag{3.4}$$

Where, l_n is random variable from a one dimensional Gaussian distribution which considers the presence of noise in the data together with differences in the number of detected genes per experiment. Furthermore, z_n is a multivariate Normal distribution of dimensionality ten, which represents remaining differences between batches indicated by s_n [15].

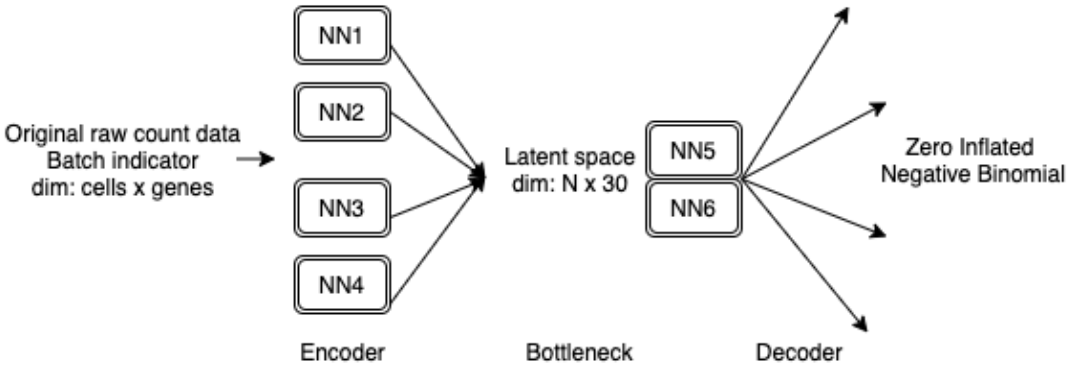
Computing the posterior distribution using Bayes' rule can be challenging as the marginal distribution in the denominator can be difficult to compute. Therefore, through variational inference [32] an approximation of the posterior distribution q is computed.

Afterwards, in the decoding step, through two additional neural networks, all points in the latent representation are transferred to match the parameters of the zero-inflated negative binomial distribution (ZINB), which is written as:

$$p(x_{ng}|z_n, s_n, l_n) \tag{3.5}$$

The essential step in this process is the representation of the data between the encoding and the decoding step. Here, each observation in the data is compressed to 30 dimensions, resulting in a matrix where the number of rows is equal to the number of observations in the original data representation and the number of columns is equal to 30. This newly obtained matrix represents the original data in latent dimensionality. A simplified graphical representation of the scVI framework is shown in figure 3.3.

Figure 3.3: scVI simplified visual representation



Important to mention is that datasets from all batches are first combined using the *GeneExpression* function in scVI. This function merges multiple datasets and overlapping genes from both datasets are retained. After this, the autoencoder is implemented on the fully appended dataset. Finally, known labels are matched to the relevant observations in latent dimensionality.

3.5 Semi-supervised learning through Graph-Based methods

A common approach to semi-supervised learning is through Graph-Based methods. This methodology is based on the idea of building a graph of all observations (labeled and unlabeled), where the observations are nodes of the graph and similarities between observations are represented by the edges between nodes. Subsequently, information from the labeled nodes is used in order to label unlabeled observations [6].

After noise removal and transferring the data into latent space, two Graph-Based semi-supervised learning methods are implemented: *Label propagation* and *Label spreading*, which are both described in detail below.

3.5.1 Label propagation

Zhu and Ghahramani, [44] propose Graph-Based semi-supervised learning through an algorithm which they define as label propagation. The main idea of the proposed algorithm is that in a Graph-Based setting, labeled data propagate labels on unlabeled data based on similarity of the nodes in the graph.

Consider $(x_1, y_1)...(x_l, y_l)$ to be the labeled observations where the labels belong to one of the available known classes. Additionally, $x_u...x_n$ is unlabeled.

Label propagation creates a graph consisting of all labeled as well as unlabeled observations. Precisely, nodes which are closer to each other get higher similarity weights, which are calculated as:

$$w_{i,j} = e^{-\frac{\sum_{d=1}^{D}(x_i^d - x_j^d)^2}{\sigma^2}} \tag{3.6}$$

In equation 3.1, d stands for the Euclidean distance between observations x_i and x_j and σ is a flexible parameter which defines to what extend d influences w. Worth to mention is that σ can vary among dimensions of x.

Subsequently labeled instances propagate information to the unlabeled instances through the edges of the graph. Resulting from this, the nodes acqurie so called *soft labels* which are probabilistic distributions over the labels. As nodes which are more similar should obtain information through propagation more easily, the following transition matrix is defined:

$$T_{i,j} = \frac{w_{i,j}}{\sum_{k=1}^{l+u} w_{kj}} \tag{3.7}$$

Whererby, T_{ij} represents the probability of moving from node i to node j, and l and u are the labelled and unlabelled observations respectively. Dimensionality of matrix T is $(l + u)$ x $(l + u)$. Additionally, in order to label nodes, the matrix Y is constructed, which stores the *soft labels* computed earlier, and is of dimensionality $(l + u) \times C$.

Row i of matrix Y defines the probabilities of node y_i belonging to each of the available classes C. Labels are propagated by updating matrix Y through the multiplication of matrix Y and matrix T. Due to the dimensionality of both matrices Y and T, matrix Y remains of dimensionality $(l+u) \times C$.

Figure 3.4: Label propagation illustration

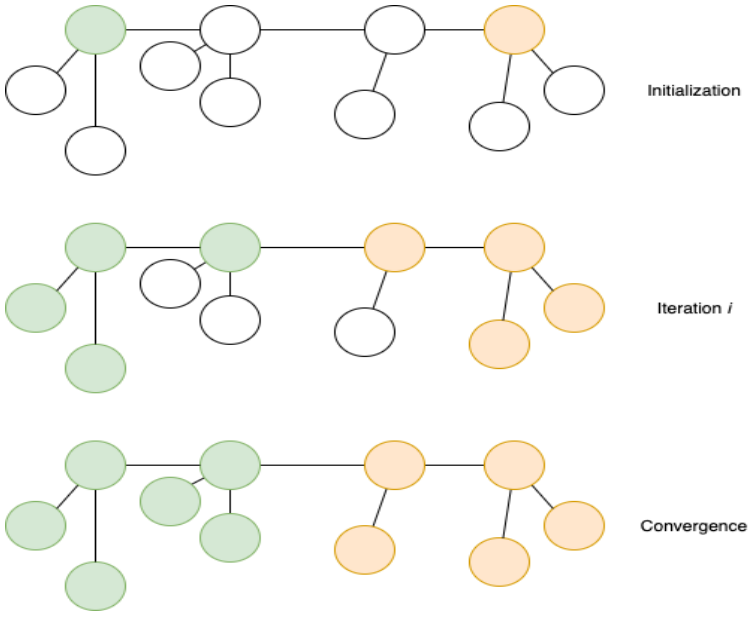


Figure 3.4 provides a visual illustration of the functioning of the label propagation methodology. In the initialization phase of the algorithm, through intermediaries, all nodes in the graph are connected, however solely two observations are labeled. Subsequently, in the next iteration nodes that are directly connected to the labeled observation are labeled, until every node in the graph is labeled. In practice all nodes in the graph are connected,

however in order to provide a simplified visual representation of the algorithm, in figure 3.4 only strongly connected nodes are shown to be connected through edges of the graph.

An algorithmic representation of label propagation is presented in algorithm 3 below. In step 1, Y, the distribution over classes together with the transition matrix T, are used to arrive at updated labels. Through step 3, the algorithm maintains the probability mass per class distributions and thus the algorithm can proceed from high density regions to low density regions, as shown in figure 3.4. Computations are performed using the *LabelPropagation* function from the *scikit-learn semi-supervised* [22] library.

Algorithm 3 Label propagation

1. Node propagation: Y ← TY
2. Row normalization of matrix Y: $\frac{Y_{ij}}{\sum_k Y_{ik}}$, k is row indicator
3. Keep labeled instances fixed
4. Repeat until convergence is reached

3.5.2 Label spreading

Similar to label propagation, label spreading utilizes $(x_1, y_1)...(x_l, y_l)$, the labeled set of observations, and seeks to assign labels to $(x_{l+1}, y_{l+1})...(x_n, y_n)$ which is the unlabeled set of observations. Additionally F' is a multitude of n x c matrices, where c is the number of possible classes an instance can belong to. $F = [F_1^T, ..., F_n^T] \in F'$ is a classification of dataset X whereby each obseravtion x_i is labeled according to step 4 in algorithm 4. Y is again a n x c matrix subject to $Y \in F'$ where Y_{ij} = 1 when x_i is labeled as $y_i = j$ and Y_{ij} = 0 when x_i is not labeled [42].

