Scalable Feature Selection Methods by Augmenting Sparse Least Squares

by

©Hanieh Marvikhorasani

A Dissertation submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Memorial University of Newfoundland

November 2019

St. John's Newfoundland

Abstract

Feature selection has been used widely for selecting a subset of genes (features) from microarray datasets, which help discriminate healthy samples from those with a particular disease. However, most feature selection methods suffer from high computational complexity when applied to these datasets due to the large number of genes present. Usually, a small subset of these genes have a contributing factor to the disease, and the rest of the genes are irrelevant to the condition. This study proposes a sparse method, Sparse Least Squares (SLS), based on singular value decomposition and least squares to filter out irrelevant features. In this thesis, we shall also consider reducing the number of features by clustering genes and selecting representative genes from each cluster based on two different metrics. These dataset size-reduction methods are incorporated into three state-of-the-art feature selection methods, namely, mRMR, SVM-RFE, and HSIC-Lasso. These methods are applied to three Inflammatory Bowel Disease (IBD) datasets and combined with support vector machines and random forest classifiers. Experimental results show that the proposed SLS method significantly reduces the running time of feature selection algorithms and improves the prediction power of the machine learning models. SLS is integrated into a novel feature selection method (DRPT), which, when combined with Support Vector Machine (SVM), is able to generate models to discriminate between healthy subjects and subjects with Ulcerative Colitis (UC) based on the expression values of genes in colon samples. The best

models were validated on two validation datasets and achieving higher predictive performance than a model generated by a recently published biomarker discovery tool (BioDiscML).

Acknowledgements

I first thank my MSc supervisor, Dr. Hamid Usefi, for his guidance and supervision during my program.

I cannot thank enough Dr. Lourdes Peña-Castillo for guiding me and answering my questions especially in the latter part of this thesis.

I also thank Dr. Todd Wareham, one of the most amazing people I've seen in my life. I have learned a lot from him during the course 'Research Methods in Computer Science,' and It was such an honor being his TA for three semesters. I never forget all his supports and kindness.

I thank my friends for always being there for me during difficult days of research and homesickness. Their supports were always heartwarming.

Last but certainly not least, I express my greatest gratitude to my parents for their endless love and support throughout my entire life, and for sacrificing so much to get me where I am today. This accomplishment would not have been possible without them.

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

SVM	Support Vector Machine
RF	Random Forest
SVM-RFE	Support Vector Machine-Recursive Feature Elimination
mRMR	Minimum Redundancy Maximum Relevance
HSIC-Lasso	Hilber-Schmit Independence Criterion-Least Absolute Shrinkage and Selection Operator
DRPT	Dimension Reduction using Perturbation Theory
SLS	Sparse Least Squares
SVD	Singular Value Decomposition
CV	Cross Validation
APS	Average Precision Score
AUC	Area Under the Curve
IBD	Inflammatory Bowel Disease
UC	Ulcerative Colitis
CD	Crohn's Disease
kNN	k-Nearest Neighbors
GEO	Gene Expression Omnibus
CA	Classification Accuracy

Chapter 1

Introduction

This chapter provides an introduction to the research done in this thesis. Section 1.1 is an overview, and Section 1.2 describes the related work. Part 1 work done in this thesis has been accepted in the Eighteenth IEEE International Conference on Machine Learning and Applications (ICMLA 2019).

1.1 Overview

Gene expression datasets usually consist of tens or hundreds of samples compared to thousands or tens of thousands of features. This impacts the performance of the classifier [52] and can also cause data overfitting [20]. To improve the performance of the classifier and reduce the dimensionality of the dataset, feature selection methods can be used to find a subset of features that are more informative and relevant to class labels [33]. So, in addition to improving the performance of the classifier, feature selection avoids overfitting. Feature selection algorithms fall into three different categories: filter methods, wrapper methods, and embedded methods. Filter methods are independent of classifiers and select a subset before performing any classification. Relief-based methods [34] such as Minimum Redundancy Maximum Relevance

(mRMR) [43], Relief [32] and ReliefF [31] are well-known filter methods. Wrapper methods [33] select a subset and estimate the score of the subset by employing the performance of the classifier. These methods are useful but have high computational complexity since many subsets of features need to be assessed by a classification algorithm [22]. Usually, evaluating each subset in wrapper methods is done using the k-fold cross-validation method. Since the k-fold cross-validation metric is used for evaluating the models in this thesis, this concept is briefly described below.

In k-fold cross-validation, data is split into k partitions of equal size called folds. A sequence of models is then generated. For example, the first model is created using the last $k-1$ folds as the training set and the first fold as a test set. The data in the training set is used for creating the model, and the model is validated on the test set. The other $k-1$ models are created with the same procedure, with one fold used as a test set, and the remaining $k-1$ folds are used for training. Each fold is used once as a test set. Figure 1.1 shows an illustration of 5-fold cross-validation.

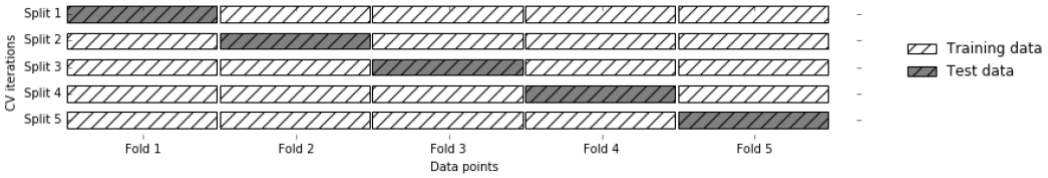

Figure 1.1: Data splitting in 5-fold cross-validation adopted from [40].

A well-known wrapper method is the Support Vector Machines Recursive Feature Elimination (SVM-RFE) [20] algorithm. This algorithm repeatedly constructs the model and eliminates features with low ranks. Filter methods are faster than wrappers and computationally suitable for large datasets [22]. In embedded methods, feature selection is incorporated into the model generation process, falling between filter and wrapper methods in terms of computational complexity. The effectiveness of different feature selection algorithms on gene datasets has been investigated in various studies

[26, 49, 58].

Microarray datasets usually contain many irrelevant genes which show little difference between cases and controls, and have no contribution to the prediction power of the model [28]. In other words, the entropy of irrelevant genes is big, and as such, these genes are not useful in discriminating between cases and controls. Not only does the existence of irrelevant features add to the computational cost of machine learning algorithms, but irrelevant features can also be viewed as noise in the dataset that negatively affects the classification algorithms.

This thesis proposes a sparse method for the removal of irrelevant features. Let $D = [A \mid \mathbf{b}]$ be a dataset where $\mathbf{b}$ is the class label and A is an $m \times n$ matrix with m rows (samples) and n columns (genes). We solve the system $AX = \mathbf{b}$ using the method of least squares and singular value decomposition, where $X = [x_1, \ldots, x_n]^T$ is the vector of unknowns. So each x_i is viewed as an assigned weight to the feature F_i. The bigger $|x_i|$, the more important the feature F_i is in connection with the outcome column $\mathbf{b}$. It then makes sense to filter out those features whose weights are very small; that is, we shrink the weights of irrelevant features to zero. This process yields a sparse method, which we call SLS (Sparse Least Squares) to reduce the size of the dataset.

SLS can be augmented to any feature selection algorithm. Of particular interest are feature selection algorithms that have great prediction power, however suffer from high computational cost. These algorithms include wrapper methods, such as SVM-RFE, and methods based on information gain, such as mRMR.

Truncated Singular Value Decomposition (SVD) has been used in the literature to reduce the size of datasets [3]. The Principal Component Regression is also based on the Truncated SVD, where eigenvectors corresponding to the largest singular values are considered. It should be noted that Truncated SVD changes the values of features,

whereas the objective of this research is to preserve features as they are and only remove the irrelevant ones.

Another method that can be used to reduce the size of large datasets is clustering. Clustering involves partitioning the data into a certain number of groups, resulting in similar data being clustered together. Clustering has three main categories: partitioning algorithms, hierarchical algorithms, and density-based algorithms. Partitioning algorithms split the data into a user-specified number of clusters (k). The number of clusters should be given as an input parameter for these algorithms. An example of this is the k-means method. Hierarchical algorithms, by comparison, create a hierarchical decomposition of the data. There are two different approaches for constructing the hierarchical decomposition, which are agglomerative (bottom-up) and divisive (top-down) approaches. The number of clusters should not be given as input for these methods, but the termination conditions for merging or dividing processes should be specified. Density-based methods, such as DBSCAN [14], construct clusters based on connectivity and density functions.

Clustering genes helps summarize the dataset and group thousand of genes into a much smaller number of clusters. It also assists in the understanding of systematic effects and predicting gene function [46]. For this reason, gene clustering has been the focus of various studies [46,53]. For this research, k-means clustering algorithm is used to cluster genes. Then, representative genes from each cluster are selected, and two different approaches are employed to select genes from each cluster. A natural way to select representative genes is to select a proportion of each cluster closest to the cluster center. The empirical results of this study reveal that a better way to select representative genes is to use entropy. Even though clustering techniques are useful in reducing the computational cost and possibly improving the prediction power of models, this study demonstrates that a sparse method based on the method of least

squares outperforms the clustering approaches.

Three Inflammatory bowel disease (IBD) datasets are used to show the effectiveness of this approach. IBD refers to a group of diseases that involve inflammation in the intestines. There are two major subtypes of this disease: Ulcerative colitis (UC) and Crohn's disease (CD) [57]. These datasets contain the expression profile of healthy and UC samples and are obtained from the Gene Expression Omnibus (GEO) database. After reducing the size of datasets using either the SLS method or a clustering approach, feature selection is applied to the reduced dataset. Finally, a model is created with the selected subset of features, using Support Vector Machine (SVM) [10] and Random Forest (RF) [7] algorithms. Empirical results show that augmenting feature selection methods with SLS reduces the computational cost while maintaining or improving the prediction power of machine learning models.

1.2 Related Work

DNA microarray datasets are essential research tools, but the problem is the amount of data they produce, which is an obstacle to the interpretation of the results [46, 58]. To better understand the data, there have been some studies on gene selection and gene clustering.

There have been studies investigating the performance of various feature selection methods on microarray datasets. In [58], the performance of feature selection methods including CFS, χ^2-Statistic, Information gain, Symmetrical uncertainty, and RielifF with the combination of multiple classifiers are examined on diffuse large B-cell lymphoma and acute leukemia datasets to show advantages and disadvantages of each feature selection approach. Their results showed that the major drawback of filter methods is that they evaluate each gene separately from others. They suggested

filter methods are good for fast analysis of data and, wrappers are suggested to use for selecting genes that can be investigated for cancer treatment. Also, the classification performance of several feature selection methods on both synthetic and real gene datasets is evaluated in [26]. Another study compares the performance of several feature selection methods to find strong genes that contribute to better performance of a particular classifier [49].

Some studies propose feature selection methods based on SVD [18] and Total Least Squares (TLS) [19] in different contexts. The authors in [9] proposed a method for selecting a subset of features utilizing SVD for speaker identification. They showed their proposed method outperforms other methods which are based on F-ratio. A feature subset selection based on TLS was proposed in [54]. They compared their method (SAB-TLS) with the subset selection algorithm (SA-LS) proposed in [18]. They showed that when the data is perturbed, their method outperforms SA-LS. SVD was also used for analyzing gene expression data [17, 39]. Authors in [39] proposed a method named robust SVD (rSVD) for analyzing microarray data, which is robust to outliers and missing values. A regression modeling based on SVD is also developed in [17] to find the association of gene expression measurements with the tumor type. This thesis proposes a sparse method based on SVD and least-squares to remove irrelevant features from the dataset and consequently to reduce the size of the dataset before applying feature selection.

Since the computational complexity of applying feature selection algorithms to gene datasets are high, clustering genes before gene selection can lower it significantly, and reduce redundancy among genes. The goal of clustering is to find a natural grouping in data without knowledge of any class labels such that features in the same cluster are more similar to each other than those from different clusters [35, 45].

There has been some previous work on comparing different methods of clustering

on various cancer datasets [46, 53]. Four different clustering algorithms, including k-means, CRC, ISA, and memISA, are examined on three brain expression datasets in [46]. Their results showed that k-means clustering was the most effective of the four methods for typical microarray brain expression datasets.

