THOMPSON RIVERS UNIVERSITY

Benchmarking penalized regression methods in machine learning for single-cell RNA sequencing data

By

Bhavithry Sen Puliparambil

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

KAMLOOPS, BRITISH COLUMBIA

April, 2022

SUPERVISORS

Dr Jabed Tomal

Dr Yan Yan

ABSTRACT

Single-Cell RNA Sequencing (scRNA-seq) technology has enabled the biological research community to explore gene expression at a single-cell resolution. By studying differences in gene expression, it is possible to differentiate cell clusters and types within tissues. One of the major challenges in a scRNA-seq study is feature selection from high dimensional data. There are several statistical and machine learning methods available to solve this problem but their performances across data sets lack systematic comparison. In this research, we benchmark different penalized regression methods which are suitable for scRNA-seq data. Results on four different scRNA-seq data sets show that Sparse Group Lasso (SGL) implemented by the SGL R package performs better than other methods in terms of area under the receiver operating curve (AUC). The computation time for different methods varies between data sets with SGL having the least average computation time. Based on our findings, we propose a new method for scRNA-seq clustering which applied SGL on a pre-selected subset of genes. These selected genes are the union of top important genes from the ridge, lasso, elastic net, and droplasso methods. The proposed method demonstrates an improvement in AUC compared to SGL and other methods as well.

Key Words: Single-Cell RNA Sequencing; Machine Learning; LASSO; Feature Selection; High Dimensional Data; R Programming Language.

ACKNOWLEDGEMENTS

Foremost, I owe a debt of gratitude to my supervisors Dr Jabed Tomal and Dr Yan Yan for guiding me through this research. Their combined knowledge is the foundation of this research, and they entrusted me to build on it. Dr Yan Yan also granted me a lab space at the TRU graduate research lab.

I would like to thank the program coordinator of MSc Data Science program, Dr Qinglin (Roger) Yu for helping me stay to on track. My sincere thanks to the faculty at Thompson Rivers University, especially Dr Piper Jackson, Dr Mohamed Tawhid, Dr Becky Wei Lin, Dr Richard Taylor, Dr Mateen Shaikh, and Dr Mila Kwiatkowska for listening and encouraging me to do better.

I acknowledge Compute Canada for hosting the 32 GB Linux remote server which is used for computation in this research. I would also like to acknowledge the funding received for this research from:

1. TRU Internal Research Fund (IRF) awarded to Dr Jabed Tomal, Department of Mathematics and Statistics, Thompson Rivers University, and Dr Yan Yan, Department of Mathematics and Statistics, Thompson Rivers University, and

2. Natural Sciences and Engineering Research Council of Canada (NSERC) fund awarded to Dr Jabed Tomal, Department of Mathematics and Statistics, Thompson Rivers University, and Dr Yan Yan, Department of Computing Science, Thompson Rivers University.

It is my privilege to thank my husband Mr Jose Mathews Kanjooparambil for his friendship and counsel that help me get through everything. Finally, I thank my parents, brother, and the god for their unconditional love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Single-cell RNA sequencing (scRNA-seq) technology is a recent trend in biological research. Researchers can now simultaneously explore thousands of cells in a tissue and their average gene expression levels, as well as the gene expression profile of each individual cell in that tissue with scRNA-seq technology (Slovin et al. [2021]). One of the many applications of scRNA-seq technology is differentiating tumour cells from normal healthy cells by comparing their molecular signatures. However, the scRNA-seq data itself is not without its challenges (Kiselev et al. [2019]). One could say it is the very definition of the curse of dimensionality ($p >> n$, where p is the number of variables and n is the number of observations) problem in machine learning (ML). For scRNA-seq data, the number of genes (variables) far exceeds the number of cells (observations). One way to solve the $p >> n$ problem is to employ feature selection.

Feature selection is the method of selecting variables (here, genes) that are more useful for predicting the target variable. Random forests, Recursive

Feature Elimination (RFE) and penalized regression are some of the commonly used feature selection methods in machine learning. Multiple studies have been published on the application of random forests for scRNA-seq data (Kaymaz et al. [2020], Pouyan and Kostka [2018]). For very high dimensional data such as scRNA-seq data, using RFE alone tend to be computationally expensive (Chen and Jeong [2007]). Furthermore, some of the penalized regression algorithms are developed specifically for scRNA-seq data and showed varied success.

Penalized regression allows feature selection for high dimensional data such as scRNA-seq by producing sparse solutions which are predictive models based on the expression of a limited number of genes. Penalized regression in machine learning has several versions such as ridge regression, Least Absolute Shrinkage and Selection Operator (LASSO) regression (Tibshirani [1996]) and a combination of ridge and lasso known as elastic net regression (Zou and Hastie [2005]). Each of these is useful for a different problem when dealing with scRNA-seq data. For instance, ridge regression is useful for bringing some of the coefficients of the model features closer to 0. To reduce the dimensions, the features close to zero are then proposed to be excluded while others are retained in the model. This notion is also known as hard thresholding. This issue is taken care of by lasso regression proposed by (Tibshirani [2011]), which can make coefficients equal to absolute zero (soft thresholding). There are many variants of lasso such as Drop lasso (Khalfaoui and Vert [2018]), Group lasso (Yuan and Lin [2006]), Sparse Group lasso (Simon et al. [2013]), and Big lasso (Zeng and Breheny [2017]). We are interested to see which of these methods are suitable for different kinds of scRNA-seq data.

There also exist several other variants of lasso regression such as fused lasso (Tibshirani et al. [2005]), adaptive lasso (Zou [2006]), and prior lasso (Jiang et al. [2016]). Fused lasso was proposed for time series or image-based data, adaptive lasso was proposed for proportional hazards regression, and prior lasso (Jiang et al. [2016]) was proposed for biological data but requires prior information to be incorporated into lasso regression. Since these algorithms are not proposed for scRNA-seq data, they are not included in this study.

Currently, there are several studies on machine learning algorithms which claim to be capable of processing high dimensional data, and some are developed specifically for scRNA-seq data. However, to our knowledge, there is no comprehensive study of how these algorithms perform in comparison with each other, specifically when dealing with scRNA-seq data. The primary objective of this study is to fill this gap in knowledge and thereby provide a comprehensive guideline as to the performance of penalized regression algorithms in terms of the prediction performance as well as the computation time. The second objective of this study is to select the top-performing algorithms as per the results from the first objective, investigate their combined performance, and propose a new algorithm that may outperform the top algorithm.

The rest of this thesis is organized as follows. Chapter 2 discusses the literature review; Chapter 3 explains the research design and proposed algorithm; Chapter 4 discusses the experimental data, presents the results, analyzes the findings and biological meanings; Chapter 5 proposes future work directions.

Chapter 2

Literature Review

This chapter briefly explains genome sequencing techniques, some of the popular machine learning and statistical methods for scRNA-seq, and the existing studies about them. The published studies vary by methods and data sets as well as the challenges of scRNA-seq data being addressed by the algorithms. It can be seen that there are very few studies that include more than one penalized regression method discussed in this research.

2.1 High Throughput Sequencing

The first draft of the human genome sequence was available in 2001 (Lander et al. [2001]). Sanger DNA sequencing was used for this project. This technology was expensive and limited in throughput. The National Human Genome Research Institute (NGHRI) funded research in genome sequencing which led to invention of many High Throughput Sequencing (HTS) tech-

nologies. HTS typically involved template preparation, clonal amplification, and repeated massive parallel sequencing (Reuter et al. [2015]). RNA sequencing (RNA-seq) is a popular HTS method. Bulk RNA-seq measures the average gene expression of a tissue sample. RNA-seq is used for transcript analysis. Illumina, Roche 454 and PacBio are some of the commercial platforms for RNA-seq.

2.2 Single Cell RNA Sequencing

scRNA-seq is more advanced than bulk RNA-seq because scRNA-seq can identify the gene expression of individual cells as well as the average gene expression of the sample. scRNA-seq is useful for identifying cellular heterogeneity. However, scRNA-seq has low capture efficiency and high dropouts compared to bulk RNA-seq. Quality control is also necessary for removing technical noise from scRNA-seq data (Chen et al. [2019]). Machine learning methods such as the drop lasso regression was designed to overcome the issue of dropout noise in scRNA-seq. Batch effect correction, normalization, imputation, dimensionality reduction, and feature selection are often used in scRNA-seq data analysis. scRNA-seq protocols can be classified into two categories, full-length transcript sequencing approaches and 3'-end or 5'-end transcript sequencing (Chen et al. [2019]). Smart-seq2, Drop-seq, STRT-seq and Chromium are some of the popular scRNA-seq methods.

2.2.1 Smart-seq2

Smart-seq2 is a full-length transcript sequencing technique. It is characterized by improved reverse transcription, template switching, and preamplification which increase both the yield and length of cDNA libraries generated from individual cells. Its limitations are the lack of strand specificity and the inability to detect nonpolyadenylated RNA (Picelli et al. [2013]).

2.2.2 Drop-seq

Drop-seq is a 3'-end transcript sequencing technique. This technology quickly profiles thousands of individual cells by enabling highly parallel analysis of individual cells by RNA-seq. Drop-seq separates individual cells into nanoliter-sized aqueous droplets, associating a different barcode with each cell's RNAs, and sequencing them all together (Macosko et al. [2015]).

2.2.3 Chromium

Chromium is another example of the 3'-end transcript sequencing technique. It is a droplet-based system that enables 3' mRNA counting of tens of thousands of single cells per sample. This technology can capture approximately 50% of cells loaded into the system. Parallel processing enables it to analyze up to eight samples per run. The droplets undergo reverse transcription, and barcoded complementary DNAs (cDNAs) are bulk amplified. Illumina short-read sequencing is applied to the resulting libraries followed by Cell Ranger which processes the sequencing data and enables automated cell clustering

(Zheng et al. [2017]).

2.2.4 STRT-seq

STRT-seq is a 5'-end transcript sequencing technique. The advantage of this technology is that it eliminates the need for known markers to classify cell types. In STRT-seq scRNA-seq expression profiles are clustered to form a two-dimensional cell map onto which expression data is projected. Three levels of the organization is integrated into this cell map: the whole population of cells, the functionally distinct subpopulations it contains, and the single cells themselves (Islam et al. [2011]).

2.3 Popular Machine Learning and Statistical Algorithms

This section discusses some of the popular machine learning and statistical algorithms, other than penalized regression algorithms, that have been discussed in academic literature, in conjunction with scRNA-seq data.

We have already seen that data sets such as scRNA-seq present the challenge of high dimensionality. There are several algorithms developed for dimensionality reduction. Principal component analysis (PCA) and Linear Discriminant Analysis (LDA) are two such dimensionality reduction methods. Abdi and Williams [2010] describe PCA as a multivariate technique that extracts information from several inter-correlated quantitative dependent variables and create a set of new orthogonal variables called principal

components. LDA finds the projection hyperplane that minimizes the interclass variance and maximizes the distance between the projected means of the classes (Xanthopoulos et al. [2013]). Recursive feature elimination is another dimesionality reduction method. However, these methods are not used in my thesis because of their computational complexity.

Support Vector Machines, Naive Bayes Classifier, Decision Trees, Neural Networks, Ensemble Methods such as Bagging, Random Forests, and Boosting are some of the machine learning algorithms used for processing high dimensional data. These algorithms have been studied widely and are not included in my thesis.

2.4 Penalized regression in published research

Evidence for use of the penalized regression methods for scRNA-seq data in the existing literature is discussed in this section. If a published study analyses at least one method of penalized regression, it is included for discussion here.

Lasso regression has been used by a study (Huynh et al. [2019]) for the identification of a small number of highly influential genes and classification of cell types in scRNA-seq data with high accuracy. Huynh et al. [2019] used lasso regression to show that multiple subpopulations of cells existed at all stages of bone marrow-derived mesenchymal stem cells (MSC) chondrogenesis. However, this study did not include any other variants of the lasso algorithm such as elasticnet or drop lasso. The next section discusses studies conducted as a comparative analysis of machine learning methods.

