Vegard Bjørgan

Classifying Combined MicroRNA Data Sets

Master Project, Spring 2019

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

Abtract

Since the first discoveries of non-coding RNA in 1993 researcher have found several thousands different microRNAs in the human genome over the last decades. MicroRNAs as a topic has become especially hot when researching cancer due to it's ability to regulate many of the protein-coding RNAs. Various machine learning methods have been employed in cancer diagnostic research. Machine learning methods can generate more accurate diagnoses or prognoses than traditional statistical methods can.

In cancer cell division become abnormal and uncontrolled which also arises from the misregulation of several genes. MicroRNAs are major regulators of gene expression and thus it is not surprising that miRNAs are actively altered in different types of cancer.

In this paper several techniques are used to classify combined microRNA data sets for both colorectal- and hepatic cancer. Techniques includes several types of normalization, feature selection, algorithms and Gene Set Enrichment Analysis. The most important features for the different classification techniques is also extracted for both diseases.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This introduction to background and motivation states where this project is situated in the field and what the key driving forces motivating this research are. An outline for the projects goal and its main contributions are also listed.

1.1 Background and Motivation

The first microRNAs (miRNAs) were discovered in Lee et al. (1993). MiRNAs are small non-coding RNAs that regulate the translation process of messenger RNAs (mRNAs) into proteins. Since then a lot of research has gone into discovering new miRNAs, finding miRNAs target mRNAs and linking miRNAs to several diseases including different types cancer. Great steps in bioinformatics, machine learning, new algorithms and increased processing power have facilitated this booming research. In addition several online sources such as mirbase.org are now available to make published miRNA sequences searchable.

Studying the relationships between miRNAs and different diseases can help us better understand the diseases, and it can help produce better diagnosis, prognosis and in recent years therapies for patients. MiRNA data sets are prone to be small in samples compared to features because of a relative high cost of producing them, thus making a classifier to separate a single data set will in many cases overfit the biases from that data set. To make a classifier for several data sets one would first have to eliminate an internal bias of each data set. This internal bias can come from several sources most of which are laboratory related and to varying degree unavoidable. The focus in this project is to examine different ways to eliminate internal biases each miRNA data set and in doing so making a more universal classifier for the miRNA data sets.

1.2 Goals and Research Questions

The goal of this project is finding a general normalization and / or external calculations such that a classifier can accurately classify microRNA samples as normal or tumor with data from several data sets, then extract which microRNAs are essential to preserve correct classification.
To help address these problems the following research questions were used.

RQ1 What are the existing solutions to classifying data with different biases in Machine learning?

RQ2 What are the methods for making gene expressions comparable?

RQ3 What Machine Learning algorithms are suitable for data low in samples and high in features?

RQ4 What pre-processing is done with miRNA expressions?

RQ5 How does the solutions found when addressing RQ1 work when considering imbalanced data sets and gene expressions?

RQ6 What implications will these findings have when creating a solution?

1.3 Research Method

As no previous work was found combining different miRNA data sets, the work was based of advise in classification of single miRNA data sets. No similar prior work also indicated the need for experiments to evaluate the advise for this thesis's problems. These advises gave a broad overview over how classification can be done and created a outline for what experiments should be done and how they should be done.

Eight data sets from patients with either colorectal cancer and hepatic cancer were used. Using these data sets a series of experiments were done testing different pre-processing combinations and algorithms. Experiments for imbalance in data sets were also done testing the previous found methods performance in data sets down to single samples. Finally a experiment were done extracting the features who's importance were highest for different methods.

1.4 Contributions

The main contributions of this project are

- Methods for combining miRNA data sets which removes internal bias to different degrees.
- An evaluation of several pre-processing and algorithm combinations when classifying combined miRNA data sets.
- Lists of miRNAs found to be related to colorectal cancer and hepatic cancer respectively.

1.5 Thesis Structure

Chapter two contains the background theory of both biological and algorithmic aspects, summary of related works and motivation. In chapter three the different normalization techniques, performance metrics, feature selection- and cross-validation techniques used in this project are listed. Chapter four contains the different experiments, their setups and the results of these. Chapter five contains a discussion of the results, conclusions found and what future work can be done. Bibliography and an appendix are the last chapters.

Chapter 2

Background Theory and Motivation

In this chapter current state-of-the-art regarding cancer classifications using microRNAs is presented. First the relevant knowledge of biology and bioinformatics are presented. Second a depth summary of relevant background in machine learning are presented. Third the research done on similar problems with similar constraints are summarized. A protocol for where this research can be found and why it is included can be found in the structured literature review protocol. The chapter ends with a motivation for possible applications and benefits for using machine learning on microRNAs.

2.1 Background Theory on Biology

This section presents current state-of-the-art in bioinformatics amd biology with regards to microRNAs. Papers in the biology and bioinformatics field were provided by supervisor. An additional source for Gene Set Enrichment Analysis was included from the paper with most cites according to Google Scholar.

2.1.1 Transcriptome, Microarrays and Sequencing Technology

The transcriptome can roughly be described as a complete set of transcripts in a cell and their quantity, i.e. it is the set of all RNA molecules in one cell. To understand the transcriptome is essential when interpreting the functional elements of the genome. Furthermore it represents the molecular constituents of cells and tissues, which allows for better understanding of development and disease. The transcriptomics seek to catalogue all species of transcripts to determine the transcriptional structure of genes and quantifying the changing expression levels for each transcript during development and under different conditions and diseases, such as cancer. The species of transcript include messenger RNA (mRNA) and small non-coding RNA called microRNA (miRNA). The transcriptional structure is expressed in terms of their start sites, 5' and 3' ends.

There exist several techniques to deduce and quantify the transcriptome, including microarray and RNA-sequencing. A microarray is typically a two-dimensional array with single-stranded DNA probes. RNA molecules from cells are transformed to complementary DNA, flourescently labelled, and allowed to hybridise to the microarrays probes. This allows for measuring the expression levels of a large number genes simultaneously by scanning the microarray with a laser.

RNA-sequencing uses high-throughput sequencing to identify which genes are active, and how much they are transcribed. To do this the RNA-sequencing must have a library of cDNA fragments with adapters attached to one of both ends. The libraries are usually constructed from a population of RNA, however several manipulation stages are involved in constructing the cDNA libraries. Each molecule is sequenced in a high-throughput manner to obtain short sequences from one end or both ends. The result is a precise discrete value for all genes expression levels.

The different techniques each have their pros and cons, and therefore one is not strictly better than the other. Microarrays expression powers are limited because of it's reliance upon existing knowledge about genome sequences. In addition comparing expression levels across different experiments can require complicated normalization methods due to both background and saturation of signals. RNA-sequencing does not have a problem with being limited to detecting transcripts that correspond to existing genomic sequence. Nor does it have any background signal because the DNA sequences can be unambiguously mapped to unique regions of the genome. Lastly the RNA-sequencing offers higher precision and requires less sample than microarrays. RNA-sequencing do face some challenges itself regarding its library construction. First of the larger RNA molecules must be fragmented into smaller pieces approximately 200-500 base pairs (bp) to be compatible with most deep-sequencing technologies. These fragmentation techniques creates a different bias in the outcome. Furthermore some manipulations during library construction also complicate the analysis of RNA-sequencing results. Lastly RNA-sequencing faces some bioinformatics challenges with storing, retrieving and process large amounts of data (Wang et al., 2009).

2.1.2 MicroRNAs

MicroRNAs are a class of short non-coding RNA sequences. The miRNAs regulate many genes by base pairing to sites in mRNAs. This makes them appealing targets for screening, diagnosis, prognosis, monitoring tumor progression, biomarker discovery and evaluation of correct treatment for patients. Bertoli et al. (2016) stated that they are specifically of interest as they regulate the expression of specific target genes, including tumor suppressors and oncogenes. That is the genes that protects a cell from one step on the path to cancer and the genes that has potential to cause cancer.

In the review Saito and Sætrom (2010), the targeting and target prediction of miRNAs are explained. Furthermore they also state that miRNAs regulate protein-coding genes post transcription. This is done by guiding a protein complex known as the RNA-induced silencing complex (RISC) to mRNAs with partial complementary to the miRNA. In general the miRNAs bind to the 3'-UTR (untranslated region) of their target mRNAs and repress protein production by destabilizing the mRNA and translational silencing. Although the full mechanism of this is not yet fully determined.

To summarize in simplistic terms: The miRNAs regulate mRNAs and represses protein production and are of interest because it also involves tumor suppressors and oncogenes.

A high regulation of a tumor suppressor could cause the gene not to protect the cell on the way to cancer and a low regulation on a oncogene could cause the gene to develop cancer.

2.1.3 Gene Set Enrichment Analysis

Gene set enrichment analysis (GSEA) is a technique that considers expression profiles from samples belonging to two classes. By considering a predefined set of genes, the GSEA aims to determine whether the members of the set of genes are randomly distributed throughout the sample or is primarily found at the top or bottom.

In GSEA there are three main steps: Calculation of an Enrichement Score(ES), estimation of significance level of ES and adjustment for multiple hypothesis testing. Whereas the first is the most important for this thesis. For calculation a predefined set of genes must first be provided. Then for a given sample the genes are put into a ranked list. This ranked list is walked through and the degree to which the predefined set is present at top or bottom i calculated. The ES is increased when a gene from the predefined set is encountered and decreased when a gene not in the predefined set is encountered. This calculation corresponds to a weighted Kolmogorov-Smirnov-like statistic (Subramanian et al., 2005). The second step is essentially validation of the ES using phenotype labels and thus create a more biologically reasonable assessment of significance. The last step is normalization of the enrichment scores and adjusting scores for when running multiple predefined sets giving multiple hypotheses.

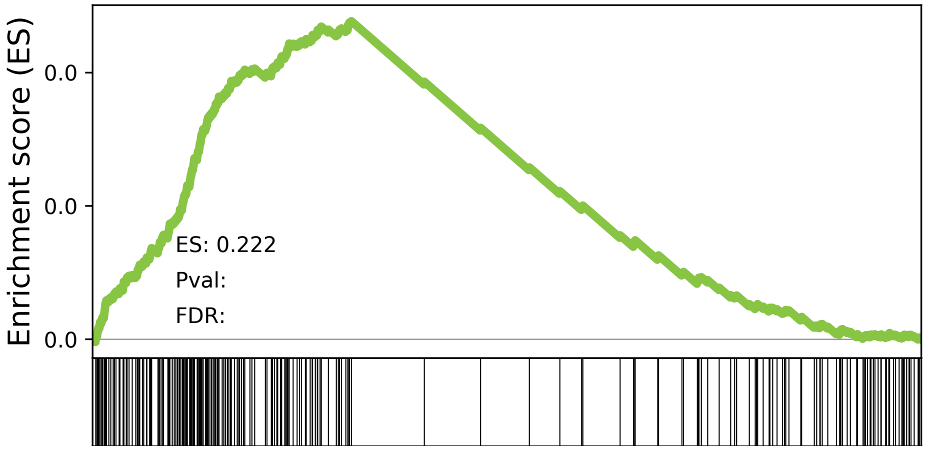

Figure 2.1: Example of GSEA enrichment score calculated on a sample. The green line indicates the enrichment score as the sample is walked through. The black bars indicate hits on the predefined set of genes. ES, P value (Pval) and false discovery rate (FDR) represents the three steps in GSEA and here only the first step is run.

2.2 Background Theory on Algorithms

In this section the relevant knowledge of the most common algorithms in classification of microRNAs are presented. Books and papers in machine learning were cherry picked with regards to material that I have previously had as syllabus in courses. In the cases where the topic had not been syllabus the paper with most citation for the specific term were chosen as a source.

2.2.1 Classification problems

A classification problem is about grouping data by particular criteria. Classification is the process where a set of input for a new observation is mapped to an output based on earlier observations whose group is known. A popular example for classification is grouping email by spam or non-spam. Classification is considered an instance of supervised learning, i.e. learning where training set of correctly identified observations are available. The corresponding unsupervised procedure is called clustering and groups observations based on some measure of inherent similarity. Classification often requires analyzing data into a set of quantifiable properties, known variously as explanatory variables or features.

2.2.2 Decision trees

Russell and Norvig (2016) stated decision trees are one of the simplest yet most successful forms of machine learning. Its rather simplistic metrics and natural flowchart like representation makes it easy to fathom and allows it to be an essential building block for more advanced algorithms.

In general a decision tree takes a vector of attributes, discrete or continuous, and returns its classification or the ”decision”. To reach a decision the decision trees performs a series of tests for the attributes. Each node in the tree represents such a test and each branch is the result of the test. The ”decision” returned lies in the trees leaf nodes. An example of this can be seen in Figure 2.2.

As with most classifiers, when constructing a decision tree classifier the set of examples i crucial. They should ideally convey a representative subset of the data for best performance. To build the decision tree the decision tree learning algorithm selects the feature that best splits the data into correct classes. This process is then repeated until all samples are correctly classified through the tree.

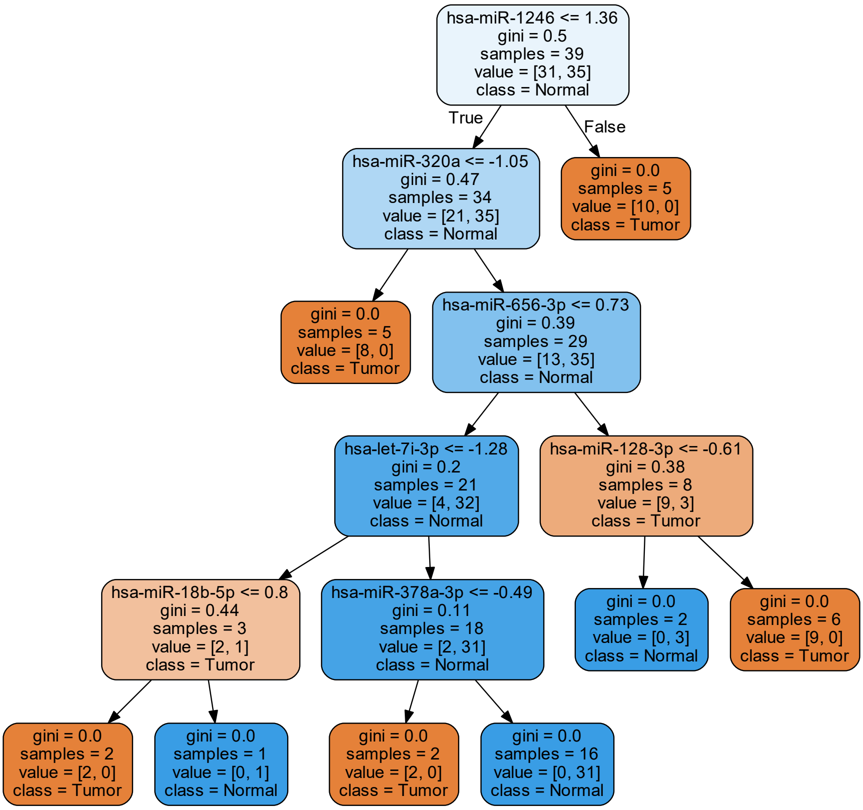

Figure 2.2: A decision tree estimator from the project. For each node the first row is the test. True is left and False is right for all tests. The second row is the gini score, further explained in section 3.1.4, which is a measure for how well this test splits the samples. The third row is the number samples which are correctly split from this test. The fourth row is the number of tumor and normal samples. The fifth row is the most common sample before a split is done. Note that the leaf nodes does not have a test.

2.2.3 Random Forest

Random forest is an ensemble machine learning technique which uses several decision trees to do its classifications or predictions. This technique is more robust than a single decision tree because it is less prone to overfitting the training data. Also Breiman (2001) suggests that because of the Strong Law of Large Numbers the random forest always converge which implies overfitting is not a problem. Furthermore the accuracy in random forest depends on the strength of each tree classifier.

To do a prediction the random forest runs the a sample trough all the decision trees in the forest and can return both probability for each class or the most common class predicted in the forest.

The random in random forest comes from the fact that there needs to be employed some randomization to make each tree independent of each other. First random forest gives each tree a random sample with replacement from training set. Second each individual tree is given a random subset of features to be used when searching for splits or tests. The key in random forest is therefore is injecting the right kind of randomness to make them accurate predictors and regressors.

Breiman (2001) concludes that random forests gives results competitive with boosting and adaptive bagging. The accuracy in random forests also indicate that they act to reduce bias, however the concrete mechanisms that reduce bias is not obvious. He also concludes the improvement in random forest are greater in larger data sets then smaller ones.

2.2.4 Ensemble techniques

Ensemble techniques are meta-algorithms that combine several machine learning techniques into one in order to improve prediction and decrease variance or bias. These meta-algorithms are either sequential or parallel.

Sequential methods are typically called boosting. In boosting the overall performance can increase by weighing samples that are misclassified with higher weight. This is done by first training a classifier on the data then create a weighted version of the data based on what samples it predicts wrong. In sequential rounds data points that are misclassified receive higher weights and data points that are correctly classified receive gets their weight decreased.

Parallel method are called bagging. In bagging algorithms such as random forest several classifiers are ran in parallel with different subsets of features and samples. The predictions of these can be voting for classification or averaged for regression.

Both bagging and boosting are mostly used with one type base classifier. Combining multiple types of classification or regression models are also possible and are commonly known as stacking. In stacking the meta-classifier is trained on the output of the base classifiers. However adding more models or layers to a classifier does not always lead to better prediction. In addition to the increased difficultly of explaining the model and its predictions. Thus can a more complex model have less real world value depending on the problem it solves.

A good example of a complex model using all techniques is the winner of a Kaggle data science competition in 2015 called Otto Group Product Classification Challenge. This model uses 33 meta features from other models and 7 features from the original data set. These features are then put into two boosting algorithms and a neural network and their predictions again are averaged.