Algorithm 4 Label spreading

1. Initialize similarity matrix W: $W_{i,j} = e^{-\frac{|x_i - x_j|^2}{2\sigma^2}}$
2. Compute S as $S = D^{-\frac{1}{2}} WD^{-\frac{1}{2}}$
3. Compute $F(t+1) = \alpha S F(t) + (1-\alpha)Y$ until convergence is reached, $0 \leqslant \alpha \leqslant 1$
4. Label each observation according to: $y_i = argmax(F_{ij})$

The abovementioned algorithm works as following. First similarities between observations are stored in similarity matrix W. Next, the affinity matrix W is normalized according to step 2 in algorithm 4. This is required for the algorithm to converge. D is a matrix of zeros, where the indices i,i are filled with the sum of row i in W. Intuitively, in step 3 of the algorithm each observation acquries information from its neighbors, computed in step 1. Also, it retains the information from itself, computed in step 2. The parameter α determines to what extent neighbouring information is considered and initial label information is retained [42].

Intuitively, label propagation and label spreading work similarly. However, differences between both algorithms occur in how the information from labeled nodes is transferred to unlabelled nodes, as presented in step 1 in algorithm 3 and steps 3 and 4 in algorithm 4.

3.6 Single-cell ANnotation using Variational Inference (scANVI)

As an extension to the scVI implementation described in section 3.4, the authors provide scANVI, a semi-supervised learning approach for cell typing based the scVI autoencoder. Similar to scVI, scANVI learns a latent embedding of the raw scRNA-seq data. However, scANVI differs to scVI as the available labels in the data determine the derivation of the latent representation. Subsequently, using a Bayesian semi-supervised approach the model produces a posterior distribution over the cell types [39]. The scANVI methodology is a benchmark methodology in this thesis.

3.7 Alternative scenarios: non-overlapping classes

In addition to previously described methodologies, scenarios in which cell types among training and test data do not perfectly overlap, are considered. Referring to table 2.1, Amacrine cells are removed from the training data. As the model is trained on merely four cell types, but the test data consists of five cell types, the statistical models are unable to assign a probability for an observation to belong to a previously unseen class during training. Therefore, it is presumed that when encountering an unrecognized cell type, the model produces relatively low probabilites for each of the classes, meaning it is insecure to which cell type the observation belongs. Therefore, when the model is less than 70% confident in it's prediction, the cell type is considered to be unrecognized. An example is shown in table 3.2.

Table 3.2: Example of prediction of unrecognized cell type

	Bipolar	**Cones**	**Muller**	**Rods**
1	3.573287e-05	0.6222095	7.6114678e-06	0.3777471
2	1.384734e-07	5.642184e-06	4.966195e-06	0.9999893

As observed, the model is rather uncertain in prediction 1, the maximum prediction is below 70% and therefore observation 1 will be predicted as an unrecognized cell type. Prediction 2 however is very certain and therefore the model will classify this observation to be a Rods cell. The choice of threshold is arbitrary, and enhanced performance could be obtained by optimizing over this threshold. A possible optimization approach would be to use k-fold cross validation as described in section 3 for different confidence thresholds.

In addition to removing one cell type from the training data, the opposite scenario, namely when one cell type is missing in the test data, is also considered. Since the training data consists of all the cell types, the classification methods remains the same. However, one would expect the model to consistently compute low probabilities for the cell type which is not present in the test data.

3.8 Evaluation

Considering a classification problem, the most straightforward evaluation metric is the accuracy ratio of a model. Considering formula 3.8, the accuracy ratio of a classification model simply means how many instances were correctly classified out of the total number of predictions. The total number of predictions can be divided into tp stand for true positives, tn is true negatives, fp is false positives and fn represent the number of false negatives.

Previously introduced terminology comes from a binary classification problem where classes can be either positive or negative. When the model predicts an instance to be positive and it is actually positive, this is considered to be a tp. Whereas if the model predicts it to be of class positive whereas it actually is of class negative, this is considered to be a fp.

Naturally, the same applies to the negative class and thus the *tn* and *fn* counts. One can extend this reasoning to a multi class classification issue, where the model classifies an instance to belong to one of multiple potential classes.

$$accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{3.8}$$

However, when dealing with imbalanced datasets using the *accuracy* metric to evaluate models does not suffice. Therefore classification results are evaluated according to additional measures:

$$precision = \frac{tp}{tp + fp} \tag{3.9}$$

Intuitively, one can think of the *precision* rate as following, when the model predicts an instance to belong to a certain class, how often is it correct for these class predictions.

$$recall = \frac{tp}{tp + tn} \tag{3.10}$$

Recall measures how often an instance which is a certain class, the model predicts it to belong to this class. As a compromise between precision and recall, the so called f1-score is developed:

$$f1 = \frac{2 * precision * recall}{precision + recall} \tag{3.11}$$

Whereby a high f1 score means both precision and recall are high, whereas lower f1 scores indicate that one of both measures or both are smaller. Simply one can have high precision but low recall, or vise versa, which on the one hand would indicate the model is performing well, but on the other hand according to other criteria the model performs undesirably. To consider previously mentioned issues, the f1-score provides a representative measure on how a model performs according to multiple classification criteria.

4 Results

Section 4 thoroughly presents the results obtained in the analysis of this thesis. Starting with section 4.1, the supervised grouped LASSO regression's performance is reported, followed by the results from the extended self-learning grouped LASSO framework presented in 4.2. 4.3 shows the transition into latent dimensionality. Also, results for a supervised grouped LASSO regression in latent space are presented in 4.4. The self-learning extension of this model is presented in the subsequent section, namely 4.5. The results from Graph-Based semi-supervised learning implementations in latent space are presented in 4.6. 4.7 displays results from the single-cell ANnotation using Variational Inference (scANVI) model, which is used as a benchmark model in this thesis. Section 4.8 provides graphical overviews of all model performances in several comparison plots. Finally, section 4.9 presents results in scenarios where one of the cell types is not present in either the training or the test data, meaning that both datasets consists of non-overlapping cell types.

4.1 Grouped LASSO

Table 4.1: Grouped LASSO regression

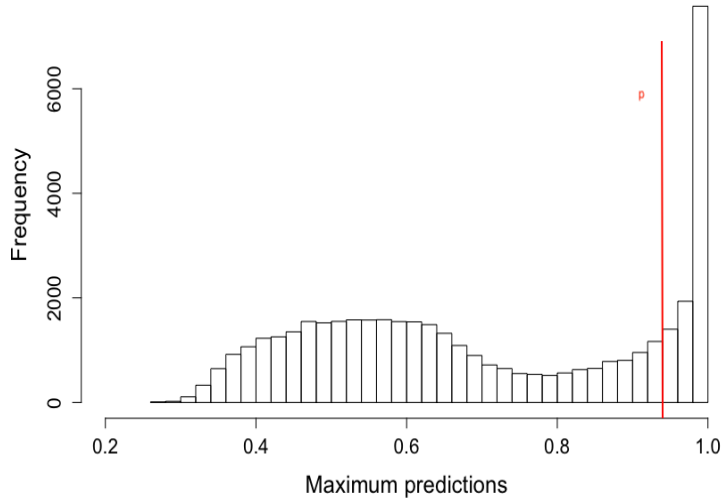
	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9663	0.7530	0.8464	4426
Bipolar	0.2501	0.9177	0.3931	6285
Cones	0.9812	0.3650	0.5321	1868
Muller	0.9347	0.9439	0.9393	1624
Rods	0.8898	0.4466	0.5947	29400
Accuracy			0.5606	43603
Macro avg	0.8044	0.6853	0.6611	43603
Weighted avg	0.8109	0.5606	0.6014	43603

Starting with a benchmark model, results for the supervised grouped LASSO regression are shown in table 4.1. Considering the most straightforward classification metric, the grouped LASSO regression achieves an accuracy of 56.06%. The support column in the presented classification reports throughout this paragraph presents the number of such celltypes in the test data.

However, as mentioned in section 3, several enhanced performance metrics are considered to be more informative. The highest precision rate is achieved for the cell type Cones, with a rate of 98.12%. This means that for all the occaisions in which the model predicted an instance to belong to class Cones, it was correct 98.12% of the time. However, the Cones class at the simultaneously shows the lowest Recall rates. Meaning that when a cell belongs to class Cones, the model could detect this is in only 36.50% of the cases. The highest recall rate is achieved for Muller cell types. Additionally, lowest precision and recall rates are achieved for Bipolar and Cones respectively.