2.4.1 Comparison studies involving penalized regression methods

Cao et al. [2021] conducted a systematic evaluation of methods for cell phenotype classification using single-cell RNA sequencing data. This study benchmarked methods such as Elastic net, Naive Bayes (NB) Classifier, and XGBoost (XGB). The result of this study concluded that Elastic Net with interactions performed best in small and medium data sets, NB worked well with medium data sets, and XGB worked well with large data sets. However, their study did not shed light on the applicability of penalized regression algorithms except Elastic net on scRNA-seq data.

Scialdone et al. [2015] evaluated five computational methods (RF, logistic regression, lasso regression, and SVM), and proposed a new classification method called Pairs using the relative expression of "marker pairs" for predicting the cell-cycle stage of single cells from their transcriptome. Their study showed that sophisticated approaches such as RF or SVM suffered from overfitting while PCA-based predictor and Pairs method achieved a strong enough regularization which captured a generalizable cell-cycle signature in the transcriptome.

2.4.2 New algorithms based on penalized regression

Apart from the major machine learning algorithms discussed earlier in this thesis, several other methods have been developed by combining those algorithms. A few of those methods are discussed below.

Hua et al. [2020a] proposed a novel method LAK for clustering single-cell RNA-seq data. LAK is a computational method that uses Lasso and K-means based feature selection methods. Their study demonstrated that LAK obtains a better performance in reliability, stability, convenience and accuracy compared with other computational approaches. Evidently, the research was about improving lasso but failed to investigate its variants.

Climente-González et al. [2019] proposed block HSIC (the Hilbert–Schmidt Independence Criterion) Lasso, a non-linear feature selector that overcomes drawbacks such as lack of parsimony, non-convexity and computational overhead related to scRNA-seq data. The results of their study showed that features selected by block HSIC Lasso retained more information about the underlying biology than those selected by other techniques. The drawback of their study is that block HSIC Lasso was only compared to HSIC Lasso and least-angle regression (LARS).

Li et al. [2011] proposed a new RNA-Seq based transcriptome assembly tool called IsoLasso. IsoLasso is a lasso algorithm with some additional constraints enforced. The study claims that IsoLasso has higher sensitivity and precision than the state-of-art transcript assembly tools.

Jiang et al. [2020] proposed a Bayesian Robit regression method with Hyper-LASSO priors (BayesHL) for feature selection in high dimensional genomic data with grouping structure. BayesHL discards more aggressively unrelated features than LASSO, and it makes feature selection within groups automatically without a pre-specified grouping structure. Results show that BayesHL outperforms alternative methods (including LASSO, group LASSO, supervised group LASSO, penalized logistic regression, random forest, neural network, XGBoost and knockoff) in terms of predictive power, sparsity and

the ability to uncover grouping structure. However BayesHL study used only Endometrial Cancer RNA-seq Data. In contrast, my thesis use 4 data sets from 3 different species to ensure the methods are being tested on a variety of data sets.

Evidently, none of the above studies discuss the applicability of many of the penalized regression methods selected for this research. More importantly, no study has investigated the combined performance of lasso algorithms.

2.5 Other ML methods in published research

Huang et al. [2021] discussed computational methods for distinguishing cell subtypes from the different pathological regions of non-small cell lung cancer on the basis of transcriptomic profiles. In their study, the random forest classifier reached a Matthew's correlation coefficient (MCC) of 0.922 by using 720 features, and the decision tree classifier reached an MCC of 0.786 by using 1880 features.

The popularity of deep learning methods for scRNA-seq data was studied by Zheng and Wang [2019]. They concluded that while Autoencoders are the dominant approach, methods based on deep generative models such as generative adversarial networks (GANs) are emerging in scRNA-seq data analysis.

Oller-Moreno et al. [2021] studied the emergence of Deep Learning algorithms and the incorporation of prior biological knowledge into Machine Learning models. They argued that prior knowledge can be seen as model

regularization that increases model generalization capabilities and stability while it supports the biological interpretation of results.

Along with many popular machine learning algorithms, neural networks also has been employed for analysing scRNA-seq data. Lin et al. [2017] tested different neural network architectures and used these to reduce the dimensionality of scRNA-seq data.

Studies were also conducted to reinforce the importance of machine learning in medical research and discussed its future potential. Asada et al. [2021] investigated the applicability of machine learning techniques such as a deep convolutional neural network, Latent semantic indexing cluster analysis which uses dimensional compression to normalize and cluster the data, and graphical lasso model for single-cell analysis. This study demonstrated the implementation and usefulness of all three methods.

2.5.1 New algorithms based on ML methods

Alquicira-Hernandez et al. [2019] proposed a new generalized method called scPred that claim to provide highly accurate classification of single cells, using a combination of unbiased feature selection from a reduced-dimension space, and machine-learning probability-based prediction method. However, this method has not been verified in comparison with existing algorithms.

Similarly, in another study, Tian et al. [2019] developed scDeepCluster which is a single-cell model-based deep embedded clustering method. This method simultaneously learns feature representation and clustering via explicit modelling of scRNA-seq data generation.

Yan et al. [2020] developed a machine learning method called Single cell Predictive markers (SPmarker) to identify novel cell-type marker genes in the Arabidopsis root. As per their sudy, SPmarker method identified hundreds of new marker genes that were not identified before. However, the performance of SPmarker has not been compared with other machine learning algorithms.

Aevermann et al. [2021] described a machine learning-based marker gene selection algorithm, NS-Forest version 2.0, which leverages the nonlinear attributes of random forest feature selection with a binary expression scoring approach to discover the minimal marker gene expression combinations that optimally capture the cell-type identity represented in complete scRNA-seq transcriptional profiles.

scRNA-seq data is often corrupted by dropout noise. As discussed earlier under penalized regression, droplasso method was developed to overcome this problem. Some studies have also been conducted to investigate the use of machine learning to impute missing data due to such dropout noise. Yang et al. [2018] proposed missing imputation for single-cell RNA (MISC) which is a robust missing data imputation model using hybrid machine learning for single-cell RNA-seq data. Their results showed that MISC improved the cell type classification.

Ensemble methods were used by Xiong et al. [2021] to address the problem of doublets in scRNA-seq data. There are multiple methods to detect doublets which happens when two cells are captured as one in scrNA-seq data. Xiong et al. [2021] proposed Chord which integrated multiple doublet detection methods and demonstrated higher accuracy.

In the study conducted by Hu et al. [2016], the Support Vector Machine

based recursive feature elimination (SVM-RFE) method of feature selection was used for the identification of key markers involved in brain development from scRNA-seq data.

2.6 Software packages for ML methods

Some studies analysed different software packages for machine learning algorithms for dealing with the challenges of scRNA-seq data. Petegrosso et al. [2020] conducted a study of machine learning and statistical methods for clustering single-cell RNA-sequencing data. Their study included clustering methods such as k-Means, BackSPIN, cellTree, CIDR, DendroSplit, ICGS, Monocle, pcaReduce, SC3, SCRAT, Seurat 1.0, and SNN-Cliq. Even though their study included many of the libraries in R, it did not consider Lasso algorithms for analysis.

Vrahatis et al. [2020] studied recent machine learning approaches for Single-Cell RNA-seq data analysis. This extensive study included 23 dimensionality reduction methods such as PCA, and LDA, 4 classification methods such as KNN, Random Forest, and bagging, 11 clustering methods such as Seurat, and Monocle, and 7 other popular analytical tools such as salmon. But even this comprehensive study conducted in 2020 also did not include penalized regression methods.

In conclusion, even though an abundance of literature has been published on machine learning and statistical methods addressing different challenges of scRNA-seq data, currently there is a definite lack of a comparative study of the penalized regression methods for feature selection of the scRNA-seq

data.

Chapter 3

Methodology

This chapter discusses the penalized regression algorithms that are selected for my study, explains how research design is developed to answer the research objectives and describes the proposed algorithm.

3.1 Penalized Regression

Simply speaking, the optimization function of penalized regression has two parts, a loss function, and a penalty. Let the regression equation be,

$$Y = \mathbf{X}\boldsymbol{\beta} + \epsilon, \tag{3.1}$$

where Y is a $n \times 1$ vector for response variable, $\mathbf{X}$ is $n \times p$ matrix for predictor variables, $\boldsymbol{\beta}$ is $p \times 1$ coefficient vector and $\epsilon \sim N(0, \sigma^2 I)$ is the error term. We consider that both $\mathbf{X}$ and Y are scaled in Eq. 3.1. The estimated penalized regression coefficients is given by,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||), \tag{3.2}$$

where $\lambda \geq 0$ is the tuning parameter to be estimated using cross-validation. $||\boldsymbol{\beta}||$ is the norm of coefficient vector $\boldsymbol{\beta}$. The first term $\frac{1}{n}||Y - \mathbf{X}\boldsymbol{\beta}||^2$ is the loss function and the second term $\lambda||\boldsymbol{\beta}||$ is the penalty. The main difference between different penalized regressions algorithms is how they apply the penalty. L1 norm and L2 norm are popular choices for penalty Khalfaoui and Vert [2018].

$$L1\ norm = ||\boldsymbol{\beta}||_1 = \sum_{i=1}^{p} |\boldsymbol{\beta}_i|, \tag{3.3}$$

$$L2\ norm = ||\boldsymbol{\beta}||_2^2 = \sum_{i=1}^{p} \boldsymbol{\beta}_i^2. \tag{3.4}$$

3.1.1 Ridge Regression

Ridge regression is a penalized regression where the penalty term is the sum of squared coefficients. It is especially useful when predictor variables are highly correlated. The estimator of ridge regression is,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_2^2), \tag{3.5}$$

where $||\boldsymbol{\beta}||_2^2$ is the L2 norm of the coefficients.

3.1.2 LASSO Regression

The Least Absolute Shrinkage and Selection Operator (LASSO) proposed by Tibshirani [1996] minimizes the residual sum of squares subject to the sum of the absolute coefficients being less than the tuning parameter. Compared to the ridge regression, which can only shrink coefficients towards 0, the lasso can make some coefficients exactly equal to zero thereby producing a more interpretable model. The estimator of lasso regression is,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1), \tag{3.6}$$

where $||\boldsymbol{\beta}||_1$ is the L1 norm of the coefficients.

3.1.3 Elastic Net

Elastic net regression proposed by Zou and Hastie [2005] is a combination of L1 and L2 penalties of lasso and ridge regression. The estimator of elastic net is,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda_1||\boldsymbol{\beta}||_1 + \lambda_2||\boldsymbol{\beta}||_2^2). \tag{3.7}$$

The elastic net regression enjoys the properties of both ridge and lasso regression. The L1 norm part of the penalty generates a sparse model and the L2 norm part of the penalty encourages greater shrinkage to large coefficients (Zou and Hastie [2003]).

3.1.4 Group LASSO

Yuan and Lin [2006] proposed Group lasso algorithm for selecting a subset of important variables which are the main effects with interactions with each

other in a regression model. Compared to the lasso model which select individual variables, group lasso select groups of variables. This is particularly useful in processing scRNA-Seq data because we would like to include or exclude the group of genes that lie in a pathway related to the outcome rather than individual genes. Assume there are $j = 1, 2, \cdots, J$ groups of variables and n observations. For each group, let $\mathbf{X}_j$ be $n \times p_j$ submatrix of $\mathbf{X}$ with columns corresponding to predictor variables in group j and $\boldsymbol{\beta}_j$ be the corresponding coefficient vector of length p_j. Then the regression equation for group lasso regression can be written as,

$$Y = \sum_{j=1}^{J} \mathbf{X}_j \boldsymbol{\beta}_j + \epsilon. \tag{3.8}$$

Note that for $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \boldsymbol{\beta}_2', \cdots, \boldsymbol{\beta}_j')'$, $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \cdots . \mathbf{X}_j)$, and $\mathbf{X}_j' \mathbf{X}_j = I_{p_j}$, the above regression equation simplifies to Eq. (3.1). For a symmetric and positive definite kernel matrix $K_j = p_j I_{p_j}$, the group lasso estimate is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\frac{1}{n} ||Y - \sum_{j=1}^{J} \mathbf{X}_j \boldsymbol{\beta}_j||^2 + \lambda \sum_{j=1}^{J} ||\boldsymbol{\beta}_j' K_j \boldsymbol{\beta}_j||^{\frac{1}{2}}), \tag{3.9}$$

where the tuning parameter is $\lambda \geq 0$.