2.2.5 Support vector machines

Support vector machines or SVMs is based around the notion of a ”margin”, the distance of either side of a hyperplane that separates two data classes. Maximizing the margin, and thus the distance between the margin and either class, has been proven to reduce the upper bound of expected generalization error (Kotsiantis et al., 2007).

If two classes are linearly separable, the optimum separating hyperplane can be found by solving a convex quadratic programming (QP) problem:

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \Phi(w) = \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1, i = 1, ..., l. \end{aligned} \tag{2.1}$$

When a optimal separating hyperplane is found, it is represented as a linear combination of the points that lie on its margin. This helps the model complexity of a SVM to be unaffected by the number of features encountered in training. Thus making it suitable to deal with learning problems with a large number of features compared to samples.

In cases where no separating hyperplane can be found a soft margin that accepts some misclassification in the training samples can be used. This soft margin is usually called C and is a hyperparameter in the SVM.

When the data is not separable, and thus no hyperplane exists that successfully separates the classes of the training set, the inseparability problem is mapped onto a higher-dimensional space. This higher-dimensional space is called the transformed feature space. In the higher-dimensional space of sufficient dimensionality any consistent training set can be made separable. A separable transformed feature space corresponds to a non linear separation in the original input space.

To map data from the input space onto a transformed feature space is however not trivial. Luckily training does only depend on dot products in the transformed feature space. This allows for a kernel function that calculates inner products in input space without ever actually determining the mapping. Once the hyperplane has been created the kernel functions can again be used to map new points into the transformed feature space for classification. Kernel functions are is the second hyperparameter used in SVMs.

There is not a given best option for a kernel function and common practice is to estimate a range of potential kernel functions with different hyperparameters in cross-validation to find the best ones. As long as the kernel function is legitimate, a SVM will operate correctly even if the creator does not know exactly what features are being used in the kernel-induced feature space.

The most common kernels are:

1. Linear : $< x, x' >$
2. Polynomial : $(\gamma < x, x' > +r)^d$
3. Radial Basis Function (RBF) : $\exp(-\gamma\|x - x'\|^2)$
4. Sigmoid : $\tanh(\gamma < x, x' > +r)$

However custom kernels are also possible. The more complex kernel functions gives further hyperparameters such as γ which is a complexity parameter.

The pros of SVM is that the training always reaches a global minimum and are good at dealing with high dimensional data even on small data sets. However the methods of SVM are binary and for multi-class classification one must first reduce the problem to a set of multiple binary classification problems. In addition picking the right kernel and parameters can be computationally intensive (Kotsiantis et al., 2007).

2.3 Structured Literature Review

2.3.1 Protocol

This thesis is based on an extensive literature search for research and review papers evaluating machine learning on miRNAs in liver and colorectal cancer. Research was independently conducted on Scopus, ACM, IEEE Xplore, ScienceDirect and CiteSeer databases using the following key words (with both extended names and abbreviations): Machine Learning AND miRNA AND cancer AND classification. In addition, reference lists of identified papers were hand searched to obtained additional articles. The search was concluded on September 2018. Finally, six papers were added on biology background from supervisor. Papers were considered for inclusion only if (IC1) they provided full text in the English language; (IC2) contained information on machine learning with miRNA; and (IC3) used data of miRNAs or gene expression. The papers were then ranked through on several quality criteria where some were dismissed because of a low score. The final resulting papers were considered eligible for the systematic review. Full list of search terms, inclusion criteria, quality criteria and paper cutoff can be found in the protocol in Appendix A.1.

2.3.2 Related Work

In this subsection the related works found through the structured literature review is summarized. The first subsection contains information found in reviews. The remaining research papers are presented in terms of how they solved four machine learning steps: choice of data, pre-processing of data, classification and validation. In each of these sections there is also tables for a better overview of similarities between the papers.

Reviews

Five papers were reviews and one (Bertoli et al., 2016) did experiments on their own. Banwait and Bastola (2015) aims to summarize various existing computational approaches and potential use of bioinformnatics in the field of cancer biology. In the role of miRNAs in human cancer they state that miRNA that are up-regulated in cancer can potentially act as oncogenes through negative regulation of tumor suppressor genes leading to uncontrolled cell proliferation, and down-regulated miRNA can act as tumor suppressors by inhibiting oncogenes or genes involved in cell proliferation and apoptosis preventing tumor development. In therapeutics they state that restoring the expression of miRNA as potential approach. They also summarize the state of the art in studies with aims to identify miRNAs. Lastly the review states how the detailed mechanism behind miRNAs as oncomiRs or tumor suppressors has not yet been achieved and points to the importance of integrating systems biology, cancer research and bioinformatics to gain a more complete and accurate picture of cancer.

Bertoli et al. (2016) reviews the use of miRNAs as biomarkers for diagnosis, prognosis and theranostics in prostate cancer. They point out the problems with circulating prostate-specific antigen (PSA) and Gleason score and propose using miRNAs as biomarkers as a more accurate prognosis. They also extracts 29 miRNAs in a meta-analysis approach with

Table 2.1: Overview of related works articles. Scores indicate to what degree different quality criteria were fulfilled in relevance to this thesis. A full overview is available in Appendix A.1.

Article ID	Author	Score
0	Brown et al. (2000)	3
1	Guyon et al. (2002)	3
2	Furey et al. (2000)	3.5
3	Önskog et al. (2011)	3
4	Liaw et al. (2002)	2
5	Banwait and Bastola (2015)	2
6	Bertoli et al. (2016)	3.5
7	Erson and Petty (2009)	2
8	Li et al. (2010)	2.5
9	Razak et al. (2016)	2.5
10	Batuwita and Palade (2008)	2.5
11	Liao et al. (2018)	3
12	Chakraborty and Maulik (2014)	3
13	Kim and Cho (2010)	4
14	Kothandan and Biswas (2015)	3
15	Ibrahim et al. (2013)	4
16	Piao et al. (2017)	4
17	Saha et al. (2015)	3
18	Saha et al. (2016)	4
19	Tran et al. (2011)	3.5
20	Wang et al. (2017)	2.7
21	Yang et al. (2015)	3
22	Iorio and Croce (2012)	3

diagnostic properties which they suggest to be used as a non-invasive blood test in prostate cancer.

Erson and Petty (2009) reviews the relationship between miRNAs and cancer, miRNA detection techniques, miRNA target identification, miRNA as cancer biomarkers. In cancer, many miRNA genes within region of genomic instability and chromosomal fragile sites are shown to have abnormal DNA copy numbers in cancer cell. A global insight into deregulated miRNA expressions in different tumors and our understanding of individual miRNA functions are being developed, e.g. miR-21 being over-expressed in multiple cancer types. As cancer biomarkers for prognosis, miR-26 levels appear low in patients who had shorter overall survival but a better response to interferon therapy in hepatocellular carcinoma patients. However exiting in most studies a larger number of patients need to be screened before miRNAs may function as reliable cancer biomarkers to be used for detection of cancer in very early stages.

Iorio and Croce (2012) reviews the dysregulation of miRNAs in cancer. This review suggests that the over-regulation of tumor suppressor genes and under-regulation of oncogenes by alterations of miRNA expressions are not exceptional but rather the rule in human cancer. There has also been shown that different types of cancer can be discriminated with high accuracy while mRNA profiles by contrast were highly inaccurate indicators of tissue or cancer type. This suggests that tumors more clearly maintain a unique tissue miRNA expression profile. In addition miRNAs are more stable due to their small size compared to long mRNAs.

Li et al. (2010) reviews three aspects of miRNA: miRNA gene finding, miRNA target prediction and regulation of miRNA genes. Although all of these aspects are interesting regulation of miRNA genes is of the highest relevance for this thesis. This review summarizes that there exists miRNA promoters that are experimentally verified and should help understand the regulation mechanism of miRNAs. These studies are done by looking at interactions between miRNA promoters and their predicted target proteins. Another study included in this review also found a case where one miRNA targets both a transcription factor and the regulating gene of that transcription factor thus having a regulated feedback loop.

Data sets

In the included related works all papers had data sets consisting of expression profiles, either mRNA or miRNA expressions, and all had samples which were either a cancer type or normal. Most papers also used paired samples from humans, i.e. both the tumor and normal sample were from the same patient. Several papers were based on data sets generated from microarrays. Some contained mRNA expressions, while others contained miRNA expressions. Others data sets were generated using RNA-sequencing technology. Some papers included several data sets, and some where focused on a single data set. A select few included both microarray data sets and RNA-sequencing data sets but used these in different experiments. Six papers had data from the cancer genome atlas (TCGA) , and five papers used data from Lu et al. work.

Kim and Cho; Tran et al. used a single microarray data set. Both originally published in Lu et al. (2005). The data set contains several types on cancer. Guyon et al.; Furey et al.; Önskog et al.; Razak et al.; Batuwita and Palade; Chakraborty and Maulik; Ibrahim et al.; Saha et al. used multiple microarray data sets. These were quite different were some focused on a single type of cancer while others had multiple cancer types. Chakraborty and Maulik; Ibrahim et al. also combined datasets with both mRNA gene expressions and miRNA. Liao et al. (2018) focused on multiple types of cancer using a single RNA-sequencing data set. Saha et al. (2015); Wang et al. (2017) focused on breast cancer using a single RNA-sequencing data set. Yang et al. (2015) used six RNA-sequencing data sets of different cancers where they extracted only paired samples. Piao et al. used both RNA-sequencing and microarray data sets. The RNA-sequencing data set were downloaded from TCGA while the microarray data set were from Lu et al.. These were though not combined but used in different experiment to show results can hold in different data sets. Brown et al. uses a single hybridization microarray data set of yeast with mRNA expression profiles. The hybridization experiment represents the ratio of expression levels of a particular gene under two different experimental conditions i.e. a measured condition divided by a reference condition. Kothandan and Biswas built their own data set from a list of genes involved in cancer using several online resources: catalog of somantic mutions (COSMIC), tumor associated gene database (TAG), miRecords and miRTARBASE. This is to find miRNAs involved in cancer pathways and thus not comparable with the other data sets.

Table 2.2: Data set overview. Data set IDs is self defined. Samples indicate how many samples and if available how many normal and tumor samples. Genes is how many genes are in the original data set. Set type is whether the data set are generated using microarrays or RNA-sequencing and also if the samples are mRNAs or miRNAs. Data type has abbreviations for the different diseases the data set are generated from. Author points to the orginal paper the data set was published or to what organization has them online i.e. The Cancer Genome Atlas (TCGA) and The European Bioinformatics Institute (ebi). Data sets from three papers Batuwita and Palade (2008), Kothandan and Biswas (2015) and Ibrahim et al. (2013) were not included in this table as their paper lacks sufficient information about their data sets.

ID	Samples	Genes	Set type	Data type	Author
DS0	79	2467	Array(m)	Yeast	Eisen(1998)
DS1	62(47/25)	2000	Array(m)	Leukemia	Golub(1999)
DS2	72(22/50)	7129	Array(m)	Colon cancer	Alon(1999)
DS3	31(17/14)	97802	Array(m)	Ovarian cancer	Furey(2000)
DS4	133(65/68)	7806	Array(m)	DLBCL	Alizadeh(2000)
DS5	66(34/32)	33491	Array(m)	Epithelial	Finak(2006)
DS6	40(22/18)	40475	Array(m)	iNFPAs	Galland(2010)
DS7	104(58/46)	19718	Array(m)	High ER	Herschkowitz(2007)
DS8	91/72/19)	40233	Array(m)	Cancer	Jones(2004)
DS9	73/55/18)	8033	Array(m)	High ER	Srlie(2001)
DS10	87(65/22)	8911	Array(m)	Metastasis	Ye(2003)
DS11	353(169/184)	315	Array(mi)	Gastric CA	E-TABM-341 / ebi
DS12	84	1569	Array(mi)	Ovarian cancer	E-TABM-343 / ebi
DS13	770(87/683)	1047	Seq(mi)	BRCA	TCGA
DS14	482(46/436)	895	Seq(mi)	LUAD	TCGA
DS15	376(45/331)	839	Seq(mi)	LUSC	TCGA
DS16	299(38/261)	857	Seq(mi)	STAD	TCGA
DS17	566(59/507)	904	Seq(mi)	THCA	TCGA
DS18	404(21/383)	765	Seq(mi)	UCEC	TCGA
DS19	83	2308	Array(m)	SRBCT	Khan(2001)
DS20	77(19/58)	7070	Array(m)	DLBCL	Shipp(2002)
DS21	334	217	Array(mi)	Cancer	Lu(2005)
DS22	215	1047	Seq(mi)	Cancer	TCGA
DS23	162(81/81)	906	Seq(mi)	HNSC	TCGA
DS24	82(41/41)	796	Seq(mi)	KICH	TCGA

Table 2.3: Data set used in overview. Contains the data sets and in what paper they were used. Only the data sets that were used in multiple papers are listed.

Data set ID	Paper by ID
DS1	1,2,12
DS2	1, 2
DS13	11,17, 20, 21
DS14	11, 21
DS16	11, 21
DS17	11, 21
DS21	12,13,16,18,19

Pre-processing of data

Regarding pre-processing of gene data, there are three main ways this can be done. First, altering the values e.g. by scaling the raw data (see section 3.1.1). Second, not use all the data e.g. using a selection either of samples and/or features. Third, adding additional data e.g. getting information from the data in an external analysis. More often than not, all of these methods are used in a way.

For feature scaling Batuwita and Palade (2008); Chakraborty and Maulik (2014); Kim and Cho (2010); Tran et al. (2011) all chose to scale features to a range between either 0 to 1 or -1 to 1. Guyon et al. (2002); Furey et al. (2000) chose to standardize their features such that the mean of each feature is 0 and the standard deviation is 1. All of these papers do though seem to think it advantageous to scale features following guidelines from e.g. Hsu et al. (2003). Brown et al. (2000) also has its raw data through a normalization algorithm such that each expression vector has euclidean length 1.

Önskog et al. looked specifically on synergistic effects between normalization, gene selection and machine learning. In their paper they implement five different normalization strategies and three different gene selection strategies. They conclude that there are significant positive effects from using normalized data on their best methods. In addition a larger number of genes selected imply better performance but that this effect decreases when there are many more genes than observations. In their experiment there were no significant improvement from including more than 200 genes.

Brown et al. (2000); Guyon et al. (2002); Furey et al. (2000); Razak et al. (2016); Batuwita and Palade (2008); Chakraborty and Maulik (2014); Kim and Cho (2010); Kothandan and Biswas (2015); Piao et al. (2017); Saha et al. (2015, 2016); Tran et al. (2011); Yang et al. (2015) all do some form of feature selection but the techniques greatly varies. The number of total features left also varies between as few as 8 up to several hundred, although the consensus on best performance seems to be at the higher end. This is due to that selection of fewer miRNAs does exclude the important interactions miRNA have on each other. Some of the selection techniques can be found in section 3.2.2.

Four papers did compare different feature selection methods in terms of performance. Chakraborty and Maulik (2014) used a kernelized fuzzy rough set (KFRS) for feature selection. In addition to study the performance of the proposed method they have also used fuzzy preference based rough set (FPRS) and consistency based feature selection (CBFS). In this experiment KFRS had the best performance. Kim and Cho (2010) tried four similarity-based methods: inverse of Euclidean distance measure, Pearson correlation, Cosine coefficient and Spearman correlation. In addition information gain, mutual information and signal-to-noise ratio were used. These were compared and results shown that cosine coefficient proved to be best for feature selection in their experiment. Saha et al. (2015) compared Gravitational Search Algorithm (GSA), signal-to-noise ratio, Welch's t-test, Wilcoxon ranksum test, Joint Mutual Information (JMI), minimum Redudancy Maximum Relevance (mRMR) and Mutual Information Feature Selection (MIFS) in conjugation with a SVM as well as with the SVM itself alone. In this experiment Gravitational Search Algorithm outperformed the other six feature selection methods. Yang et al. (2015) removed all samples where the sum of expression levels for that sample were less than 10 in raw sequencing data expression. For selection of features seven feature selection algorithms were tested but failed to compare the algorithms in real data due to lack of a gold standard.

Three papers did not mention any feature scaling or feature selection. Liao et al. (2018) uses IsomiR expressions and no scaling or selection is explained. However using IsomiR expressions this is not directly comparable to the other papers. Wang et al. (2017); Ibrahim et al. (2013) does not specify that they use any specific scaling or feature selection though the last mentioned has feature selection methods in their related works.

Table 2.4: Normalization overview. This table shows what type of normalization were done by which papers

Normalization type	Papers by ID
Standardization	0, 1, 2
MinMax	10, 12, 13, 19
Tested Several	3
None / Not mentioned	4, 5, 6, 11, 14, 15, 16, 17, 18, 20, 21

Table 2.5: Feature selection overview. This table shows which papers utilized feature selection and if they tested multiple techniques.

Feature Selection technique	Papers by ID
Single feature selection technique	0, 1, 2, 9, 10, 12, 13, 14, 16, 17, 18, 19, 21
Multiple techniques	3, 12, 13, 17, 21
None	11, 20, 15

Classification

MiRNA gene expressions being high dimensional in features and low in samples somewhat limits what type of classifiers that can effectively give accurate predictions. Suitable methods are those who use some form of regularization and the primarily methods are SVMs, KNNs and ensemble methods e.g. random forest. This is also reflected in the selected related literature. In addition to what kind of classifiers are used, it is interesting to see what parameters are chosen or optimized.

Razak et al. (2016) focused solely on a random forest classifier. This classifier is not sensitive to outliers or noise (Liaw et al., 2002). Gini index (section 3.1.4) is chosen as splitting criteria . The number of estimators or trees in the forest is however not revealed in the paper.