There have been some recent works on feature selection based on feature clustering [12,27,50]. Features are clustered using hierarchical clustering, and a filter method is applied to each cluster to rank features and select representative features in [12]. They tested their algorithm on different UCI machine learning datasets using the kNN classifier. Also, another feature selection based on clustering was proposed in [27]. Their proposed method was an enhancement in the SVM-RFE gene selection method. There are three stages in their proposed method: clustering genes with k-means, creating a representative set, and then ranking representative genes with SVM-RFE. They applied their method on various cancer datasets and compared it with different feature selection methods. Their results showed that the proposed method decreases computational complexity and redundancy among genes. A clustering-based feature selection method was proposed in [50]. First, irrelevant features were removed from the dataset, and features were clustered with minimum spanning tree method, and then representative features were selected from clusters. We study the idea of clustering gene before feature selection in more details. We use two metrics for removing redundant features and reducing the size of the dataset using clustering. Genes are first clustered with the k-means clustering algorithm, and then representative genes are selected from each cluster. In this approach, representative genes are selected based on two different metrics, consisting of Euclidean distance and entropy.

1.3 Outline of the Thesis

This thesis extends the idea of redundant features removal using clustering and also proposes a sparse method for the removal of irrelevant features before the feature selection step. Both approaches are explained in detail in Chapter 2. In Chapter 3, we compare the effect of augmenting SLS method and the other approach to different feature selection methods. Chapter 4 presents a case study. In the end, Chapter 5 concludes this thesis.

Chapter 2

Methodology

Microarray datasets usually contain a significant number of irrelevant and redundant genes that negatively impact feature selection and machine learning models, both computationally and prediction power. For this reason, in this chapter, a method for reducing the size of the dataset by removing potentially uninformative features is proposed.

2.1 SLS Method

Let $D = [A \mid \mathbf{b}]$ be a dataset where $\mathbf{b}$ is the class label and A is an $m \times n$ matrix with m rows (samples) and n columns (genes). The i-th column (feature) of D is denoted by $\mathbf{F}_i$. Our objective is to remove those columns of A that do not have a significant impact on $\mathbf{b}$. We consider the linear system $AX = \mathbf{b}$, where $X = [x_1, \ldots, x_n]^T$ is the vector of unknowns. Since the system $AX = \mathbf{b}$ may not have exact solutions, instead we find the unique solution with the smallest 2-norm that satisfy the least squares problem over all X:

$$||AX - \mathbf{b}||_2, \tag{2.1}$$

This minimization problem is known as the method of least squares and its solutions is defined via singular value decomposition (SVD) of A. Recall that the SVD of an $m \times n$ matrix A is of the form $A = USV^T$, where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and $S = \text{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$ is an $m \times n$ diagonal matrix. Also recall that the Moore-Penrose inverse of A is the $n \times m$ matrix $A^+ = VS^{-1}U^T$, where $S^{-1} = \text{diag}(\sigma_1^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0)$.

It is well-known that the least squares solutions can be given in terms of the Moore-Penrose inverse, see [55].

Theorem 2.1.1 (All Least Squares Solutions)**.** *Let A be an $m \times n$ matrix and $\boldsymbol{b} \in \mathbb{R}^m$. Then all the solutions of $min_X||AX - \boldsymbol{b}||_2$ are of the form $y = A^+\boldsymbol{b} + q$, where $q \in \ker(A)$. Furthermore, the unique solution whose 2-norm is the smallest is given by $z = A^+\boldsymbol{b}$.*

We view a solution $X = \begin{bmatrix} x_1, \ldots, x_n \end{bmatrix}^T$ to problem (2.1), as a weight vector. In other words, we can approximate the label column b as a linear combination $x_1\mathbf{F}_1 + \cdots + x_i\mathbf{F}_i + \cdots + x_n\mathbf{F}_n$. Intuitively, the larger $|x_i|$ the more impact the feature $\mathbf{F}_i$ has on $\mathbf{b}$. As such, we filter out those features whose corresponding weight is less than a threshold as irrelevant features. In other words, we shrink the weights of irrelevant features to zero. This process yields a sparse method which we call SLS to reduce the size of dataset.

Next, the feature selection algorithm is applied to the reduced dataset. In other words, SLS is augmented to the existing feature selection methods.

One can tune the threshold parameter depending on the dataset, although our

Algorithm 1: Augmenting SLS to feature selection algorithms

Data: $D = [A \mid \mathbf{b}]_{m\times(n+1)}$
Result: Subset of features
F_i: i-th column of A;
$X = A^+ \times \mathbf{b}$, where A^+ is the Moore-Penrose inverse of A ;
Threshold $=0.4 * \max(|X|)$;
Irrelevant $=\{i \mid x_i < \text{Threshold}\}$;
Index $=\{1, \ldots, n\} \setminus$ Irrelevant;
$\hat{D} = [A(\text{Index}) \mid \mathbf{b}]$;
Apply feature selection algorithm to the reduced dataset $\hat{D}$;

experiments show that even a soft threshold is enough to reduce computational times of feature selection algorithms and as well increase the prediction power of classifiers on selected features. The procedure of augmenting SLS to any feature selection algorithm is described in Algorithm 1.

2.2 Removing Redundant Features Using Clustering

Clustering algorithms have been applied to microarray datasets frequently. Clustering is useful to find natural grouping among genes, so genes in each cluster are similar to each other and different from genes in other groups. We use gene clustering for reducing the size of the dataset by removing redundant features. Genes are first clustered using the k-means algorithm, and then representative genes are selected from each cluster. We use Euclidean distance and entropy as two criteria to rank the genes in each cluster, and then high ranked genes are selected from each cluster. The idea of clustering genes and selecting representative genes from clusters using Euclidean distance was proposed in [27], and they incorporated this clustering step to the SVM-RFE feature selection method. A slightly different approach is taken in this thesis.

The difference between the two works is that we select more than one representative gene from each cluster, and also we incorporate another criterion, entropy, in addition to Euclidean distance for selecting representative genes. Entropy was used in [12] to find the similarity between features. Another difference is an evaluation of the effect of adding this dataset size reduction step on the performance of three well-known feature selection methods using two classifiers is made. Experiments show a comprehensive comparison between different combinations of dataset size reduction methods, feature selection methods, and classifiers. Euclidean distance was used in [27] to find the closest gene from each cluster center as representative of the cluster.

Entropy comes from information theory [48] and is calculated as below:

$$H = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{2.2}$$

Entropy is a function of probabilities (p_i) which $\sum_{i=1}^{m} p_i = 1$, and $p_i \geq 0$. Considering the probabilities of all observations (m) for each random variable, in [2], Section 2.2.1 they proved that the minimum entropy is occurred when there is one nonzero p_i. Also, [2] in Section 2.3 showed that the maximum entropy occurs when all the probabilities are identical and the entropy would be equal to $\log_2 m$ (logarithm of the number of observations).

Entropy has been applied in various research areas: physics, chemistry, and also bioinformatics. For example, entropy is used in [16] for feature elimination in microarray expression data. The entropy of a gene is calculated as following:

$$H(G) = -\sum_{i=1}^{m} f_i \log_2 f_i \tag{2.3}$$

Let G be a gene and $s_1, ..., s_m$ normalized expression values of G for m samples. Cumulative expression values of G are calculated as $\bar{s} = \sum_{i=1}^{m} s_i$ and f_i is defined as $f_i = s_i/\bar{s}$. $H(G)$ is defined as the entropy of gene G. According to (2.3), $H(G)$ is at its maximum when gene G has identical expression values over all m samples, which means this gene is not expressed differentially in any sample. The more the expression value of a gene is expressed differently in all samples, the lower $H(G)$ is, and minimum entropy occurs where only the gene is expressed in one sample. So genes with the lowest entropy are selected as representative genes from each cluster.

The framework of this approach is divided into four phases: (i) gene clustering where genes are clustered into a specified number of groups. The number of clusters is specified in advance using 5-fold cross validation. In phase (ii) representative genes selection where genes inside each cluster are ranked based on the selected criteria and are selected from each group. In phase (iii) feature selection is performed on reduced dataset consisting of representative genes. Finally in phase (IV) we evaluate the selected features using two classifiers. In the following sections, we describe each phase of this approach in details.

2.2.1 Gene Clustering

In gene clustering, genes with similar expression profiles are clustered into the same gene groups. Genes belonging to the same cluster contain partially redundant information; however, genes of different clusters contain different information. It is important to select good clustering algorithms in this study but there is no best clustering algorithm especially for gene datasets. There have been some previous studies on comparing different clustering algorithms on cancer datasets [46, 53]. k-means clustering algorithm was used as the gene clustering algorithm in different studies [27] due to its simplicity and efficiency. For this reason, the candidate gene clustering algorithm

for this project is k-means.

2.2.2 Gene Representative

Since genes clustered into the same group are supposed to have a similar profile and function, selecting representative genes from each cluster can reduce redundancy among genes. So, after clustering the genes, 10% of genes from each cluster are selected based on two different metrics as representative genes. One of the metrics is Euclidean distance, so the distance of each gene from the cluster center is calculated. Then all genes of the corresponding cluster are ranked based on calculated distances in ascending order. Lastly, 10% of genes with smaller distances are chosen as representative genes of the cluster.

When selecting representative genes based on entropy, genes of each cluster are ranked based on calculated entropy in ascending order. In the end, 10% of genes with the smallest entropies are chosen as representative genes of the cluster. According to entropy formula, the more different the expression values of a gene across all samples are, the lower the entropy is [63]. So, in selecting representative genes based on entropy, 10% of genes with smaller entropy are chosen from each cluster.

2.2.3 Gene Selection

After gene clustering and gene representation selection step, all the chosen representative genes are collected as a representative set. Next, the number of genes in the original dataset (D) is reduced to the number of representative genes. It means genes that are not in the representative set are removed from the original dataset. After this step, the size of the dataset is reduced. The number of samples of reduced datasets is the same as the original one, but the number of genes is reduced to the number of selected representative genes. Next, the feature selection method is applied to the

reduced dataset. Figure 2.1 shows a schematic representation of this approach, along with feature selection and model creation steps.

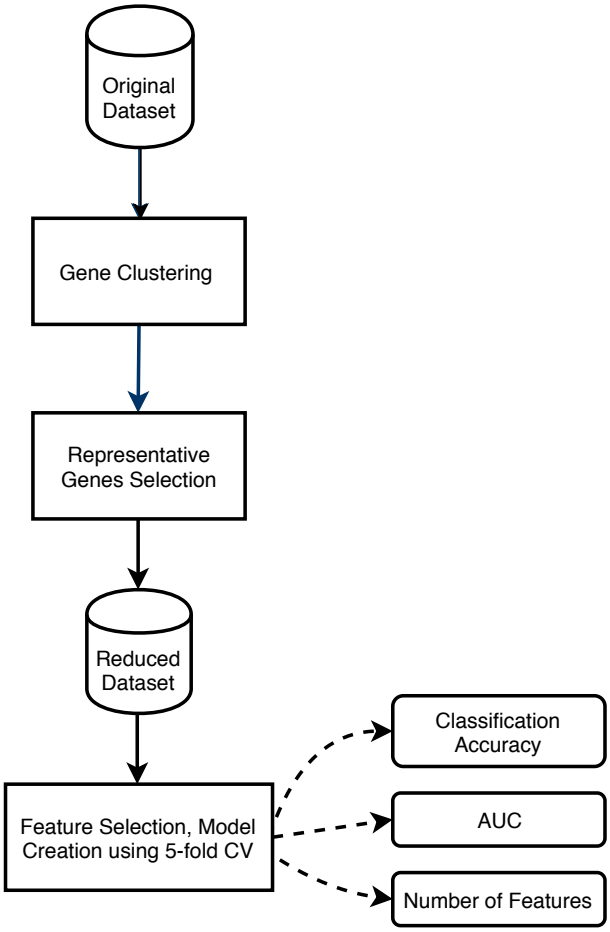

Figure 2.1: Removing redundant features using clustering and performing 5-fold CV for feature selection and model creation on the reduced dataset

Three well-known feature selection algorithms: mRMR, SVM-RFE, and Hilber-Schmit Independence Criterion Least Absolute Shrinkage and Selection Operator (HSIC-Lasso) [60] are examined in this research, and they are described as follows.

mRMR This method selects features with the highest relevance to the class labels and lowest redundancy among genes. Both maximum-relevance and minimum-redundancy criteria on this method are based on mutual information.

SVM-RFE This algorithm is a classic gene selection methods that utilizes an external estimator to assign weights to features. These weights generated by an estimator are used as ranking criteria. In each step of feature elimination, the lowest-ranked features are removed from the current subset of features. Feature elimination is a recursive procedure and is repeated until the specified number of features is selected.