3.1.5 Sparse Group LASSO

The shortcoming of group lasso is that while it gives a sparse set of groups, all the coefficients in a group will be nonzero if that group is included in the model. But sometimes both sparsity of groups and variables within each group are desired in a model simultaneously. In the case of scRNA-seq data, identifying some important genes in the biological pathways is of interest.

Simon et al. [2013] proposed sparse group lasso as a solution to this specific problem. The estimator of sparse group lasso is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \sum_{j=1}^{J} \mathbf{X}_j \boldsymbol{\beta}_j||^2 + \lambda \sum_{j=1}^{J} ||\boldsymbol{\beta}_j' K_j \boldsymbol{\beta}_j||^{\frac{1}{2}} + \alpha\lambda||\boldsymbol{\beta}||_1)), \quad (3.10)$$

where the tuning parameter is $\lambda \geq 0$, and $\alpha \in [0, 1]$ is a convex combination of the lasso and group lasso penalties. When $\alpha = 0$, the model reduces to group lasso and when $\alpha = 1$ it reduces to lasso which makes this approach remarkably similar to elastic net regression.

3.1.6 Drop LASSO

scRNA-seq data is often corrupted by dropout noise. Corruption of data here means incorrect values. For instance, dropout noise occurs when scRNA-seq fails to detect some genes even though they are expressed in the cell (Khalfaoui and Vert [2018]). Consequently, those genes (columns) will have zeros in the data set which is incorrect. Khalfaoui and Vert [2018] proposed Drop Lasso as a better-adapted solution to data corrupted by dropout noise. It is a combination of the dropout regularisation technique proposed by Srivastava et al. [2014] and lasso proposed by Tibshirani [1996]. It creates a sparse linear model robust to the noise by artificially augmenting the training set with new examples corrupted by dropout. First, a random permutation of rows is chosen from matrix $\mathbf{X}$ with n observations and p predictor variables. Then each of the chosen rows in $\mathbf{X}$ undergo an element-wise multiplication with a random dropout mask (vector of 1s and 0s) of length p to create a new dropout corrupted matrix $\mathbf{X}_{drop}$. The Drop Lasso estimator is calculated as,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \mathbf{X}_{drop}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1), \tag{3.11}$$

where the tuning parameter $\lambda \geq 0$.

3.1.7 Big LASSO

Zeng and Breheny [2017] proposed Big lasso algorithm for handling ultra-high dimensional and large scale data in R. Their approach handles out-of-core computation seamlessly by loading data into memory only when necessary while model fitting. This is done with the help of memory-mapped files which stores massive data on the disk. The Big lasso algorithm also possesses efficient feature screening rules which can accelerate the computation (Zeng and Breheny [2017]). The estimator of Big lasso regression is,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(\frac{1}{n}||Y - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1). \tag{3.12}$$

The major differences between Big lasso and lasso are in out-of-core computation and parallel processing.

3.2 Clustering

Clustering is the process of grouping data into clusters so that objects of the same cluster has high similarity (Rani[1] and Rohil [2013]). Two popular clustering methods are hierarchical clustering and K-Means clustering which are described below.

3.2.1 Hierarchical Clustering

In hierarchical clustering, objects are grouped into a hierarchy of clusters by nested sequential partitioning (Rani[1] and Rohil [2013]). These clusters are then graphically represented as a tree-like diagram known as dendrogram. The dendrogram is very useful in deciding the optimal number of clusters. In this study, we use hierarchical clustering to group genes into clusters prior to applying group lasso and SGL. This eliminates the need to have deep knowledge of genes to group them for processing in the algorithms.

3.2.2 K-Means Clustering

K-Means clustering algorithm starts by randomly placing k centroids in p data points scattered in the n-dimensional space (Hartigan and Wong [1979]). Clusters are formed by assigning data points to the nearest centroid. The algorithm progress iteratively by moving the centroids at each step such that the clustering error is minimized. K-Means clustering when applied in conjunction with Lasso improved prediction accuracy (Hua et al. [2020b]). In my study, K-means clustering is employed to cluster cells at the final step of the proposed methods. Note that the hierarchical clustering is used to group the genes while K-means clustering is used to cluster cells.

3.3 K-Fold Cross-Validation

Cross-validation is a data partitioning method used to estimate the prediction error of models and to tune model parameters (Bates et al. [2021]). We will

use K-Fold Cross-Validation (CV) for tuning the hyperparameters for all the algorithms in this study. In K-Fold CV, the data are first divided into k subsets of cells. One of the k subsets is used as the test set and the remaining $k-1$ subsets are used as the training set. Then prediction error is calculated for k repeats by selecting a different test set each time. The average error for k repeats is used as the prediction performance of the model. In this study, we use 10-fold CV for measuring the performance of the algorithms.

3.4 Metrics

This section explains the metric used to measure performance of my algorithms.

3.4.1 ROC AUC

A receiver operating characteristic curve (ROC) is a graph that plots the true positive rate (TPR) on the vertical axis and the false positive rate (FPR) on the horizontal axis (Park et al. [2004]). ROC is used to evaluate the diagnostic ability of a binary classification model. The area under the ROC curve is known as ROC AUC. The AUC value reflects the overall ranking performance of a classifier. The AUC is theoretically and empirically better than the accuracy metric for evaluating the classifier performance and discriminating an optimal solution during the training of a classifier. However, the computational cost of AUC is high relative to other evaluation metrics (Hossin and Sulaiman [2015]).

3.5 Research design

As the first step of the analysis, each data set is pre-processed to be compatible for the use in different R packages. With the processed data, we verify how each algorithm performs in terms of AUC and computation time when dealing with scRNA-seq data. We use the same performance measures (AUC and computation time) and CV for all data sets to ensure a fair comparison. In this step, a 10-fold Stratified CV is conducted to fine-tune the hyperparameters for each algorithm, and then the performance metrics for all the algorithms are calculated. We used hierarchical clustering for grouping variables before Group Lasso and Sparse Group Lasso. After comparing the performance metrics, better-performing algorithms are selected and combined to form a new algorithm. Cell clustering (using K-Means) with the final selected genes from the proposed algorithm is used to identify how well the cell groups in each data are differentiated by those genes. Finally, the performance metrics of the new algorithm are compared with that of the top-performing algorithm. Fig. 3.1 illustrates the proposed algorithm.

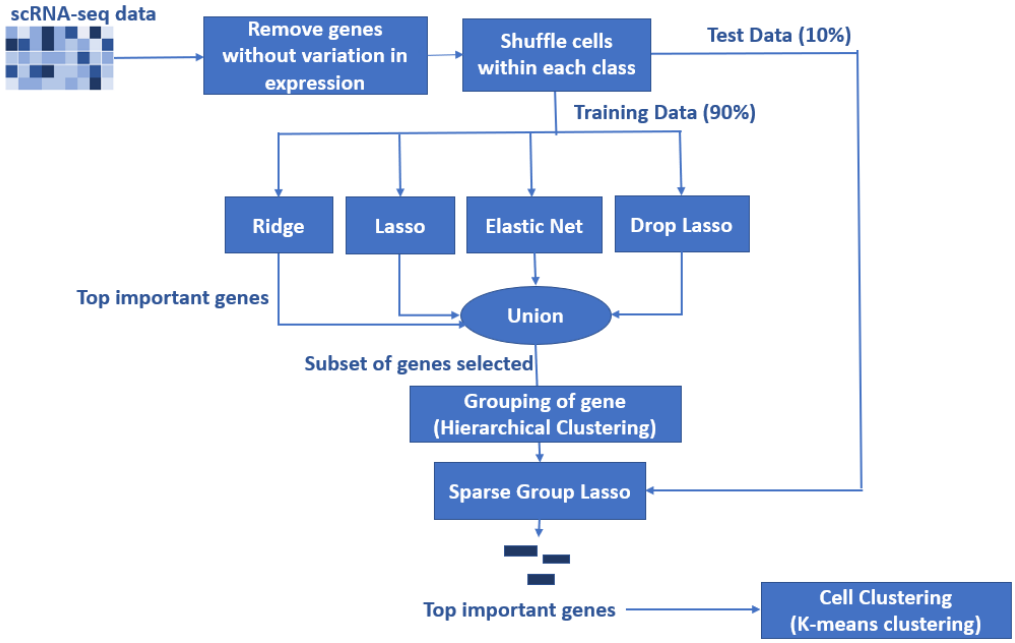

Figure 3.1: Schematic of the proposed algorithm. In this algorithm, there is a significant reduction in the number of genes prior to the execution of SGL. Once the final set of genes are selected by the algorithm, it is used to cluster cells in the data set.

Algorithm 1 Steps implementing the proposed algorithm

1. Load data to R and assign classes 1 and 0 to the two selected group of cells to form a binary classification problem.
2. Remove genes with no variability in expression across all cells.
3. Shuffle cells within each class to randomize data points.
4. Split 90% of the data set into training and 10% into testing data for 10-fold CV.
5. Repeat the steps below for the 10-fold CV
   (a) Fit ridge, lasso, elastic net, and drop lasso.
   (b) Select the top important genes for each algorithm. The top important genes are the genes which have coefficients above a cut off (e.g. mean of absolute value of the coefficients).
   (c) Form a gene pool by taking the union of the top important genes from all the 4 models. For example, Fig. 4.1 and Fig. 4.2 represent gene pool of data set GSE123818 and GSE71585, respectively.
   (d) Fit SGL with new gene pool grouped by hierarchical clustering algorithm.
   (e) Save the coefficients of SGL.
6. Find the average of coefficients for each gene across 10 folds and sort the genes.
7. Visualize the gene vs coefficients plot and select the final set of genes which are above the elbow of the curve.
8. Cluster all the cells by applying K-means clustering on the top important genes.

Chapter 4

Experiments and Results

In this chapter, we discuss the experimental data, results of the analysis, final subset of important genes selected with the new algorithm, their biological functions, and clustering of cells with this final subset of genes for each of the data set separately.

4.1 Experimental Data

In this study, 4 scRNA-seq data sets from 3 different species (Human, mouse and plant) are used. The first two data sets (GSE60749 and GSE71585) are selected from a collection of 40 curated scRNA-seq data sets from conquer website (http://imlspenticton.uzh.ch:3838/conquer/) created by Soneson and Robinson [2017]. Conquer website is a collection of consistently processed, analysis-ready and well documented publicly available scRNA-seq data sets. There are currently 40 data sets on conquer website, each having

Table 4.1: Experimental data sets

Data set	Genes	Cells	Cell Labels	Organism	Source
GSE60749	224444	183:84	mESCs : Dgcr8 -/- mESCs	Mus musculus	conquer
GSE71585	24058	79:57	Ntsr1 tdTpositive cells : Ntsr1 tdTnegative cells	Mus musculus	conquer
GSE81861	57241	272:160	Colorectal tumor cells : Normal mucosa cells	Homo sapiens	GEO
GSE123818	27629	1099:1099	Shortroot - knockout cells : Wild-type cells	Arabidopsis Thaliana	GEO

counts and transcripts per million (TPM) estimates for genes and transcripts, as well as quality control and exploratory analysis reports. The other two data sets (GSE81861 and GSE123818) for this study are downloaded from Gene Expression Omnibus (GEO). Tab. 4.1 shows the data sets with GEO accession numbers, number of genes, number of cells in each class, species and the source from where data was accessed.