Five papers focused on SVMs. Guyon et al. (2002) only focuses on a linear kernel and leaves the non-linear kernels as future work. In this paper the diagonal factor C is set to 100 because the problem is insensitive to the value of C as the training data set are linearly separable down to just a few features. They also concludes that the number of features selected matter more than the classifier used when compared to Golub et al. (1999) classifier and Fisher's linear discriminant. Furey et al. (2000) focus on a polynomial kernel with default parameters except for the C which is tuned in a hold-one-out cross-validation. Saha et al. (2015); Kothandan and Biswas (2015); Batuwita and Palade (2008) used a single SVM with the RBF kernel. Saha et al. preset parameters γ and C to 0.5 and 2.0 respectively. Kothandan and Biswas optimized the parameters through a exhaustive grid search. Batuwita and Palade used a more complex method to optimize the parameters. Initially they find the optimal C with a linear kernel called $\widetilde{C}$, using this $\widetilde{C}$ the remaining γ parameter can be found in a linear search and corresponding C using (2.2). The derivation of this relationship can be found in (Keerthi and Lin, 2003).

$$log_2C = log_2\widetilde{C} - (1 + log_2\gamma) \tag{2.2}$$

Brown et al. (2000) solves a multi-class classification problem (dealing with different cancer illnesses) by using several classifiers. In this paper SVM with a higher dimensional kernel outperforms Parzen windows, Fisher's linear discriminant, two decision tree classifiers, and SVMs that use the simple dot product kernel. Önskog et al. (2011) tested decision trees with both gini index and information gain, neural networks with one hidden layer and no hidden layer, and SVM with linear, polynomial and RBF kernels. SVM with RBF kernel had the highest accuracy. Kim and Cho (2010) used KNNs with Euclidean distance, Pearson correlation, cosine coefficients and Spearman correlation, Multi-layer Perceptron and SVM with a linear kernel. Lowest error rate came from KNN with Euclidean distance. Tran et al. (2011) used a SVM with linear, polynomial with degree 3 and RBF kernel. RBF had best performance in terms of F1-score and AUC. This SVM also outperformed other classifiers which included decision trees, bayesian networks and backpropagation neural networks. Yang et al. (2015) used the classification algorithms of logistic regression, random forest and SVM with RBF kernel. They conclude that logistic regression is unsuitable for the high dimension and small sample data, and that random forest performed better than SVM.

Liao et al. (2018) used both a random forest and libD3C, an ensemble classifier, were the latter gave the best results. Piao et al. (2017) used C4.5 decision tree and SVM as base classifiers in their own ensemble with multiple independent feature subsets then uses averaging to produce a classification. This classifier is compared with random forest, bagging and boosting using the same base classifiers and finds their ensemble to outperform the other ensembles. Wang et al. (2017) uses random forest, eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine(LightGBM). LightGBM outperformed the other classifiers in several aspects.

Chakraborty and Maulik (2014) proposes using a Transductive SVM (TSVM). This is a semisupervised SVM that utilizes unlabeled data. Traditional supervised learning or inductive learning is more general, and presumable harder, than transductive learning.

The TSVM outperformed inductive SVM, Naive bayes and a KNN. The TSVM had a RBF kernel and used a grid search with cross validation to find the optimal parameters. Ibrahim et al. (2013) used random forest with 10 decision trees and SVMs as base classifiers. Then tries to improve these using two semisupervised machine learning approaches called self-learning and co-training. Saha et al. (2016) used a two step approach. The first step uses a multiobjective optimization technique in combination with multiple classifiers automatic determines classifier type. The second step has two different approaches. First a frequency based approach and the second approach is an simple ensemble approach. Both of which is developed to combine the outputs of the solutions obtained from the first stage. Classifiers in the first stage included random tree, random forest, Sequential Minimal Optimization (SMO) and Logistic regression.

Table 2.6: Classification technique overview. Overview over which papers used particular algorithms. Algorithms not specified were either labeled as an ensemble technique or a simple learner (a non ensemble technique).

Classification algorithm	Papers by ID
Random Forest	9, 11, 16, 20
SVM	1, 2, 3, 10, 12, 14, 17
Simple learners	0, 3, 13, 19, 21
Other ensemble techniques	11, 16, 20

Table 2.7: Parameter selection overview. This table gives a overview over which papers optimized their parameters through cross-validation and which used presets.

Parameter Optimization	Papers by ID
Cross-validation	2(onlyC), 3, 10, 12, 14, 19
Preset or multiple presets	0, 1, 2, 9, 13, 15, 17, 18, 21
None / Not stated	9, 11, 16, 20

Metrics and Validation

There are several metrics that can be used to measure performance in a classification of miRNA expression levels. These depend on whether it is a binary or a multiclass classification, and on what type of balance classes there is in the data set. The most popular performance metrics, including all used in this thesis, can be found in section 3.1.2.

Furey et al. (2000) only used the metric of False Positives(FP), False Negatives (FN), True Positives (TP) and True Negatives (TN).

Önskog et al. (2011) used error rate, i.e. the percentage of misclassified observations in a test set, as a measure of performance. Error rate was adjusted by dividing by the theoretical error rate obtained by randomly assigning classes given the distribution of the two classes. Chakraborty and Maulik (2014); Kim and Cho (2010); Wang et al. (2017); Liao et al. (2018); Saha et al. (2015) used overall accuracy (ACC) as metric. The two latter also calculated Matthew's Correlation Coefficient (MCC) to determine the trade-off of sensitivity and specificity. In addition Saha et al. (2015) also calculated F-measure and AUC, while Wang et al. (2017) used logistic loss.

Kothandan and Biswas (2015) chose to not use accuracy as class imbalance existed in the dataset. Hence, performance measures were chosen in compliance with the cross-validation rate and MCC.

Razak et al. (2016); Piao et al. (2017); Tran et al. (2011); Yang et al. (2015) used AUC obtained from a ROC curve as metrics. Piao et al. also calculated accuracy, sensitivity and specificity. Tran et al. also used precision, recall and F-measure. Yang et al. also calculated positive predictive value (PPV) and negative predictive value (NPV).

Ibrahim et al. (2013); Saha et al. (2016) used precision, recall and F-measure.

Three papers used slightly different metrics than the others. Brown et al. (2000) defines cost savings of using the learner procedure M as $S(M) = C(N) - C(M)$ where $C(N) = fp(N) + 2 \times fn(N)$ false negatives is higher weighted because the number of positive examples are low. Samples are tested against the null learning procedure which classifies all data as negative. Guyon et al. (2002) used error, reject, extremal margin and median margin. Error rate is the fraction of samples that are misclassified with its compliment the success rate. The rejection rate is samples that are rejected (low confidence) complemented by acceptance rate. Batuwita and Palade (2008) used $G{-}mean = \sqrt{SE * SP}$, where SE is the proportion of positive samples correctly classified and SE is the proportion of negative samples correctly classified.

Table 2.8: Overview over used performance metrics by papers.

Metric	Papers by ID
ROC (AUC)	9, 16, 17, 19, 21
F-score	15, 17, 18, 19
ACC/ Error rate	3, 11, 12, 13, 17, 20
MCC	11, 14, 17
Other	0, 1, 2, 10, 20

Table 2.9: Overview over how results were validated.

Validation technique	Papers by ID
K-fold	0, 3, 12, 13, 14, 16, 20, 21
LOOCV	1, 2, 9, 16, 18, 19
Stratified K-fold	10, 17
None / Not stated	11, 15

Depending on what specific problem an experiment tries to solve and how much data is available, several different ways to validate the results are also used. Gene data is prone to low amount of samples which again makes low confidence when going from e.g. 50-100 samples to an infinite solution. The most common techniques are using a form of Cross-Validation (CV) and, depending on the experiment and goals, do a statistical test to show confidence in the results. The CV techniques are explained in section 3.3.

Furey et al. (2000); Razak et al. (2016); Tran et al. (2011); Saha et al. (2016); Guyon et al. (2002) applied Leave-one-out Cross-Validation (LOOCV) to asses feasibility and validity. The results was then averaged to produce an estimate of the accuracy of the system. Guyon et al. also computed metrics for each value in a separate test set.

Kothandan and Biswas (2015); Chakraborty and Maulik (2014); Yang et al. (2015); Brown et al. (2000) all used a K-fold cross validation with respectively 10, 5, 5 and 3 folds. Chakraborty and Maulik only used this to optimize SVM parameters. Yang et al. replicated this K-fold 100 times to average out results. Brown et al. also repeated this procedure for best classifiers to show relatively low standard deviation in results.

Saha et al. (2015); Batuwita and Palade (2008) applied a Stratified K-fold CV. With 10 and 5 folds respectively.

Two papers combined cross-validation folds to validate their experiments. Önskog et al. (2011) used an inner 10-fold CV for optimization, and an outer 5-fold CV to estimate final classification performance. Piao et al. (2017) applied LOOCV and 10-fold CV to gain two separate results. In addition, the experiment was repeated 50 times where results of each method were recorded and finally averaged.

Two papers also applied additional statistical tests to show significance in their results. Chakraborty and Maulik used a t-test and Wilcoxon signed rank test at 5% significance level. Saha et al. (2015) used the non-parametric test Friedman test with 5% significance level. This showed statistical significance of their results produced by their proposed method with respect to the results of other methods.

2.3.3 Reported performance and confidence

As the previous section and tables 2.1-2.9 has shown there are several factors which makes it hard to compare results and draw conclusions about how this type of work should be done. There is some key points most papers explicitly agrees on. First scaling raw gene expressions do increase the performance of a classifier. Secondly several papers put fourth the notion that feature selection is necessary to achieve somewhat good results. Third a classifier that handles high dimensional data should be chosen. Fourth cross validation should be used both for optimizing parameters and to establish a correct performance estimate.

For DS1 three papers did classify this data set. Guyon et al. (2002); Furey et al. (2000) both achieved perfect classification using SVMs and Chakraborty and Maulik (2014) achieved 98.89 % accuracy. For DS2 two papers used this data set. Furey et al. (2000) had 6 missclassifications of 72 samples and Guyon et al. (2002) held a 98% accuracy.

The next three data sets were from TCGA and all authors used different number of genes and subsets of samples of this data. Some split the data into paired samples while others combined data from patients from other sets to balance the imbalanced set of samples TCGA provides. Four papers used DS13. Liao et al. (2018); Saha et al. (2015) achieved approximately 95% accuracy using random forest and SVM respectively. Wang et al. (2017); Yang et al. (2015) achieved perfect classification using LightGBM and random forest respectively. Two papers used DS14. Liao et al. (2018) using IsomiR expressions and random forest achieved approximately 92% accuracy while Yang et al. (2015) achieved 65% AUC score using a SVM. Two papers used DS16. Liao et al. (2018) using

IsomiR expressions and random forest achieved approximately 94% accuracy while Yang et al. (2015) achieved 71% AUC score using a SVM and logistic regression.

DS21 were used by five papers. This is a data set that holds multiple types of cancer and most papers also used different subsets of the original data set. Kim and Cho (2010) had a 95% accuracy using a KNN and feature selection. Piao et al. (2017); Saha et al. (2016) had approximately 98% accuracy using self defined ensembles. Chakraborty and Maulik (2014); Tran et al. (2011) also had approximately 98% accuracy using SVMs.

2.4 Motivation

There are two main reasons for motivation behind classifying miRNA. The first is in the context of miRNA and its potential applications. The second is the technical challenge of classifying such data, especially when using multiple data sets at once with different errors and biases in their technology.

During the recent decade the number of diseases that are linked to misregulation of miRNA has dramatically increased. For cancer approximately 50% of miRNA genes are localized in genomic regions that are associated cancer. MiRNA expression profiling has been shown to be associated with tumor development, progression and response to therapy. This suggests that there are potential clinical use of miRNA in diagnostic, prognostic and possibly as a therapeutic tool. Several studies has already shown potential use of miRNA for diagnosis and prognosis. There are also potential use of miRNA as oncogenes and oncosupressor genes that can improve disease response and cure rates. MiRNA-based anticancer therapies have also recently been exploited, either alone or in combination with current targeted therapies.

In terms of the classification challenge there are several interesting aspects. Each sample has some individual differences associated with age, sex, ethnicity etc. However the larger difference is usually between data sets. This is differences connected to what lab has made the data sets, how each sample has been preserved and what technologies are used to deduce and quantify the transcriptome. Most samples also come in pairs such that one sample has been harvested from the disease site and one sample has been harvested from neighbouring healthy tissue. For different types of cancer both of these samples may come from different sites e.g. for colorectal cancer the colon is quite a large organ and tumors may form in different parts of the colon. The tumors may also be at different stages questioning how tumor is the tumor sample and normal is the normal sample. Even through all of these individual differences, some of which can be rectified by normalizing the data, the individual data sets can often be linearly separable and quite easy to classify.

The data sets are also quite large in terms of features (miRNAs) and small in samples which increases the importance of trying to combine data sets. These dimensions also suggest that the majority of features are not necessarily useful for classification. In addition the features can be validated by looking at the known relations between miRNA and cancer.

Chapter 3

Method

This chapter contains all necessary information and techniques to reproduce experiments in chapter 4. The implementations of these were used through scikit-learns API (Pedregosa et al., 2011) and are easily found in their documentation. First different techniques for scaling data and performance metrics are explained. Secondly six feature selection techniques are presented. Thirdly the different techniques used for cross-validation are explained.

3.1 Feature Scaling and Performance Metrics

3.1.1 Normalization Techniques

Min-max normalization:

$$z = \frac{x_i - \min(x)}{[\max(x) - \min(x)]}$$

This scaling technique can also be slightly adjusted to scale data into a range. However it has some drawbacks in that it is sensitive to outliers.

Z-score normalization:

$$z = \frac{x_i - \mu}{\sigma}$$

Standardization is better than min-max normalization to outliers but do not guaranty a specific range. In general this scaling works better the more or less the features look like standard normally distributed data.

Robust scaling:

$$z = \frac{x_i - \widetilde{x}}{q_3 - q_1}$$

Removes the median and scales the data according to the quantile range (usually to 1st and 3rd quantile). This method is more robust to outliers than normal standardization.

3.1.2 Performance Metrics

When considering a boolean classification problem a given sample can be classied to one of four cases: 1. a correctly positive case True Positive (TP), 2. a false positive case False Positive (FP), 3. a correctly negative case True Negative (TN) or, 4. a false negative case False Negative (FN). These are the basic building blocks for most performance metrics in a binary classification problem.

Sensitivity / Recall

$$sn = \frac{TP}{TP + FN} \tag{3.1}$$

Specificity

$$sp = \frac{TN}{FP + TN} \tag{3.2}$$

Precision / Positive predictive value (PPV)

$$precision = \frac{TP}{TP + FP} \tag{3.3}$$

Negative Predicitve Value (NPV)

$$NPV = \frac{TN}{TN + FN} \tag{3.4}$$

Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.5}$$

Balanced accuracy

$$BACC = \frac{sn + sp}{2} \tag{3.6}$$

Matthew’s correlation coefficient

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}} \tag{3.7}$$

F1 Score

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.8}$$

G-mean

$$G - mean = \sqrt{SE * SP} \tag{3.9}$$

where SE is the proportion of positive samples correctly classified and SE is the proportion of negative samples correctly classified.

3.1.3 Receiver Operating Characteristic

A Receiver Operating Characteristic (ROC) curve is a graphical plot that shows the diagnostic ability of a binary classifier. The curve can also be used as a metric through its Area Under Curve (AUC) score. The ROC Curve is created by plotting the True Positive Rate (Sensitivity) in function of the False Positive Rate (100-Specificity) for different cutoff points. This implies that a perfect ROC AUC score is a curve that curves all the way to the top left corner of the plot (Schoonjans, 2018).

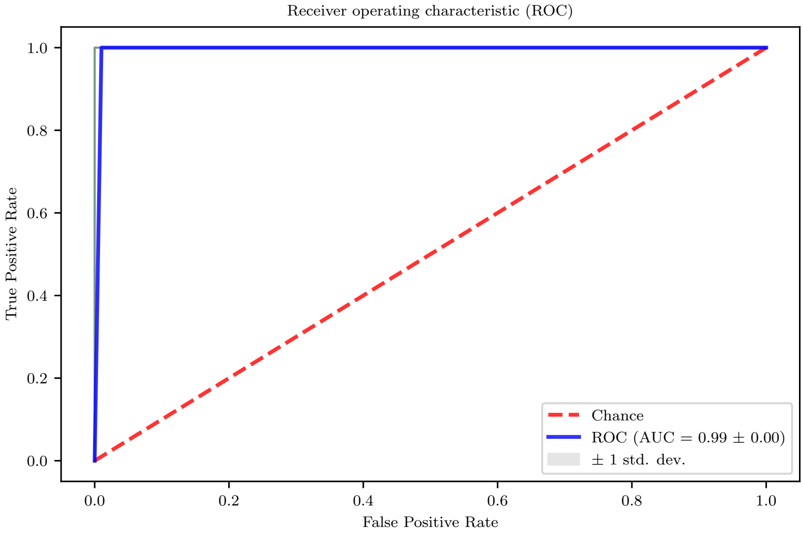

Figure 3.1: A ROC curve example. The vertical axis is the Sensitivity and the horizontal axis is the inverse Specificity. The further the ROC curve (blue line) is to the top left corner the better ROC AUC score.

3.1.4 Random Forest Splitting Criteria

Random forests decision trees has two criteria in which it marks a feature as a good splitting condition. In general these are just calculations too minimize the amount of decision nodes by splitting on a feature that correctly separates the data.