HSIC-Lasso This method is a recently proposed kernel-based mRMR algorithm. This method considers the non-linear dependency between input features and output values. HSIC-Lasso finds non-redundant features with a high dependency on output using specific kernel functions.

2.3 Validation Techniques

To evaluate how selected genes can help differentiate between healthy and disease samples, having a test dataset that has not been seen by our model is essential. So, in this thesis, the function of repeated stratified k-fold cross-validation (RepeatedStratifiedKFold) from Scikit-learn machine learning library [42] is used for feature selection and model construction. The stratified version is used to avoid dataset shift that is one of the drawbacks of using cross-validation. Stratification is utilized to ensure that the proportions between classes are the same in each fold as they are in the original dataset. The number of splits and the number of repetitions are set to 5 and 3, respectively (n_splits=5, n_repeats=3). So the original data sample is randomly partitioned into five sub-samples that one fold is used for testing, and the remaining four folds are used for feature selection and training. The process is repeated five times, so each subsample used precisely once as testing data. Since we used the repeated version of stratified k-fold CV, the whole process of cross-validation is repeated three times, and the optimal number of features and all evaluation metrics are reported as

average over 15 iterations.

2.4 Evaluation Measures

Selecting evaluation metrics is vital to measure the performance of the classifier. According to [38], there is no best metric for evaluating the machine learning model. For example, Classification Accuracy (CA) is not enough for model evaluation since it does not give us information about the False Negatives or False Positives, especially when our data is unbalanced. For this reason, we reported other well-known metrics, including Recall, Specificity, Area Under the Curve (AUC), and Average Precision.

2.5 Datasets

Inflammatory bowel disease (IBD) refers to a group of diseases that involve inflammation in intestines. There are two major subtypes for this disease, which are Ulcerative colitis (UC) and Crohn's disease (CD) [57]. Due to the complexity of this disease, the diagnosis is challenging. However, several symptoms, including abdominal pain, diarrhea, and bloody stool, have been considered as diagnostic indicators for IBD [5]. Development of IBD is affected by genetic and environmental factors [15]. For example, studies show that having a family member who is affected by this disorder is a significant risk factor for developing IBD [6, 23]. Since genetic factors have been proved to be highly related to IBD and its subtypes, there have been different studies for identifying genes that are expressed differently in healthy and diseased individuals. Although both subtypes share disease-causing genes, some genes are expressed differently in UC and CD patients.

Three datasets, containing gene expression profile of healthy and UC samples, have been obtained from the Gene Expression Omnibus database (GEO). Datasets were

downloaded under accession numbers GSE11223 [41], GSE3365 [8], and GSE22619 [24, 37] are described in Table 2.1.

Table 2.1: Summary of datasets

Dataset	Sample size	Number of features	Number of UC	Number of controls
GSE3365	68	13,299	26	42
GSE11223	49	18,626	25	24
GSE22619	20	22,189	10	10

2.5.1 Data Collection and Pre-processing

Data Collection

For each dataset, GEO2R [4] was used to retrieve the mapping between probe IDs and gene symbols. Probe IDs without a gene mapping were removed from further processing. Expression values for the mapped probe IDs were obtained using the Python package GEOparse *.

Data Pre-processing

The data pre-processing has the following stages:

(i) Calculating the expression value of each gene by taking the average of expression values of all probes mapped to it.

(ii) Handling missing values with k-Nearest Neighbors (k-NN) imputation method (kNNImputer) [51] from the "missingpy" library in Python †.

*https://pypi.org/project/GEOparse/
†https://pypi.org/project/missingpy/

kNNImputer uses k-NN to impute missing values so each missing value is imputed by utilizing the values from nearest neighbors. The number of neighbors was set to 2 (n-neighbors=2) and uniform weight was used.

2.6 Summary

In this chapter, we proposed the SLS method and introduced two clustering-based approaches for reducing the number of features. Then, we introduced validation techniques and the metrics for measuring the predictive performance of the models. In the end, we explained the datasets and preprocessing steps of datasets.

Chapter 3

Experiments

The goal of this chapter is to compare the effect of augmenting the proposed SLS method and two dataset dimensionality reduction approaches to different feature selection methods. Section 3.1 explains the parameter setting for clustering and feature selection methods. Section 3.2 presents the results of applying all three feature selection methods on the datasets using SVM and RF. Section 3.3 reports the results of applying feature selection on the reduced datasets using described approaches (SLS and clustering) and compares them. Section 3.4 and Section 3.5 show SLS preserves informative features while it removes irrelevant ones and also reduces the computational cost significantly. At the end, Section 3.6 summarizes this chapter.

3.1 Parameter setting

Since the number of clusters (k) is a key parameter in k-means, it should be explicitly specified. We employ repeated stratified 5-fold cross-validation to find the optimal number of clusters for each microarray dataset. Fig. 3.1 shows the experimental results for all datasets. According to average accuracy curve, the optimal number of clusters for GSE3365 is 150 (CA = 94.98%), for GSE11223 is 100 (CA= 76.37%) and for

GSE22619 is 10 and 80 (CA= 75%).

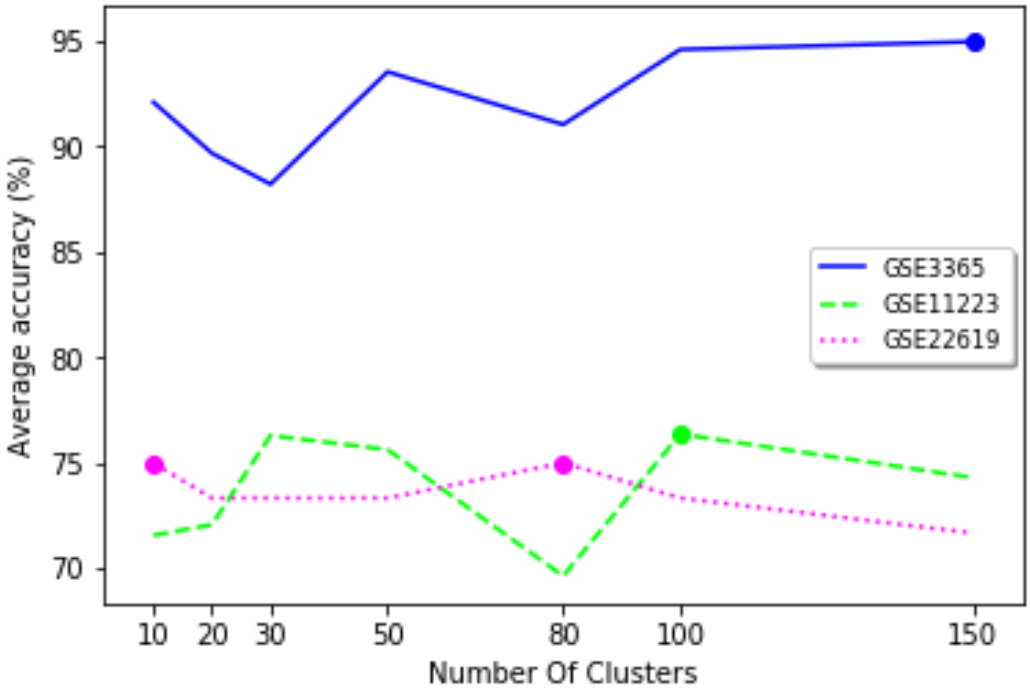

Figure 3.1: Accuracy vs. cluster number for all datasets

The number of features (f) to be selected should be given as an input parameter to all the chosen feature selection methods for this study. Because the computational complexity of the feature selection methods is affected by f, this parameter is set to $f = 20$ for all algorithms.

In all experiments, all codes are implemented with Python 3.6 and also the Python software packages of mRMR and HSIC-Lasso are used. SVM-RFE is also available in Scikit-learn. In addition, all experiments are conducted on a PC with an Intel® Core™ i7-4790 CPU (3.60GHz x 8), and 16GB of RAM.

mRMR, HSIC-Lasso, and SVM-RFE feature selection methods are applied on all three microarray datasets described earlier. Two well-known classification algorithms: SVM, and RF are used for model creation, which are adopted from Scikit-learn.

3.2 Feature Selection On The Whole Datasets

To evaluate the performance of each combination of methods, all feature selection methods are applied on the whole datasets without any filtering step. Tables 3.1 and 3.2 show the experimental results of applying only feature selection methods on the whole datasets using two different classifiers.

CA and AUC are reported as average over 15 iterations of repeating stratified 5-fold cross-validation for three times. Also, the optimal number of features (#f) are presented for each method. This number is the number of features that yielded the highest CA in all iterations. The reported running time is the total run time after 15 iterations, which includes the running time of the classifier as well.

Table 3.1: Performance of feature selection methods on the whole datasets using SVM

Method	#f	CA (%)	AUC	Runtime (Sec)
Dataset: GSE3365				
mRMR	17	77.60 ± 11.40	0.81 ± 0.16	8,153
SVM-RFE	17	93.93 ± 5.10	0.97 ± 0.04	4,711
HSIC-Lasso	13	97.40 ± 4.74	0.97 ± 0.04	77
Dataset: GSE11223				
mRMR	6	57.13 ± 14.62	0.64 ± 0.23	2,959
SVM-RFE	2	100	1.0	4,597
HSIC-Lasso	19	66.66 ± 5.77	0.72 ± 0.16	92
Dataset: GSE22619				
mRMR	12	73.33 ± 19.97	0.75 ± 0.23	1,325
SVM-RFE	10	71.66 ± 22.88	0.76 ± 0.29	1,815
HSIC-Lasso	9	75.0 ± 21.12	0.80 ± 0.25	72

Table 3.2: Performance of feature selection methods on the whole datasets using RF

Method	#f	CA (%)	AUC	Runtime (Sec)
Dataset: GSE3365				
mRMR	18	77.0 ± 11.53	0.81 ± 0.14	8,359
SVM-RFE	20	92.20 ± 7.96	0.96 ± 0.05	4,792
HSIC-Lasso	19	80.0 ± 7.79	0.80 ± 0.11	69
Dataset: GSE11223				
mRMR	20	63.33 ± 14.0	0.63 ± 0.13	2,954
SVM-RFE	2	100	1.0	4,586
HSIC-Lasso	20	80.0 ± 20	0.81 ± 0.20	96
Dataset: GSE22619				
mRMR	7	68.33 ± 22.09	0.75 ± 0.27	1336
SVM-RFE	9	90.0 ± 12.67	0.94 ± 0.10	1832
HSIC-Lasso	10	73.33 ± 24.02	0.74 ± 0.25	72

3.3 Feature Selection On The Reduced Datasets

To show how reducing the number of features affects the model prediction power and running time, three categories of experiments are conducted using described methods in Section 2.1 and Section 2.2. These three categories are: reducing the size of the dataset with (i) clustering genes and selecting representative genes based on Euclidean distance, (ii) clustering genes and selecting representative genes based on entropy, (iii) finding irrelevant genes and removing them from datasets using SLS method. In the first two categories, genes are clustered into the optimal number of clusters for each dataset; this number was specified in advance. After selecting 10 percent of genes from each cluster, a subset of features containing representative genes is created. Next, the number of genes is reduced to the number of representative genes. In the third category of experiments, the proposed SLS method is applied to the dataset, and

then the size of the dataset is reduced by removing irrelevant genes. In the next step, each feature selection algorithm is performed on the reduced datasets by adopting a repeated stratified 5-fold CV for finding the best subset of genes and creating the model based on them and evaluating the model. So in each iteration of CV, 20 genes are selected, and 20 models are generated using the selected genes on the training dataset. For example, the first model is created using the first selected feature, and the fifth model is created using the first five selected features. For each model, all the evaluation metrics are calculated and saved, which are the results of validating the model on the test set. When all iterations of CV are finished, all evaluation metrics are averaged. For instance, all the CAs achieved from the models created based on a subset of 5 features in each iteration are averaged together. Then, the highest average CA, along with corresponding feature number, recall, specificity, AUC, and APS, are reported.

In the following sections, experimental results on each dataset are discussed. Firstly, for each dataset, 12 different combinations are examined. They contain comparing the results of the combinations of three feature selection methods with two classifiers for each representative gene selection approach. Then, the results of applying the feature selection methods on the reduced dataset by SLS are reported. To show which approach of reducing the size of the dataset works better, the results corresponding to each approach are reported separately. Firstly, a comparison of the results of applying feature selection methods on the reduced dataset using two representative gene selection approaches is made. Then it is compared with the results of applying these methods on the reduced dataset by SLS.