The first data set GSE60749, downloaded from Gene Expression Omnibus, is of species Mus musculus. This data set was generated in the study of gene expression variability in pluripotent stem cells (PSCs) by single-cell expression profiling of PSCs under different chemical and genetic perturbations conducted by Kumar et al. [2014]. Gene expression levels are quantified

as transcripts per million reads (TPM).

For our research, we selected 183 individual v6.5 mouse embryonic stem cells (mESCs) and 84 Dgcr8 -/- mESCs that lack mature miRNAs (knockout of a miRNA processing factor). The 183 individual mESCs are assigned to class 1 and 84 Dgcr8 -/- mESCs are assigned to class 0 to create the binary classification problem. The data set included 22443 genes initially which was reduced to 15508 after data preprocessing in which all the genes with no variance in expression. Some of the genes in this data set are noncoding piRNAs with numbers as names. Such numbers are converted to text by prefixing them with 'RNA_', before loading data to R to make sure that the names are not converted as dates.

The second data set GSE71585 is also of the species Mus musculus (mouse). Our research included mouse species data because numerous clinical trials are conducted on mice prior to human trials. GSE71585 data set was generated by scRNA-Seq of adult mouse primary visual cortex in a study conducted by Tasic et al. [2016] where the objective was to understand cell-type diversity in the nervous system. There are 1809 cells and 24057 genes in the data set. Gene expression levels are quantified as transcripts per million reads (TPM). Out of all the cells, 79 Ntsr1_tdTpositive_cell are assigned to class 0 and 57 Ntsr1_tdTnegative_cell are assigned to class 1. After removing the genes which did not vary in expression across all cells, we left only 17870 genes.

The third data set GSE81861 is of species homo sapiens and is from the analysis of transcriptional heterogeneity in colorectal tumours (Li et al. [2017]). Intratumoral heterogeneity is a major obstacle to cancer treatment and a significant confounding factor in bulk-tumor profiling. Therefore, Li

et al. [2017] conducted an analysis of transcriptional heterogeneity in colorectal tumors and their microenvironments using scRNA–seq. There are 272 primary colorectal tumor cells and 160 matched normal mucosa cells. Gene expression levels are quantified as fragments per kilo-base per million reads (FPKM). A binary classification problem was created by assigning 272 primary colorectal tumor cells to class 1 and 160 matched normal mucosa cells to class 0. The data was then transposed to form a matrix of 432 rows (cells) and 57242 columns (genes). Standardization and normalization were not carried out for this data set because it negatively affected the performance of Lasso algorithms in preliminary analysis. All the genes which were not expressed (0 values) or equally expressed among all cells were removed, thereby reducing the number of genes to 38090.

Our last data set GSE123818 belongs to the plant species Arabidopsis Thaliana (thale cress). This data is obtained from the study of Spatiotemporal Developmental Trajectories in the Arabidopsis Root by Denyer et al. [2019]. The objective of this study was to distinguish Arabidopsis root cells by developmental fate and time. The study generated mRNA profiles of 6-day-old wild-type (wt) and shortroot-knockout (shr) Arabidopsis thaliana roots by deep sequencing of single cell and bulk RNA libraries (wild type only), in duplicate (bulk & wild-type single cell) and singlicate (shr-3), using Illumina NextSeq. From these 4727 wt cells and 1099 shr cells are selected at random for our research. Class 0 was assigned to wt cells and class 1 was assigned to shr cells to create the binary classification problem. Class 0 has 0 and class 1 has 1 assigned to the value of response variable. There are 27629 genes in the data set which were reduced to 24075 after removing the genes which did not vary in expression.

4.2 Results

The first objective of this study is to compare the performance of the selected methods. The results of this comparison in terms of average cross-validated AUC (CV-AUC) and computation time are shown in Tab. 4.2 and Tab. 4.4 respectively. Fig. 4.3 shows the average CV-AUC by algorithm across all 4 data sets. From Tab. 4.2 and Fig. 4.3 we observe that the top 5 algorithms in the order of importance are SGL, grplasso, droplasso, biglasso, and Lasso. As evident from Tab. 4.3, the variance of CV-AUC is close to 0 for all methods when rounded to 2 decimal points. Notice that SGL and grplasso outperform all other methods in terms of average CV-AUC, whereas ridge regression algorithm has the least average CV-AUC. This could be because grplasso and SGL incorporate grouping of genes information into model selection, whereas ridge regression treats all the genes equally. On performing Friedman test (a non parametric statistical test) with the CV-AUC results of the 7 methods, we found statistically significant difference between their performance at $p - value = 15\%$. A post-hoc Nemenyi test revealed that the difference in performance is between SGL and ridge regression. However, more data sets need to be analysed in future to verify these results.

The average computation time is the least for SGL and biglasso, while the most time-consuming algorithm is ridge regression. Ridge regression use all genes in its final model making the computation complex. On the other hand, SGL can make an entire group of genes, as well as some of the genes within selected groups, zero (0) resulting in a sparse matrix and lesser computation time. It is notable that computation time for GSE81861 data set is higher compared to that of GSE60749 for most of the algorithms due

to larger number of non zero coefficients in the former data set.

The second objective of this study is to combine the top performing lasso algorithms in order to improve the prediction AUC and gene selection. From the discussion of the results of the first objective, we see that SGL and grplasso are good candidates for forming a new algorithm. In terms of gene selection, SGL performs better than grplasso. SGL could identify the top important genes in one fold of 10-Fold CV, whereas grplasso takes multiple folds to get the same result. In other words, the results of top important genes changed for each fold of 10-fold CV for group lasso compared to SGL. SGL is therefore chosen over grplasso for the new computational algorithm. SGL achieve better AUC than biglasso in comparable time for data sets of size 63 MB to 252 MB when tested on a computer with 32 GB processor. Since biglasso did not show significant improvement in computation time compared to SGL, biglasso is not included in the new method.

In the new algorithm, we select the ridge, lasso, elastic net, and droplasso to form a filter which creates a gene pool with the number of genes in it significantly reduced when compared to the whole. The gene pool is formed by taking a union of top important genes from 4 algorithms because we observed that the top important genes have some variations between algorithms. A union of top important genes is, therefore, more likely to capture the important differentially expressed genes. The gene pool thus formed is grouped and used as input to SGL and then the AUC of SGL fit is calculated.

We execute the hierarchical clustering to group the genes in the gene pool and then run SGL. The use of hierarchical clustering avoids the need to have extensive knowledge about the genes in order to group them. We noted that each data set had a different number of gene groups and each gene

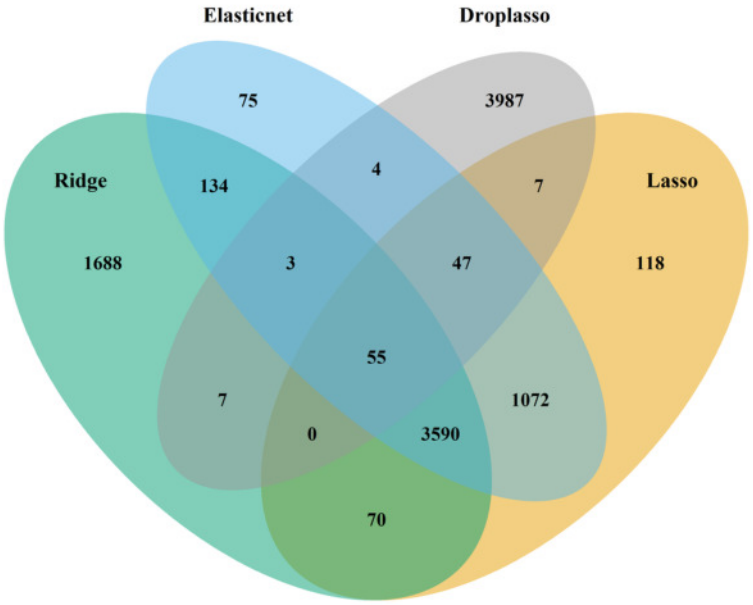

Figure 4.1: The gene pool of data set GSE123818 formed by taking the union of top important genes from ridge, lasso, elastic net, and droplasso.

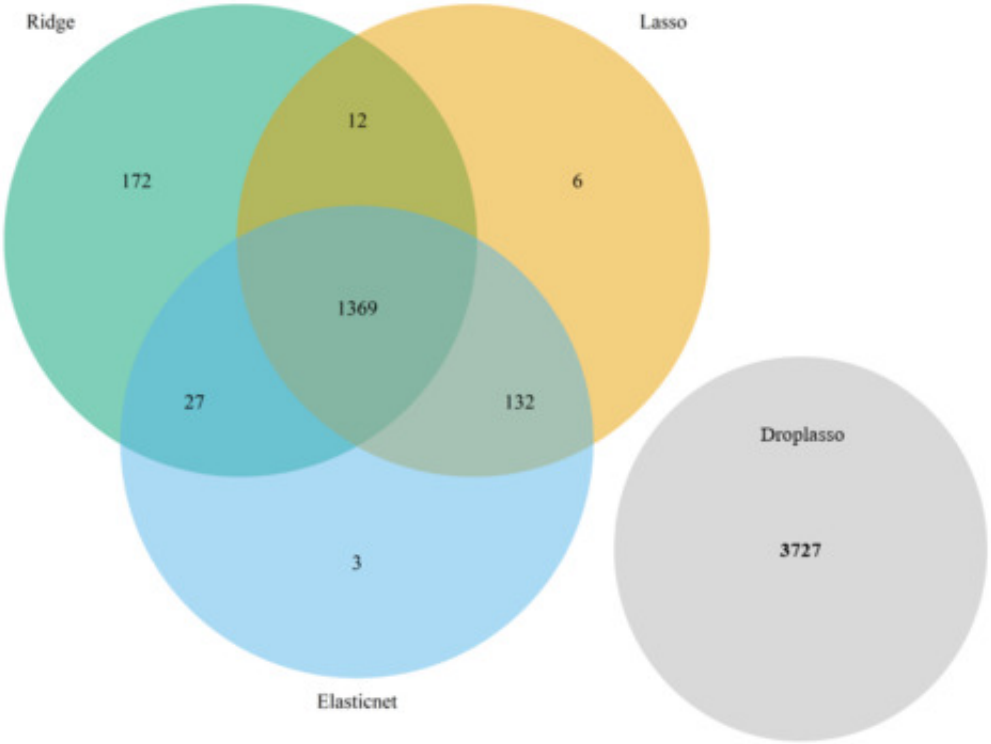

Figure 4.2: The gene pool of data set GSE71585 formed by taking the union of top important genes from the ridge, lasso, elastic net, and droplasso. Note that for this data set, the top important genes of droplasso have no intersection with the other 3 algorithms. In the proposed algorithm, gene pool is formed with the union of the top important genes from the 4 algorithms rather than an intersection because there may not always be an intersection due to the difference in regularization used.

group had different number of genes in them. This is acceptable given the differences between cells and species. There is a reduction of 61% to 92% in the number of genes in the gene pool compared to the whole for the 4 data set used in this study. Since hierarchical clustering and SGL are executed with a significantly reduced number of genes in the new algorithm, it enables us to execute these steps with an 8 GB processor for two data sets (GSEGSE60749, GSE71585) and get the same AUC as obtained with the 32 GB processor. It needs to be mentioned here that this new computational algorithm can be used for other high dimensional data sets as well for feature selection. Furthermore, the grouping of genes in the proposed algorithm can be done without expert knowledge of genes due to the use of hierarchical clustering. We note that the AUC of the proposed algorithm shown in Tab. 4.5 is equal to or better than that of SGL. The final selection of genes is found using a genes vs coefficients plot of SGL fit. Fig. 4.4, 4.5, 4.6, and 4.7 show the top important genes of 4 data sets. Note that these plots include the first 20 genes of the top important genes only for improved readability, and do not show all the genes above the cut off. In the validation step, we use K-Means clustering for cell clustering with the final selection of genes for each data set. The cell clustering of 4 data sets are shown in Fig. 4.8, 4.9, 4.10, and 4.11.