Entropy

Entropy for information gain contains two steps. First the entropy of the target feature is calculated.

$$E(S) = \sum_{j=1}^{J} -p_j \log_2 p_j \tag{3.10}$$

where j is each class and p is the probability of that class. Second the entropy of the child nodes are calculated, then added proportionally, to get the total entropy for the split.

$$E(S, X) = \sum_{c \in X} P(c)E(c) \tag{3.11}$$

where c is the classes of X and $P(c)$ is the probability of that class. This entropy is subtracted from the step one entropy. The result is the Information Gain, or decrease in entropy.

$$IG(S, X) = E(S) - E(S, X) \tag{3.12}$$

The higher the information gain the better it is as a splitting condition.

Gini Importance

Nembrini et al. (2018) explains the revival of gini importance to the fact that it is more unbiased to number of categories and almost as fast as standard impurity importance. In addition they give this definition of Gini impurity:

$$\hat{\Gamma}(t) = \sum_{j=1}^{J} \hat{\phi}_j(t)(1 - \hat{\phi}_j(t)), \tag{3.13}$$

where $\hat{\phi}_j(t)$ is the class frequency for class j in the node t. The decrease of impurity is the difference between a nodes impurity and the weighted sum of the impurity measures of the two child nodes (the Gini index).

3.2 Principal Component Analysis and Feature Selection Methods

3.2.1 Principal Component Analysis

A Principal Component Analysis (PCA) is a statistical procedure that transform features into a set of principal components. In mathematics this procedure is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance of the data is in the first principal component. This is often used as a technique to view high dimensional data and get insight into how linearly separable it is.

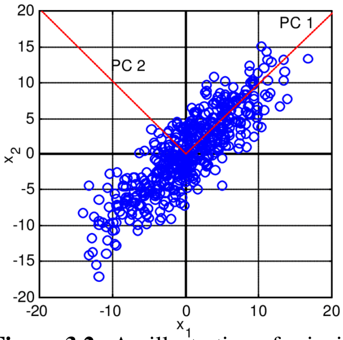

Figure 3.2: An illustration of principal components in two dimensional data.

3.2.2 Feature Selection

Score Based

Selects the features that on average differs least between classes. Then removes the selected feature until a desired subset with features that on average differs most between classes.

Recursive Feature Elimination

Recursive Feature Elimination (RFE) uses an external supervised learning classifier that can provide information about how important a feature is, then recursively prunes the least important feature until a predefined number of features are left.

Symmetrical Uncertainty

Symmetrical Uncertainty (SU) is calculated as follows (Singh et al., 2014):

$$SU(X,Y) = \frac{2 \times MI(X,Y}{E(X) + E(Y)} \tag{3.14}$$

where E is the Entropy and MI is the Mutual Information,

$$MI(X,Y) = E(X) - E(X,Y) = E(X,Y) - E(X|Y) - E(Y|X) \tag{3.15}$$

χ^2-Test

χ^2-test measures dependence between stochastic variables which requires non-negative features. Thus will this test find features that are most likely to be independent of classes and therefore irrelevant for classification. The χ^2 is calculated as follows:

$$\chi^2 = \sum \frac{(f_0 - f_e)^2}{f_e} \tag{3.16}$$

where f_0 is the feature count and f_e is the expected count.

Signal-to-Noise Ratio

Signal-to-Noise Ratio (SNR) is defined as follows:

$$SNR = \frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2} \tag{3.17}$$

where μ is the mean values and σ is the standard deviation. In the case of feature selection the denoted 1 and 2 represents the classes and each feature thus has a SNR score.

Fisher Score

Fisher score selects each feature independently according to their scores under the Fisher criterion.

$$FS = \frac{\sigma^2_{between}}{\sigma^2_{within}} \tag{3.18}$$

i.e. this is the ratio of variance between the two classes divided by variance within the class.

3.3 Cross-Validation

Cross-validation is model validation techniques to asses how the results of a model performs in unseen data. In general, a model is first trained on one part of the data and tested on another. This helps asses how well the trained model generalizes to a independent data set. However simply doing a train-test split is often not random enough to give an accurate statistics on how well the model performs in unseen data. Thus slightly more advanced validation techniques are required. There are three main reasons for doing cross-validation. First cross-validation is essential to help us evaluate the quality of a model. Secondly it helps choosing the model which performs best on unseen data. Third it help avoiding overfitting and underfitting of the data. Overfitting is when a model is over trained to the training data such that it captures unnecessary noise and is fitted to patterns that does not generalize to the training set. Underfitting is when a model is under trained and does not capture essential patterns to predict in unseen data.

3.3.1 Leave-one-out Cross-Validation

Leave-one-out cross-validation is done by leaving a single sample out as the test set and the remaining data is used as the training set. Then this process is repeated for each sample until all samples has been a test sample. The predictions for each iteration can then be averaged to estimate how the model would perform on unseen data. The drawbacks of this technique is that it is computationally heavy and should only be used if there is small amounts of samples and the model is fast to retrain in each iteration.

3.3.2 K-fold Cross-Validation

K-fold cross-validation is essentially the generalized version of leave-one-out cross-validation. Here the data are split into K-folds. Then one fold is selected as the validation set and the remaining fold are selected as a training set. This process is repeated for each fold such that all folds has been run once as the validation set. The accuracy for each fold can then be collected and averaged to estimate actual accuracy in unseen data.

3.3.3 Stratified K-fold Cross-Validation

The stratification of a K-fold implies that the number of classes within each fold is similar. For binary classification this means that the number of positive and negative cases in each fold is similar. This is highly useful for smaller data sets, imbalanced data sets and in the cases of multiclass classification.

Chapter 4

Experiments and Results

In this chapter experiments and the results of these are presented. The first experiment is a overview of the data sets and how to combine these. The second experiment looks into how scaling and feature selection should be done for combined data sets. The third experiment looks into algorithm performances on the combined and individual data sets. The fourth experiment relates to the problem of imbalanced data sets. The fifth and last experiment looks into which features were of the highest importance for the different methods used in previous experiments.

4.1 Data sets

The classification experiments will use different data sets from colorectal- and liver cancer. These data sets have few samples but gather several hundred to thousands of microRNAs. The samples are either labeled as 'normal' or 'tumor'. For colorectal cancer samples different types of tissue are used e.g. rectal, ascending and sigmoid. These were initially split into separate groups but PCA plots show these are quite comparable.

$$n_i = \log_2(\frac{(c_i + 0.5)}{\sum_j c_j} \times 10^6) \tag{4.1}$$

The samples are generated using different technologies. One data set is made using microarray technology and the rest is generated using RNA-sequencing technology. These different technologies are not inherently comparable, therefore equation (4.1) is used to normalize gene sequencing data to comparable values to microarray data. Equation (4.1) accounts for differences in library sizes and the relative value RPM (Reads per million) is then log-transformed to stabilize the variance as variance increases with mean. Here n_i is a normalized sequencing miRNA and c_i is a unnormalized miRNA. Log normalized values are preferred as sequencing values are absolutes which leaves us to wonder if the sample was twice as large or if it had twice us much miRNA. Furthermore as sequencing technology picks up a lot more miRNAs, only miRNAs with at least a mean of 1.0 in normalized n_i values is kept. A overview over each data set can be found in table 4.1 where miRNAs are already filtered.

Table 4.1: Table over data sets. In each data set samples are the number of rows and number of miRNAs are number of columns. Technology refers to what technology were used to generate the data set. Type refers to what type of disease the data set has. HCC - Heptatocellular carcinoma and CRC - Colorectal cancer.

Name	**ID**	**Samples**	**MiRNAs**	**Technology**	**Type**
Hepmark-Microarray	D_1	146	396	Microarray	HCC
Hepmark-Tissue	D_2	150	472	RNA-seq	HCC
Hepmark-Paired-Tissue	D_3	37	381	RNA-seq	HCC
ColonCancer_GCF-2014-295	D_4	92	424	RNA-seq	CRC
GuihuaSun-PMID_26646696	D_5	66	425	RNA-seq	CRC
PublicCRC_GSE46622	D_6	15	441	RNA-seq	CRC
PublicCRC_PMID_23824282	D_7	57	485	RNA-seq	CRC
PublicCRC_PMID_26436952	D_8	51	433	RNA-seq	CRC

Density plots gives us an idea of how well the equation works to make the different technologies comparable. Figure 4.1 gives us such a plot for Hepmark-Microarray, Hepmark-Tissue and Hepmark-Paired-Tissue. In general, the ideal plot is overlapping lines equally stretched in width and with equal peaks. Although this is not exactly the case as is easily recognizable in figure 4.2, they still do pair up quite well. The separation of samples in -1 is due to the microarray set having fewer transcribed miRNAs and thus when combined with the other sets have a bit more ”missing” miRNA. The missing miRNA is filled in as -1 because missing certain miRNAs can itself be a biomarker for tumor. It was discovered in later experiments that this filling for missing values worsened the overall performance in classification for the colorectal data sets.

In figure 4.2 there is also one sample in light blue from the Hepmark-Tissue data set that deviates a lot from the other samples in that data sets. The density plots helped identify such samples. These are samples that had been contaminated during the process of making the data sets. These were simply removed from the data sets when found and the source code appendix A.2 includes which samples were excluded.

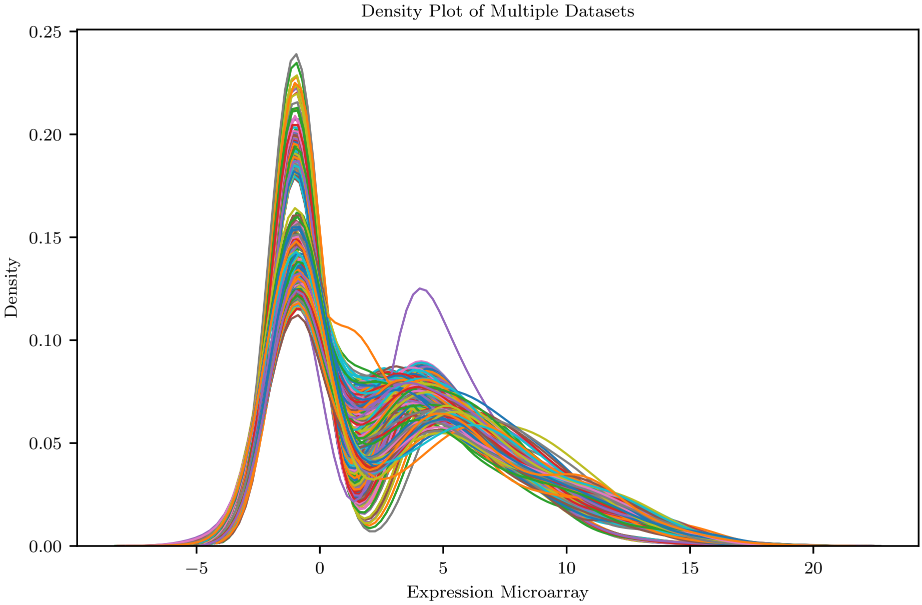

Figure 4.1: A density plot of hepmark data sets. Each line represents a sample and its values, the higher the line is for some value the more common the value is in the sample.

Another observation to be made is although both plots in figure 4.2 have the same peak around 4 the microarray data has few values around 1 which indicates that the microarray data is more sensitive to be present or absent. Which also can be expected as the technology looks for specific miRNA while sequencing technology will register all miRNA in the library.

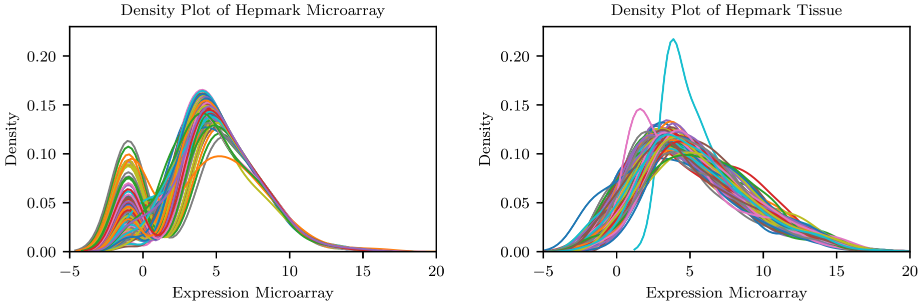

Figure 4.2: A density plot of hepmark data sets microarray and tissue. Each line represents a sample and its distribution of values, the higher the line is at a value the more common the value is in the sample.

4.2 Scaling and Feature Selection of Data Sets

As mentioned in section 3.1.1, there exists several ways to scale data points. Often such normalization of the data points are necessary to remove inherent bias in the data. All of the related work articles concerns themselves with scaling a single data set. However when several data sets should be combined and the classifier should work for even a single sample there is not clear cut for what is the best way to scale such data.

4.2.1 Problem Description

Generally for miRNA transcription the highest differences is often between data sets and not the differences between tumor and normal samples. Subsequently there is no one scaler to handle all transcription data as the errors from both lab and samples differs. Thus each data set should be scaled individually as can be seen in figure 4.3, however this will only be applicable for full data sets. Therefore it is important to see what scaling gives best accuracy for different combinations of samples both in terms of set size and imbalance.

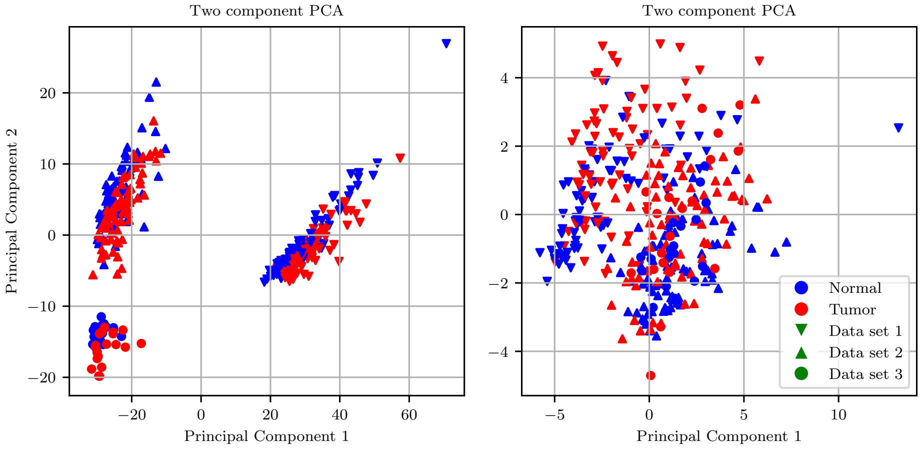

Figure 4.3: Principal component analysis for hepmark data sets. Unscaled to the left and one standard scaler per data set to the right. Here all three hepmark data sets were used.

4.2.2 Scaler Performances

In this thesis the normalization techniques tested are the same as mentioned in section 3.1.1. The scalers were tested on full data sets scaling each set individually. Then they were ran in a stratified 10-fold cross validation using AUC as performance metric. A SVM with RBF kernel and parameters optimized in a 5-fold cross validation and Random forest with 200 estimators were used as test classifiers. All permutations of sets with the same disease were tested to get a better picture.

The results, table 4.2, shows that a min-max normalization adjusted to range from -1 to 1 performed slightly better than both standardization and robust scaling for the hepmark data sets and colorectal data sets. The same tendency were also seen in the other permutations of these two groups. An interesting finding here was that unscaled colorectal data sets performed better than the normalized ones. To further inspect this a similar experiment were done replacing the stratified 10-fold with a leave one data set out setup leading to results summarized in table 4.3. These results indicate that the colorectal cancer data sets were a lot more similar than the hepatic cancer data sets and that scaling are not necessarily advantageous.

Table 4.2: Normalization results for different scaling on combined data sets.

	Random Forest			
Data sets	MinMax	Standard	Robust	Unscaled
D_1 D_2 D_3	$\mathbf{0.94 \pm 0.04}$	0.93 ± 0.05	0.93 ± 0.05	0.93 ± 0.05
D_4 D_5 D_6 D_7 D_8	0.89 ± 0.15	0.79 ± 0.18	0.78 ± 0.18	$\mathbf{0.91 \pm 0.15}$
	SVM			
Data sets	MinMax	Standard	Robust	Unscaled
D_1 D_2 D_3	$\mathbf{0.95 \pm 0.04}$	$\mathbf{0.95 \pm 0.04}$	$\mathbf{0.95 \pm 0.04}$	0.94 ± 0.05
D_4 D_5 D_6 D_7 D_8	0.86 ± 0.13	0.74 ± 0.24	0.73 ± 0.23	$\mathbf{0.91 \pm 0.11}$

Table 4.3: Normalization results for leave one data set out.

	Random Forest		SVM	
Data sets	MinMax	Unscaled	MinMax	Unscaled
D_1 D_2 D_3	$\mathbf{0.94 \pm 0.03}$	0.83 ± 0.16	$\mathbf{0.95 \pm 0.03}$	0.90 ± 0.05
D_4 D_5 D_6 D_7 D_8	0.79 ± 0.22	$\mathbf{0.83 \pm 0.23}$	0.73 ± 0.18	$\mathbf{0.75 \pm 0.30}$

4.2.3 Feature Selection Performances

For feature selection the same experimental setup was used but here fixing it to a min-max normalization from -1 to 1. The feature selection experiment was also done using the filling for missing miRNA which left a larger number of features. This lowered the overall performance for the baseline but having more features for the feature selection might give the feature selection methods an additional edge. To facilitate feature selection RFE with cross-validation was used with an estimator of the same type of classifier. For random forest a random forest with equal amount of estimators were used. For SVM a linear kernel were used for feature selection because the RFE algorithms requires information about feature importances which is not available in a RBF kernel.