In addition to considering individual class performance, in order to compare model performance, summarizing scores in one evaluation metric can be advantageous. As the f1-score measures a compromise between precision and recall rates, considering the average f1-score, which is computed as the average of f1-scores over the respective classes, as a general performance measure for the model seems reasonable. Aforementioned model reaches an average f1-score of 66.11%. On the individual class level, the highest f1-score is achieved for Muller cell types, 93.93%. Additionally, lowest f1-scores refer to Bipolar cells, which an f1-score of 39.31%.

Figure 4.1: Maximum predicted probability per instance



Important for the subsequent self-learning implementation is the confidence of the predictions of the grouped LASSO regression. As shown in figure 4.1. A significant number of predictions exceed the confidence threshold of 95%, meaning that the self-learning framework will have a sufficient increase in training data during the self-learning iterations.

4.2 Self-learning grouped LASSO

Results from the self-learning framework regarding a grouped LASSO regression in the original representation of the data are presented. The model converges after 4 iterations, meaning

Table 4.2: Self-learning grouped LASSO regression

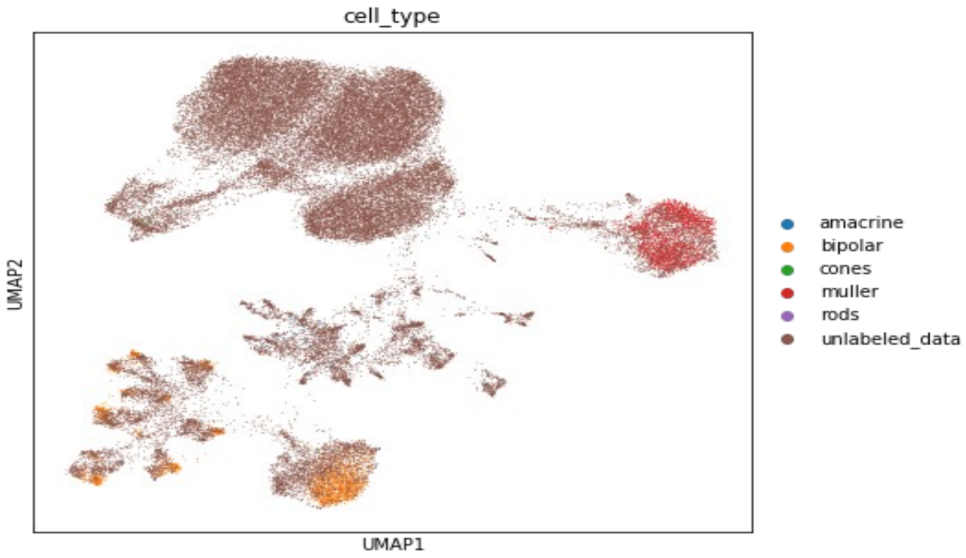
	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9799	0.6954	0.8135	4426
Bipolar	0.3102	0.9182	0.4637	6285
Cones	0.9974	0.2114	0.3489	1868
Muller	0.9702	0.9039	0.9359	1624
Rods	0.9068	0.6154	0.7332	29400
Accuracy			0.6606	43603
Macro avg	0.8329	0.6688	0.6590	43603
Weighted avg	0.8345	0.6606	0.6936	43603

that in iteration 5 there are no more predictions exceeding the confidence threshold and thus the learning is stopped and the final model is returned.

Similar to section 4.1, Cones cell types show a superior precision rate, namely 99.74%. However for this cell type, recall rates are low at 21.14%. Highest recall rates are achieved for the Bipolar cell types at the 91.82% level. A general accuracy rate of 66.06% is reached, together with a macro average f1-score of 65.90%. Similar to results presented in 4.1, on an individual cell type level, the highest f1-score is achieved for Muller cell types, 93.59%. The lowest f1-score reported in table 4.2 refers to Cones cell type. For this cell type, the model achieves an f1-score of 34.89%.

4.3 Latent representation

Figure 4.2: Latent representation

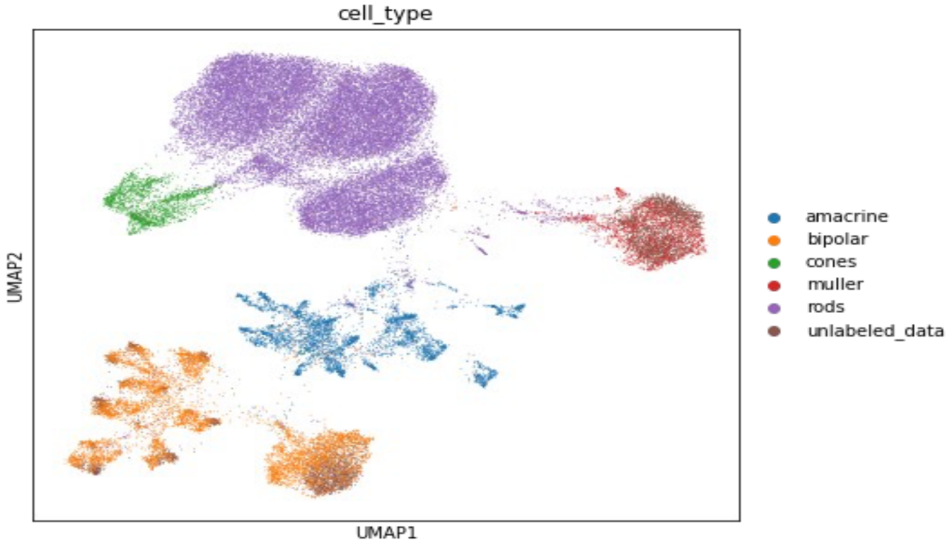


Utilizing the scVI autoencoder reduces the dimensionality from the original data to solely 30 dimensions. Additionally, subsequent to obtaining the latent representation of the original raw scRNA-seq count data, scVI provides visualization tools for the obtained latent data in

two dimensional plots. Figure 4.2 shows the latent representation when the training and test data are integrated, and only the labels for the training data are shown.

For Muller and Bipolar cell types, due to the relatively large amount of observations in the training data, both cell types can clearly be seen in figure 4.2. Important in this figure, and the essence of this thesis, is the fact that the training and test data seem to complement one another in boundaries of the clusters. This is a confirmation that a semi-supervised learning framework for cell typing is reasonable. Additionally, results in figure 4.3 support the findings from earlier research [29] presented in figure 1.1.

Figure 4.3: Latent representation - reversed labelling



Due to the small number of observations, Amacrine, Cones and Rods are difficult to notice in figure 4.2. A close look at the figure, however, reveals the presence of Amacrine cells in the middle of the figure, whereas Cones seem to appear in the top left. Additionally, in the largest cluster on at the top of the figure, few purple observations can be observed, indicating the presence of Rods cells. Figure 7.1 in Appendix A, displays a clearer intergration of both datasets, and support the previously mentioned expansion of cluster boundaries.

In order to support previous claims regarding expansion of decision boundaries for specific cell clusters, figure 4.3 is provided. When reversing the labelling of figure 4.2, thus use labels from the test data and remaining the training data as unlabeled, one can see the clear division of the clusters labelled by the cell types. From figures 4.2 and 4.3 it can be concluded that scVI is a suitable dimensionality reduction tool as a preprocessing step in the self-learning pipeline of this thesis. Moreover, previously shown plots support the assumption that integration of multiple datasets expands cluster boundaries.

Important for the biological scope of this thesis is that there seems to be very little overlap among clusters. When referring to the purpose of cell typing, this means that cells are fairly easily separated, which is of great importance during individualized treatment plans.

Pertinent is that figures 4.2 and 4.3 portray the latent representation in solely two dimensions whereas the actual latent data consists of 30 dimensions. This implies that separations between clusters can in fact be much more explicit than can be shown in two dimensional plots. Visualizing the data however remains useful in order to determine whether the transition to a latent representation of the data is a beneficial intermediate step in cell type identification.

4.4 Grouped LASSO in latent representation

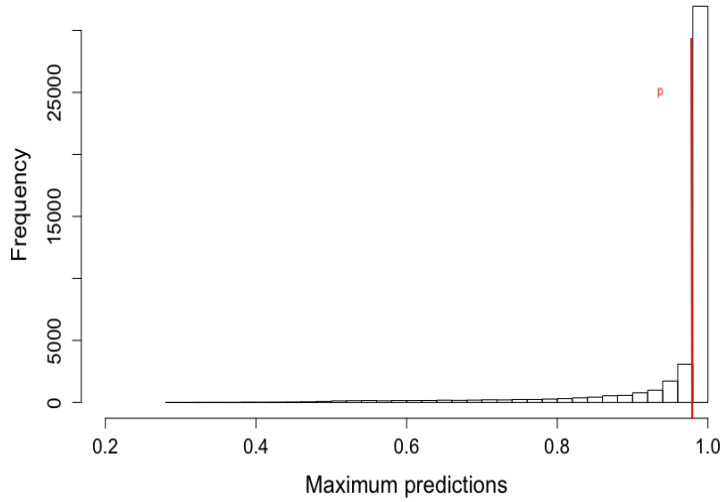
Results from modelling the latent representation of the data are shown below. A grouped LASSO regression in the latent dimensionality achieves an accuracy of 94.83%. Also a macro average f1-score of 90.12% is reported in table 4.3. Additionally, the highest level of precision is 96.92%, which is referring to the Rods cell type. Muller cells show the highest individual recall rate, namely 98.95%.