3.3.1 Experiments on GSE3365 Dataset

This dataset contains 127 samples, including 59 CD, 26 UC, and 42 healthy subjects. Only samples from UC and healthy classes were selected for this study. The expression levels of 13,300 genes were measured using Affymetrix Human Genome U133A Array. To show which method of selecting representative genes is better, genes are clustered into 150 groups, and 1,262 representative genes are selected based on described clustering-based approaches. Then, the dataset's features are reduced to the number of representative genes. In each iteration of repeated CV, 54 samples are used for feature selection and training, and the remaining 14 samples are used for testing. Results of all combinations of feature selection methods and classification algorithms on the dataset reduced by clustering-based approaches are reported in Tables 3.3 and 3.4. They present the number of selected features, average Classification Accuracy (CA), Recall (Rec), Specificity (Spec), the Area Under the Curve (AUC), Average Precision Score (APS), and Runtime.

Table 3.3: Experimental results of applying feature selection methods on GSE3365 using SVM and RF classifiers after reducing the size of the dataset by clustering genes and selecting representative genes based on Euclidean distance.

Dataset: GSE3365							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	19	83.53 ± 9.94	62.0 ± 23.49	96.66 ± 5.72	0.90 ± 0.11	0.90 ± 0.10	105
SVM-RFE	15	95.93 ± 4.86	93.55 ± 9.46	97.59 ± 4.99	0.97 ± 0.04	0.97 ± 0.03	66
HSIC-Lasso	19	92.93 ± 7.29	85.33 ± 17.67	97.59 ± 4.99	0.96 ± 0.05	0.96 ± 0.05	35
Classifier: RF							
mRMR	19	83.84 ± 6.51	63.87 ± 22.11	95.88 ± 7.71	0.90 ± 0.09	0.87 ± 0.09	121
SVM-RFE	9	92.13 ± 5.85	84.66 ± 13.38	96.75 ± 5.57	0.96 ± 0.04	0.93 ± 0.07	65
HSIC-Lasso	12	91.06 ± 9.20	81.77 ± 22.88	96.85 ± 5.41	0.94 ± 0.07	0.91 ± 0.10	34

According to these two tables, we can conclude that the CAs, and AUCs of mRMR and SVM-RFE using both classifiers are higher when we use Euclidean distance. When

Table 3.4: Experimental results of applying feature selection methods on GSE3365 using SVM and RF classifiers after reducing the size of the dataset by clustering genes and selecting representative genes based on entropy.

Dataset: GSE3365							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	19	80.4 ± 8.37	58.88 ± 17.07	93.61 ± 7.80	0.82 ± 0.11	0.80 ± 0.13	105
SVM-RFE	15	91.06 ± 8.16	89.77 ± 16.44	92.03 ± 8.61	0.96 ± 0.05	0.95 ± 0.07	65
HSIC-Lasso	19	97.4 ± 4.74	97.33 ± 7.03	97.5 ± 5.17	0.99 ± 0.03	0.98 ± 0.05	27
Classifier: RF							
mRMR	17	77.0 ± 9.59	57.77 ± 21.51	88.98 ± 11.17	0.80 ± 0.100	0.75 ± 0.12	106
SVM-RFE	15	90.60 ± 6.43	81.77 ± 11.87	96.01 ± 7.19	0.96 ± 0.05	0.93 ± 0.11	67
HSIC-Lasso	14	92.53 ± 6.35	84.44 ± 17.16	97.5 ± 4.99	0.98 ± 0.03	0.96 ± 0.06	28

using HSIC-Lasso, the entropy is a better metric for reducing the size of the dataset in terms of CA and AUC.

After removing irrelevant features using SLS, 588 features remained on the dataset. The results of applying feature selection methods using both classifiers on this reduced dataset are reported in Table 3.5. Comparing the results of all three tables, it can be seen that CA and AUC achieved from applying feature selection on the reduced dataset using SLS method are superior or comparable with the highest results of two former tables. Only for the combination of SVM-RFE and SVM, a lower CA= 94.06 is obtained as compared with the same combination when using Euclidean distance for selecting representative genes. In terms of AUC, both approaches achieve comparable results.

Regarding runtimes, mRMR and HSIC-Lasso are the slowest and fastest methods, respectively. Also, comparing the running of three tables, the running times belonging to the last table are the lowest, and the reason is that the number of features remained is smaller when the SLS method for the dataset dimensionality reduction is used.

Regarding recall and specificity, all three tables show higher specificity than recall. One potential reason is the number of healthy subjects is almost twice that of UC

samples in the dataset, so the generated models are biased to the healthy class, which yields higher specificities.

Table 3.5: Experimental results of applying feature selection methods on GSE3365 using SVM and RF classifiers after reducing the size of the dataset with SLS method.

Dataset: GSE3365							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	18	97.13 ± 5.23	93.66 ± 11.11	99.06 ± 3.22	0.99 ± 0.00	0.99 ± 0.01	31
SVM-RFE	16	94.06 ± 5.71	89.77 ± 12.50	96.85 ± 8.32	0.97 ± 0.04	0.96 ± 0.04	18
HSIC-Lasso	12	97.86 ± 3.66	96.0 ± 8.28	99.16 ± 3.22	0.99 ± 0.02	0.99 ± 0.02	11
Classifier: RF							
mRMR	16	87.73 ± 10.22	71.33 ± 21.44	94.53 ± 7.37	0.93 ± 0.06	0.91 ± 0.08	30
SVM-RFE	16	94 ± 5.73	88.22 ± 14.63	95.92 ± 7.61	0.97 ± 0.04	0.93 ± 0.08	20
HSIC-Lasso	18	94.46 ± 5.92	90.66 ± 14.86	96.75 ± 5.57	0.99 ± 0.01	0.98 ± 0.02	11

3.3.2 Experiments on GSE11223 Dataset

This dataset contains 202 samples, which only 49 samples were selected out of it, including 25 UC and 24 healthy samples. Selected samples are biopsies from the uninflamed sigmoid colon of UC patients and healthy donors. The expression levels of 18,626 genes were measured using Agilent-012391 Whole Human Genome Oligo Microarray G4112A. Tables 3.6, 3.7 and 3.8 show the experimental results on this dataset. Firstly, genes are clustered into 100 groups, and 1,821 representative genes are selected based on two clustering approaches. Then, the number of features in the dataset is decreased to the number of representative genes. In each iteration of repeated CV, 39 samples are used for feature selection and training, and the remaining 10 samples are used for testing. Results of all 12 combinations of feature selection methods and classification algorithms using two approaches of representative gene selection are reported in Tables 3.6 and 3.7.

Table 3.6: Experimental results of applying feature selection methods on GSE11223 using SVM and RF classifiers after reducing the size of the dataset by clustering genes and selecting representative genes based on Euclidean distance.

Dataset: GSE11223							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	5	51.80 ± 10.73	54.66 ± 34.19	49.66 ± 38.19	0.49 ± 0.19	0.63 ± 0.15	146
SVM-RFE	11	68.8 ± 12.82	66.66 ± 22.25	71.66 ± 19.60	0.65 ± 0.17	0.72 ± 0.14	87
HSIC-Lasso	9	89.0	90.0 ± 14.14	87.5 ± 17.67	0.89 ± 0.07	0.92 ± 0.06	30
Classifier: RF							
mRMR	14	53.93 ± 15.29	45.33 ± 25.59	62.66 ± 16.99	0.50 ± 0.14	0.60 ± 0.11	147
SVM-RFE	10	92.0 ± 11.46	89.33 ± 16.67	94.66 ± 11.87	0.95 ± 0.05	0.96 ± 0.05	89
HSIC-Lasso	4	100	100	100	1.0	1.0	31

According to these tables, higher CA and AUC are achieved when entropy was used as the representative selection metric. Considering the 6 combinations of feature selection and classification methods when using Euclidean distance, the combination of HSIC-Lasso and RF yields the highest results (CA= 100%, AUC= 1.0). When using entropy for selecting features from each cluster, the combination of HSIC-Lasso and SVM-RFE with both classifiers yields the highest CA(100%) and AUC(1.0). In both cases, mRMR performance on this dataset using both classifiers is inferior in comparison with the other feature selection methods.

After removing irrelevant features using SLS, only 452 features remained on the dataset. The results of applying feature selection methods on the dataset reduced by SLS are presented in Table 3.8. Comparing the results of all three tables, it can be seen that CA and AUC achieved from applying feature selection methods on the dataset reduced by SLS, are higher than that of the other two approaches. Only for the combination of HSIC-Lasso with SVM, the results belong to the SLS approach are comparable with the results of the same combination belong to the entropy approach.

Comparing the recall and specificity of all tables, the highest results are achieved

from the SLS method.

Table 3.7: Experimental results of applying feature selection methods on GSE11223 using SVM and RF classifiers after reducing the size of the dataset by clustering genes and selecting representative genes based on entropy.

Dataset: GSE11223							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	15	53.80 ± 14.0	52.0 ± 23.66	55.33 ± 18.46	0.53 ± 0.17	0.63 ± 0.13	142
SVM-RFE	1	100	100	100	1.0	1.0	82
HSIC-Lasso	5	100	100	100	1.0	1.0	30
Classifier: RF							
mRMR	19	56.86 ± 18.23	50.66 ± 21.20	63.33 ± 25.88	0.55 ± 0.20	0.65 ± 0.13	144
SVM-RFE	1	100	100	100	1.0	1.0	100
HSIC-Lasso	3	100	100	100	1.0	1.0	29

Table 3.8: Experimental results of applying feature selection methods on GSE11223 using SVM and RF classifiers after reducing the size of the dataset with SLS method

Dataset: GSE11223							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	15	79.06 ± 15.70	81.33 ± 17.67	76.66 ± 26.02	0.82 ± 0.15	0.84 ± 0.13	19
SVM-RFE	2	100	100	100	1.0	1.0	14
HSIC-Lasso	3	99.33 ± 2.58	100	98.66 ± 5.16	0.99 ± 0.03	0.99 ± 0.03	10
Classifier: RF							
mRMR	13	64.73 ± 13.50	53.33 ± 25.25	76.66 ± 18.28	0.67 ± 0.18	0.72 ± 0.13	21
SVM-RFE	2	100	100	100	1.0	1.0	16
HSIC-Lasso	1	100	100	100	1.0	1.0	12

In terms of running times, HSIC-Lasso is the fastest, and the higher running times belong to mRMR. Also, since removing irrelevant features reduced the size of dataset more than the other two approaches did for this dataset, the running times of this approach are lower than other approaches. For example, comparing the running time of applying mRMR with SVM in three tables, we can see that the running time decreased from 146 seconds in Table 3.6 to 19 seconds in Table 3.8.

3.3.3 Experiments on GSE22619 Dataset

This dataset contains 20 samples, including 10 UC and 10 normal samples. The expression levels of 22,189 genes were measured using Affymetrix Human Genome U133 Plus 2.0 Array. Results of all 12 combinations of feature selection methods and classification algorithms using two approaches of representative gene selection are reported in Tables 3.9 and 3.10.

Table 3.9: Experimental results of applying feature selection methods on GSE22619 using SVM and RF classifiers after reducing the size of the dataset by clustering genes and selecting representative genes based on Euclidean distance.

Dataset: GSE22619							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	18	73.33 ± 17.59	73.33 ± 31.99	73.33 ± 25.81	0.83 ± 0.24	0.9 ± 0.14	81
SVM-RFE	18	78.33 ± 15.99	80.0 ± 25.35	76.66 ± 31.99	0.81 ± 0.27	0.9 ± 0.15	54
HSIC-Lasso	18	70.0 ± 25.35	63.33 ± 44.18	76.66 ± 25.81	0.68 ± 0.31	0.81 ± 0.19	20
Classifier: RF							
mRMR	14	70.0 ± 23.52	56.66 ± 37.16	83.33 ± 30.86	0.72 ± 0.24	0.79 ± 0.18	82
SVM-RFE	6	76.66 ± 24.02	66.66 ± 36.18	86.66 ± 29.68	0.75 ± 0.26	0.85 ± 0.16	56
HSIC-Lasso	11	63.33 ± 24.76	63.33 ± 39.94	63.33 ± 35.18	0.57 ± 0.31	0.70 ± 0.22	22

Genes are clustered into 10 groups, and 2,214 representative genes are selected based on two clustering approaches. Then, the number of features in the dataset is decreased to the number of representative genes. In each iteration of repeated stratified CV, 16 samples are used for feature selection and training, and the remaining 4 samples are used for testing. According to Tables 3.9 and 3.10, when using entropy as the representative selection metric, for almost all combinations, the results are superior to the Euclidean distance approach. Only for the combination of SVM-RFE and SVM, the CA belonging to the Euclidean distance approach is better.