Figure 4.4 gives an overview over this process. First normalization of all data sets is done. Seconding data is split in a stratified 10-fold cross validation. Then ran through three steps for feature selection: 1. finding the best linear kernel parameters, 2. run RFE with this optimal linear kernel and 3. reducing the feature space of training and test data. Thirdly classification were done in two steps: 1. grid search for optimal parameters for the RBF kernel (This also trains the classifier). 2. run predictions on the test data.

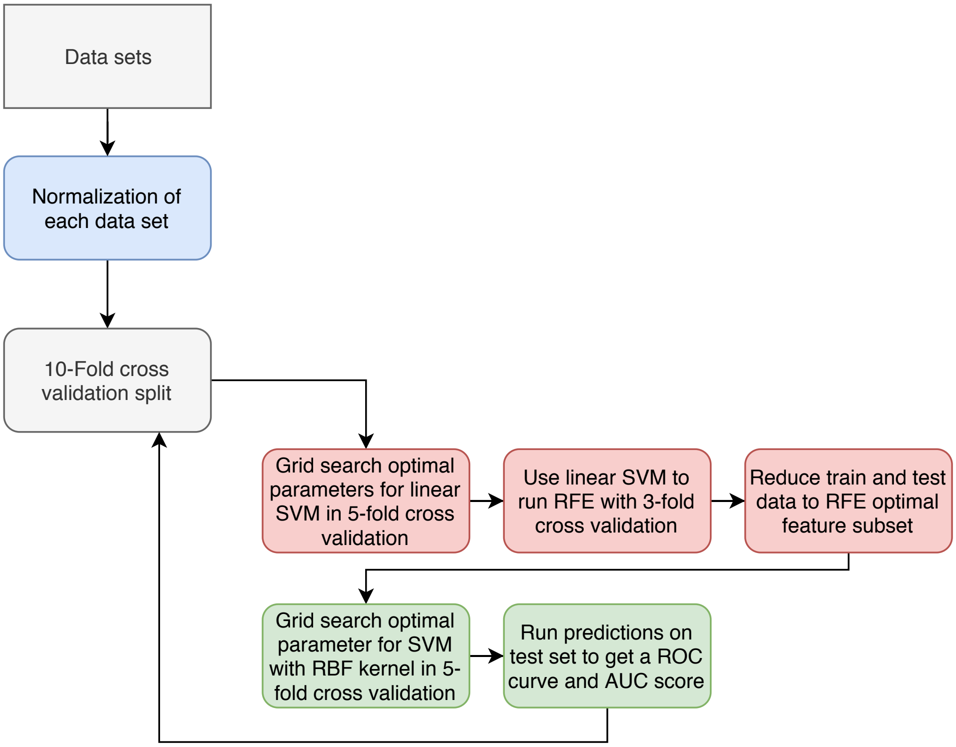
Data sets
Normalization of each data set
10-Fold cross validation split
Grid search optimal parameters for linear SVM in 5-fold cross validation
Use linear SVM to run RFE with 3-fold cross validation
Reduce train and test data to RFE optimal feature subset
Grid search optimal parameter for SVM with RBF kernel in 5-fold cross validation
Run predictions on test set to get a ROC curve and AUC score

Figure 4.4: Overview over feature selection done with SVM. Blue boxes are normalization, red boxes are feature selection and green boxes are classification. Note that linear SVM were used for feature selection while RBF kernel were used in classification.

To adapt this model to random forest none of the grid searches were used and the linear SVM kernel were swapped with a random forest classifier to be used as estimator for the RFE algorithm.

In table 4.4 the number of features selected in RFE for each iteration of the outer stratified 10-fold cross validation is presented along with the score of each fold. The high deviation between folds in the colon data set is a topic to be returned to in section 4.2.4. From this table there are no clear correlation between loss of performance in selecting a fewer features but an indication that there exist data in the training fold that are more similar and more easily separable.

In table 4.5 the overall performance of doing feature selection versus no feature selection is presented. From this data there seems to be a loss of performance from doing feature selection on SVM and random forest does not necessarily benefit either.

For random forest the impact of feature selection is quite clear. As the number of features decreases the amount of randomness in sub-feature space each estimator has is decreased. This results in a model that, depending on the cross validation split, is more overfitted to the training data and thus less robust to the test data. In cases were the split

Table 4.4: Number of features selected in RFE for both SVM and random forest with the respective AUC score for each fold. Hepmark data sets refer to D_1 D_2 D_3 and Colon data sets refer to D_4 D_5 D_6 D_7 D_8.

	Random Forest			
	Hepmark		Colon	
Iteration	Features	AUC score	Features	AUC score
0	701		578	
1	41	0.90	338	1.00
2	66	0.87	83	1.00
3	41	0.93	78	0.71
4	161	1.00	448	0.63
5	146	0.98	568	0.57
6	66	0.89	243	0.98
7	146	0.93	413	0.99
8	541	0.89	318	0.94
9	111	0.97	508	1.00
10	286	1.00	463	1.00
	SVM			
	Hepmark		Colon	
Iteration	Features	AUC score	Features	AUC score
0	701		578	
1	11	0.95	118	1.00
2	91	0.90	408	1.00
3	21	0.94	423	0.31
4	71	1.00	318	0.60
5	216	0.98	528	0.67
6	131	0.94	228	0.99
7	346	0.89	323	0.97
8	96	0.91	318	0.86
9	111	1.00	208	1.00
10	41	1.00	13	0.96

of training data is representative for the testing data this results in higher accuracy but in the cases were this is not the case it results in lower accuracy. This can also be seen in the full ROC curves in appendix A.4.1.

For SVM the impact is not as easily explainable. An obvious source of error might the use of a linear kernel to select features. The linear kernel does generally perform worse than the RBF kernel. Thus one might conclude that the linear kernel allowed for more error than the RBF kernel. There has been studies that show feature selection may be valuable analysis to include in preprocessing operations for classification by SVM (Pal and Foody, 2010). However from this experiment the conclusion were that the extra effort of doing feature selection is not valuable towards overall performance.

Another problem with feature selection for these types of experiment is that the selected features will be the ones that separate the classes best and not necessarily the ones

Table 4.5: Feature selection results comparing scores from doing feature selection to not doing feature selection.

	Random Forest	
Data sets	No feature selection	Feature selection
D_1 D_2 D_3	0.932 ± 0.05	$\mathbf{0.936 \pm 0.05}$
D_4 D_5 D_6 D_7 D_8	$\mathbf{0.898 \pm 0.15}$	0.882 ± 0.16
	SVM	
Data sets	No feature selection	Feature selection
D_1 D_2 D_3	$\mathbf{0.954 \pm 0.04}$	0.951 ± 0.04
D_4 D_5 D_6 D_7 D_8	$\mathbf{0.844 \pm 0.21}$	0.836 ± 0.22

related to the overall problem. For example did Guyon et al. (2002) point out that a particular challenging problem with their colon cancer data set is that the tumor samples and normal samples differ in cell composition. Thus was the best split that tumor samples were generally rich in epithelial(skin) cells while normal samples held a variety of cells. This split of tumor and normal is not informative for tracking cancer related genes.

4.2.4 Algorithm Performances and Data Sets

As shown in previous sections the two main algorithms used for classification is random forest and SVM. The initial choice of these are connected to the use of similar classifications from related works. These have themselves several parameters that can be tuned. Random forest has number of estimators, choice of splitting criteria and number of max features. SVM has the choice of kernel, error term C, complexity γ, degree of polynomial and coefficient. Though not all of these are available for every kernel. In addition to this there were some attempts to create bagging and boosting ensembles to achieve even better performance. Table 4.6 gives a short summary of this.

Table 4.6: Overview of algorithm performances in a 10-fold cross validation using the two combined data sets.

	Algorithm performance			
Data sets	Random Forest	SingleSVM	BaggingSVM	BoostingSVM
Hepmark	0.932 ± 0.05	$\mathbf{0.954 \pm 0.04}$	0.88 ± 0.05	0.92 ± 0.05
Colon	$\mathbf{0.898 \pm 0.15}$	0.844 ± 0.21	0.67 ± 0.16	0.76 ± 0.37

To select best parameters for random forest a simple experiment were done. A grid search through parameters in table 4.7 on the Hepmark data set finding the optimal parameters in each of the 10 folds. In this experiment some deviation from standard parameters gave a little improvement in score. The standard parameter of max features (square root) were best in all folds. In the criteria parameter gini and entropy were best in five folds each but gini had a slightly higher increase of performance compared to entropy in its folds. For estimators between 100 and 500 were best in all folds. Going forward a preset of parameters were used setting estimators to 200, criterion to gini and max features to square root of number of features. There are obviously faults to this procedure by not doing a full grid search through all parameters every time but the additional cost in processing time, approximately one hour on a single 4.4GHz core, seemed not to be worth it. For comparison this preset is only slightly worse 0.932 to the optimal 0.941 having the same standard deviation while being close to instantaneous.

Table 4.7: Random forest grid search parameters. Square root and log_2 indicates that the square root and log_2 of number of features are used. Real numbers indicate the percentage of max features that can be used.

Parameter	Values
Estimators	10, 50, 100, 200, 500
Criterion	gini, entropy
Max features	$\surd$, $\log_2$, 0.5, 1.0

The grid search used on the SVM classifier used all parameters in table 4.8. In most folds both the polynomial and RBF kernels performed best. RBF dominated the colon data sets and polynomial of degree 3 dominated the hepmark data sets. Neither linear kernel or sigmoid were best in any folds of the combined data sets. Head-to-head RBF kernel performed slightly more consistent than the polynomial kernel. Thus to speed up was a reduced parameter list were created that only considered the RBF kernel. The full parameter search also performed worse than the reduced. Scoring 0.954 ± 0.04 on the reduced to 0.946 ± 0.05 on the full parameters list for the hepmark data sets and scoring 0.844 ± 0.21 on the reduced to 0.839 ± 0.21 on the full parameters list for the colon data sets. The ROC curves from these runs can be seen in the appendix A.4.1.

Table 4.8: SVM grid search parameters. C is used in all kernels. γ is used in RBF, polynomial and sigmoid. Coef is used as an independent term in polynomial and sigmoid kernels. Degree is only used in the polynomial kernel.

Parameter	Values
Kernel	Linear, Polynomial, Sigmoid, Radial Bias Function
C	0.1, 1, 5, 10
γ	0.1, 0.01, 1e-3, 1e-4, 1e-5
Coef	0, 1
Degree	1, 2, 3

Table 4.9: Algorithm performances overview. D_7 is a tumor only data set and thus has no score by itself. The best scores for each data set are in bold.

	Algorithm performance on individual data sets			
Data sets	Random Forest	SingleSVM	BaggingSVM	BoostingSVM
D_1	0.92 ± 0.10	$\mathbf{0.94 \pm 0.07}$	0.92 ± 0.08	0.92 ± 0.07
D_2	0.92 ± 0.09	$\mathbf{0.94 \pm 0.07}$	0.94 ± 0.08	0.84 ± 0.11
D_3	$\mathbf{1.00 \pm 0.00}$	0.23 ± 0.39	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
D_4	$\mathbf{1.00 \pm 0.00}$	0.95 ± 0.11	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
D_5	$\mathbf{0.66 \pm 0.16}$	0.51 ± 0.18	0.48 ± 0.17	0.40 ± 0.16
D_6	$\mathbf{0.75 \pm 0.38}$	0.58 ± 0.45	0.34 ± 0.47	0.17 ± 0.37
D_8	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$

The data sets were of quite different difficulties to classify. Table 4.9 has a summary of how the different algorithms performed on the individual data sets. An important note here is that a small sample count of some of these data sets makes them harder to classify alone. As an example SingleSVM performed better for combined data sets, D_2 and D_3 gave 0.95 ± 0.06 and D_6 and D_8 gave 0.97 ± 0.08 the latter is also the case for BaggingSVM and BoostingSVM. While all but one data set could be classified with 0.95 or above AUC score, data set D_5 were consistently hard to classify. The best performance on this set were achieved by the random forest preset achieving 0.66 ± 0.16. Excluding this data set from the overall colon data sets gives an ROC AUC score of 0.99 ± 0.03 and 0.98 ± 0.03 for random forest and SingleSVM respectively.

BaggingSVM and BoostingSVM were introduced as an attempt to improve the initial performances from random forest and SingleSVM. The bagging classifier has 3 parameters that can be tuned: base estimator, number of estimators and max features to be given to each estimator. For boosting, Adaptive boost classifier (Adaboost) were used. This also has 3 parameters that can be tuned: base estimator, number of estimators and learning rate. The base estimators for both of these could also be tuned in the same manner as the SingleSVM. However tuning the base estimator much is somewhat counter intuitive for the overall algorithm as the idea is to use SVM as a weak learner in the same way random forest uses decision trees. Thus were the base estimators kept to the linear and RBF kernel.

$$Binom(0, 9, 0.5) = P(X \leq 0) = 0.001953125. \tag{4.2}$$

The hypothesis, equation 4.2, was there that there would be some improvement over random forest or SingleSVM in a single or combination of data sets. However in none of the nine combinations from tables 4.6 and 4.9 were bagging and boosting strictly better. Thus was the hypothesis rejected and further testing of bagging and boosting is left to future work.

4.3 Imbalance Problem

The next experiment is to see how well other data sets can be used to classify an unknown data set. This will be done by training on all other available data sets of the same disease and then sampling a subset of the testing set. In addition the same procedure will be done using the GSEA signatures of the training data sets and testing how well these can be used to classify the testing subset.

To test this the random forest and SingleSVM classifiers from the previous section were used. Leaving one data set out as a test set, the remaining data sets were used to train the classifiers. The training sets were scaled individually using standard scaler and minmax scaler. Then for each sample in the test set, take 0,1,2,4,8,16 and all positive samples and 0,1,2,4,8,16 and all negative samples from the test set. These combinations of samples were scaled using standard scaling, minmax scaling, closest scaler and nonscaled and saved as results in a csv file. Closest scaler is a self defined scaling using the same scaling as the closest full training set.

These results were extracted onto heatmaps based on scaling of training set, scaling of test set and algorithm. On the heatmap the mean of each combination was generated for each data set and the combined sets for liver and colorectal cancer. The mean was the ROC(AUC) score where such is applicable (from at least two samples of both tumor and normal), and balanced accuracy where ROC score was not applicable.

To generate GSEA signatures a data set is first ran through a R function(page 67) that extracts the miRNAs that negatively and positively correlated with tumor samples for that data set. The negatively correlated genes are used as a normal gene set while the positively correlated genes are used as a tumor gene set and saved in a Gene Matrix Transposed (GMT) file. This GMT file can be used in a GSEA package for python to run single or multiple samples though for an enrichment score per gene set in the GMT. Thus are two enrichment scores per training set generated for the test set.

These enrichment scores can be used in multiple ways. The first intuitive way was to evaluate the normal enrichment scores to the tumor enrichment scores. However in this process there were found some on average the normal enrichment scores had a higher mean than the tumor enrichment scores. Instead the enrichment scores from the training samples were used to train a SVM classifier, optimized with the same parameters used in table 4.8, that could be used on the test samples enrichment scores. This process is illustrated in figure 4.5.

One strictly advantageous feature of using the enrichment scores is that these are not scaled and thus are not reliant on the set scaling. Thus should these not be affected by imbalance from the test data set. The only real variable in this scoring is the creation of the original gene sets to be used in GSEA. This allows for saving of the enrichment scores and new generation of heatmaps are fairly fast compared to the multiple types of scaling, algorithms and possibly feature selection used in the other heatmap generation. It is however true that SVM used to solve biases in enrichment scores also must be optimized but this training set only has two times the number of training sets as features, specifically four for hepmark and up to eight for colorectal, while the other has several hundred and possibly thousands for other miRNA data sets.

The results from both hepmark and colorectal data sets can be found in figures 4.6 and 4.7. The best heatmap from the non-GSEA heatmap approach is shown alongside the GSEA heatmap for each disease. All other heatmaps can be found on page 64. In these heatmaps the results gained inside the 25 squares with at least 2 positive and 2 negative samples differed a lot with the ones obtained outside. As a results the inner area refers to these squares and the outer area to the area outside these squares. One should also note that the performance metric of AUC is valid inside all of the inner area while the outer area uses balanced accuracy and accuracy. This change in performance metric also makes the methods which should not be affected by imbalance such as GSEA and unscaled has a slightly higher performance inside this area.

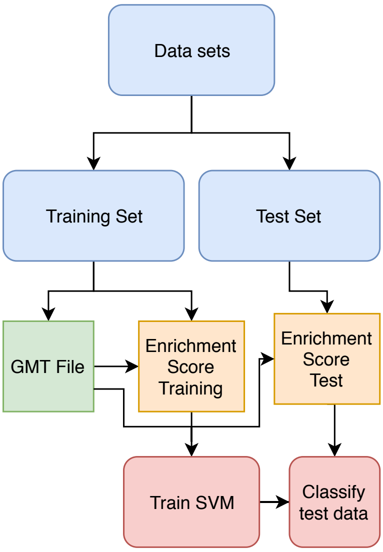

Figure 4.5: GSEA model with SVM classification.

In the hepmark data sets SVM proved the best algorithm in all scalings, furthermore minmax scaling was best for the inner area. In the outer area minmax and closest scaling were imbalanced towards tumor and GSEA and unscaled performed pretty equal with a slight edge to GSEA which had the highest average performance of 0.787 in accuracy and balanced accuracy with lower deviation that unscaled. The best for inner and outer can be seen in figure 4.6.

In the colorectal data sets random forest were best in most cases. In the inner area minmax scaling was the worst and the remaining three were pretty equal but the best was the GSEA approach with an average of 0.77 AUC score. For the outer area minmax performed worst followed by GSEA and unscaled. The best performance in the outer area was actually a SVM with closest scaling scoring an average of 0.78 in balanced accuracy. In figure 4.7 the two best performing heatmaps for the colon data set can be found.