4.3 Application

As shown in Fig. 4.8, the data set GSE60749 is clustered with two classes well separated. Tab. 4.6 lists the final selection of genes for this data set. The top important genes of this data set identified by the new algorithm are

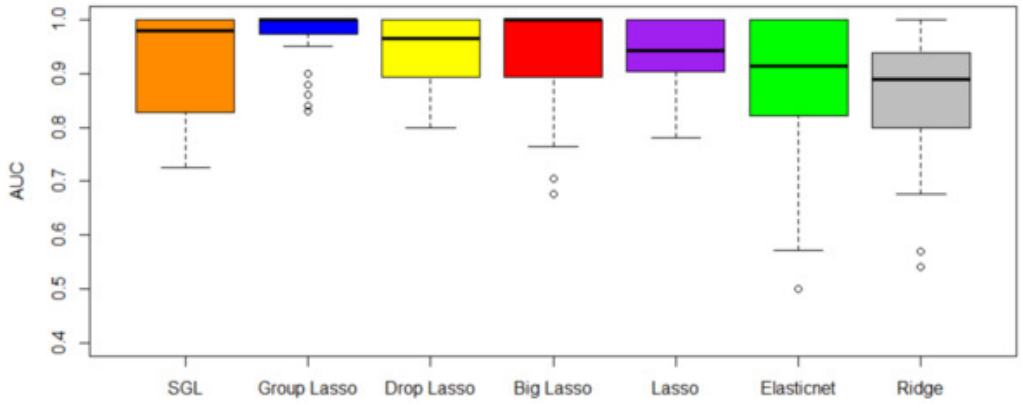

Figure 4.3: Average cross-validated AUC for the 4 data sets. Even though group lasso has better AUC than SGL, SGL is better in terms of gene selection. Selecting the differentially expressed genes is of more importance for a scRNA-seq data set compared to the prediction performance via AUC.

Table 4.2: Comparison of algorithm using CV-AUC.

Algorithm	R Package	GSE60749	GSE71585	GSE81861	GSE123818
Sparse Group Lasso	SGL	1	**0.98**	**0.92**	0.83
Group Lasso	grplasso	1	**0.98**	0.87	**0.99**
Drop Lasso	droplasso	0.99	0.94	0.87	**0.97**
Big Lasso	biglasso	1	**1**	0.80	0.95
Lasso	glmnet	1	0.96	0.85	0.94
Elastic Net	glmnet	1	0.63	0.86	0.93
Ridge	glmnet	0.99	0.84	0.71	0.90

Table 4.3: Variance of CV-AUC.

Algorithm	R Package	GSE60749	GSE71585	GSE81861	GSE123818
Sparse Group Lasso	SGL	0	0.0018	0.0018	0.0012
Group Lasso	grplasso	0	0.0018	0.0017	0.0000
Drop Lasso	droplasso	0.0004	0.0047	0.0006	0.0003
Big Lasso	biglasso	0	0	0.0061	0.0061
Lasso	glmnet	0	0.0029	0.0017	0.0003
Elastic Net	glmnet	0	0.0406	0.0021	0.0007
Ridge	glmnet	0.0004	0.0042	0.0089	0.0003

Table 4.4: Comparison of algorithm using computation time (Seconds).

Algorithm	R Package	GSE60749	GSE71585	GSE81861	GSE123818
Sparse Group Lasso	SGL	6.53	**1.73**	**2.97**	5.66
Group Lasso	grplasso	**1.12**	2.51	29.66	3.78
Drop Lasso	droplasso	13.57	7.18	59.33	**3.36**
Big Lasso	biglasso	3.11	4.77	7.23	20.30
Lasso	glmnet	3.18	2.76	13.39	48.54
Elastic Net	glmnet	3.57	2.66	13.59	51.51
Ridge	glmnet	58.07	26.71	3.77	17.58

Table 4.5: Comparison of performance (AUC) between SGL with all genes and SGL using new algorithm.

data set	All Genes	Gene Pool	SGL	New algorithm
GSE60749	224444	5965	1	1
GSE71585	24058	5448	0.98	1
GSE81861	57241	5823	0.92	0.94
GSE123818	27629	10857	0.83	0.85

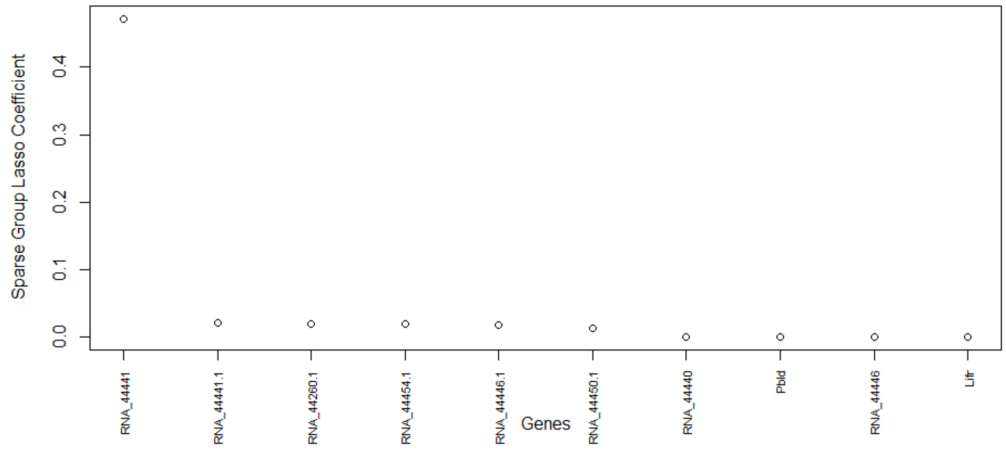

Figure 4.4: Genes vs coefficients plot of data set GSE60749. Here piRNA 44441 is the top important gene.

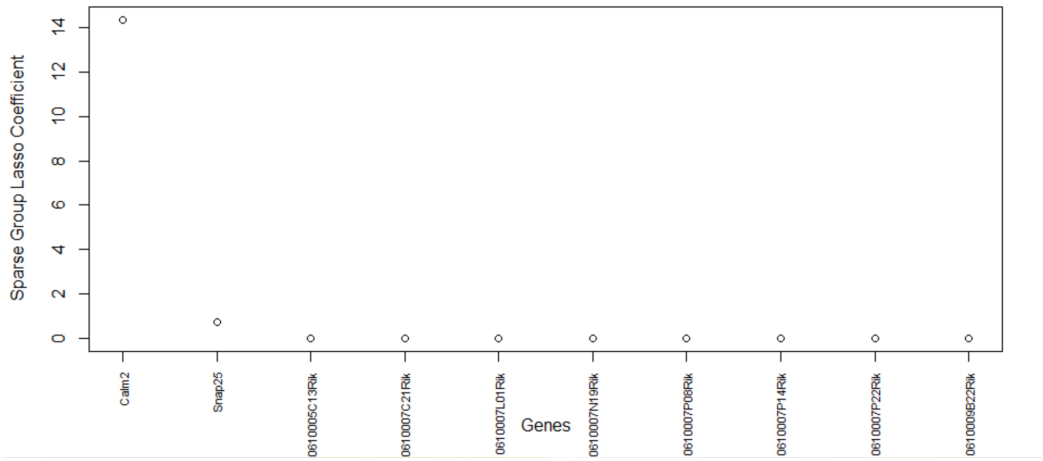

Figure 4.5: Genes vs coefficients plot of data set GSE71585. Calm2 and Spap25 are the top important genes.

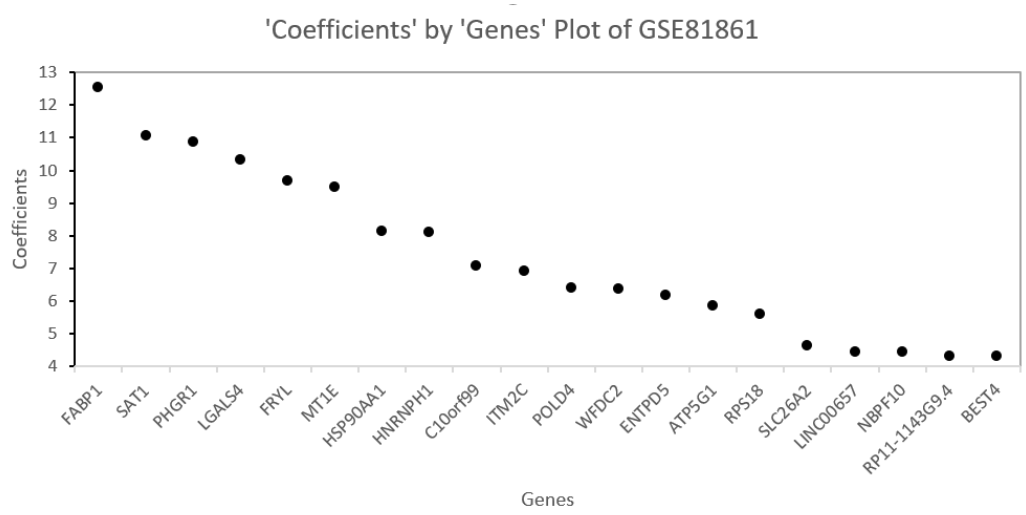

Figure 4.6: Genes vs coefficients plot of data set GSE81861. FABP1 and SAT1 are the first 2 top important genes.

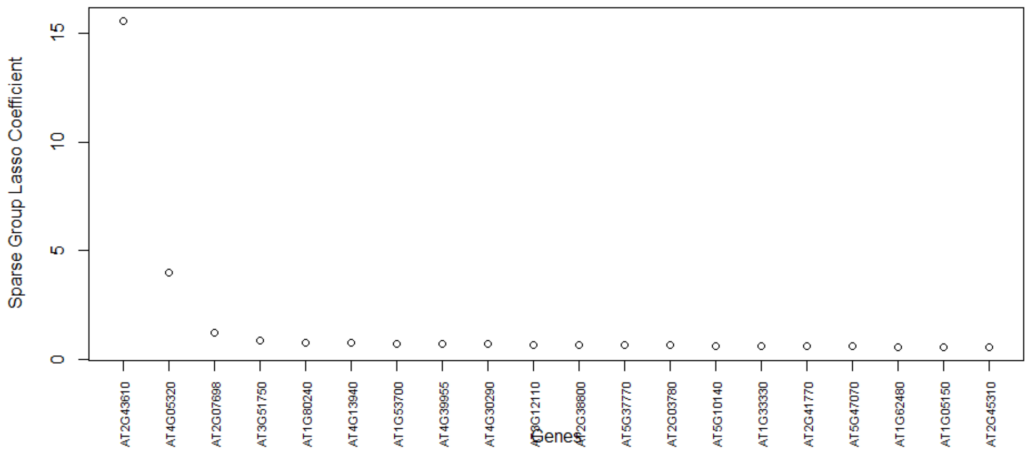

Figure 4.7: Genes vs coefficients plot of data set GSE123818. AT2G43610 and AT4G05320 are the top 2 genes.