Comparing results in terms of AUC, we can see that the combinations of both mRMR and SVM-RFE with SVM yields better results when selecting representative

Table 3.10: Experimental results of applying feature selection methods on GSE22619 using SVM and RF classifiers after reducing the size of the dataset by clustering genes and selecting representative genes based on entropy.

Dataset: GSE22619							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	6	78.33 ± 22.88	63.33 ± 39.94	93.33 ± 17.59	0.75 ± 0.29	0.86 ± 0.16	78
SVM-RFE	17	73.33 ± 22.09	66.66 ± 30.86	80.0 ± 31.62	0.75 ± 0.26	0.83 ± 0.17	50
HSIC-Lasso	1	88.33 ± 20.84	83.33 ± 30.86	93.33 ± 17.59	0.83 ± 0.30	0.91 ± 0.17	16
Classifier: RF							
mRMR	10	80.0 ± 16.90	63.33 ± 35.18	96.66 ± 12.90	0.75 ± 0.25	0.85 ± 0.15	80
SVM-RFE	16	83.33 ± 18.09	70.0 ± 31.62	96.66 ± 12.90	0.89 ± 0.15	0.90 ± 0.13	52
HSIC-Lasso	1	81.66 ± 24.02	73.33 ± 37.16	90.0 ± 28.03	0.85 ± 0.24	0.87 ± 0.17	17

genes based on Euclidean distance. The remaining combinations achieve better results when using entropy.

After removing irrelevant features using SLS, only 321 features remained on the dataset. The results of applying feature selection methods on the dataset reduced by SLS are presented in Table 3.11.

Table 3.11: Experimental results of applying feature selection methods on GSE22619 using SVM and RF classifiers after reducing the size of the dataset with SLS method.

Dataset: GSE22619							
Method	**#f**	**CA (%)**	**Rec (%)**	**Spec(%)**	**AUC**	**APS**	**Runtime (Sec)**
Classifier: SVM							
mRMR	6	95.0 ± 10.35	90 ± 20.70	100	0.98 ± 0.06	0.98 ± 0.04	9
SVM-RFE	3	80.0 ± 19.36	80.0 ± 31.62	80.0 ± 36.83	0.88 ± 0.22	0.92 ± 0.14	8
HSIC-Lasso	19	91.66 ± 20.41	100	83.33 ± 40.82	1.0	1.0	10
Classifier: RF							
mRMR	4	86.66 ± 12.90	73.33 ± 25.81	100	0.93 ± 0.10	0.94 ± 0.08	11
SVM-RFE	9	90.0 ± 15.81	83.33 ± 30.86	96.66 ± 12.90	0.95 ± 0.11	0.96 ± 0.08	10
HSIC-Lasso	15	87.5 ± 21.37	82.14 ± 24.86	92.85 ± 26.72	0.87 ± 0.25	0.91 ± 0.17	11

Comparing the results of all three tables, it can be seen that CA and AUC achieved from applying feature selection methods on the datasets without irrelevant features,

are higher than that of the other two clustering approaches.

In terms of running times, HSIC-Lasso is the fastest, and the higher running times belong to mRMR. Also, since removing irrelevant features using SLS reduces the size of this dataset more than the other two approaches did, the running times belonging to this method are lower than other approaches. For example, comparing the running time of applying mRMR with SVM in three tables, we can see that the running time decreased from 81 seconds in Table 3.9 to 9 seconds in Table 3.11.

3.4 SLS Method Improves the Predictive Power of the Models

To show how removing irrelevant features from the dataset can affect the predictive power of the model and reduce the computational complexity, we compare the results of applying feature selection methods on the whole datasets and also on the datasets reduced by SLS method.

Tables 3.12, 3.13, and 3.14 show the results of applying feature selection methods on GSE3365, GSE11223, and GSE22619 respectively. According to results, higher CA and AUC are achieved when we used SLS, and running time has been reduced significantly. Only for GSE11223, the results of applying SVM-RFE using both classifiers are comparable with and without augmenting SLS (CA = 100, AUC= 1.0).

3.5 SLS Method Preserves Informative Features

Since the results obtained from using the SLS method are better than the representative selection approaches, in this section, we compare features selected by the

Table 3.12: Experimental results of applying feature selection methods with and without augmenting SLS using SVM on GSE3365.

Dataset: GSE3365			
Method	**#f**	**CA (%)**	**AUC**
Classifier: SVM			
mRMR	17	77.60 ± 11.40	0.81 ± 0.16
SLS+mRMR	18	97.13 ± 5.23	0.99 ± 0.007
SVM-RFE	17	93.93 ± 5.10	0.97 ± 0.04
SLS+SVM-RFE	16	94.06 ± 5.71	0.97 ± 0.04
HSIC-Lasso	13	97.40 ± 4.74	0.97 ± 0.04
SLS+HSIC-Lasso	12	97.86 ± 3.66	0.99 ± 0.02
Classifier: RF			
mRMR	18	77.0 ± 11.53	0.81 ± 0.14
SLS+mRMR	16	87.73 ± 10.22	0.93 ± 0.06
SVM-RFE	20	92.20 ± 7.96	0.96 ± 0.05
SLS+SVM-RFE	16	94 ± 5.73	0.97 ± 0.04
HSIC-Lasso	19	80.0 ± 7.79	0.80 ± 0.11
SLS+HSIC-Lasso	18	94.46 ± 5.92	0.99 ± 0.01

feature selection methods with and without augmenting the SLS method to show this method preserves important features. Only comparisons for the feature selection methods that yield high predictive performance on the whole dataset are made. For example, by looking at Tables 3.1, 3.2, and 3.5, it can be seen that SMVRFE and HSIC-Lasso obtained higher results on GSE3365 considering all three tables. So only these two methods with and without augmenting SLS are re-performed on GSE3365. For GSE11223 and GSE22619, SVM-RFE is re-performed. For this purpose, the feature selection method is applied to the original dataset and also on the reduced dataset by SLS. The stratified 5-fold CV is utilized for feature selection and model creation, and in each iteration of CV, 20 features are selected. Then the features selected in all iterations (N=100) for each feature selection method are compared. Table 3.15 shows a summary of comparing selected features. This table presents the dataset, feature selection method with and without augmenting SLS, number of unique features

Table 3.13: Experimental results of applying feature selection methods with and without augmenting SLS using SVM on GSE11223.

Dataset: GSE11223			
Method	**#f**	**CA (%)**	**AUC**
Classifier: SVM			
mRMR	6	57.13 ± 14.62	0.64 ± 0.23
SLS+mRMR	15	79.06 ± 15.70	0.82 ± 0.15
SVM-RFE	2	100	1.0
SLS+SVM-RFE	2	100	1.0
HSIC-Lasso	19	66.66 ± 5.77	0.72 ± 0.16
SLS+HSIC-Lasso	3	99.33 ± 2.58	0.99 ± 0.03
Classifier: RF			
mRMR	20	63.33 ± 14.0	0.63 ± 0.13
SLS+mRMR	13	64.73 ± 13.50	0.67 ± 0.18
SVM-RFE	2	100	1.0
SLS+SVM-RFE	2	100	1.0
HSIC-Lasso	20	80.0 ± 20	0.81 ± 0.20
SLS+HSIC-Lasso	1	100	1.0

out of a total of 100 selected features, and number of overlapping features between two approaches (feature selection with and without augmenting SLS). The number of overlapping features is obtained from getting the intersection of unique features selected by each feature selection method and the augmented version.

Results belonging to GSE3365 show that HSIC-Lasso selects more repetitive features in different runs than SVM-RFE. For example, after running HSIC-Lasso on the whole dataset (without SLS), 51 features out of all 100 selected features are unique. However, after running SVM-RFE on the same dataset, 79 features are unique, which is equal to less repetitive features. Comparing the results of the last column for GSE3365, the number of overlapping features, it can be seen that SVM-RFE selected more identical features while running on the original and reduced dataset (N=47) than HSIC-Lasso (N=39). The highest number of overlapping features belongs to perform-

Table 3.14: Experimental results of applying feature selection methods with and without augmenting SLS using SVM on GSE22619.

Dataset: GSE22619			
Method	**#f**	**CA (%)**	**AUC**
Classifier: SVM			
mRMR	12	73.33 ± 19.97	0.75 ± 0.23
SLS+mRMR	6	95.0 ± 10.35	0.98 ± 0.06
SVM-RFE	10	71.66 ± 22.88	0.76 ± 0.29
SLS+SVM-RFE	3	80.0 ± 19.36	0.88 ± 0.22
HSIC-Lasso	9	75.0 ± 21.12	0.80 ± 0.25
SLS+HSIC-Lasso	19	91.66 ± 20.41	1.0 ± 0.0
Classifier: RF			
mRMR	7	68.33 ± 22.09	0.75 ± 0.27
SLS+mRMR	4	86.66 ± 12.90	0.93 ± 0.10
SVM-RFE	9	90.0 ± 12.67	0.94 ± 0.10
SLS+SVM-RFE	9	90.0 ± 15.81	0.95 ± 0.11
HSIC-Lasso	10	73.33 ± 24.02	0.74 ± 0.25
SLS+HSIC-Lasso	15	87.5 ± 21.37	0.87 ± 0.25

ing SVM-RFE on the whole and reduced GSE22619 dataset, which is 54. The reason that the number of overlapping features for GSE11223 is lower than other datasets is that when SVM-RFE is performed on the GSE11223 and the reduced one, the number of unique features is low in each experiment (with and without SLS). So it justifies the lower number of overlapping features for this dataset.

3.6 SLS Method Reduces Computational Complexity Significantly

To show how reducing the size of the dataset decreases the time complexity of feature selection methods, the running time of applying feature selection methods on the whole dataset and the reduced dataset using SLS is compared in this section. Based on previous results, there is no significant difference between the running time of

Table 3.15: Comparison of 100 selected features after performing feature selection methods using stratified 5-fold CV on the whole dataset and the dataset reduced by the SLS method.

Dataset	Method	# of unique features	# of overlapping features
GSE3365	HSIC-Lasso	51	39
	SLS+HSIC-Lasso	47	
	SVM-RFE	79	47
	SLS+SVM-RFE	78	
GSE11223	SVM-RFE	27	22
	SLS+SVM-RFE	29	
GSE22619	SVM-RFE	65	54
	SLS+SVM-RFE	62	

applying each method using different classifiers. Therefore, only the running times of feature selection methods with and without using SLS method using SVM are compared here. According to Table 3.16, the running time of all combinations has a reduction of above 84% after the removal of irrelevant features.

3.7 Summary

In this chapter, the proposed SLS method and two clustering-based dataset size-reduction approaches were described. These reduction steps were augmented to three feature selection methods, namely mRMR, SVM-RFE, and HSIC-Lasso. Then the results of applying feature selection methods and the augmented version of them on three IBD microarray datasets using support vector machine and random forest classifiers were compared. The comparison results showed that the proposed SLS method is more effective in the removal of irrelevant features and preserves informative genes.

It also reduces the computational complexity of feature selection methods significantly.

Table 3.16: Comparison of running time (sec) reductions of performing feature selection methods on the whole dataset and the dataset reduced by the SLS method.

Method	**Dataset**	**Whole Dataset**	**Reduced Dataset**	**Reduction** (%)
mRMR	GSE3365	8,153	31	99.6
SVM-RFE		4,711	18	99.6
HSIC-Lasso		77	11	85.7
mRMR	GSE11223	2,959	21	99.2
SVM-RFE		4,597	16	99.6
HSIC-Lasso		92	12	86.9
mRMR	GSE22619	1,325	11	99.1
SVM-RFE		1,815	10	99.4
HSIC-Lasso		72	11	84.7

Chapter 4

Case Study

Feature selection and machine learning are useful for creating models to help the diagnosis of certain diseases. In this chapter, as a case study, we use a recently developed feature selection method, DRPT [1], which augments SLS (described in Section 2.1) for identifying subjects with IBD. We show that DRPT combined with SVM, is able to generate models to discriminate between healthy subjects and subjects with Ulcerative Colitis (UC) based on the expression values of genes in colon samples. We compared the predictive performance of our best models with a model generated by BioDiscML (a biomarker discovery tool) on two validation datasets and showed that our model achieves higher average precision.

4.1 Datasets

Various datasets, containing expression profile of healthy and Ulcerative Colitis (UC) subjects, were obtained from the Gene Expression Omnibus database (GEO). Table 4.1 shows the datasets used.