Table 4.3: Grouped LASSO regression in latent representation

	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9127	0.9482	0.9301	4426
Bipolar	0.9603	0.9331	0.9465	6285
Cones	0.8107	0.7430	0.7754	1868
Muller	0.8059	0.9895	0.8883	1624
Rods	0.9692	0.9624	0.9658	29400
Accuracy			0.9483	43603
Macro avg	0.8918	0.9152	0.9012	43603
Weighted avg	0.9493	0.9483	0.9483	43603

As in section 4.1, the confidence with which predictions are made by the model is important. Figure 4.4 displays the confidence by which predictions of the grouped LASSO regression in latent dimensionality were made. As shown, a vast amount of predictions exceed

Figure 4.4: Maximum predicted probability per instance



the confidence threshold of the self-learning algorithm. This is favourable for the the self-learning algorithm, as it implies that sufficient predictions can be augmented to the training data in subsequent iterations. Due to the high confidence of the predictions by the model, a confidence threshold of 99% is chosen for the self-learning algorithm.

4.5 Self-learning grouped LASSO in latent representation

Table 4.4: Self-learning grouped LASSO regression in latent representation

	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9230	0.9627	0.9424	4426
Bipolar	0.9700	0.9482	0.9590	6285
Cones	0.9753	0.6991	0.8144	1868
Muller	0.8830	0.9907	0.9338	1624
Rods	0.9695	0.9788	0.9742	29400
Accuracy			0.9612	43603
Macro avg	0.9442	0.9159	0.9248	43603
Weighted avg	0.9619	0.9612	0.9604	43603

Table 4.4 portraits results from the self-learning grouped LASSO regression in latent space. The algorithm has iterated ten times, which is equal to the predefined maximum number of iterations this implies that the model made predictions after iteration 10 with confidence exceeding 99%, however in order to keep computational time of the algorithm reasonable, and suitable to scale to larger datasets, a maximum of ten iterations is predefined.

The returned model produces the following results: 96.12% accuracy and a 92.48% macro average f1-score. A 97.53% precision rate for Cones and 99.07% recall for Muller cell types. On average the model achieves a precision rate of 94.42%, whereas a 91.59% recall rate is obtained when averaging over the five cell types.

4.6 Graph-Based semi-supervised learning in latent space

Table 4.5: Label propagation in latent representation

	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9715	0.9241	0.9472	4426
Bipolar	0.9681	0.9601	0.9641	6285
Cones	0.9809	0.1097	0.1974	1868
Muller	0.9446	0.9772	0.9607	1624
Rods	0.9310	0.9903	0.9597	29400
Accuracy			0.9410	43603
Macro avg	0.9592	0.7923	0.8058	43603
Weighted avg	0.9431	0.9410	0.9265	43603

The obtained evaluation metrics when utilizing label propagation in latent space are shown in table 4.5. An overall accuracy of 94.10% is achieved. However, the average f1-score is limited to 80.58%. On an indiviual cell type level, precision rates are highest for Cones, namely 98.09%, whereas Rods cells perform best in terms of recall, at a rate of 99.03%.

Label spreading in latent space produces the results presented in table 4.6. Best performing cell type in terms of precision rate are Rods, with a 98.27% rate. Muller cells have the

highest recall rates at 98.89%. On average an f1-score of 91.84% is reached, whereas the algorithm classifies correctly in 95.54% of it's predictions.

Table 4.6: Label spreading in latent representation

	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9176	0.9530	0.9349	4426
Bipolar	0.9443	0.9585	0.9514	6285
Cones	0.9392	0.8603	0.8980	1868
Muller	0.7251	0.9889	0.8367	1624
Rods	0.9827	0.9593	0.9709	29400
Accuracy			0.9554	43603
Macro avg	0.9018	0.9440	0.9184	43603
Weighted avg	0.9591	0.9554	0.9563	43603

4.7 Single-cell ANnotation using Variational Inference (scANVI)

In order to appropriately valuate the methodologies implemented in this thesis, results are compared to the work by Xu et al. [39] who present an alternative semi-supervised learning approach to cell typing, as discussed in section 3. Classification performance of scANVI method is presented in table 4.7.

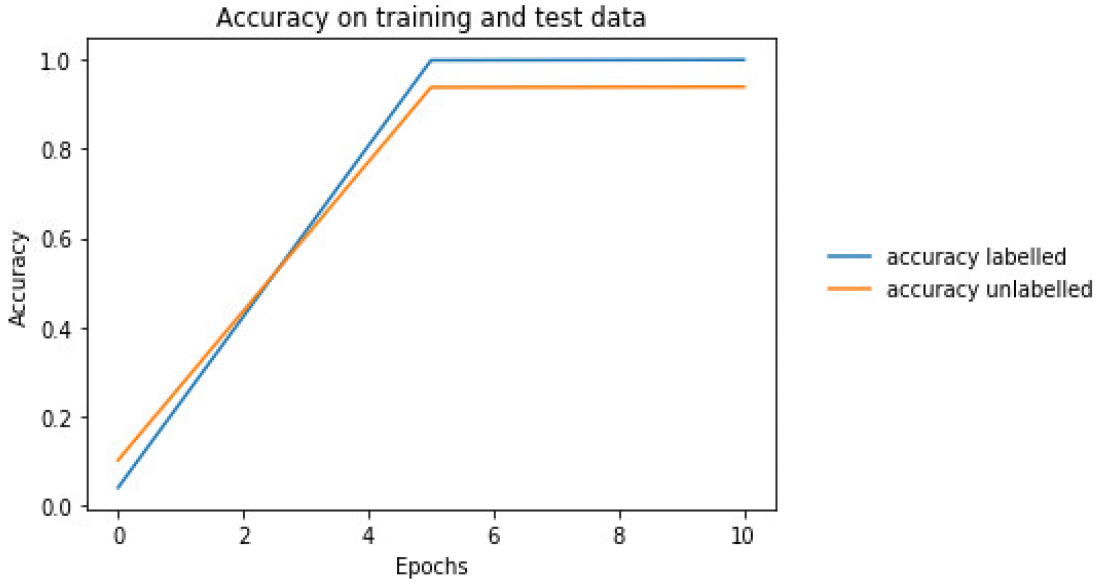
Table 4.7: scANVI

	Precision	**Recall**	**F1-score**	**Support**
Amacrine	0.9337	0.9361	0.9349	4426
Bipolar	0.8522	0.9232	0.8863	6285
Cones	0.6458	0.8431	0.7312	1868
Muller	0.9033	0.9840	0.9419	1624
Rods	0.9881	0.9461	0.9666	29400
Accuracy			0.9387	43603
Macro avg	0.8646	0.9265	0.8922	43603
Weighted avg	0.9452	0.9387	0.9408	43603

At the individual cell type level, the highest precision rates are achieved for Rods cells at 98.81%. Additionally, in terms of recall rates, scANVI is best performing on the Muller cell type and reaches a recall rate of 98.40%. 93.87% of the predictions made are correct. Additionally, an average f1-score of 89.22% is observed. Previously mentioned scANVI methodology is trained for 10 epochs. Evaluation of the model training is displayed in figure 4.5.