44441, 44260, 44454, 44446, 44450, 44440, Pbld, Lifr, Hist2h4 and AK203176 as given in Tab. 4.6. One interesting finding is that 44441, 44260, 44454, and 44440 genes are non-coding RNAs called piRNA. This indicates a possibility of an association between knockout of miRNA processing factor and piRNAs which has not been studied before. The exact function of this RNA is still unknown. piRNAs are found in humans and rats, with major clusters occurring in syntenic locations. Although their function must still be resolved, the abundance of piRNAs in germline cells and the male sterility of Miwi mutants suggest a role in gametogenesis Girard et al. [2006].

The new algorithm is able to cluster primary visual cortex cell groups in the data set GSE71585 well (Fig. 4.9). Tab. 4.7 lists the final selection of genes. The top 2 differentially expressed genes in Ntsr1 (neurotensin receptor 1) tdT (tdTomato - an exceptionally bright red fluorescent protein) positive cells and Ntsr1 tdT negative cells are Calm2 and Snap25. Calm2 gene is active in cortex, frontal lobe and a few other organs. It enables N-terminal myristoylation domain binding, calcium ion and protein binding. It is active in the pathways of Alzheimer's disease and Glycogen Metabolism. As per NCBI, several infants with severe forms of long-QT syndrome (LQTS) who displayed life-threatening ventricular arrhythmias together with delayed neurodevelopment and epilepsy were found to have mutations in either this gene or another member of the calmodulin gene family (cal [2022]). Snap25 gene enables syntaxin-1 binding activity. It is present in 10 different biological pathways. NCBI records show that it is used to study attention deficit hyperactivity disorder, obesity, schizophrenia, and type 2 diabetes mellitus (sna [2022]). The human ortholog of this gene is implicated in Down syndrome and congenital myasthenic syndrome 18. The strong association between the lack of tdT protein in Ntsr1 cells and these genes identified by our computa-

Table 4.6: Final selection of genes from GSE60749

Gene	Function	Source
44441	piRNA. Function unknown.	ENA
44260	piRNA. Function unknown.	NCBI
44454	piRNA. Function unknown.	RNA Central
44446	Predicted gene. Function unknown.	RGD
44450	Predicted gene. Function unknown.	RGD
44440	piRNA. Function unknown.	piRNAdb
Pbld	Predicted to enable identical protein binding activity and isomerase activity. Predicted to be involved in maintenance of gastrointestinal epithelium; negative regulation of SMAD protein signal transduction; and negative regulation of transforming growth factor beta receptor signaling pathway.	NCBI
Lifr	Predicted to enable several functions, including ciliary neurotrophic factor receptor binding activity; growth factor binding activity; and leukemia inhibitory factor receptor activity. This gene has also been discussed in 9 pathways including ESC pluripotency pathways.	NCBI
Hist2h4	It encodes a replication-dependent histone that is a member of the histone H4 family (basic nuclear proteins responsible for the nucleosome structure of the chromosomal fiber in eukaryotes). This gene is found in Type II interferon signaling (IFNG) pathway.	NCBI
AK203176	Predicted to enable GTP binding activity; double-stranded RNA binding activity; and ubiquitin protein ligase binding activity. Acts upstream of or within cellular response to interleukin-4.	NCBI

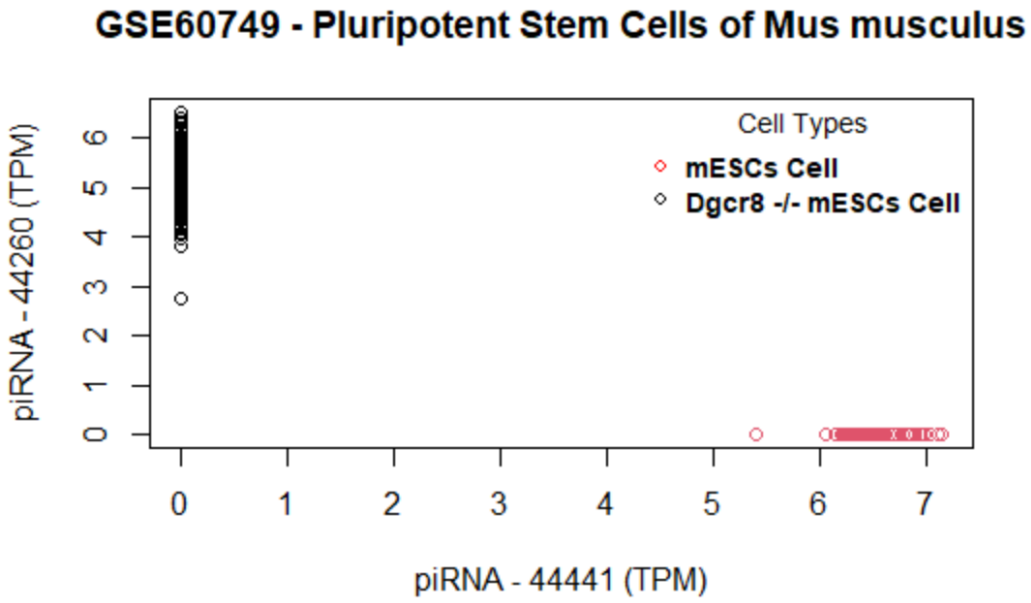

Figure 4.8: Cell clustering with final selection of genes for data set GSE60479. The top important gene (piRNA 44441) alone can differentiate two cell groups perfectly.

tional algorithm merits further study.

GSE81861 data set cell groups are clustered with some overlap in classes (Fig. 4.10). Tab. 4.8 lists the final selection of genes for this data set. The top important genes in this data set are FABP1, SAT1, PHGR1, LGALS4, FRYL, MT1E, HSP90AA1, and HNRNPH1. FABP1 gene enables long-chain fatty acids binding activity. It is involved in 13 biological pathways including metabolism, and Peroxisome proliferator-activated receptor (PPAR) signaling pathway fab [2022]. The second gene SAT1 is also involved in 13 biological pathways including metabolism. It catalyzes the acetylation of spermidine and spermine. Defects in this gene are associated with keratosis follicularis spinulosa decalvans (KFSD) sat [2022]. LGALS4 gene codes galectins which are a family of beta-galactoside-binding proteins implicated in modulating

Table 4.7: Final selection of genes from GSE71585

Gene	Function	Source
Calm2	This gene enables calcium-dependent protein binding activity. It is involved in Alzheimer's disease pathway and Glycogen metabolism pathway. It is also involved in several important processes including regulation of response to tumor cell. Human ortholog of this gene is implicated in long QT syndrome 15.	NCBI
Snap25	This gene enables syntaxin-1 binding activity. It is used to study attention deficit hyperactivity disorder; obesity; schizophrenia; and type 2 diabetes mellitus. Human ortholog of this gene is implicated in Down syndrome and congenital myasthenic syndrome 18. It is found in 10 different pathways.	NCBI
0610005C13Rik	This gene is expressed in several structures, including heart; intestine; liver; lung; and metanephros.	NCBI
0610007C21Rik	This gene is replaced with name Atraid. It is predicted to be involved in several processes, including negative regulation of osteoblast proliferation; positive regulation of bone mineralization; and positive regulation of osteoblast differentiation.	NCBI

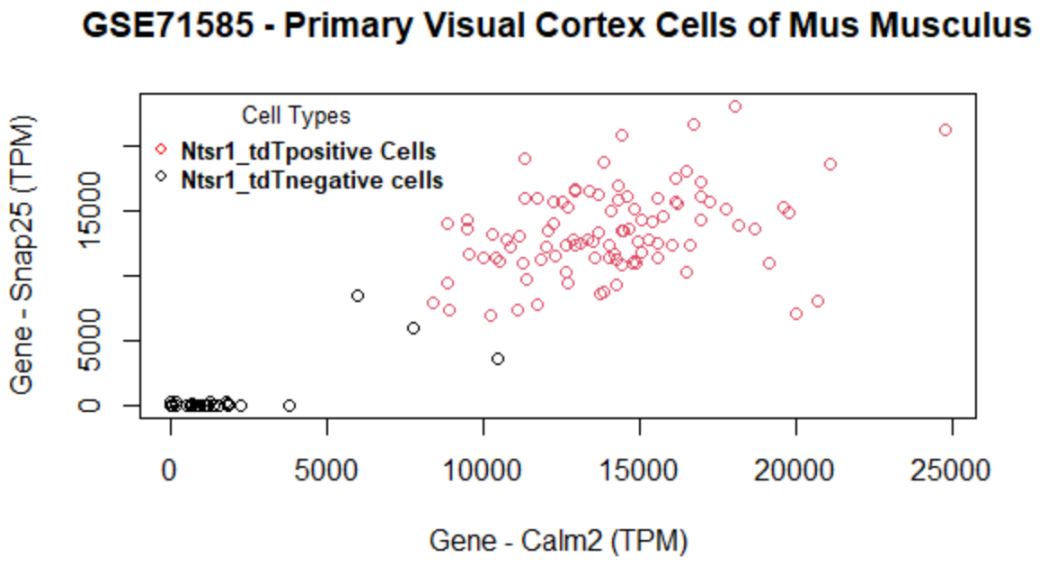

Figure 4.9: Cell Clustering (K-Means) with final selection of genes for data set GSE71585. The top important genes Snap25 and Calm2 are able to cluster two cell groups.

cell-cell and cell-matrix interactions. This gene is underexpressed in colorectal cancer lga [2022]. Similarly, HSP90AA1 gene is an important gene found in 115 pathways such as signaling by EGFR, EGFRvIII, and ERBB2in Cancer hsp [2022]. HNRNPH1 gene found in 12 pathways may be associated with hereditary lymphedema type I. Knockdown of heterogeneous nuclear ribonucleoprotein H1 (HNRNPH1) by siRNA inhibits the early stages of HIV-1 replication in 293T cells infected with VSV-G pseudotyped HIV-1 (hnr [2022]). Looking at Tab. 4.8, we also find a remarkable connection between the top important genes and HIV-1. Four genes (SAT1, MT1E, HSP90AA1, and HNRNPH1) out of 8 important genes from a colorectal tumor also have strong interaction with HIV-1 proteins. The association between cancer and HIV-1 has been widely studied by medical researchers (König et al. [2008], Nunnari et al. [2008], Corbeil et al. [1995], Alfano et al. [2013]). Evidently,

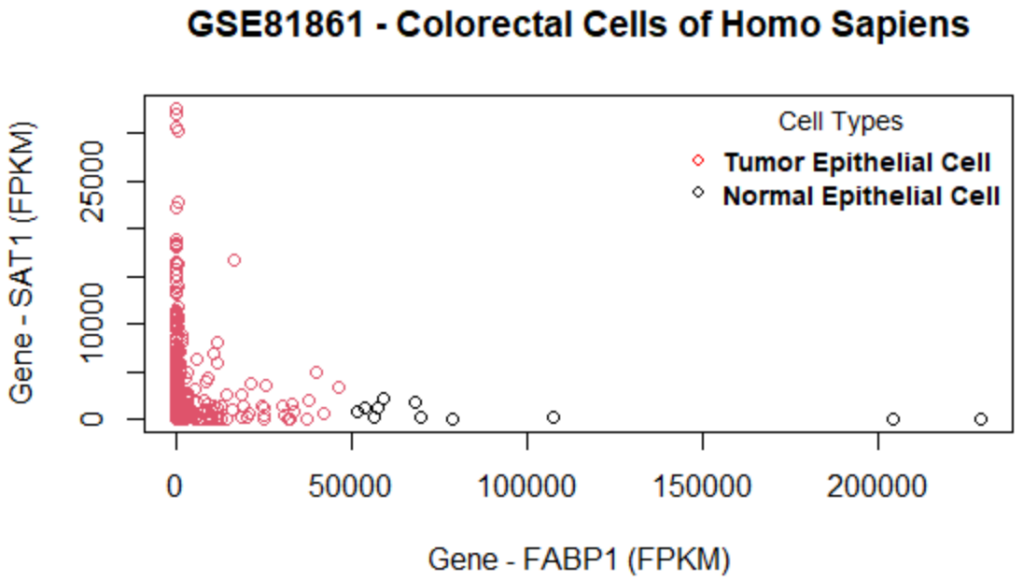

Figure 4.10: Cell Clustering (K-Means) with final selection of genes for data set GSE81861. There is some overlap between cell groups along the axis of top important genes SAT1 and FABP1.

the new algorithm is able to select a highly relevant subset of genes from the given samples of human colorectal cancer cells.