Accession number	# of selected controls	# of selected cases	Description of selected samples	Platform	# of genes (features)
GSE1152 [62]	4	4	mucosal biopsies from uninflammed colonic tissues of UC and control patients	Affymetrix Human Genome U133A Array and Affymetrix Human Genome U133B Array	19,353
GSE11223 [41]	24	25	biopsies from the uninflamed sigmoid colon of UC patients and healthy donors	Agilent-012391 Whole Human Genome Oligo Microarray G4112A	18,626
GSE22619 [24, 37]	10	10	biopsies from the sigmoid colon of siblings (healthy and diseased siblings)	Affymetrix Human Genome U133 Plus 2.0 Array	22,189
GSE75214-active [56]	11	74	biopsies from the inflamed colon of UC patients and from the colon of healthy donors	Affymetrix Human Gene 1.0 ST Array	20,358
GSE75214-inactive [56]	11	23	biopsies from the uninflamed colon	Affymetrix Human Gene 1.0 ST Array	20,358

Table 4.1: Summary of datasets

4.1.1 Data Collection and Pre-processing

Data collection and preprocessing is same as what described in Section 2.5.1. We used GSE1152, GSE11223, and GSE22619 for training and model selection, and GSE75214-active, and GSE75214-inactive for validation. Active means all samples are inflamed and inactive refers to uninflamed samples. Training datasets were merged by taking the genes present in all of them. The merged dataset has 39 UC samples and 38 normal subject, and 16,313 genes. Since the range of expression values of genes belonging to each dataset were different, we normalized the expression values of the final merged dataset and validation dataset by calculating Z-scores per sample.

4.1.2 Model Generation

To create a model to discriminate between UC patients from healthy subjects, we selected the features (genes) using recently introduced feature selection method based

on perturbation theory (DRPT). Let $D = [A \mid \mathbf{b}]$ be a dataset where $\mathbf{b}$ is the class label and A is an $m \times n$ matrix with n columns (genes) and m rows (samples). There is only a limited number of genes that are associated with the disease, and as such, a majority of genes are considered irrelevant. DRPT considers the solution $\mathbf{x}$ of the linear system $A\mathbf{x} = \mathbf{b}$ with the smallest 2-norm. Hence, $\mathbf{b}$ is a sum of $x_i F_i$ where F_i is the i-th column of A. Then each component x_i of $\mathbf{x}$ is viewed as an assigned weight to the feature F_i. So the bigger the $|x_i|$ the more important F_i is in connection with $\mathbf{b}$. DRPT then filters out features whose weights are very small compared to the average of local maximums over $|x_i|$'s. After removing irrelevant features, DRPT uses perturbation theory to detect correlations between genes of the reduced dataset. Finally, the remaining genes are sorted based on their entropy. The selected features were assessed using 5-fold cross-validation and support vector machines (SVMs) as the classifier. First, we performed DRPT 100 times on the training dataset to generate 100 subsets of features. Then, to find the best subsets, we performed 3 repetitions of stratified 5-fold cross-validation (CV) on the training dataset. We utilized the APS as the evaluation metric to determine the best subset of genes among those 100 generated subsets. The 10 subsets with the highest mean APS over the folds were chosen for creating the final models. For each of the selected subset of features, we created a final model using all samples in the training dataset. To evaluate the prediction performance of each model, we validated it on both validation datasets. In this step, we utilized the precision-recall curve as a performance metric to assess the performance of the models on unseen data. An additional model was created using the genes most frequently selected as features.

4.2 Results

We performed DRPT 100 times on the training dataset for selecting 100 subsets of features. Then we performed 5-fold CV to find the subsets with the highest mean APS over the folds. The range of APS for the 100 subsets is between 0.82 and 0.97, with an average of 0.91 ± 0.03. Table 4.2 shows the 10 subsets with the highest cross-validated APS and the number of features on each subset.

Table 4.2: 10 top subsets of features with the highest mean of APS on the training dataset after performing stratified 5-fold CV for three times.

Subset	**APS**	**#of Features**
Subset 10	0.97	42
Subset 51	0.97	47
Subset 58	0.97	32
Subset 83	0.97	39
Subset 5	0.96	37
Subset 16	0.96	30
Subset 33	0.96	27
Subset 55	0.96	22
Subset 62	0.96	46
Subset 74	0.96	50

We selected the top four subsets with the highest mean APS, which are subsets 10, 51, 58, and 83 and created final models based on them. Each final model was created using all samples of the training dataset and the features of the corresponding subset. To evaluate the prediction performance of the model, it was tested on the validation datasets, and the precision-recall curve was plotted for model assessment. To identify the most relevant genes to discriminate between healthy and UC subjects, we looked at the number of times genes were selected by DRPT. On 100 DRPT runs, 211 genes were selected at least once. The upper plot on Figure 4.1 shows the number of times each feature was selected, and the lower plot shows normal quantile-quantile (Q-Q) plot. If DRPT selected the genes randomly, then the points on the normal Q-Q plot

would have formed a straight line. Based on the plot, we observed that the genes that deviate the most from the normal distribution are those selected more than 31 times over 100 DRPT runs. We considered these genes as highly relevant and created a fifth model using 32 genes selected by DRPT at least 31 times over the 100 runs. Figures 4.2, and 4.3 present the precision-recall curves of all the five models tested on GSE75214-active and GSE75214-inactive datasets.

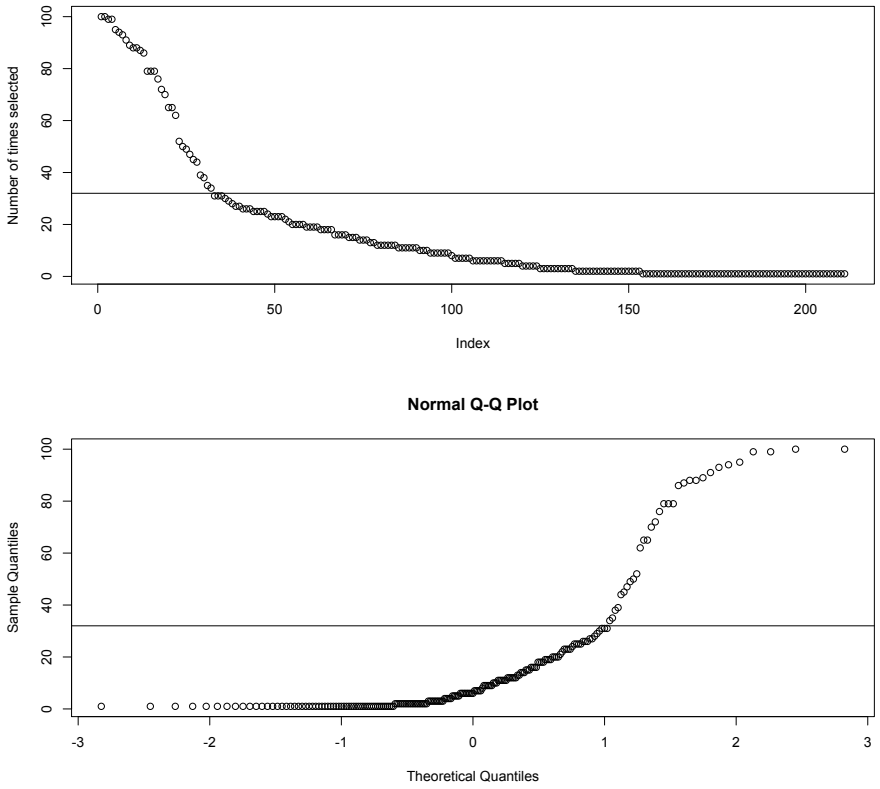

Figure 4.1: Quantile-Quantile plot

The average precision approximates the area under the precision-recall curve. We used average precision to summarize and compare the performance of the various models [40]. All five final models achieved better predictive performance on the validation dataset GSE75214-active with an average APS of 0.97 ± 0.03, while the average APS of the five final models on the validation dataset GSE75214-inactive is

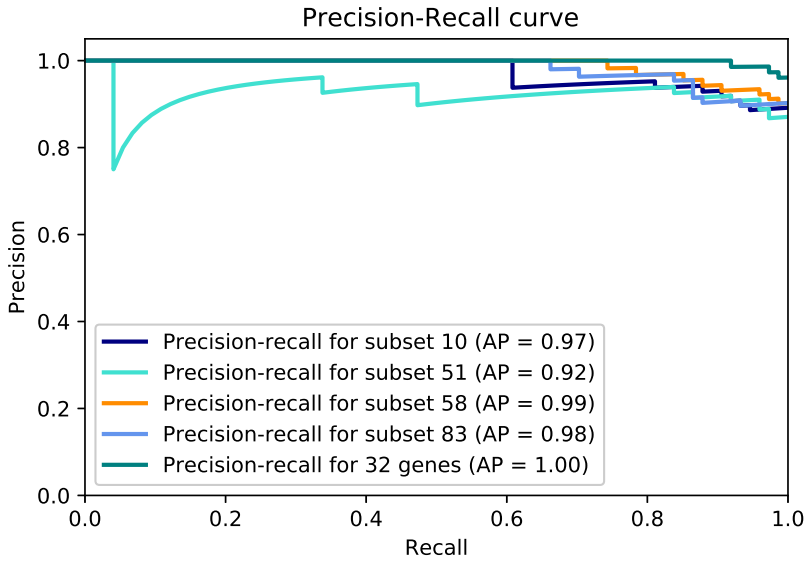

Figure 4.2: Precision-Recall Curve of testing the models created using features of the best subsets on GSE75214-active.

0.60 ± 0.06. Comparing the best model for each validation dataset, the models created with 32 most repeated genes and subset 83 obtained the highest APS on GSE75214-active and GSE75214-inactive, respectively. However, based on a Friedman test [13] ($p-value = 0.17$), there is no statistically significant difference among the five models.

We also performed DRPT on GSE11223, and 100 subsets of features were selected. This dataset contains only biopsies from the uninflamed sigmoid colon. After performing the 5-fold CV to find the best subsets which are subsets with the highest APS, 13 subsets had the highest APS (AP = 1.0). The range of APS for all 100 subsets is between 0.84 and 0.1, with an average of 0.97 ± 0.02. We created 13 final models based on those 13 subsets. Each final model was created using all the samples of GSE11223 and the features of the corresponding subset. For evaluating the performance of the model, it was tested on validation datasets, and the precision-recall curve was plotted for model assessment. Figures 4.4 and 4.5 show the precision-recall curves of validating the final models on each validation dataset. All 13 models achieved better

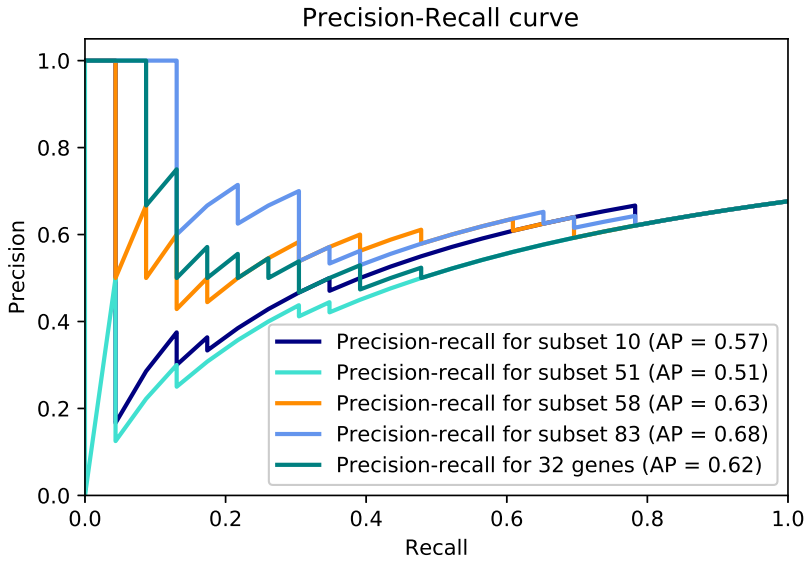

Figure 4.3: Precision-Recall Curve of testing the models created using features of the best subsets on GSE75214-inactive.

predictive performance on GSE75214-active with an average APS of 0.79 ± 0.57 than on GSE75214-active with an average APS of 0.60 ± 0.05. So we observed that even training on inactive datasets did not improve the prediction power of the final models for predicting inactive samples.