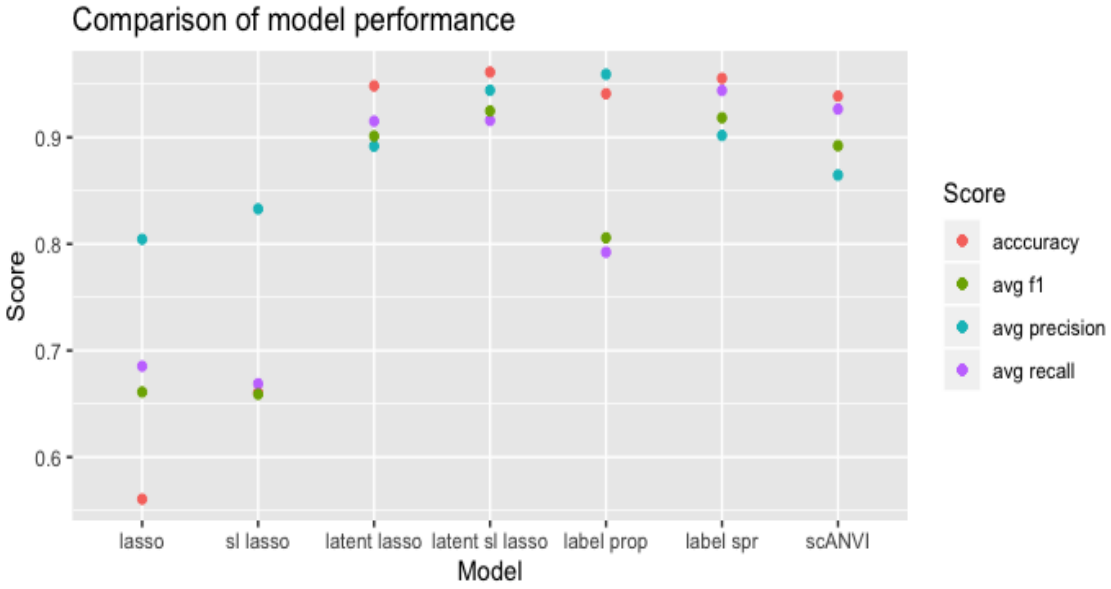
As can be observed, the scANVI model has the ability to learn rapidly. In the first 5 epochs of training, a clear increase of accuracy levels on both the labelled and unlabelled data are observed. However subsequent to epoch 5, no additional performance is achieved, and the learning stagnates. Furthermore a small dispersion between accuracy on labelled and unlabelled data is observed, which indicates a minor bias of the model towards the training data.

Figure 4.5: scVI model optimization



4.8 Model comparison

Figure 4.6: Performance comparison across models



In order to properly assess model performance in imbalanced classification problems, one needs to evaluate according to multiple criteria, therefore, figure 4.6 shows model performance of all models according to accuracy, precision, recall and f1-score.

In figure 4.6 one can observe that generally, the transition to latent space is beneficial for all four shown evaluation criteria. Additionally, models trained in latent dimensionality seem

Figure 4.7: Precision rate across models

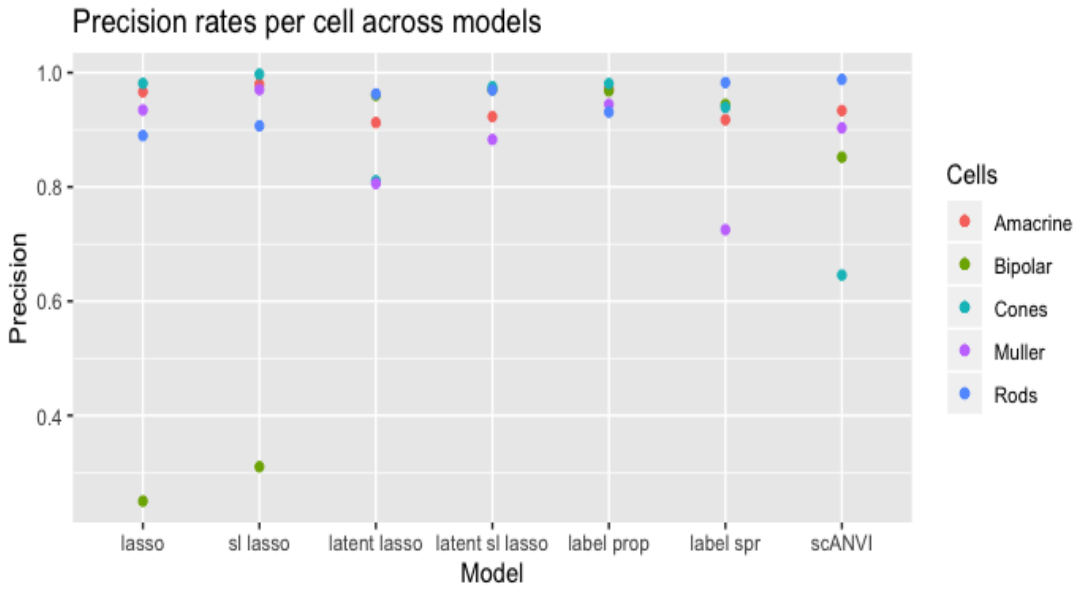
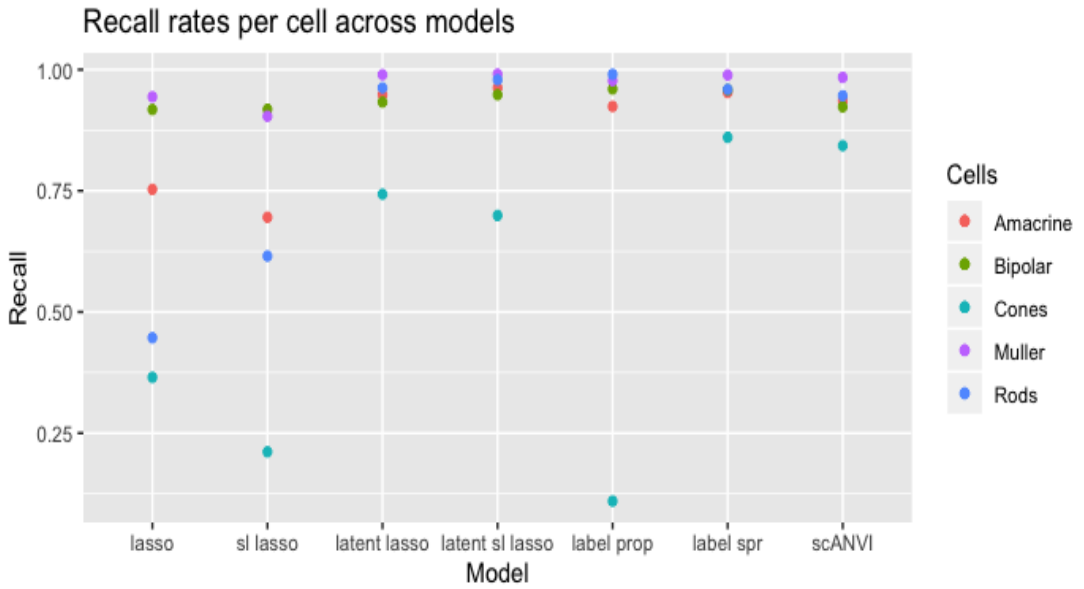


Figure 4.8: Recall rate across models



to show less dispersion across the four evaluation criteria compared to models performed in the original data representation.

However, considering the eventual purpose of cell typing, namely to target specific cells during treatment, one must also evaluate performance of models on a individual cell type level. Figure 4.7 shows precision rates for each cell type per model implemented in this thesis.

It can be observed that on an individual cell type level, all models performed in latent dimensionality show less dispersion across individual precision rates. Additionally, label propagation in latent space produces precision rates which are close for each individual cell

type. Additionally, when considering precision rates on individual cell type level, one must also evaluate individual recall rates, as show in figure 4.8.

Figure 4.8 shows that except for label propagation, all models in latent space generally show higher recall rates on an individual cell type level. In addition, and also with the exception of the label propagation model, dispersion between individual recall rates is narrower for models trained in latent dimensionality.

4.9 Non-overlapping classes

Table 4.8: Grouped LASSO in latent dimensionality for non-overlapping classes

	Precision	**Recall**	**F1-score**	**Support**
Unrecognized cell type	0.2337	0.1640	0.1927	4426
Bipolar	0.6456	0.9102	0.7554	6285
Cones	0.8210	0.5771	0.6778	1868
Muller	0.6483	0.9864	0.7824	1624
Rods	0.9617	0.9111	0.9357	29400
Accuracy			0.8236	43603
Macro avg	0.6621	0.7098	0.6688	43603
Weighted avg	0.8246	0.8236	0.8175	43603

Table 4.9: Self-learning grouped LASSO latent dimensionality for non-overlapping classes

	Precision	**Recall**	**F1-score**	**Support**
Unrecognized cell type	0.4868	0.0251	0.0477	4426
Bipolar	0.6048	0.9391	0.7358	6285
Cones	0.9409	0.5541	0.6974	1868
Muller	0.7374	0.9889	0.8448	1624
Rods	0.9511	0.9815	0.9661	29400
Accuracy			0.8603	43603
Macro avg	0.7442	0.6977	0.6584	43603
Weighted avg	0.8457	0.8602	0.8236	43603

All previous obtained results represent a scenario in which the training and test data perfectly overlap in the cell types the respective datasets contain. In bioinformatics, this is however seldom the case. Therefore, to attain a more realistic representation of cell type identification, scenarios in which training and test data do not perfectly overlap in cell types are considered.