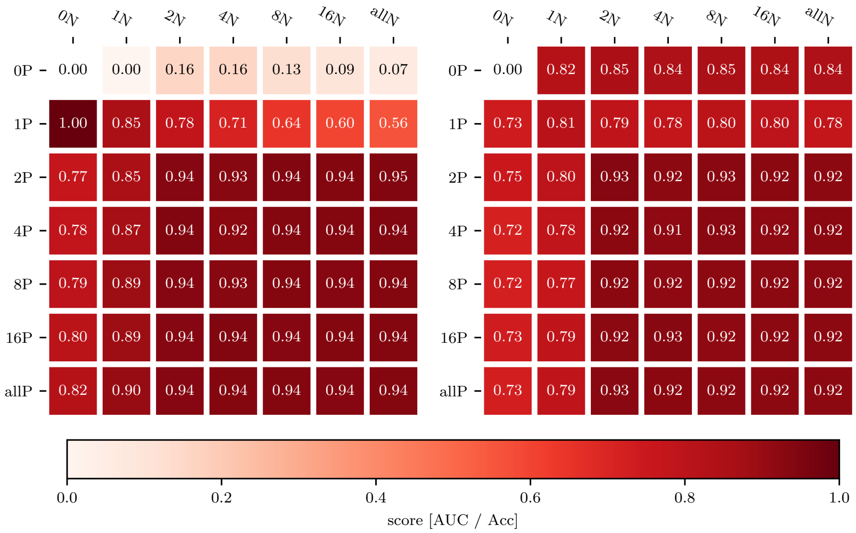

Figure 4.6: SVM and GSEA heatmaps for hepmark data sets. The left heatmap contains SVM scores from data scaled to range -1 to 1, while the right heatmap contains scores on classification based of GSEA scores.

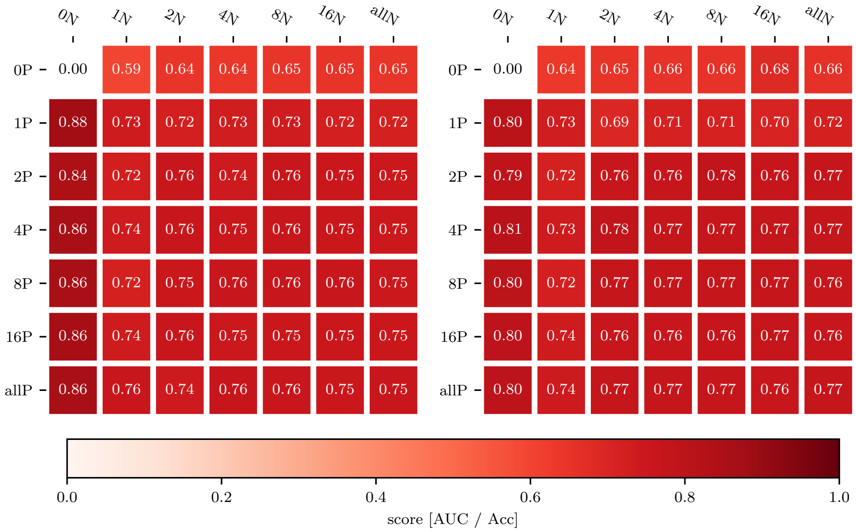

Figure 4.7: Random forest and GSEA heatmaps for colorectal data sets. The left heatmap contains random forest scores from data that are unscaled, while the right heatmap contains scores on classification based of GSEA scores.

Assuming the methods are equally good the hypothesis can be formulated as a coin flip. Over the two data sets a total of 96 tiles can be compared. The methods that has no scaling, unscaled and GSEA, is compared first. Choosing the best overall algorithm for unscaled for each of the data sets. In 72 out of the 96 tiles GSEA is equal or better than unscaled giving us equation (4.3) which concludes that GSEA outperforms non-GSEA method with unscaled data.

$$Binom(72, 96, 0.5) = P(X \leq 72) = 0.999.. \tag{4.3}$$

For the methods with scaling, closest scaler is compared to GSEA for the outer area while minmax scaler is compared to GSEA for the inner area. Also this time using the best overall algorithm for closest and minmax scaler in each of the two combined data sets. This gives us equations (4.4) and (4.5) respectively. For the outer area the closest scaler outperformed the GSEA approach in 17 tiles for the colorectal data set and in 6 tiles for the hepmark data set. For the inner area minmax scaler beats the GSEA approach in all 25 tiles for the hepmark data set and is worse in all tiles for the colorectal data sets. Thus concluding that neither method is better than the other.

$$Binom(23, 46, 0.5) = P(X \leq 23) = 0.56 \tag{4.4}$$

$$Binom(25, 50, 0.5) = P(X \leq 25) = 0.56 \tag{4.5}$$

4.4 MiRNA Feature Importance

The next experiment is to look into the most important features used for classification in the previous sections across different methods. This is done to look into similarities in selected miRNA. One interesting part of this is whether the most important miRNAs used for classification has known connections to the disease. Another interesting part is if the selected miRNAs found by the different methods are the same of if they found many different ones.

For the GSEA method the miRNAs found for the tumor gene sets and transported to the GMT file is used. In random forest an attribute called ”feature_importances” is used. This attribute calculates importance of each feature by first calculating the importance of each feature in each tree estimator then sum this and divide it by the number of tree estimators. The importance of each feature in a single tree is the normalized total reduction of the criterion brought by that feature. In this case this would be the features gini importance in that tree. One downside to this is that the computed importance of each feature becomes quite small in a large forest and that it does not indicate if the feature is more correlated with disease or normal.

For SVM there is no attribute available to indicate feature importance for most kernels. One possible solution to this is to remove features one by one and see how this affects the performance. The problems with this is the amount of computation time because of the relative high number of features and the base time required to train the model. Instead by using a linear kernel the coefficients for each feature can be used. Thus is the feature importance generated by this model not the same as the one used in the primary SVM model. The coefficients in the linear kernel is the weight assigned each feature. This means that a positive coefficient indicates that the feature is up-regulated in disease and a negative coefficient indicate that the feature is down-regulated in disease. This is illustrated in figure 4.8.

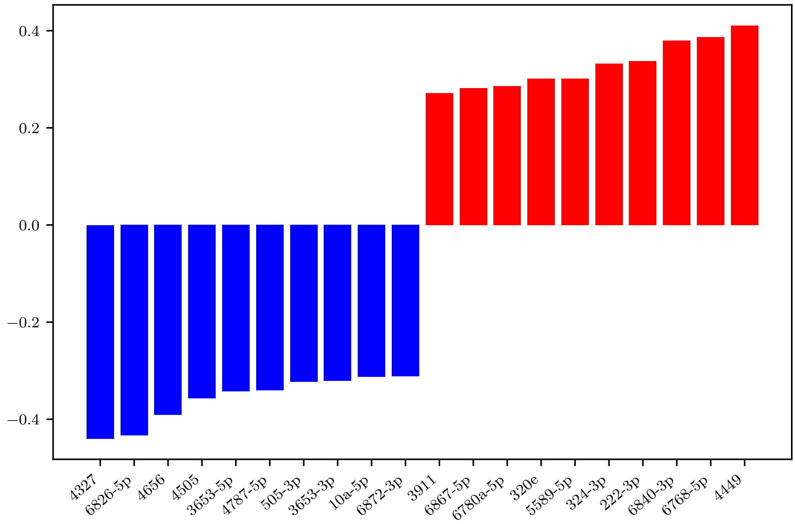

Figure 4.8: Feature importance in SVM for hepmark. Red bars are positively correlated with tumor i.e. up-regulated and blue bars are negatively correlated with tumor. Features has their prefix (hsa-miR-) removed to make space.

To validate connections between miRNAs and diseases Human microRNA Disease Database (HMDD) is used. As this has to be done manually not all miRNAs were checked but the top 15 features for each method were used to validate. These results are summarized in tables 4.10 and 4.11.

To give GSEA a scoring in these tables rank GSEA is invented. Rank GSEA is calculated by giving each feature in the gene set sequence a number as it appears in the gene set. In addition to separate between normal and tumor gene sets the normal gene sets are given negative numbers while the tumor gene sets are positive numbers. For the cases where a feature is in several gene sets the smallest number is always chosen.

To select the top 15 features for each method the following were done. For random forest the highest values in the calculated feature importance. For SVM the highest positive coefficients were used. For GSEA the top features from each tumor gene set such that they total 15, e.g. if you have three tumor gene sets and no duplicates in the top five in each of these the top 15 features is then the top five features in each of those three sets.

In some cases the scoring between GSEA and the other methods does not make sense as either the GSEA does not have the feature in any of its gene sets or in the case were the gene set contains features that were excluded either in the reads per million normalization or a feature that is only present in some of the data sets. In these cases these values will be filled with N/A.

For the HMDD results it has been chosen to use the labels ”Yes”, ”No” and ”Other”. The first label, Yes, is if the searched miRNA is connected to the disease (HCC for hepmark sets and CRC for colon sets). Other is if the searched miRNA is connected to diseases but not the one in question. No is if the searched miRNA has no hits in the database.

In these two tables there are a couple of important observations. First is to what degree does the features extracted from each method overlap i.e. the feature is indicated as important by more than one method. Second at what rate it the features marked as important actually known to be related to the disease. Third is there a lot of mismatch between scoring of the same feature.

In table 4.10 the answer to the first question is that there are only five features that overlaps between methods. To the second question the random forest approach has all its 15 top rated features as known targets while GSEA and SVM has 12 and 7 respectively. In terms of scoring one important observation is that most of the high ranking random forest features were in negative for GSEA and SVM which indicates that these miRNAs are down-regulated in tumor and may be a indication for why there is few overlapping features between methods. Scores from SVM and GSEA also do agree which features are up- and down-regulated in HCC in all but three cases and in these three cases there is the SVM score that is positive and and GSEA that is negative and the SVM score is lower than 0.05 in absolute value. To further check this the miRNAs hsa-miR-200b-3p, hsa-miR-200a-3p and hsa-miR-96-5p had their regulation confirmed by previous findings manually in HMDD.

In table 4.11 the amount of overlap between methods are somewhat higher. Also here the random forest approach has the best prediction of miRNAs related to the disease with 14 of its 15 features being related, while GSEA and SVM had 13 and 12 respectively. The scoring of features from different methods also seem to be more correlated than the previous table. There are also more up-regulated or positive scored features. GSEA and SVM does not agree to the same degree which features are either up- or down-regulated as for HCC. The features that had a fairly high score (above 0.5) for SVM but were mismatched in regulation based on GSEA rank were checked manually for regulation in HMDD. These miRNA were hsa-miR-138-5p, hsa-miR-143-3p, hsa-miR-143-5p and hsa-miR-363-3p. The discovery was that the GSEA had the right regulation in all four cases.

Each of the three feature importance strategies had their uses. Random forest were the most accurate in terms of finding related targets. GSEA were the second best in accuracy and also had the correct notion which way the feature in question was regulated but does not have a rank for all features and does not enforce strict ordering of features. SVM was the most inaccurate but has a ordering of all features and its regulation.

Table 4.10: Feature importance for hepmark data sets. MiRNA is the feature. Rank GSEA refers to the what number the feature is in the gene set, positive values for tumor and negative values for normal gene signature. Rank RF is the features position in the sorted feature importance list from random forest. Score SVM is the linear kernels coefficient for the particular feature. Related to disease is whether the feature is linked to HCC in HMDD. The table is sorted by Rank RF.

MiRNA	Rank GSEA	Rank RF	Score SVM	Related to Disease
hsa-miR-200b-3p	-2	1	-0.17	Yes
hsa-miR-200a-3p	-18	2	-0.12	Yes
hsa-miR-96-5p	4	3	0.16	Yes
hsa-mir-130b-3p	7	4	-0.05	Yes
hsa-miR-30a-3p	-24	5	-0.12	Yes
hsa-miR-224-5p	23	6	0.12	Yes
hsa-miR-30a-5p	-17	7	-0.13	Yes
hsa-miR-483-5p	-21	8	-0.17	Yes
hsa-miR-199a-3p	-5	9	-0.07	Yes
hsa-miR-199a-5p	-6	10	-0.12	Yes
hsa-miR-221-3p	15	11	0.12	Yes
hsa-miR-452-5p	10	12	0.05	Yes
hsa-miR-30d-5p	33	13	0.09	Yes
hsa-mir-21-5p	3	14	0.22	Yes
hsa-miR-25-3p	17	15	0.08	Yes
hsa-mir-15b-5p	6	16	0.007	Yes
hsa-mir-1269a	1	17	0.15	Yes
hsa-mir-3651	4	20	-0.06	Yes
hsa-mir-93-5p	5	21	-0.004	Yes
hsa-miR-182-5p	3	32	0.02	Yes
hsa-miR-222-3p	21	36	0.34	Yes
hsa-miR-183-5p	2	38	0.09	Yes
hsa-miR-15a-5p	31	50	0.23	Yes
hsa-miR-320e	71	52	0.30	Yes
hsa-mir-6090	1	65	0.15	No
hsa-miR-1290	38	87	0.25	Yes
hsa-miR-147b	5	89	0.17	Other
hsa-miR-324-3p	128	92	0.33	Yes
hsa-miR-1180-3p	7	113	0.08	Yes
hsa-mir-3665	2	127	0.022	No
hsa-miR-5589-5p	N/A	140	0.30	No
hsa-miR-6768-5p	32	170	0.38	No
hsa-miR-1185-1-3p	N/A	219	0.26	Other
hsa-miR-4449	21	229	0.41	Other
hsa-miR-6840-3p	83	299	0.38	No
hsa-miR-939-5p	85	371	0.23	Yes
hsa-miR-6780a-5p	N/A	383	0.28	No
hsa-miR-6867-5p	N/A	387	0.28	Other
hsa-miR-10b-3p	1	546	0.77	Yes
hsa-miR-3911	N/A	575	0.27	No
hsa-miR-183-3p	2	N/A	N/A	Yes

Table 4.11: Feature importance for colon data sets. MiRNA is the feature. Rank GSEA refers to the what number the feature is in the gene set, positive values for tumor and negative values for normal gene signature. Rank RF is the features position in the sorted feature importance list from random forest. Score SVM is the linear kernels coefficient for the particular feature. Related to disease is whether the feature is linked to CRC in HMDD. The table is sorted by Rank RF.

MiRNA	Rank GSEA	Rank RF	Score SVM	Related to Disease
hsa-miR-181d-5p	15	1	0.02	Other
hsa-miR-93-5p	47	2	0.26	Yes
hsa-miR-92a-3p	7	3	0.27	Yes
hsa-miR-584-5p	4	4	0.31	Yes
hsa-miR-25-3p	23	5	0.88	Yes
hsa-miR-21-3p	3	6	0.34	Yes
hsa-miR-378a-3p	-3	7	-0.48	Yes
hsa-miR-31-5p	1	8	0.03	Yes
hsa-miR-9-5p	-6	9	-0.27	Yes
hsa-miR-1-3p	-4	10	-0.53	Yes
hsa-miR-20a-3p	47	11	0.20	Yes
hsa-miR-147b	-6	12	0.02	Yes
hsa-miR-30a-5p	-3	13	-0.07	Yes
hsa-miR-424-3p	7	14	0.64	Yes
hsa-miR-182-5p	6	15	0.26	Yes
hsa-miR-135b-5p	2	18	0.32	Yes
hsa-miR-183-5p	5	19	0.09	Yes
hsa-miR-224-5p	3	27	0.18	Yes
hsa-miR-125a-3p	N/A	42	0.69	Yes
hsa-miR-7641	26	54	0.80	Other
hsa-miR-138-5p	-15	66	0.63	Yes
hsa-miR-27b-5p	N/A	74	0.54	Yes
hsa-miR-21-5p	1	76	0.05	Yes
hsa-miR-10a-5p	N/A	83	0.60	Other
hsa-miR-181a-2-3p	N/A	86	0.74	Yes
hsa-miR-143-3p	-34	90	0.71	Yes
hsa-miR-143-5p	-16	181	0.49	Yes
hsa-miR-1271-5p	N/A	195	0.67	Yes
hsa-miR-323a-3p	N/A	227	0.63	Other
hsa-miR-210-3p	N/A	233	0.56	Yes
hsa-miR-363-3p	-31	235	0.54	Yes
hsa-miR-33a-3p	N/A	306	0.63	Yes
hsa-miR-549a	2	N/A	N/A	No
hsa-miR-135b-3p	1	N/A	N/A	Yes
hsa-miR-503-5p	4	N/A	N/A	Yes
hsa-miR-31-3p	3	N/A	N/A	Yes
hsa-miR-1273d	5	N/A	N/A	Other

Chapter 5

Discussion and Conclusion

In this chapter a discussion regarding previous experiments is done and conclusions to problem of combining miRNA data sets. In addition a overview of contributions and possible future work is included.

5.1 Discussion

Looking back at all experiments there are several important observations to be made. In the first experiment there were already several steps of pre-processing done. These preparation processes removes quite a bit of the miRNAs in the original data sets. Through the filtering following the reads per million equation the miRNAs from the colorectal data sets were reduced from around 2438 unique miRNAs to 578 and the hepmark data sets were reduced from 2259 unique miRNAs to 701. In this step some information in low expressed miRNAs may have been lost that might further increase performance in classification. In addition there was a filling of missing values as -1 compared to the removal of the miRNA from the combined data set. Initial tests pointed towards using the filling method however later tests on the colorectal data sets shown that this were lowering performance quite a bit.