4.2.1 BioDiscML

BioDiscML [36] is a biomarker discovery software that uses machine learning methods to analyze biological datasets. To compare the prediction power of our models with BioDiscML, we ran the software on our training dataset. Note that 2/3 of the samples (N=52) were utilized for training and the remaining 1/3 (N=25) for testing. Since the software generates thousands of models, and we required only one model, we specified the number of best models as 1 in the config file (numberOfBestModels=1). One best model out of all models was created based on the 10-fold cross-validated Area Under Precision-Recall Curve (numberOfBestModelsSortingMetric= TRAIN-10CV-

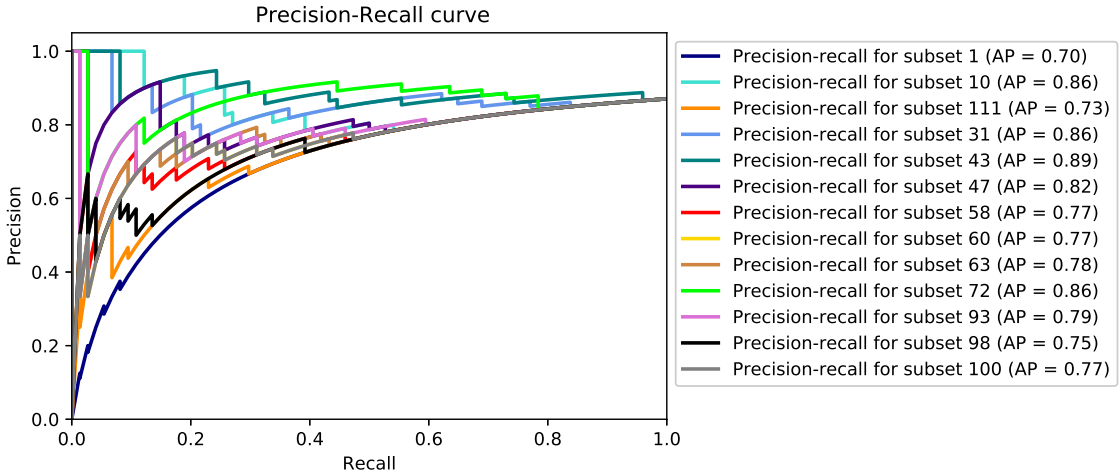

Figure 4.4: Precision-Recall Curve of testing the models created using features of the best subsets of GSE11223 on GSE75214-active.

AUPRC) on the training set. We used Weka 3.8 [21, 25, 59] to evaluate the model generated by the software, on both validation datasets. Selected features by BioDiscML are C3orf36, ADAM30, SLS6A3, FEZF2, and, GCNT3. In order to be able to use the model in Weka, we loaded the training dataset as it was created by BioDiskML, which was one of the outputs of the software. This dataset has 6 features, including selected genes and class labels, and 52 samples. We also prepared test datasets by reducing the number of features to the selected features by BioDisckML. After loading the training and test dataset in Weka explorer, we loaded the model, and we entered the classifier configuration as "weka.classifiers.misc.InputMappedClassifier -I -trim -W weka.classifiers.trees.RandomTree – -K 3 -M 1.0 -V 0.001 -S 1" which is the classifier's set up in the generated model by BioDiscML. The average AUPRC resulted from running the model on both GSE75214-active and GSE75214-inactive datasets was 0.798 and 0.544, respectively. Comparing the results of validating our final models and the model created by BioDiskML on validation datasets, we observed that we achieved better AUPRC on both datasets (AUPRC = 1 on the active dataset, AUPRC = 0.68 on the inactive dataset). In terms of time complexity, subset selection

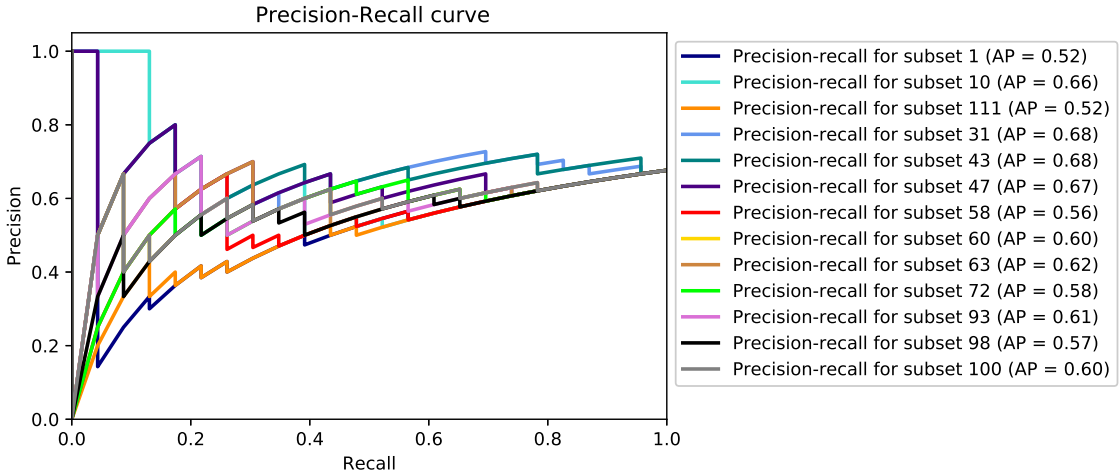

Figure 4.5: Precision-Recall Curve of testing the models created using features of the best subsets of GSE11223 on GSE75214-inactive.

by DRPT and final model creation and validation, took 3 minutes, while the running time of BioDiskML to create all the models and output the best final model was 1,890 minutes.

4.2.2 Analyzing the Most Repeated Genes

We used Ensembl REST API (Version 11.0) [61] to find the associated phenotypes with each gene belonging to the subset of the 32 most repeated genes. The corresponding results are shown in Table 4.3.

Among 32 genes, FAM118A has a phenotypic association with IBD and its subtypes. Patients suffering for a long time of IBD are more susceptible to develop colorectal cancer [29]. TFRC among our genes is associated with colorectal cancer. IBD patients are more prone to develop cardio vascular disease which this disease is associated with blood pressure and cholestrol [47]. We found LIPF, MMP2, DMTN and PPP1CB associated with blood pressure and cholestrol. We also used bedtools [44] to find out whether or not identified 241 IBD-associated SNPs [11] are located on our selected genes. First, we utilized Ensembl's BioMart [30] website

Table 4.3: Phenotypes associated with the 32 most repeated genes among 100 subsets

Gene Symbol	Associated Phenotypes	# of Repetition
CWF19L1	Spinocerebellar ataxia, autosomal recessive 17; depressive disorder, Major	100
FCER2	Blood protein levels; post bronchodilator FEV1	100
MMP2	Multicentric Osteolysis-Nodulosis-Arthropathy (MONA) spectrum disorders; cholesterol, HDL; lip and oral cavity carcinoma; body height; winchester syndrome	99
PPP1CB	Noonan Syndrome-like disorder with loose anagen hair 2; Heel bone mineral density; Blood pressure; basophils asopathy with developmental delay; short stature and sparse slow-growing hair	99
RPL23AP32	Attention deficit disorder with hyperactivity; body Height	95
ZNF624	None	94
REG1B	Contrast sensitivity; Body Mass Index	93
TFRC	Breast ductal adenocarcinoma; esophageal adenocarcinoma; thyroid carcinoma; clear cell renal carcinoma; prostate carcinoma; pancreatic cancer; gastric adenocarcinoma; hepatocellular carcinoma; lung adenocarcinoma; rectal adenocarcinoma; basal cell carcinoma; colorectal adenocarcinoma; squamous cell lung carcinoma; head and neck squamous cell carcinoma; colon adenocarcinoma; iron status biomarkers (transferrin levels); mean corpuscular hemoglobin concentration; red cell distribution width; combined immunodeficiency; red blood cell traits; high light scatter reticulocyte percentage of red cells; reticulocyte fraction of red cells; Immunodeficiency 46	91
FAM118A	**Chronic inflammatory diseases** (ankylosing spondylitis, Crohn's disease, psoriasis, primary sclerosing cholangitis, ulcerative colitis); Glucose; Peanut allergy (maternal genetic effects); Heel bone mineral density	89
CFHR2	Macular degeneration; blood protein levels; feeling miserable; alanine aminotransferase (ALT) levels after remission induction therapy in actute lymphoblastic leukemia (ALL); asthma	88
KRT8	Cirrhosis; familial cirrhosis; hepatitis C virus; suseptibility to, cirrhosis, cryptogenic cirrhosis, noncryptogenic cirrhosis; susceptibility to, gamma glutamyl transferase levels, cancer (pleiotropy)	88
PRELID1	Body fat distribution; heel bone mineral density; activated partial thromboplastin time	87
ZNF92	None	86
ABHD2	Itch intensity from mosquito bite adjusted by bite size; gut microbiota; Obesity-related traits; coronary artery disease; advanced age related macular degeneration; squamous cell lung carcinoma; pulse pressure	79
C16orf89	None	79
CAB39L	Hemoglobin S; erythrocyte count; pancreatic neoplasms	79
SPATC1L	None	76
DUOXA2	Familial thyroid dyshormonogenesis; thyroglobulin synthesis defect	72
MESP1	None	70
MAML3	Social science traits; intelligence (MTAG); chronic mucus hypersecretion; borderline personality disorder; bongenital heart malformation	65
PITX2	Axenfeld-Rieger syndrome; ring dermoid of cornea; iridogoniodygenesis type 2; peters anomaly; familial atrial fibrillation; rieger anomaly; stroke; ischemic stroke; cataract; PITX2-related eye abnormalities; phosphorus; cognitive decline rate in late mild cognitive impairment; creatinine; intraocular pressure; incident atrial fibrillation; wolff-parkinson-white pattern; parkinson disease; early onset atrial fibrillation; anterior segment sygenesis 4	65
DMTN	Total cholesterol levels; LDL cholesterol	62
ASF1B	None	52
PGF	Mood instability; blood protein levels	50
BEX4	None	49
ODF1	Body weight; body mass index; glucose; IgA nephropathy; Chronic lymphocytic leukemia; type 2 diabetes; erythrocyte indices	47
PTGR1	Body height; menarche; monocyte count; blood protein levels	45
ZNF35	None	44
LIPF	Maximal midexpiratory flow rate; blood protein levels; respiratory function tests; blood pressure	39
SLC25A13	Citrullinemia type II; neonatal intrahepatic cholestasis due to citrin deficiency; citrin deficiency; citrullinemia type I; bone mineral density	38
BARX2	Type 2 diabetes; breast cancer; night sleep phenotypes; response to cyclophosphamide in systemic lupus erythematosus with lupus nephritis; stroke	35
C2orf42	None	34

http://www.ensembl.org/biomart/martview/ to retrieve the required information about the selected 32 genes. We obtained the chromosome name, the genomic position of all human genes which are the start position of the genes in chromosome coordinates, and the end position of the gene in chromosome coordinates. The version we used is Ensembl Release version 98 - September 2019. Table 4.4 shows a sample of the data downloaded from Ensembl's BioMart website.

Table 4.4: A sample of a data downloaded from Ensembl's Biomart

Chromosome/scaffold name	**Gene name**	**Gene start (bp)**	**Gene end (bp)**
15	DUOXA2	45114326	45118421
14	PGF	74941834	74955626
19	FCER2	7688758	7702146
.	.	.	.
.	.	.	.
.	.	.	.

Then we created a BED file with four columns, including the name of the chromosome, start and endpoints, and gene names using downloaded data from Ensembl's BioMart. We also created a BED file including chromosome name, the start position of each SNP in chromosome coordinates, end position of each SNP in chromosome coordinates, and SNP's rs number using 241 IBD-associated SNPs data. We calculated the start position of each SNP by subtracting 1 from the end position of it, which was available. Then intersectBed utility was used to figure out the intersection of the two BED files. The intersection results showed that none of the IBD-associated SNPs were located on selected genes.

4.3 Summary

In this chapter, DRPT feature selection method combined with SVM was used to generate models to discriminate subjects with UC and healthy ones based on gene

expression values of genes in colon samples. The best five generated models were validated on two validation datasets, and better predictive performance was achieved on the GSE75214-active dataset with an average APS of 0.97 ± 0.03. Comparing the performance of our best model with the model generated by BioDiscML showed that a higher average precision score was achieved on both validation datasets. We also analyzed the 32 most repeated genes over 100 runs of DRPT. We found 6 genes out of the 32 genes that have support from literature to be associated with IBD. Surprisingly, none of our genes harbor already known SNPs associated with IBD.