First, one cell type, namely the Amacrine cells are removed from the training data. Subsequently, the model is trained, and evaluated on the test data, which contains a cell type which the model is not trained on. Results for a grouped LASSO regression are shown in table 4.8. When predicting an instance to be an unrecognized cell type, in 23,37% of the predictions the model actually encountered an unrecognized cell type, which is the Amacrine cell type in this scenario. Additionally, when the model encounters an Amacrine cell, which is a cell type it is not trained on, and labels the instance to be an unrecognized cell type, in 16.40% of the

predictions the model is correct. On average, performance over all classes show an accuracy rate of 82,36%. Additionally, an average f1-score of 66,88% is obtained.

Through self-learning, for the unrecognized cell type, a precision rate of 48.68% is reached. A recall rate of 2.51% is observed. Also, 86.03% of the predictions made are correct. Whereas an average f1-score of 65.84% is reported. Results are shown in table 4.9. In addition to

Table 4.10: Confusion matrix, grouped LASSO

	Predicted cell type				
	Amacrine	Bipolar	Cones	Muller	Rods
Bipolar	92	5919	129	35	110
Cones	3	87	1526	5	247
Muller	3	4	5	1602	10
Rods	473	344	954	1205	26424

removing one cell type from the training dataset, the scenario in which one cell type is removed from the test data is considered. As there is one cell type which is not present in the test data, but is present in the training data, solely reporting classification reports does not suffice. Therefore, the confusion matrix in table 4.10 portraits model performance when encountering a test dataset with non-overlapping cell types.

Table 4.11: Grouped LASSO in latent dimension for non-overlapping classes

	Precision	**Recall**	**F1-score**	**Support**
Bipolar	0.9315	0.9417	0.9366	6285
Cones	0.5838	0.8169	0.6809	1868
Muller	0.5627	0.9865	0.7166	1624
Rods	0.9863	0.8988	0.9405	29400
Accuracy			0.9054	39177
Macro avg	0.6129	0.7288	0.6549	39177
Weighted avg	0.9408	0.9054	0.9182	39177

In total the model predicts 571 cell to be Amacrine cells, whereas there are zero of such cells present in the test data. Out of total of 39,177 predictions, this comprises 1.46% of the predictions made. Additionally, table 4.11 shows an average f1-score of 65.49% for the remaining cell types, together with a total of 90.45% of correct classifications.

Table 4.12: Confusion matrix, self-learning grouped LASSO

	Predicted cell type				
	Amacrine	Bipolar	Cones	Muller	Rods
Bipolar	86	6006	47	23	123
Cones	6	31	1610	1	220
Muller	4	3	2	1606	9
Rods	379	187	257	248	28329

Classification performance for the semi-supervised learning framework, specifically through self-learning, are shown in tables 4.12 and 4.13. By observing the confusion matrix, one concludes that a total of 475 cells are predicted to be Amacrine cells, which totals to 1.21% of all predictions made.

Specifically, for the remaining cell types an average precision rate of 72.95% is obtained, together with a slightly higher average recall rate of 75.40%. In total the model accurately predicts in 95.85% of the predictions. Moreover, an average f1-score of 74.07% is reported.

Table 4.13: self-learning grouped LASSO in latent dimension for non-overlapping classes

	Precision	**Recall**	**F1-score**	**Support**
Bipolar	0.9645	0.9556	0.9600	6285
Cones	0.8403	0.8619	0.8510	1868
Muller	0.8552	0.9889	0.9172	1624
Rods	0.9877	0.9636	0.9755	29400
Accuracy			0.9585	39177
Macro avg	0.7295	0.7540	0.7407	39177
Weighted avg	0.9715	0.9585	0.9647	39177

5 Discussion

Section 5 extensively discusses the obtained results and provides a critical assessment of the methodology implemented in this thesis. Starting with 5.1, remarkable findings are discussed and whether a certain model shows superior performance is determined. In 5.2 the choice of methodology is discussed and criticized. Finally, 5.3 discusses ethical aspects of the work and the impact this thesis has on society, the research in cell type identification and research in bioinformatics in general. Additionally, suggestions for future research are also presented in 5.3.

5.1 Results

When solely interpreting accuracy levels, the self-learning grouped LASSO regression in latent dimensionality shows superior performance, with correct classifications in 96.12% of the observations in the test data (table 4.4). When, however, considering more advanced classification metrics, alternative models perform favourably. Specifically, the label propagation algorithm performed in latent dimensionality shows the highest average precision rate over the five cell types, namely 95.92% (table 4.5). Additionally, when considering recall rates, label spreading in latent space performs best, with a score of 94.40% on average (table 4.6). In terms of average f1-score, the self-learning grouped LASSO regression in latent space performs favourably compared to all alternative models, including the scANVI benchmark.

As mentioned earlier, it is rather inadequate to compare models according to one evaluation metric. Research has namely shown that there is a tradeoff between precision and recall rates. It is likely that as one metric increases the other one decreases [5]. An example of such a scenario can be observed between tables 4.5 and 4.6, where table 4.6 shows higher average recall rates, but at the expense of a lower precision rate. If one however, still aims at determining a single model as best performing, comparing performance according to average f1-score is reasonable. According to this evaluation metric, the self-learning grouped LASSO regression in latent diemensionality performs best.

Considering the discussion presented in section 2, on the characteristics and complexities that come with raw scRNA-seq data, it is not surprising that for all four evaluation metrics, a model performed on the data represented in latent space shows superior performance. Results as such can be considered to be a confirmation of enhanced performance in latent dimensionality rather than modelling on the original data. Outperformance by modelling in

latent space supports using proposed scVI methodology in order to capture the challenges accompanied by working with scRNA-seq data, such as high dropout rates and the batch effect.

Worth mentioning is that generally the self-learning framework seems to be effective. Starting with the original data representation, self-learning grouped LASSO regression outperforms the supervised grouped LASSO regression in terms of accuracy and average precision (table 4.1 and table 4.2). Subsequently in latent dimensionality self-learning outperforms supervised learning in terms of accuracy, average precision, average recall and average f1-score (table 4.3 and table 4.4).

Considering the eventual purpose of cell type identification, namely to target specific cells during individualized drug treatments, designating one model as the best performing model remains troublesome. Firstly, one might be tempted to look at accuracy rates, in such a case a self-learning grouped LASSO regression in latent dimensionality is superior. However, because some cell types may be more important than others, precision rates can be vital.

As an extension of average precision rates, one must consider precision rates at the individual cell type level. Perhaps surprising is the finding that when aiming to target Amacrine, Cones, and Muller cells, one should implement a self-learning grouped LASSO regression in the original data representation, which yields a precision rate of 97.99%, 99.74% and 97.02% respectively (table 4.2). However, when aiming to target Bipolar cells, the self-learning grouped LASSO regression in latent space performs best (table 4.4). Highest precision rates for Rods cells are achieved by using the scANVI methodology (table 4.7).

Regarding recall rates, different models show superior performance on an individual cell type level. Both for Amacrine and Muller cell types, optimal performance is obtained by implementing a self-learning grouped LASSO regression in latent space (table 4.4). Considering Bipolar and Rods cell types, it is observed that label propagation in latent dimensionality is performing best (table 4.5). Label spreading in latent dimensionality is recommended to be implemented when one is interested in the highest recall rates for Cones cells (table 4.6).

When comparing the models implemented in this thesis to the scANVI methdology benchmark (table 4.7), outperformance according to multiple criteria is achieved. When considering label spreading (table 4.6), this methodology outperforms the scANVI in accuracy, average precision, average recall and average f1-score by 1.67%, 3.27%, 1.75%, 2.62% respectively.

5.2 Method

Generally, in terms of replicability the methodologies implemented in this thesis are easily reproduced, especially considering the frequent usage of existing libraries in *R* [30] and *Python* [35]. Advantageous of the use of *R* [30] and *Python* [35] software is the open-source characteristic of the methodologies, meaning implemented methodologies can be assessed in detail when referring to the source code.

Considering reliability, results among the six models implemented in this thesis show no great dispersion. Nor are the results considered to be surprising, and thus coincide with the theoretical argumentation presented in sections 1 and 3. Generally, transforming the original raw count scRNA-seq data into latent dimensionality using scVI methodology is beneficial. Considering previously mentioned argumentation, one could say that the results and therefore the methods implemented in this thesis are reliable. Still, this study refers to solely one training and test data, therefore the possibility that results are achieved through coincidentally chosen datasets with great overlap, is present.

It is important to address the validity of the conducted study. Generally, as expected, different gene expressions are correlated to the different cell types. I find that using gene expression levels per individual cell make accurate modelling and cell type identification

possible. Finding different clusters of cell types according to gene expression and modelling on these gene readings is therefore a valid study.