The cell clusters in GSE123818 have more overlap than the rest of the data sets. The clustering result is shown in Fig. 4.11. Tab. 4.9 lists the final selection of genes. The top important genes of this data set as found with the new algorithm are AT2G43610, AT4G05320, AT2G07698, and AT3G51750. One of five polyubiquitin genes in Arabidopsis thaliana, AT2G43610 gene is found in growth and developmental stages such as root development (AT2 [2022a]). AT4G05320 gene encodes the highly conserved 76-amino acid protein ubiquitin which is attached to proteins targeting degradation (AT4 [2022]). AT2G07698 gene is expressed during the seed development stage (AT2 [2022b]). AT3G51750 codes a hypothetical protein that is involved

Table 4.8: Final selection of genes from GSE81861

Gene	Function	Source
FABP1	This gene encodes the fatty acid binding protein found in liver. Biological pathways - 13, such as Peroxisome proliferator-activated receptor (PPAR) signaling pathway.	NCBI
SAT1	The protein encoded by this gene is a rate-limiting enzyme in the catabolic pathway of polyamine metabolism. Biological pathways - 13. It has HIV-1 interaction and KFSD.	NCBI
PHGR1	It is a protien coding gene with biased expression in colon and small intestine.	NCBI
LGALS4	The galectins are implicated in modulating cell-cell and cell-matrix interactions. The expression of this gene is restricted to small intestine, colon, and rectum. It is underexpressed in colorectal cancer.	NCBI
FRYL	This gene is predicted to be involved in cell morphogenesis and neuron projection development. It is predicted to be active in the site of polarized growth.	NCBI
MT1E	Biological pathways - 5, such as Zinc homeostasis, Copper homeostasis. HIV-1 Tat upregulates the interferon-responsive gene expression of Metallothionein, an effect that likely facilitates the expansion of HIV-1 infection.	NCBI
HSP90AA1	The protein encoded by this gene aids in the proper folding of specific target proteins. Biological pathways - 115, such as programmed cell death, innate immune system. It has strong interactions with HIV-1 proteins.	NCBI
HNRNPH1	This gene may be associated with hereditary lymphedema type I. Biological pathways - 12, such as mRNA processing. Knockdown of HNRNPH1 inhibits the early stages of HIV-1 replication in 293T cells.	NCBI

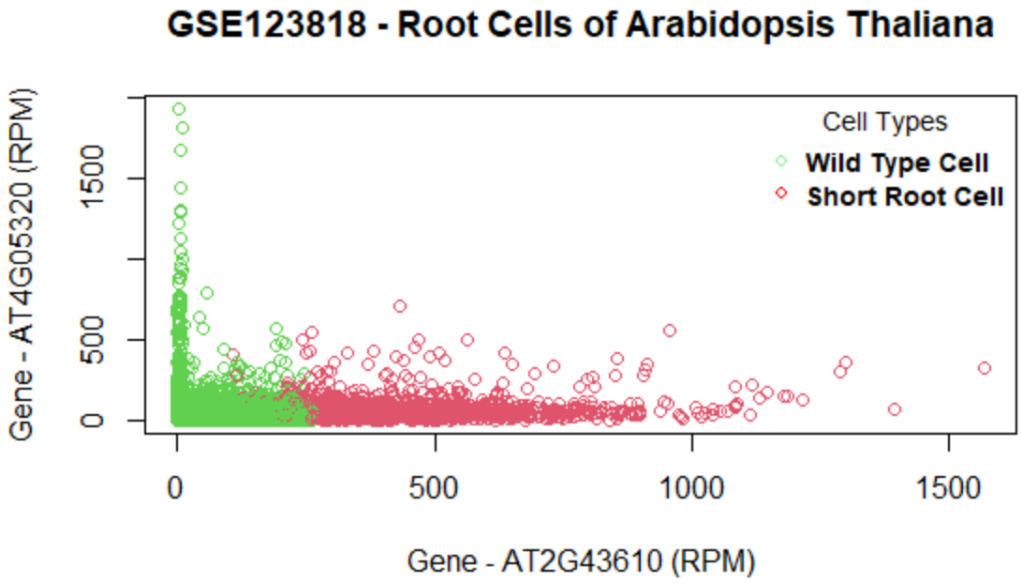

Figure 4.11: Cell Clustering (K-Means) with final selection of genes for data set GSE123818. There is notable overlap between the short root knockout and wild type cell groups of Arabidopsis Thaliana.

in root and seed development (AT3 [2022]). All the top important genes selected by the new computational algorithm are related to growth and developmental stages in Arabidopsis thaliana. We recommend further study of these genes in relation to root development and degradation.

Table 4.9: Final selection of genes from GSE123818

Gene	Function	Source
AT2G43610	This gene is found in growth and devlopmental stages such as root development. It enables chitinase activity and protein binding.	TAIR
AT4G05320	One of five polyubiquitin genes in Arabidopsis thaliana. These genes encode the highly conserved 76-amino acid protein ubiquitin that is covalently attached to substrate proteins targeting most for degradation. This gene enables mRNA binding, protein tag, ubiquitin protein ligase binding. The mRNA is cell-to-cell mobile.	TAIR
AT2G07698	This gene is expressed in growth and developmental stages such as seed and seedling development. It enables ADP binding, ATP binding, poly(U) RNA binding, and zinc ion binding.	TAIR
AT3G51750	This gene expressed during initial leaves visible stages and flowering stages. The biological processes associated with this gene are cellular lipid metabolic process, response to inorganic substance, response to light stimulus, root development, and seed development.	TAIR

Chapter 5

Conclusion

This research is conducted to address the need for a comparative analysis of penalized regression algorithms for scRNA-seq data. The algorithms chosen to study were ridge, Lasso, Elastic Net, Drop Lasso, Group Lasso, Sparse Group Lasso and Big Lasso regression. The research used a varied set of scRNA-seq data from 3 different species (Mus Musculus, Homo Sapiens, and Arabidopsis Thaliana) for analysis. The size of the data set varied from 63 MB to 252 MB. At the end of the analysis, it was found that the Sparse Group Lasso performed better compared to other methods in terms of average CV-AUC, computation time and selection of differentially expressed genes.

This study explored the possibility of improving the performance of the top algorithm by combining it with others. Upon analysing top important genes from all algorithms, it was found that the selection of genes may vary between algorithms to the point that there may not be any intersection in top important genes from different algorithms. Based on this result, a union of the top important genes from 4 algorithms (ridge, lasso, elastic net and

droplasso) was used to form a gene pool that had a significantly reduced number of genes which was then used as the input to Sparse Group Lasso. As evident from the results and discussion, the proposed algorithm that uses the sparse group lasso with a reduced set of genes can select a highly relevant subset of genes that are strongly associated with the cell clusters in scRNA-seq samples. The CV-AUC of the proposed algorithm was found better than that of the Sparse Group Lasso algorithm. Here we recognize that lasso algorithms have many hyperparameters which can be customized to arrive at different results. The output of grplasso and SGL packages can also change depending on the number of groups in the input data and the type of grouping used. This research has used scRNA-seq data from 3 species that are most frequently used in biomedical research. However, the proposed algorithm may need to be tested on data from more species.

5.1 Contributions

The major contributions of this thesis can be summarized as follows:

- The thesis produced a reliable guideline, about the comparative performance of penalized regression algorithms, which might be useful for the bioinformatics research community.
- The thesis has proposed a new algorithm that is a combination of lasso algorithms that showed better AUC that the top-performing lasso algorithm (SGL).
- The proposed algorithm does not require deep knowledge in grouping of genes in scRNA-seq data and yet identified a highly relevant set of

differentially expressed genes.

5.2 Future perspective

This research can take different directions in future. Some of them are listed below.

1. Current research can expand to include more algorithms and related R packages, such as Seagull developed by Klosa et al. [2020] which also implement lasso, group lasso and sparse group lasso algorithms.

2. Another direction worth exploring is verifying the R packages such as msgl (Vincent and Hansen [2014]) which can implement multinomial classification.

3. In the proposed algorithm, all of the training data is used as input to each of the 4 lasso algorithms. Instead, training data itself can be divided into 4 parts and one part each fed to the 4 lasso algorithms. Theoretically, this should reduce computation further since each algorithm processes only one-fourth of the training data. How this additional step would affect relevant gene selection need to be studied.

4. A scRNA-seq data set with 1.3 million cells was published by 10x Genomics [2017]. According to 10x Genomics, this is the largest scRNA-seq data set available as of date. Even this colossal data set is of size 3.93 GB only. This thesis research can be extended to such large data sets when the scRNA-seq data set of Gigabyte (GB) size becomes more commonly available in the future.

5. The performance of the proposed algorithm for scRNA-seq data with highly imbalanced classes may be verified against other methods created for rare class identification such as the ensemble of phalaxes method (Tomal et al. [2015, 2016]).

Bibliography

Shaked Slovin, Annamaria Carissimo, Francesco Panariello, Antonio Grimaldi, Valentina Bouché, Gennaro Gambardella, and Davide Cacchiarelli. Single-cell rna sequencing analysis: a step-by-step overview. *RNA Bioinformatics*, pages 343–365, 2021.

Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

Yasin Kaymaz, Florian Ganglberger, Ming Tang, Francesc Fernandez-Albert, Nathan Lawless, and Timothy Sackton. Hierfit: Hierarchical random forest for information transfer. *bioRxiv*, 2020.

Maziyar Baran Pouyan and Dennis Kostka. Random forest based similarity learning for single cell rna sequencing data. *Bioinformatics*, 34(13):i79–i88, 2018.

Xue-wen Chen and Jong Cheol Jeong. Enhanced recursive feature elimination. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 429–435. IEEE, 2007.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal*

of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective series b statistical methodology. *Journal of the Royal Statistical Society*, 73(3):273–282, 2011.

Beyrem Khalfaoui and Jean-Philippe Vert. Droplasso: A robust variant of lasso for single cell rna-seq data. *arXiv preprint arXiv:1802.09381*, 2018.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of computational and graphical statistics*, 22 (2):231–245, 2013.

Yaohui Zeng and Patrick Breheny. The biglasso package: A memory-and computation-efficient solver for lasso model fitting with big data in r. *arXiv preprint arXiv:1701.05936*, 2017.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Yuan Jiang, Yunxiao He, and Heping Zhang. Variable selection with prior information for generalized linear models via the prior lasso method. *Journal of the American Statistical Association*, 111(513):355–376, 2016.

Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. Initial sequencing and analysis of the human genome. 2001.

Jason A. Reuter, Damek V. Spacek, and Michael P. Snyder. High-throughput sequencing technologies. *Molecular Cell*, 58(4):586–597, 2015. ISSN 1097-2765. doi: https://doi.org/10.1016/j.molcel.2015.05.004. URL https://www.sciencedirect.com/science/article/pii/S1097276515003408.

Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell rna-seq technologies and related computational data analysis. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00317. URL https://www.frontiersin.org/article/10.3389/fgene.2019.00317.

Simone Picelli, Åsa K Björklund, Omid R Faridani, Sven Sagasser, Gösta Winberg, and Rickard Sandberg. Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature methods*, 10(11):1096–1098, 2013.

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. *Cell*, 161(5):1202–1214, 2015.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler,

Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):1–12, 2017.

Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson. Characterization of the single-cell transcriptional landscape by highly multiplex rna-seq. *Genome research*, 21(7):1160–1167, 2011.