In the scaling experiments the most surprising result was how well the unscaled data did perform. Especially in the colorectal data sets. In an attempt to understand why the PCA of the unscaled and scaled is shown in figure 5.1. Here the first observation was the likeness of data sets D_4, D_7 and D_8 which actually overlaps quite well without any scaling. These sets already contributes 235 of the 348 samples in the combined colorectal data set. Close to perfect classification, 0.99 AUC score, were achieved from both unscaled and scaled when excluding D_5. Thus it seems that the actual performance mostly relied on the classification of D_5. As D_5 held the majority of remaining samples, 95 out of 113, the best technique was to distinguish the samples from this data set rather than making them more relatable to the others.

Another important mention is that the unscaled data is not entirely unscaled as it has been through the reads per million process though this process has not eliminated any

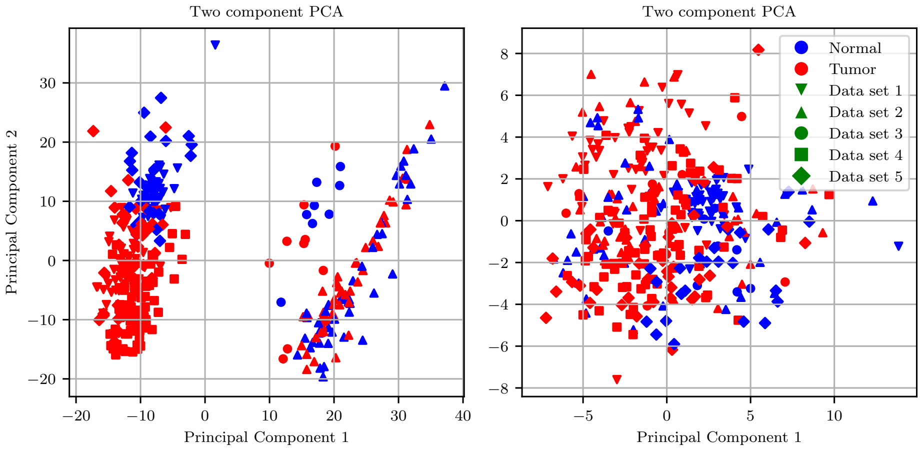

Figure 5.1: Principal component analysis for colorectal data sets. Unscaled to the left and one MinMax scaler per data set to the right.

biases inside the data sets themselves.

For the feature selection experiments there were few advantages found for doing feature selection. Neither did it improve performance nor provide a stable subset of features that correctly separated the classes. Highly related to the number of initial features this may have been differently if a lower threshold had been set in the initial pre-processing. In addition, for the case of SVM, the use of a linear kernel to reduce the feature space of a RBF kernel should be questioned. In the same manner as mentioned in the miRNA feature importance section it is possible to use the RBF kernel to do feature selection. However this would require the implementation of a ranking of features once the classifier is trained and then run this ranking at each iteration a feature is removed. An example of this RFE-SVM for feature selection can be found in Guyon et al. (2002).

In the algorithm experiments few parameters were tested for both baggingSVM and boostingSVM. SVMs are not necessarily the best base estimators for bagging and boosting and other combinations should also have been tested. Due to some of the discoveries in related works section 2.3.3 the more complex models were not necessarily favored. Thus were the experiments to some degree favored to keep models as simple as possible.

The imbalance experiments were the most time consuming work as this required setup of GSEA for enrichment scores. This again required some analysis of single data sets to extract gene sets that made up the GMT files which was easiest achieved using R. Enrichment scores did though prove to be quite good at classifying and had the best overall performance in this experiment. As for the non-GSEA approach random forest had better scores in the more ambiguous data while SVM had the highest measured performance in balanced full data sets.

The score sheets generated in the imbalance experiments are themselves good objects of investigation. These allows us to select specific data sets, normalization strategies and algorithms. In general the performance greatly improve when there is at least two positive and two negative samples if normalized data is used. The use of a closest scaling proved only to be effective when data sets were quite similar and did not outperform the unscaled tests. For data set D_5 no algorithm, scaling or method gave good performance. The lack of general difference between the tumor and normal class for this data set was also discovered when creating gene signatures for GSEA. Here the information about pairing of samples had to be utilized to find genes that differed between the classes. At the very least this data set was much more difficult to classify than the other.

For the miRNA importance for classification three different methods were implemented. All three selected features that were mostly related to their respective disease based on information from HMDD. However all methods also did differ more than expected in what features were selected of a given importance. For instance SVM and GSEA found a feature, hsa-miR-10b-3p, in HCC that random forest had close to its bottom. Although random forest had the highest accuracy it would by itself still miss out on certain features and lack the information of which way the feature is regulated in tumor.

5.2 Conclusions

Combining different miRNA data sets for classification is a challenge that I found no one easy solution to. Here is the general findings and authors suggestions for doing so.

Related works gave several pointers for this problem. Random forest and Support Vector Machines were the algorithms that are best for classification of data that are high dimensional in features and low dimensional in samples. Normalization and feature selection were suggested when working with gene expressions and should help improve performance. The general approach to remove some bias in the data sets would be normalization of the data. However this does not remove all bias for miRNA data sets. Working with multiple transcriptome technologies a reads per million formula, equation (4.1), were used to make gene expressions between technologies comparable and remove the absolutes that were present in raw RNA-seq data and turn them into relatives. Combining the data sets the missing miRNAs between data sets are excluded from the combined data set.

The first improvement to the procedure might be to lower the mean requirement in reads per million allowing more features in the data set to begin with. This can also benefit feature selection at later stages. Although unscaled data has performed really well in these experiments the general approach should be to normalize the data as this is what makes the data sets properly comparable. The process of doing reads per million is not enough by itself as the value ranges for each miRNA would still slightly differ between sets. Among scalers the scaling to range -1 to 1 had the best performance in these experiments.

For feature selection the benefits in these experiments were none. In general the suggestion should be to keep feature selection to an informed level. Not selecting features because they differ between classes alone, but because they are actually linked to the problem. This would bypass the problem of classifying tumor based of skin tissue instead of miRNAs that are linked to cancer.

Gene set enrichment analysis were a great tool for classifying data sets. This procedure also avoided the problems of normalization and thus were unaffected by smaller and unbalanced data sets. GSEA also has potential improvement both in its creation of enrichment scores, i.e. the GMT file, and classification using the enrichment scores. In this project only one set of GMT files were created using the statistical significance difference in expression levels from one data set at a time.

5.3 Contributions

In this thesis I have provided several useful insights into using multiple miRNA data sets to classify miRNA samples into tumor or normal. These include:

- Methods for combining miRNA data sets which removes internal bias to different degrees.
- Evaluation of several pre-processing and algorithm combinations when classifying combined miRNA data sets.
- Lists of miRNAs found to be related to colorectal and hepatic cancer respectively.

In addition several parts of the generated material is provided with the delivery:

- Source code.
- Generated enrichment scores for data sets.
- Generated GMT files.
- Generated score sheets.

5.4 Future Work

Multiple adjustments can be made to further enhance performance including trying other ensemble classifiers, use new feature selection estimators, finding alternatives to read per million normalization and testing the methods introduced in this thesis in new data sets of same and different diseases.

In GSEA gene sets can be made to score a single samples correlation with multiple diseases at once. This has potential application for looking into likeness in different diseases

There are additional information in most data sets that are not utilized here. The data for most data sets include paired samples. This information could be used to create a pair-based method for an extreme value of normalization, which should remove most technology and individual differences. This has somewhat more limited in practical use as it requires both the normal and tumor sample from the same individual. One potential use could be a prognosis using cox regression.

Bibliography

Banwait, J. K., Bastola, D. R., 2015. Contribution of bioinformatics prediction in microrna-based cancer therapeutics. Advanced drug delivery reviews 81, 94–103.

Batuwita, R., Palade, V., 2008. An improved non-comparative classification method for human microrna gene prediction. In: BioInformatics and BioEngineering, 2008. BIBE 2008. 8th IEEE International Conference on. IEEE, pp. 1–6.

Bertoli, G., Cava, C., Castiglioni, I., 2016. Micrornas as biomarkers for diagnosis, prognosis and theranostics in prostate cancer. International journal of molecular sciences 17 (3), 421.

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

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., Haussler, D., 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proceedings of the National Academy of Sciences 97 (1), 262–267.

Chakraborty, D., Maulik, U., 2014. Identifying cancer biomarkers from microarray data using feature selection and semisupervised learning. IEEE journal of translational engineering in health and medicine 2, 1–11.

Erson, A. E., Petty, E. M., 2009. mirnas and cancer: New research developments and potential clinical applications. Cancer biology & therapy 8 (24), 2317–2322.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16 (10), 906–914.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science 286 (5439), 531–537.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine learning 46 (1-3), 389–422.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al., 2003. A practical guide to support vector classification.

Ibrahim, R., Yousri, N. A., Ismail, M. A., El-Makky, N. M., 2013. mirna and gene expression based cancer classification using self-learning and co-training approaches. In: Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on. IEEE, pp. 495–498.

Iorio, M. V., Croce, C. M., 2012. Microrna dysregulation in cancer: diagnostics, monitoring and therapeutics. a comprehensive review. EMBO molecular medicine 4 (3), 143–159.

Keerthi, S. S., Lin, C.-J., 2003. Asymptotic behaviors of support vector machines with gaussian kernel. Neural computation 15 (7), 1667–1689.

Kim, K.-J., Cho, S.-B., 2010. Exploring features and classifiers to classify microrna expression profiles of human cancer. In: International Conference on Neural Information Processing. Springer, pp. 234–241.

Kothandan, R., Biswas, S., 2015. Identifying micrornas involved in cancer pathway using support vector machines. Computational biology and chemistry 55, 31–36.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering 160, 3–24.

Lee, R. C., Feinbaum, R. L., Ambros, V., 1993. The c. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. cell 75 (5), 843–854.

Li, L., Xu, J., Yang, D., Tan, X., Wang, H., 2010. Computational approaches for microrna studies: a review. Mammalian Genome 21 (1-2), 1–12.

Liao, Z., Li, D., Wang, X., Li, L., Zou, Q., 2018. Cancer diagnosis through isomir expression with machine learning method. Current Bioinformatics 13 (1), 57–63.

Liaw, A., Wiener, M., et al., 2002. Classification and regression by randomforest. R news 2 (3), 18–22.

Lu, J., Getz, G., Miska, E. A., Alvarez-Saavedra, E., Lamb, J., Peck, D., Sweet-Cordero, A., Ebert, B. L., Mak, R. H., Ferrando, A. A., et al., 2005. Microrna expression profiles classify human cancers. nature 435 (7043), 834.

Nembrini, S., Knig, I. R., Wright, M. N., 2018. The revival of the gini importance? Bioinformatics 34 (21), 3711–3718.
URL `http://dx.doi.org/10.1093/bioinformatics/bty373`

Önskog, J., Freyhult, E., Landfors, M., Rydén, P., Hvidsten, T. R., 2011. Classification of microarrays; synergistic effects between normalization, gene selection and machine learning. BMC bioinformatics 12 (1), 390.

Pal, M., Foody, G. M., 2010. Feature selection for classification of hyperspectral data by svm. IEEE Transactions on Geoscience and Remote Sensing 48 (5), 2297–2307.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Piao, Y., Piao, M., Ryu, K. H., 2017. Multiclass cancer classification using a feature subset-based ensemble from microrna expression profiles. Computers in biology and medicine 80, 39–44.

Razak, E., Yusof, F., Raus, R. A., 2016. Classification of mirna expression data using random forests for cancer diagnosis. In: Computer and Communication Engineering (ICCCE), 2016 International Conference on. IEEE, pp. 187–190.

Russell, S. J., Norvig, P., 2016. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,.

Saha, I., Bhowmick, S. S., Geraci, F., Pellegrini, M., Bhattacharjee, D., Maulik, U., Plewczynski, D., 2015. Analysis of next-generation sequencing data of mirna for the prediction of breast cancer. In: International Conference on Swarm, Evolutionary, and Memetic Computing. Springer, pp. 116–127.

Saha, S., Mitra, S., Yadav, R. K., 2016. A multiobjective based automatic framework for classifying cancer-microrna biomarkers. Gene Reports 4, 91–103.

Saito, T., Sætrom, P., 2010. Micrornas–targeting and target prediction. New biotechnology 27 (3), 243–249.

Schoonjans, F., Nov 2018. Roc curve analysis with medcalc.
URL `https://www.medcalc.org/manual/roc-curves.php`

Singh, B., Kushwaha, N., Vyas, O. P., 2014. A feature subset selection technique for high dimensional data using symmetric uncertainty. Journal of Data Analysis and Information Processing 2 (04), 95.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., et al., 2005. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences 102 (43), 15545–15550.

Tran, D. H., Ho, T. B., Pham, T. H., Satou, K., 2011. Microrna expression profiles for classification and analysis of tumor samples. IEICE TRANSACTIONS on Information and Systems 94 (3), 416–422.

Wang, D., Zhang, Y., Zhao, Y., 2017. Lightgbm: an effective mirna classification method in breast cancer patients. In: Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics. ACM, pp. 7–11.

Wang, Z., Gerstein, M., Snyder, M., 2009. Rna-seq: a revolutionary tool for transcriptomics. Nature reviews genetics 10 (1), 57.

Yang, S., Guo, L., Shao, F., Zhao, Y., Chen, F., 2015. A systematic evaluation of feature selection and classification algorithms using simulated and real mirna sequencing data. Computational and Mathematical Methods in Medicine 2015.

Appendix A

Appendix

A.1 Appendix: Structued litterature review protocol

Full list of search terms, ICs and QCs and table with paper cutoffs

A.2 Code and user guide

A short explaination of experimental setup python and packages and the usage aswell as a github reference. Github - vegabj/Mastersproject

A.3 Additional tables

Data set ID	Paper by ID
DS0	0
DS1	1,2,12
DS2	1, 2
DS3	2
DS4	3
DS5	3
DS6	3
DS7	3
DS8	3
DS9	3
DS10	3
DS11	9
DS12	9
DS13	11,17, 20, 21
DS14	11, 21
DS15	11
DS16	11, 21
DS17	11, 21
DS18	11
DS19	12
DS20	12
DS21	12,13,16,18,19
DS22	16
DS23	21
DS24	21

Table A.1: Data set ID and article ID relations.