Important to mention is the fact that labels in the training data can be subject to errors. Shekhar et al., [28] labeled the cells in the training data of this thesis by analyzing patterns of gene expressions and combining this with biological knowledge to define the cell types. Although Shekhar et al., [28] are confident about their labelling method, the methdology is based on statistical models which can be subject to errors. When interpreting plots 4.2 and 4.3, one can observe some overlap among especially the Amacrine, Muller, Cones and Rods cell types. Considering this slight overlap in clusters, there is a possibility that some labels in the training data were labelled as a cell type which is very similar but different than the true cell type.

Although the use of methodologies is well argued in sections 1 and 3, it remains important to critically analyze my own work, and especially assess the drawbacks of the methodologies chosen. Critical discussions of the self-learning algorithm and the transition of the original data into latent dimensionality are presented below.

A benefit and a simultaneous drawback of the self-learning framework is its simplicity. Where on the one hand, the simplicity of the model allows for straightforward model interpretation, on the other hand, the self-learning framework as expected, is accompanied by several drawbacks. Firstly, this type of learning is heavily dependent on the supervised classifier one chooses to use. If one chooses a supervised model which is not suitable for the work, though self-learning, one attempts to develop an improved methodology of which the fundamental model is unsuited. Secondly, and related to the earlier mentioned drawback is that predictions in the self-learning framework can be very confident but false. In such a case it could occur that the algorithm is learning from itself but from wrongly classified instances, leading to an improperly trained model. This can eventually lead to inferior performance of a self-learning model compared to the solely supervised version of the model. Enhanced development of the self-learning framework would include optimization over the confidence threshold which determines when classifications are considered to be true labels and appended to the training data.

Also the Graph-Based semi-supervsied learning and scANVI methodology are subject to discussion. Firstly, Graph-Based learning seems straightforward as presented in figure 3.3, when the data is subject to solely two dimensions. In my work however, actual dimensionality is 30, meaning the data could be overlapping and less easiliy seperated in multiple dimensions [27]. Secondly, the scANVI is heavily dependent on neural network implementations, which as mentioned earlier are hardly interpretable [39]. Model interpretability is of great essence in any statistical implementation, but especially in a bioinformatics domain.

Considering the choice of methods, several alternative semi-supervised learning methods such as low-density separation and generative models [6] have not been implemented, nor have they been thoroughly considered. In order to produce a comprehensive model evaluation study and provide a broadly oriented thesis, my work could have been extended by implementing a wider diversification of learning methods.

A vital intermediate step in this thesis is the transition from original data to latent dimensionality. Where at first one might see clear benefits in this data processing step, a critical note is presented below. Naturally it is in one's interest to modify models in order to obtain improved results in order to arrive at an optimal model. Clearly in this thesis one can observe that transferring the original data representation into latent space enhances model performance. Important, however, is that this increase in performance comes at the cost of model interpretability. Meaning that, when modelling in original data, one could clearly interpret which genes are important determinants in the cell typing of the dataset. By transferring into latent space, one loses this possibility to easily interpret, and analyze which genes define differences among cell types. An analysis of which genes are important determinants in cell types is still possible when using scVI, however much less straightforward compared to models developed on the original data representation.

Related to this, is that one should carefully consider the audience of this thesis work. Naturally, the statistical research community is usually interested in obtaining the best possible results. For the bioinformatics field this might be different, and models with high interpretability might be favoured, as these models will eventually be used in personalized drug treatment plans.

Considering the loss of straightforward variable interpretability due to the transition into latent space, arguments against the use of certain models become less evident, and an even wider range of models than presented earlier could have been considered. Usually one argues against the use of deep learning methods as this methodology is hardly interpretable. But, as models computed in latent dimensionality are also hard to analyze, this argument loses its credibility. Recently, neural networks and deep learning approaches to classification have proven to be very effective in the machine learning community, this thesis however refrains from using state-of-the art methods as such.

Reference to academic sources is widely present throughout the study. All methodologies are elaborated upon and explained using scientific papers that have proven their credibility. Important to mention is that this research is highly dependent on open source libraries in *R* [30] and *Python* [35]. Specifically the grouped LASSO regressions are performed using the *glmnet* [9] package in *R* [30], whereas scVI [15] is a *Python* [35] implementation. Both packages are supported by academic papers which enhances their credibility. Still, open-source implementations come at the risk that packages can be published without being critically assessed. Therefore, although unlikely, the possibility of computational and statistical flaws in packages used, always remains present.

5.3 The work in a wider context

5.3.1 Ethical considerations

As the results from this thesis might be used in personalized drug treatment plans, a note on ethical considerations of the work conducted is vital. One must be aware that every statistical model is inevitably prone to errors. When dealing with a theoretical statistical problem, a misclassification by an implemented model might be undesirable but not crucial. A statistical error or misinterpretation in a personalized drug treatment plan, however, might impact human lives. Therefore, although results from this thesis look promising, the reader of this work must always apply logical reasoning in addition to statistical modelling, rather than blindly following model recommendations. Additionally, it is essential that data privacy is guaranteed. Especially in the scope of medical data, one could be working with data that reveals personal information about a patient. When working with such data, researchers should always be aware of the sensitivity of the data.

5.3.2 Societal implications

Implications for society resulting from this thesis are important. Cell type identification is a crucial step in individualized drug treatment and previous shown results are an important contribution to this field of research. Therefore, the current work on cell type identification might be an important step forward in individualized drug treatment research, which can lead to severe cost reductions for health care systems worldwide.

5.3.3 Future reserach

Although results from this thesis look promising, still, much research in cell type identification is required. A simple additional research direction would be to assess the robustness of the models developed in this thesis by examining a larger extent of datasets and determine how the models perform. Also, when two datasets complement each other in decision

boundaries, three, four and even a larger number of datasets could potentially complement one another in decision boundaries of clusters. Using multiple datasets to determine decision boundaries can therefore be a relevant study.

Additionally, a research direction for cell typing would be to investigate cell type identification when classes of cells do not perfectly overlap between the training and the test data. To some extent, these scenarios have been explored in this thesis, as presented in section 4.9. However, much more work on these scenarios can be implemented. One could think of how models perform when multiple classes among training and test datasets do not overlap. Scenarios as such are realistic, as datasets seldom perfectly overlap in cell types.

Additional research can be conducted with regard to bulk data. This type of data expresses gene expressions per cell type on an aggregate level, and this would therefore be a different way of modelling gene expression data. A thorough study on how semi-supervised learning performs by using bulk data would therefore be highly relevant and contributing to the research in cell type identification.

Also, enhanced evaluation methods when dealing with cell type identification methods can be considered. As the eventual purpose of this thesis is to improve drug treatment by targeting essential cell types, some cells are more important than others. Therefore in evaluation metrics, one could consider a more severe error when essential cells are misclassified then when non-essential cell types are misclassified. However, to do so, one must know which cell types are essential in drug treatment, which requires much more biological reserach, which is beyond the scope of this thesis.

6 Conclusion

This thesis aims to develop semi-supervised learning algorithms that present enhanced performance over existing cell type identification methodologies. To a large extent this aim has been achieved. Compared to similar cell typing studies, such as scANVI, the label spreading model in latent dimensionality performs favourable according to multiple criteria. In addition, answers to the four research questions stated in the introduction have been obtained.

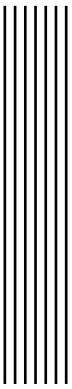
Firstly, this thesis shows that both self-learning as well as Graph-Based semi-supervised learning methods are suitable approaches to cell type identification studies. Both in the original data representation as well as in latent dimensionality, self-learning as well as Graph-Based methods are straightforward to implement through computations in *R* [30] and *Python* [35].

Secondly, compared to recent work on cell tying, namely scANVI, my work performs favourably. Compared to scANVI (table 4.7), label spreading (table 4.6) in latent dimensionality shows an increase of 1.67% in accuracy, average precision is increased by 3.27%, average recall is increased by 1.75% and finally an increase of 2.62% in average f1-score is obtained.

Thirdly, I find that semi-supervised learning outperforms supervised learning in cell typing. Compared to a solely supervised classifier, in the original data representation, the self-learning framework achieves an increase in accuracy of 10%, whereas average precision increases by 2.85% (table 4.1 and table 4.2). In latent dimensionality, semi-supervised learning approaches achieve an increase in accuracy of 1.29%, whereas average precision, average recall and average f1-score are increased by 6.74%, 2.88%, 2.36% respectively (table 4.3, table 4.4, table 4.5 and table 4.6).