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Petros Xanthopoulos, Panos M Pardalos, and Theodore B Trafalis. Linear discriminant analysis. In *Robust data mining*, pages 27–33. Springer, 2013.

Nguyen PT Huynh, Natalie H Kelly, Dakota B Katz, Minh Pham, and Farshid Guilak. Single cell rna sequencing reveals heterogeneity of human msc chondrogenesis: Lasso regularized logistic regression to identify gene and regulatory signatures. *bioRxiv*, page 854406, 2019.

Xiaowen Cao, Li Xing, Elham Majd, Hua He, Junhua Gu, and Xuekui Zhang. A systematic evaluation of methods for cell phenotype classification using single-cell rna sequencing data. *arXiv preprint arXiv:2110.00681*, 2021.

Antonio Scialdone, Kedar N Natarajan, Luis R Saraiva, Valentina Proserpio, Sarah A Teichmann, Oliver Stegle, John C Marioni, and Florian Buettner. Computational assignment of cell-cycle stage from single-cell transcriptome data. *Methods*, 85:54–61, 2015.

Jiao Hua, Hongkun Liu, Boyang Zhang, and Shuilin Jin. Lak: lasso and k-means based single-cell rna-seq data clustering analysis. *IEEE Access*, 8:129679–129688, 2020a.

Héctor Climente-González, Chloé Agathe Azencott, Samuel Kaski, Makoto Yamada, et al. Block HSIC Lasso: model-free biomarker detection for ultra-high dimensional data. *Bioinformatics*, 35(14):i427–i435, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz333. URL https://doi.org/10.1093/bioinformatics/btz333.

Wei Li, Jianxing Feng, and Tao Jiang. Isolasso: a lasso regression approach to rna-seq based transcriptome assembly. *Journal of Computational Biology*, 18(11):1693–1707, 2011.

Lai Jiang, Celia MT Greenwood, Weixin Yao, and Longhai Li. Bayesian hyper-lasso classification for feature selection with application to endometrial cancer rna-seq data. *Scientific reports*, 10(1):1–16, 2020.

Guo-Hua Huang, Yu-Hang Zhang, Lei Chen, You Li, Tao Huang, and Yu-Dong Cai. Identifying lung cancer cell markers with machine learning methods and single-cell rna-seq data. *Life*, 11(9):940, 2021.

Jie Zheng and Ke Wang. Emerging deep learning methods for single-cell rna-seq data analysis. *Quantitative Biology*, 7(4):247–254, 2019.

Sergio Oller-Moreno, Karin Kloiber, Pierre Machart, and Stefan Bonn. Algorithmic advances in machine learning for single-cell expression analysis. *Current Opinion in Systems Biology*, 25:27–33, 2021.

Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell rna-seq data. *Nucleic acids research*, 45(17):e156–e156, 2017.

K Asada, K Takasawa, H Machino, S Takahashi, N Shinkai, A Bolatkan, K Kobayashi, M Komatsu, S Kaneko, K Okamoto, et al. Single-cell analysis

using machine learning techniques and its application to medical research. biomedicines 2021, 9, 1513, 2021.

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. *Genome biology*, 20(1):1–17, 2019.

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

Haidong Yan, Qi Song, Jiyoung Lee, John Schiefelbein, and Song Li. Identification of cell-type marker genes from plant single-cell rna-seq data using machine learning. *bioRxiv*, 2020.

Brian Aevermann, Yun Zhang, Mark Novotny, Mohamed Keshk, Trygve Bakken, Jeremy Miller, Rebecca Hodge, Boudewijn Lelieveldt, Ed Lein, and Richard H Scheuermann. A machine learning method for the discovery of minimum marker gene combinations for cell type identification from single-cell rna sequencing. *Genome research*, 31(10):1767–1780, 2021.

Mary Qu Yang, Sherman M Weissman, William Yang, Jialing Zhang, Allon Canaann, and Renchu Guan. Misc: missing imputation for single-cell rna sequencing data. *BMC systems biology*, 12(7):55–63, 2018.

Ke-Xu Xiong, Han-Lin Zhou, Jian-Hua Yin, Karsten Kristiansen, Huan-Ming Yang, and Gui-Bo Li. Chord: Identifying doublets in single-cell rna sequencing data by an ensemble machine learning algorithm. *bioRxiv*, 2021.

Yongli Hu, Takeshi Hase, Hui Peng Li, Shyam Prabhakar, Hiroaki Kitano, See Kiong Ng, Samik Ghosh, and Lawrence Jin Kiat Wee. A machine

learning approach for the identification of key markers involved in brain development from single-cell transcriptomic data. *BMC genomics*, 17(13): 19–29, 2016.

Raphael Petegrosso, Zhuliu Li, and Rui Kuang. Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Briefings in bioinformatics*, 21(4):1209–1223, 2020.

Aristidis G Vrahatis, Sotiris K Tasoulis, Ilias Maglogiannis, and Vassilis P Plagianakos. Recent machine learning approaches for single-cell rna-seq data analysis. In *Advanced Computational Intelligence in Healthcare-7*, pages 65–79. Springer, 2020.

Hui Zou and Trevor Hastie. Regression shrinkage and selection via the elastic net, with applications to microarrays. *JR Stat Soc Ser B*, 67:301–20, 2003.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Yogita Rani[1] and Harish Rohil. A study of hierarchical clustering algorithm. *ter S & on Te SIT*, 2:113, 2013.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

Jiao Hua, Hongkun Liu, Boyang Zhang, and Shuilin Jin. Lak: lasso and k-means based single-cell rna-seq data clustering analysis. *IEEE Access*, 8:129679–129688, 2020b.

Stephen Bates, Trevor Hastie, and Robert Tibshirani. Cross-validation: what does it estimate and how well does it do it? *arXiv preprint arXiv:2104.00673*, 2021.

Seong Ho Park, Jin Mo Goo, and Chan-Hee Jo. Receiver operating characteristic (roc) curve: practical review for radiologists. *Korean journal of radiology*, 5(1):11–18, 2004.

Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

Charlotte Soneson and Mark D Robinson. Bias, robustness and scalability in differential expression analysis of single-cell rna-seq data. *bioRxiv*, page 143289, 2017.

Roshan M Kumar, Patrick Cahan, Alex K Shalek, Rahul Satija, A Jay DaleyKeyser, Hu Li, Jin Zhang, Keith Pardee, David Gennert, John J Trombetta, et al. Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature*, 516(7529):56–61, 2014.

Bosiljka Tasic, Vilas Menon, Thuc Nghi Nguyen, Tae Kyung Kim, Tim Jarsky, Zizhen Yao, Boaz Levi, Lucas T Gray, Staci A Sorensen, Tim Dolbeare, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature neuroscience*, 19(2):335–346, 2016.

Huipeng Li, Elise T Courtois, Debarka Sengupta, Yuliana Tan, Kok Hao Chen, Jolene Jie Lin Goh, Say Li Kong, Clarinda Chua, Lim Kiat Hon, Wah Siew Tan, et al. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nature genetics*, 49(5):708–718, 2017.

Tom Denyer, Xiaoli Ma, Simon Klesen, Emanuele Scacchi, Kay Nieselt, and Marja CP Timmermans. Spatiotemporal developmental trajectories in the arabidopsis root revealed using high-throughput single-cell rna sequencing. *Developmental cell*, 48(6):840–852, 2019.

Angélique Girard, Ravi Sachidanandam, Gregory J Hannon, and Michelle A Carmell. A germline-specific class of small rnas binds mammalian piwi proteins. *Nature*, 442(7099):199–202, 2006.

Calm2 calmodulin 2 [mus musculus (house mouse)] - gene - ncbi, 2022. URL https://www.ncbi.nlm.nih.gov/gene/12314.

Snap25 synaptosomal-associated protein 25 [Mus musculus (house mouse)] - Gene - NCBI, 2022. URL https://www.ncbi.nlm.nih.gov/gene/20614.

Fabp1 fatty acid binding protein 1, liver [Mus musculus (house mouse)] - Gene - NCBI, 2022. URL https://www.ncbi.nlm.nih.gov/gene/14080.

SAT1 spermidine/spermine N1-acetyltransferase 1 [Homo sapiens (human)] - Gene - NCBI, 2022. URL https://www.ncbi.nlm.nih.gov/gene/6303.

LGALS4 galectin 4 [Homo sapiens (human)] - Gene - NCBI, 2022. URL https://www.ncbi.nlm.nih.gov/gene/3960.

HSP90AA1 heat shock protein 90 alpha family class A member 1 [Homo sapiens (human)] - Gene - NCBI, 2022. URL https://www.ncbi.nlm.nih.gov/gene/3320.

HNRNPH1 heterogeneous nuclear ribonucleoprotein H1 [Homo sapiens (human)] - Gene - NCBI, 2022. URL https://www.ncbi.nlm.nih.gov/gene/3187.

Renate König, Yingyao Zhou, Daniel Elleder, Tracy L Diamond, Ghislain MC Bonamy, Jeffrey T Irelan, Chih-yuan Chiang, Buu P Tu, Paul D De Jesus, Caroline E Lilley, et al. Global analysis of host-pathogen interactions that regulate early-stage hiv-1 replication. *Cell*, 135(1):49–60, 2008.

Giuseppe Nunnari, Johanna A Smith, and René Daniel. Hiv-1 tat and aids-associated cancer: targeting the cellular anti-cancer barrier? *Journal of Experimental & Clinical Cancer Research*, 27(1):1–8, 2008.

Jacques Corbeil, Louise A Evans, Paul W McQueen, Eva Vasak, Paul D Edward, Douglas D Richman, Ronald Penny, and David A Cooper. Productive in vitro infection of human umbilical vein endothelial cells and three colon carcinoma cell lines with hiv-1. *Immunology and cell biology*, 73(2):140–145, 1995.

Massimo Alfano, Francesca Graziano, Luca Genovese, and Guido Poli. Macrophage polarization at the crossroad between hiv-1 infection and cancer development. *Arteriosclerosis, thrombosis, and vascular biology*, 33(6): 1145–1152, 2013.

The arabidopsis information resource (tair), 2022a. URL https://www.arabidopsis.org/servlets/TairObject?type=locus&name=At2g43610. on: www.arabidopsis.org.

The arabidopsis information resource (tair), 2022. URL https://www.arabidopsis.org/servlets/TairObject?type=locus&id=126703. on: www.arabidopsis.org.

The arabidopsis information resource (tair), 2022b. URL https://www.arabidopsis.org/servlets/TairObject?type=locus&name=At2g07698. on: www.arabidopsis.org.

The arabidopsis information resource (tair), 2022. URL https://www.arabidopsis.org/servlets/TairObject?type=locus&name=At3g51750. on: www.arabidopsis.org.

Jan Klosa, Noah Simon, Pål Olof Westermark, Volkmar Liebscher, and Dörte Wittenburg. Seagull: lasso, group lasso and sparse-group lasso regularization for linear regression models via proximal gradient descent. *BMC bioinformatics*, 21(1):1–8, 2020.

Martin Vincent and Niels Richard Hansen. Sparse group lasso and high dimensional multinomial classification. *Computational Statistics & Data Analysis*, 71:771–786, 2014.

10x Genomics. 1.3 million brain cells from e18 mice. single cell gene expression dataset by cell ranger 1.3.0., 2017. URL https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons?

Jabed H Tomal, William J Welch, and Ruben H Zamar. Ensembling classification models based on phalanxes of variables with applications in drug discovery. *The Annals of Applied Statistics*, 9(1):69–93, 2015.

Jabed H Tomal, William J Welch, and Ruben H Zamar. Exploiting multiple descriptor sets in qsar studies. *Journal of Chemical Information and Modeling*, 56(3):501–509, 2016.

Appendix A

Source Codes

The R codes for the proposed method is in the GitHub Repository below,

https://github.com/bhavithry/Benchmarking-LASSO-R