A.4 Additional plots

A.4.1 ROC curves

ROC Curves from Feature Selection

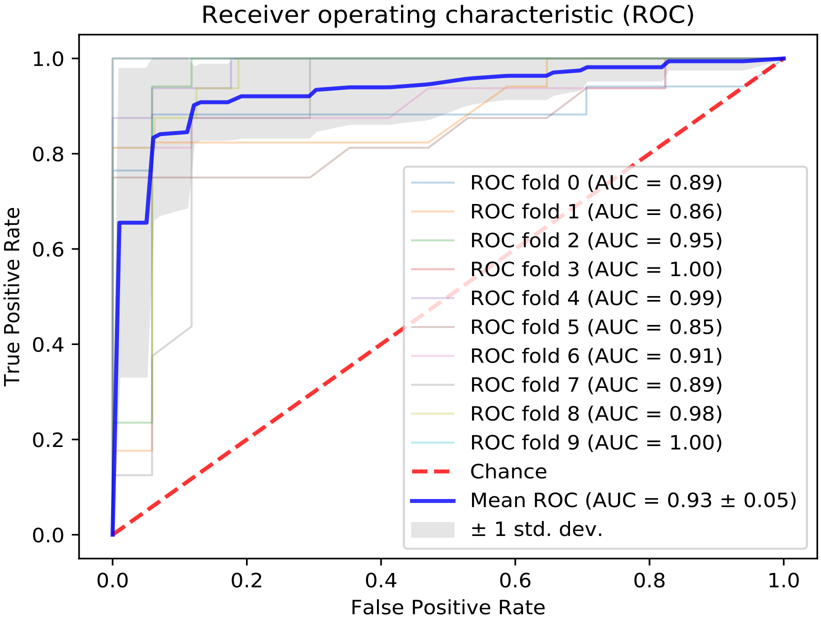

Figure A.1: Baseline ROC Random Forest Hepmark

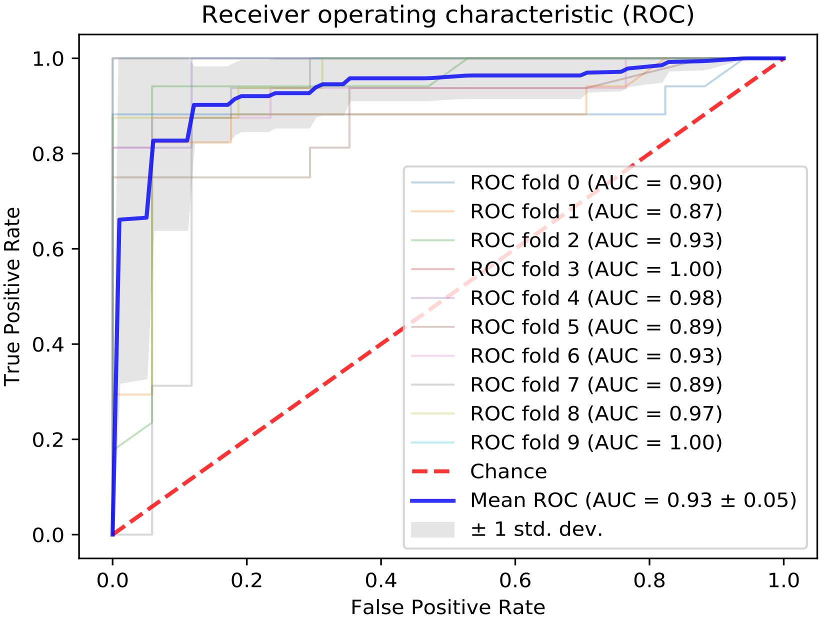

Figure A.2: Feature Selection ROC ROC Random Forest Hepmark

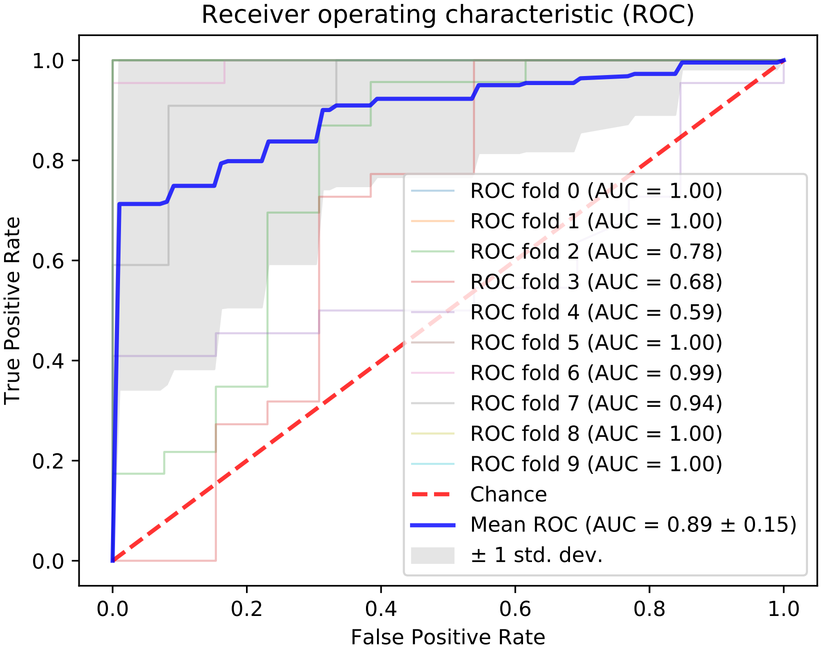

Figure A.3: Baseline ROC Random Forest Colon

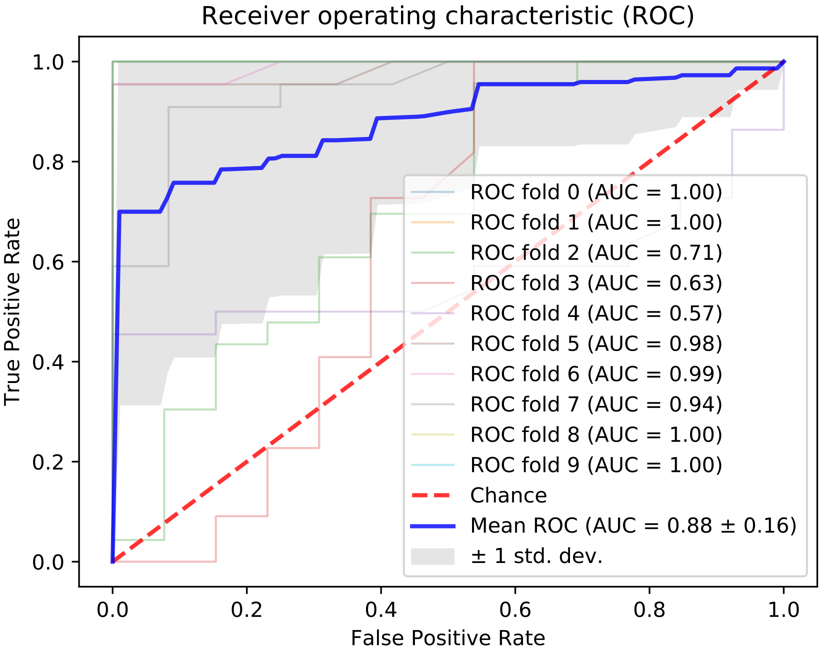

Figure A.4: Feature Selection ROC ROC Random Forest Colon

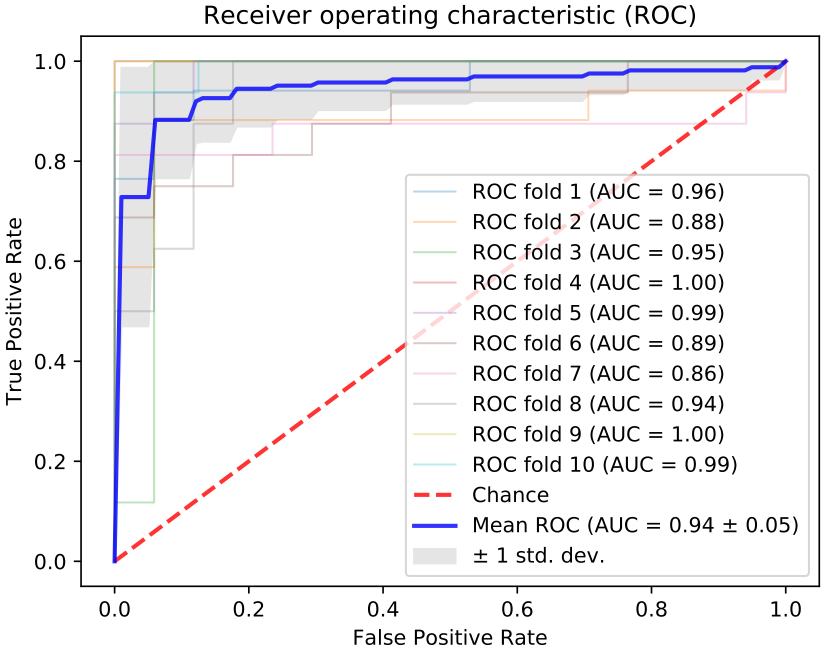

Figure A.5: Full parameter search SVM hepmark.

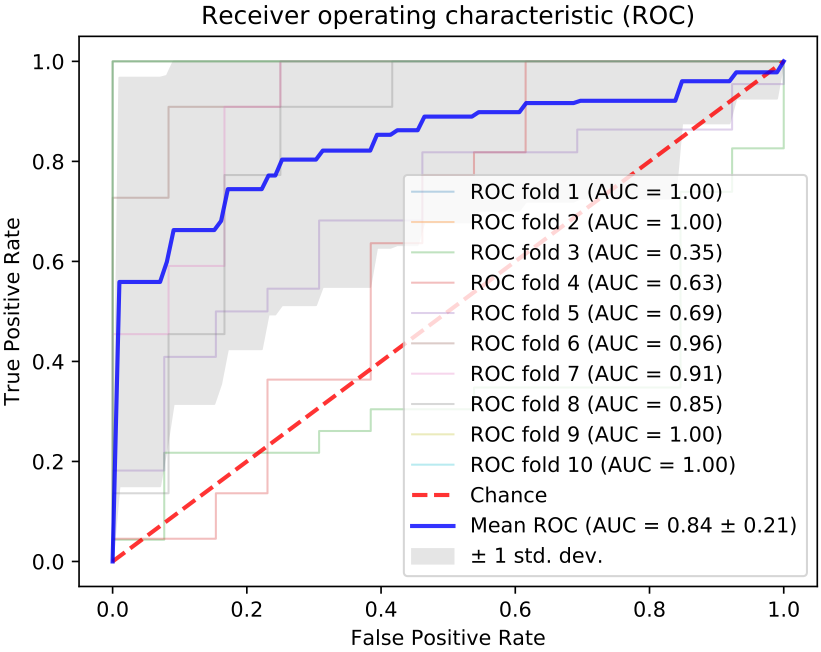

Figure A.6: Full parameter search SVM colon.

A.4.2 Heatmaps

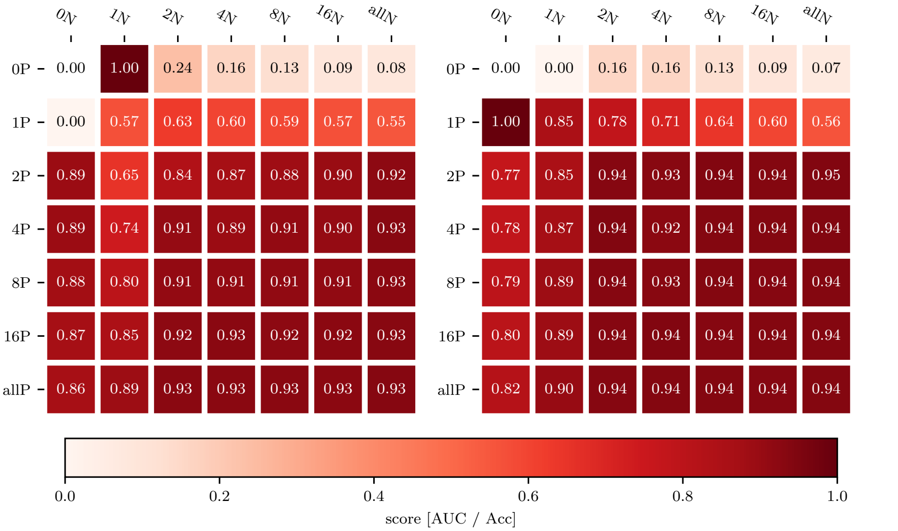

Figure A.7: Heatmaps for hepmark minmax scaling. Random forest to the left. SVM to the right.

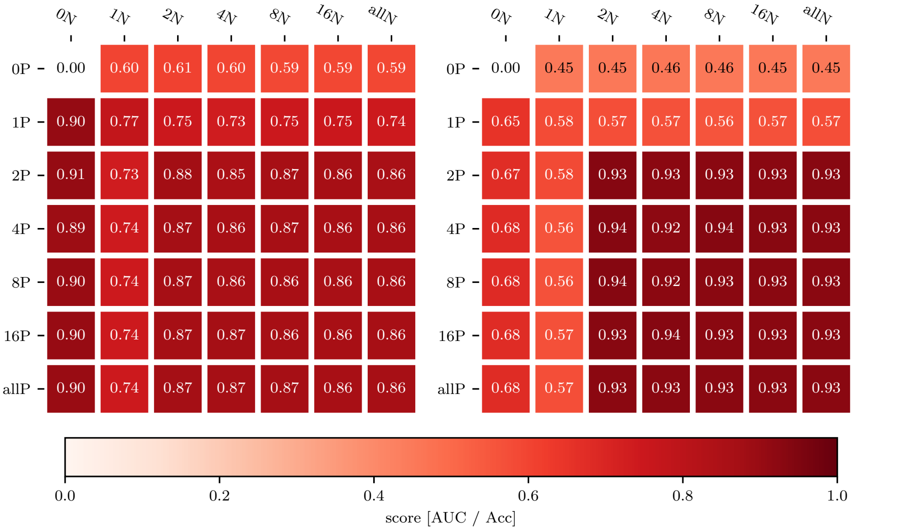

Figure A.8: Heatmaps for hepmark closest scaling. Random forest to the left. SVM to the right.

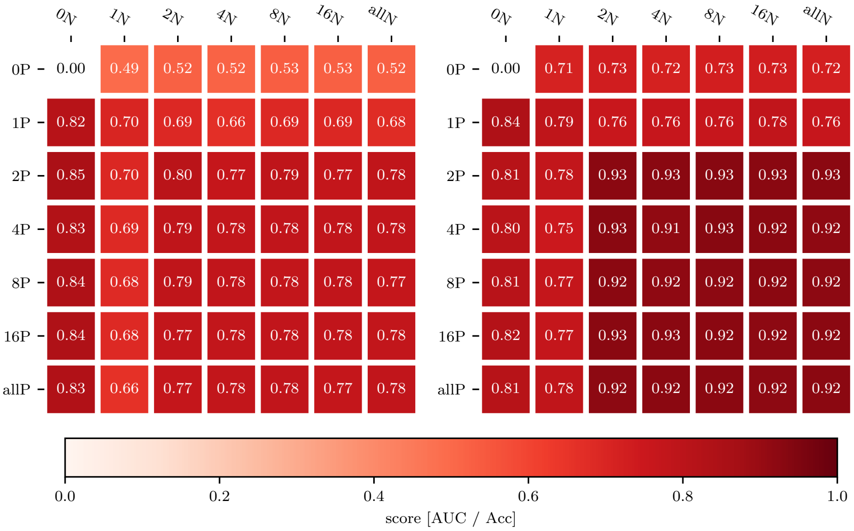

Figure A.9: Heatmaps for hepmark unscaled. Random forest to the left. SVM to the right.

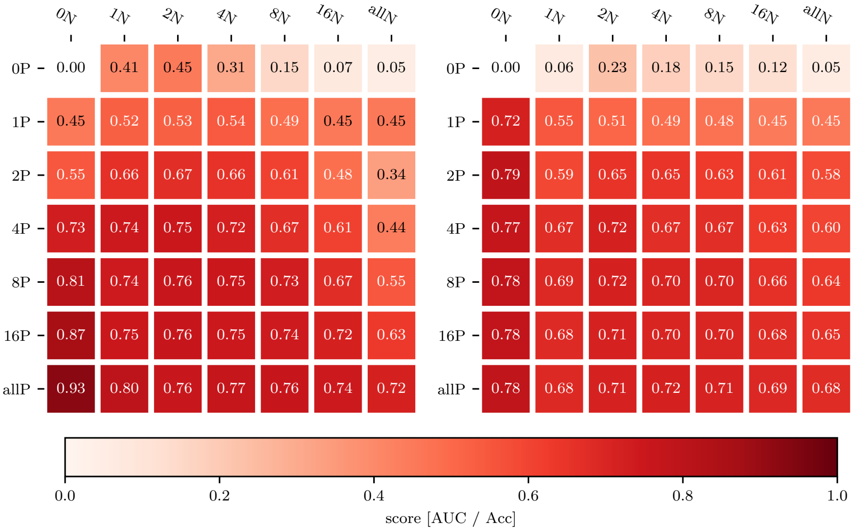

Figure A.10: Heatmaps for colorectal minmax scaling. Random forest to the left. SVM to the right.

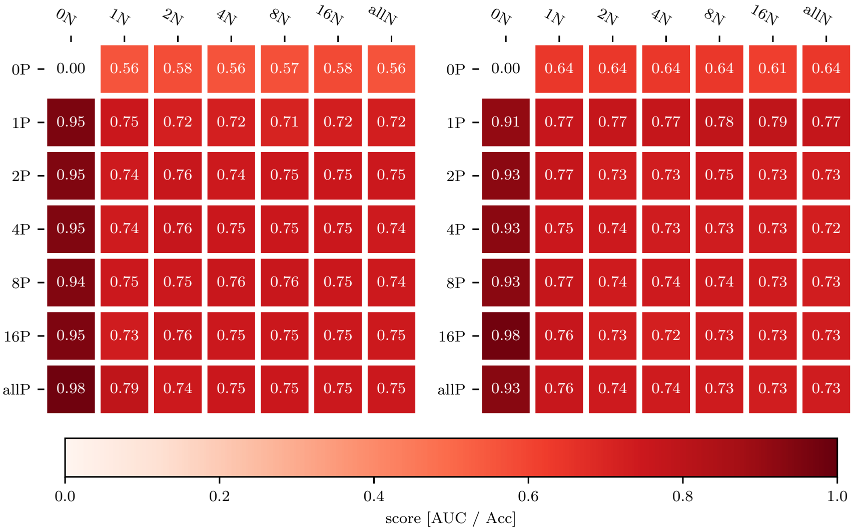

Figure A.11: Heatmaps for colorectal closest scaling. Random forest to the left. SVM to the right.

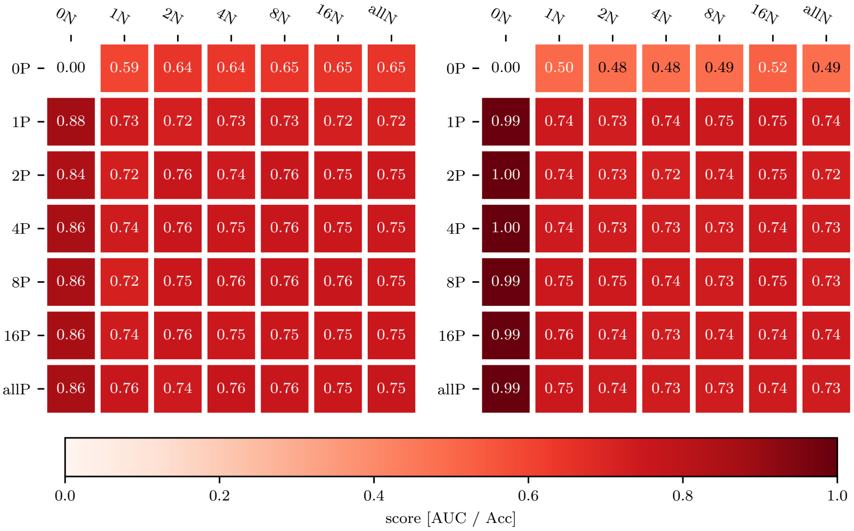

Figure A.12: Heatmaps for colorectal unscaled. Random forest to the left. SVM to the right.

A.4.3 Density plots?

TODO

A.4.4 PCA plots

TODO

A.5 Code sniplets

A.5.1 extract miRNAs R

```
dup.cor <- duplicateCorrelation(count.voom,
    design=design, block=block)
# Estimate the correlation between duplicate spots (regularly
# spaced replicate spots on the same array) or between
# technical replicates from a series of arrays.
fit <- lmFit(count.voom, design=design,
    block=block, correlation=dup.cor$consensus.correlation)
# Fit linear model for each gene given a series of arrays
fit <- eBayes(fit)
# Given a microarray linear model fit, compute moderated
# t-statistics, moderated F-statistic, and log-odds of
# differential expression by empirical Bayes moderation
# of the standard errors towards a common value.
topTab <- topTable(fit, coef=2, p.value=0.05, number=”inf”, sort.by=”p”)

\section{Github references}
```