Chapter 5

Conclusions

In this thesis, we proposed a sparse method (SLS) based on singular value decomposition and least squares to filter out irrelevant features from the dataset and reducing the size of it. We also reduced the size of the dataset by clustering features with the k-means algorithm and selecting representative features from each cluster using two metrics, namely Euclidean distance and entropy. We augmented our method and two other approaches to three well-known feature selection methods to select subsets of genes from three IBD microarray datasets. Then, we created models based on selected features by each method using support vector machine and random forest classifiers. The results showed that the proposed SLS method outperforms two other approaches in terms of the prediction power of the models and the computational time of the feature selection algorithms. We also showed in Section 3.4 that SLS removed irrelevant features while preserving informative features. A novel feature selection algorithm (DRPT) combined with SVM was used to create a model to differentiate between healthy subjects and subjects with Ulcerative Colitis. We validated the best models on two validation datasets. One of them contained biopsies from the inflamed colon and the other one biopsies from the uninflamed colon. First, we created models us-

ing the best subsets selected from a training dataset which contains samples from both inflamed and uninflamed subjects. Then, we tested the models on both validation datasets. The results showed that the predictive performance of the best five final models were better on the validation dataset GSE75214-active, with an average APS of 0.97 ± 0.03. Comparing the best model for each validation dataset, the model created with all the samples of training dataset and 32 most repeated genes in 100 runs of DRPT, achieved the highest APS = 1.0 when tested on GSE75214-active. Also, the model created with subset 83 yielded the higher APS = 0.68 when tested on GSE75214-inactive. So we concluded that predicting inflamed samples are easier than uninflamed ones. To support our conclusion, we also created models using the best subsets of features selected from GSE11223, which contains biopsies from the uninflamed sigmoid colon of UC patients, and we tested them on both validation datasets. The results showed that even training on uninflamed samples did not improve the prediction power of the final models for predicting inactive samples. We also compared our best model with the model generated by BioDiskML, which is a recently developed biomarker discovery software, and we showed that we achieved higher predictive performance. After analyzing 32 genes that were selected repeatedly in 100 runs of DRPT, we found 6 genes that have an association with IBD.

Bibliography

[1] M. Afshar and H. Usefi. High-Dimensional Feature Selection for Genomics Datasets. Submitted.

[2] Badr Albanna, Christopher Hillar, Jascha Sohl-Dickstein, and Michael DeWeese. Minimum and maximum entropy distributions for binary systems with known means and pairwise correlations. *Entropy*, 19(8):427, 2017.

[3] Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[4] Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, et al. NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1):D991–D995, 2012.

[5] Gabrio Bassotti, Elisabetta Antonelli, Vincenzo Villanacci, Marianna Salemme, Manuela Coppola, and Vito Annese. Gastrointestinal motility disorders in inflammatory bowel diseases. *World Journal of Gastroenterology: WJG*, 20(1):37, 2014.

[6] Daniel C Baumgart and Simon R Carding. Inflammatory bowel disease: cause and immunobiology. *The Lancet*, 369(9573):1627–1640, 2007.

[7] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] Michael E Burczynski, Ron L Peterson, Natalie C Twine, Krystyna A Zuberek, Brendan J Brodeur, Lori Casciotti, Vasu Maganti, Padma S Reddy, Andrew Strahs, Fred Immermann, et al. Molecular classification of Crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells. *The Journal of Molecular Diagnostics*, 8(1):51–61, 2006.

[9] Sandipan Chakroborty and Goutam Saha. Feature selection using singular value decomposition and QR factorization with column pivoting for text-independent speaker identification. *Speech Communication*, 52(9):693–709, 2010.

[10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[11] Katrina M de Lange, Loukas Moutsianas, James C Lee, Christopher A Lamb, Yang Luo, Nicholas A Kennedy, Luke Jostins, Daniel L Rice, Javier Gutierrez-Achury, Sun-Gou Ji, et al. Genome-wide association study implicates immune activation of multiple integrin genes in inflammatory bowel disease. *Nature Genetics*, 49(2):256, 2017.

[12] Zeinab Dehghan and Eghbal G Mansoori. A new feature subset selection using bottom-up clustering. *Pattern Analysis and Applications*, 21(1):57–66, 2018.

[13] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[15] Claudio Fiocchi. Inflammatory bowel disease: Etiology and pathogenesis. *Gastroenterology*, 115(1):182–205, 1998.

[16] Cesare Furlanello, Maria Serafini, Stefano Merler, and Giuseppe Jurman. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics*, 4(1):54, 2003.

[17] Debashis Ghosh. Singular value decomposition regression models for classification of tumors from microarray experiments. In *Biocomputing 2002*, pages 18–29. World Scientific, 2001.

[18] Gene Golub, Virginia Klema, and Gilbert W Stewart. Rank degeneracy and least squares problems. Technical report, Stanford University, Dept of Computer Science, 1976.

[19] Gene H Golub and Charles F Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980.

[20] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[21] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[22] Mark A Hall and Lloyd A Smith. Practical feature subset selection for machine learning. *Proceedings of the 21st Australasian Computer Science Conference ACSC'98*, pages 181–191, 1998.

[23] Leena Halme, Paulina Paavola-Sakki, Ulla Turunen, Maarit Lappalainen, Martti Färkkilä, and Kimmo Kontula. Family and twin studies in inflammatory bowel disease. *World Journal of Gastroenterology: WJG*, 12(23):3668, 2006.

[24] Robert Häsler, Zhe Feng, Liselotte Bäckdahl, Martina E Spehlmann, Andre Franke, Andrew Teschendorff, Vardhman K Rakyan, Thomas A Down, Gareth A Wilson, Andrew Feber, et al. A functional methylome map of ulcerative colitis. *Genome research*, 22(11):2130–2137, 2012.

[25] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. Weka: A machine learning workbench. 1994.

[26] Jianping Hua, Waibhav D Tembe, and Edward R Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409–424, 2009.

[27] Xiaojuan Huang, Li Zhang, Bangjun Wang, Fanzhang Li, and Zhao Zhang. Feature clustering based support vector machine recursive feature elimination for gene selection. *Applied Intelligence*, 48(3):594–607, 2018.

[28] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier, 1994.

[29] Eun Ran Kim and Dong Kyung Chang. Colorectal cancer in inflammatory bowel disease: the risk, pathogenesis, prevention and diagnosis. *World Journal of Gastroenterology: WJG*, 20(29):9872, 2014.

[30] Rhoda J Kinsella, Andreas Kähäri, Syed Haider, Jorge Zamora, Glenn Proctor, Giulietta Spudich, Jeff Almeida-King, Daniel Staines, Paul Derwent, Arnaud

Kerhornou, et al. Ensembl biomarts: a hub for data retrieval across taxonomic space. *Database*, 2011, 2011.

[31] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine Learning Proceedings 1992*, pages 249–256. Elsevier, 1992.

[32] Kenji Kira, Larry A Rendell, et al. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, volume 2, pages 129–134, 1992.

[33] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[34] Igor Kononenko. Estimating attributes: analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182. Springer, 1994.

[35] Martin HC Law, Mario AT Figueiredo, and Anil K Jain. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[36] Mickael Leclercq, Benjamin Vittrant, Marie-Laure Martin-Magniette, Marie-Pier Scott-Boyer, Olivier Perin, Alain Bergeron, Yves Fradet, and Arnaud Droit. Large-scale Automatic Feature Selection for Biomarker Discovery in High-Dimensional OMICs data. *Frontiers in Genetics*, 10:452, 2019.

[37] Patricia Lepage, Robert Häsler, Martina E Spehlmann, Ateequr Rehman, Aida Zvirbliene, Alexander Begun, Stephan Ott, Limas Kupcinskas, Joël Doré, Andreas Raedler, et al. Twin study indicates loss of interaction between microbiota and mucosa of patients with ulcerative colitis. *Gastroenterology*, 141(1):227–236, 2011.

[38] Jake Lever, Martin Krzywinski, and Naomi Altman. Points of significance: classification evaluation, 2016.

[39] Li Liu, Douglas M Hawkins, Sujoy Ghosh, and S Stanley Young. Robust singular value decomposition analysis of microarray data. *Proceedings of the National Academy of Sciences*, 100(23):13167–13172, 2003.

[40] Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.

[41] Colin L Noble, Alexander R Abbas, Jennine Cornelius, Charlie W Lees, Gwo-Tzer Ho, Karen Toy, Zora Modrusan, Navneet Pal, Fiona Zhong, Sreedevi Chalasani, et al. Regional variation in gene expression in the healthy colon is dysregulated in ulcerative colitis. *Gut*, 57(10):1398–1405, 2008.

[42] 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. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[43] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(8):1226–1238, 2005.

[44] Aaron R Quinlan and Ira M Hall. BEDtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[45] Sebastian Raschka. *Python machine learning*. Packt Publishing Ltd, 2015.

[46] Alexander L Richards, Peter Holmans, Michael C O'Donovan, Michael J Owen, and Lesley Jones. A comparison of four clustering methods for brain expression microarray data. *BMC bioinformatics*, 9(1):490, 2008.

[47] DM Schulte, K Paulsen, K Türk, B Brandt, S Freitag-Wolf, I Hagen, R Zeuner, JO Schröder, W Lieb, A Franke, et al. Small dense LDL cholesterol in human subjects with different chronic inflammatory diseases. *Nutrition, Metabolism and Cardiovascular Diseases*, 28(11):1100–1105, 2018.

[48] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[49] Paulo JS Silva, Ronaldo F Hashimoto, Seungchan Kim, Junior Barrera, Leônidas O Brandão, Edward Suh, and Edward R Dougherty. Feature selection algorithms to find strong genes. *Pattern Recognition Letters*, 26(10):1444–1453, 2005.

[50] Qinbao Song, Jingjie Ni, and Guangtao Wang. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):1–14, 2011.

[51] Daniel J Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2011.

[52] Yuchun Tang, Yan Qing Zhang, and Zhen Huang. Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(3):365–381, 2007.

[53] Anbupalam Thalamuthu, Indranil Mukhopadhyay, Xiaojing Zheng, and George C Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412, 2006.

[54] Sabine Van Huffel and Joos Vandewalle. Subset selection using the total least squares approach in collinearity problems with errors in the variables. *Linear Algebra and its Applications*, 88:695–714, 1987.

[55] Charles F Van Loan and Gene H Golub. *Matrix computations*. Johns Hopkins University Press, 1983.

[56] Maaike Vancamelbeke, Tim Vanuytsel, Ricard Farré, Sare Verstockt, Marc Ferrante, Gert Van Assche, Paul Rutgeerts, Frans Schuit, Séverine Vermeire, Ingrid Arijs, et al. Genetic and transcriptomic bases of intestinal epithelial barrier dysfunction in inflammatory bowel disease. *Inflammatory Bowel Diseases*, 23(10):1718–1729, 2017.

[57] Kelli L VanDussen, Ta-Chiang Liu, Dalin Li, Fadi Towfic, Nir Modiano, Rachel Winter, Talin Haritunians, Kent D Taylor, Deepti Dhall, Stephan R Targan, et al. Genetic variants synthesize to produce paneth cell phenotypes that define subtypes of Crohn's disease. *Gastroenterology*, 146(1):200–209, 2014.

[58] Yu Wang, Igor V Tetko, Mark A Hall, Eibe Frank, Axel Facius, Klaus FX Mayer, and Hans W Mewes. Gene selection from microarray data for cancer classification—a machine learning approach. *Computational Biology and Chemistry*, 29(1):37–46, 2005.

[59] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[60] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation*, 26(1):185–207, 2014.

[61] Andrew Yates, Kathryn Beal, Stephen Keenan, William McLaren, Miguel Pignatelli, Graham RS Ritchie, Magali Ruffier, Kieron Taylor, Alessandro Vullo, and Paul Flicek. The Ensembl REST API: Ensembl data for any language. *Bioinformatics*, 31(1):143–145, 2014.

[62] Alexandra Zahn, Christoph Moehle, Thomas Langmann, Robert Ehehalt, Frank Autschbach, Wolfgang Stremmel, and Gerd Schmitz. Aquaporin-8 expression is reduced in ileum and induced in colon of patients with ulcerative colitis. *World Journal of Gastroenterology: WJG*, 13(11):1687, 2007.

[63] Federico Zambelli, Francesca Mastropasqua, Ernesto Picardi, Anna Maria D'Erchia, Graziano Pesole, and Giulio Pavesi. RNentropy: an entropy-based tool for the detection of significant variation of gene expression across multiple RNA-Seq experiments. *Nucleic Acids Research*, 46(8):e46–e46, 2018.