And fourthly, strong evidence is found that noise removal and dimensionality reduction is beneficial in cell type identification. Models evaluated in latent dimensionality consistently outperform models computed on the original data representation.

As our work produces accurate cell typing methodologies, the developed models can be used to target essential cell types in individualized drug treatment programs. Therefore, this work contributes to an essential step in individualized drug treatment research.



Bibliography

[1] Tamim Abdelaal, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel JT Reinders, and Ahmed Mahfouz. "A comparison of automatic cell identification methods for single-cell RNA sequencing data". In: *Genome biology* 20.1 (2019), p. 194.

[2] Jose Alquicira-Hernandez, Anuja Sathe, Hanlee P Ji, Quan Nguyen, and Joseph E Powell. "scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data". In: *Genome Biology* 20.1 (2019), pp. 1–17.

[3] Matthew Amodio, David Van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, et al. "Exploring single-cell data with deep multitasking neural networks". In: *Nature methods* (2019), pp. 1–7.

[4] Dor Bank, Daniel Greenfeld, and Gal Hyams. "Improved training for self training by confidence assessments". In: *Science and Information Conference*. Springer. 2018, pp. 163–173.

[5] Michael Buckland and Fredric Gey. "The relationship between recall and precision". In: *Journal of the American society for information science* 45.1 (1994), pp. 12–19.

[6] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[7] Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. "Single-cell RNA-seq denoising using a deep count autoencoder". In: *Nature communications* 10.1 (2019), pp. 1–14.

[8] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson. "Semi-supervised network traffic classification". In: *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. 2007, pp. 369–370.

[9] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1.

[10] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[11] Trevor Hastie and Junyang Qian. "Glmnet vignette". In: *Retrieve from http://www. web. stanford. edu/~ hastie/Papers/Glmnet_Vignette. pdf. Accessed September* 20 (2014), p. 2016.

[12] Nathaniel D Heintzman, Gary C Hon, R David Hawkins, Pouya Kheradpour, Alexander Stark, Lindsey F Harp, Zhen Ye, Leonard K Lee, Rhona K Stuart, Christina W Ching, et al. "Histone modifications at human enhancers reflect global cell-type-specific gene expression". In: *Nature* 459.7243 (2009), pp. 108–112.

[13] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. "scmap: projection of single-cell RNA-seq data across data sets". In: *Nature methods* 15.5 (2018), p. 359.

[14] Cosmin Lazar, Stijn Meganck, Jonatan Taminau, David Steenhoff, Alain Coletta, Colin Molter, David Y Weiss-Solıs, Robin Duque, Hugues Bersini, and Ann Nowé. "Batch effect removal methods for microarray gene expression data integration: a survey". In: *Briefings in bioinformatics* 14.4 (2013), pp. 469–490.

[15] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. "Deep generative modeling for single-cell transcriptomics". In: *Nature methods* 15.12 (2018), pp. 1053–1058.

[16] Aaron Lun, Davide Risso, and K Korthauer. "SingleCellExperiment: S4 classes for single cell data". In: *R package version* 1.0 (2018).

[17] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets". In: *Cell* 161.5 (2015), pp. 1202–1214.

[18] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[19] M Morgan. "Biocmanager: Access the Bioconductor Project Package Repository". In: *R package version* 1.4 (2018).

[20] Samuel M Mwalili, Emmanuel Lesaffre, and Dominique Declerck. "The zero-inflated negative binomial regression model with correction for misclassification: an example in caries research". In: *Statistical methods in medical research* 17.2 (2008), pp. 123–139.

[21] Kamal Nigam, Andrew McCallum, and Tom Mitchell. "Semi-supervised text classification using EM". In: *Semi-Supervised Learning* (2006), pp. 33–56.

[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[23] Emma Pierson and Christopher Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis". In: *Genome biology* 16.1 (2015), p. 241.

[24] Munir Pirmohamed and B Kevin Park. "Genetic susceptibility to adverse drug reactions". In: *Trends in pharmacological sciences* 22.6 (2001), pp. 298–305.

[25] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. "A general and flexible method for signal extraction from single-cell RNA-seq data". In: *Nature communications* 9.1 (2018), pp. 1–17.

[26] Fabio Roli and Gian Luca Marcialis. "Semi-supervised PCA-based face recognition using self-training". In: *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer. 2006, pp. 560–568.

[27] Shrutika S Sawant and Manoharan Prabukumar. "A review on graph-based semi-supervised learning methods for hyperspectral image classification". In: *The Egyptian Journal of Remote Sensing and Space Science* (2018).

[28] Karthik Shekhar, Sylvain W Lapan, Irene E Whitney, Nicholas M Tran, Evan Z Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z Levin, James Nemesh, Melissa Goldman, et al. "Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics". In: *Cell* 166.5 (2016), pp. 1308–1323.

[29] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. "Comprehensive integration of single-cell data". In: *Cell* 177.7 (2019), pp. 1888–1902.

[30] R Core Team et al. "R: A language and environment for statistical computing". In: (2013).

[31] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[32] Michael E Tipping and Neil D Lawrence. "Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis". In: *Neurocomputing* 69.1-3 (2005), pp. 123–141.

[33] Koen Van den Berge, Fanny Perraudeau, Charlotte Soneson, Michael I Love, Davide Risso, Jean-Philippe Vert, Mark D Robinson, Sandrine Dudoit, and Lieven Clement. "Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications". In: *Genome biology* 19.1 (2018), p. 24.

[34] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. "Dimensionality reduction: a comparative". In: *J Mach Learn Res* 10.66-71 (2009), p. 13.

[35] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Vol. 620. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[36] Bin Wang, Bruce Spencer, Charles X Ling, and Harry Zhang. "Semi-supervised self-training for sentence subjectivity classification". In: *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer. 2008, pp. 344–355.

[37] Xiangyu Wang, David Dunson, and Chenlei Leng. "No penalty no tears: Least squares in high-dimensional linear models". In: *International Conference on Machine Learning*. 2016, pp. 1814–1822.

[38] Grant R Wilkinson. "Drug metabolism and variability among patients in drug response". In: *New England Journal of Medicine* 352.21 (2005), pp. 2211–2221.

[39] Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I Jordan, and Nir Yosef. "Harmonization and annotation of single-cell transcriptomics data with deep generative models". In: *bioRxiv* (2019), p. 532895.

[40] Qian Xu and Qiang Yang. "A survey of transfer and multitask learning in bioinformatics". In: *Journal of Computing Science and Engineering* 5.3 (2011), pp. 257–268.

[41] Haiqin Yang, Zenglin Xu, Irwin King, and Michael R Lyu. "Online learning for group lasso". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 1191–1198.

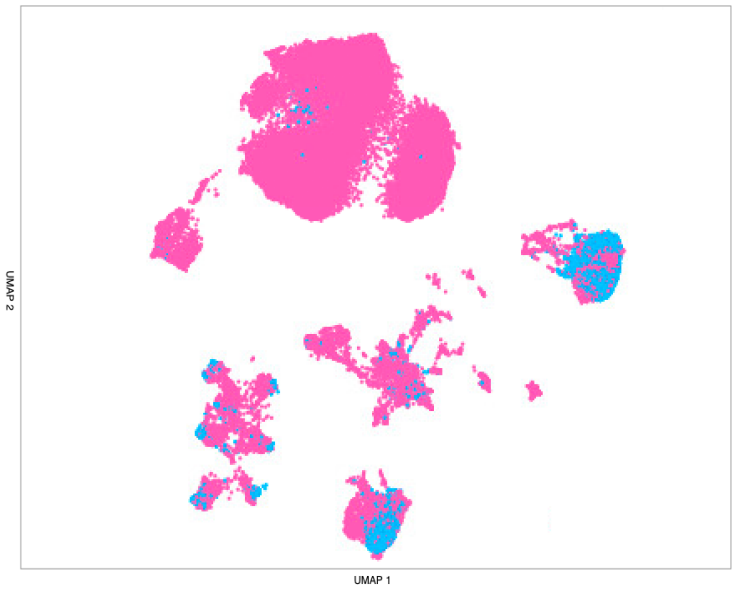
[42] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency". In: *Advances in neural information processing systems*. 2004, pp. 321–328.

[43] Xiaojin Jerry Zhu. *Semi-supervised learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2005.

[44] Xiaojin Zhu and Zoubin Ghahramani. "Learning from labeled and unlabeled data with label propagation". In: (2002).

7 Appendix A

Figure 7.1: Training and test data harmonization in latent dimensionality



Training and test datasets harmonization in latent dimensionality obtained by implementing *scVI*. Training data is represented by blue observations, whereas pink indicates observations belonging to the test data. Visualization is done according to UMAP mappings.