NTNU – Trondheim
Norwegian University of
Science and Technology

Separating pseudo-microRNAs from true microRNAs

Frederik Klokk Holst

Master of Science in Computer Science
Submission date: June 2013
Supervisor: Pål Sætrom, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Preface

This master thesis has been completed at Laboratorisenteret at St. Olavs hospital in Trondheim spring of 2013. It has been a continuation of the work stared in [Holst, 2012].

I would like to thank my advisor Pål Sætrom for all his help and feedback throughout this semester, and the bioinformatics group for many interesting talks about various topics in the bioinformatics world. I would also like to thank Birgit Longva and Jorun Klokk Holst, for feedback and discussions that have helped in shaping the content of this thesis.

Abstract

MicroRNAs are small RNA molecules that regulate gene expression in cells. They are derived from hairpin shaped RNA transcripts, and about 50 % of microRNA genes are localized in genomic regions that are associated with cancer. There are numerous other natural occurring RNA molecules that also take shape as hairpins. Being able to distinguish between these molecules and real microRNAs is vital to understand the nature of microRNAs.

The goal of this thesis has been to construct a classifier that based on existing features is able to predict whether a hairpin shaped RNA molecule is a microRNA or a pseudo-microRNA. In addition the features in use have been analyzed to see which of these features are the most important for Microprocessor processing, and microRNA classification.

I present a classifier that is able to distinguish between real and pseudo-microRNAs with high certainty for mus musculus microRNAs. This classifier is based on feature information constructed from the output of another classifier that predicts the Microprocessor cut site of microRNAs. The features used by this classifier have been analyzed using feature elimination. Indications show that there are specific positions within the flanking regions of a microRNA substrate that are important for Drosha recognition of the substrate. Feature analysis has also been performed for the microRNA classifier, and discoveries were made that indicate that microRNAs can be distinguished from other hairpin RNAs by the fact that microRNAs have one clear cut site candidate where the other hairpin shaped RNAs might have many possible candidates. This information will hopefully further assists the search for novel microRNAs, and also to help reanalyze existing microRNAs to verify that they are in fact microRNAs.

Acronyms

A	Adenine
C	Cytosine
DGCR8	DiGeorgio Syndrome Critical Region 8
DNA	Deoxyribonucleic acid
FN	False Negative
FP	False Positive
G	Guanine
HMM	Hidden Markov Model
hsa	Homo sapiens
MFE	Minimum Free Energy
miRNA	microRNA
mmu	Mus musculus
mRNA	messenger RNA
ncRNA	non-coding RNA
nt	Nucleotide
pre-miRNA	precursor microRNA
pri-miRNA	primary microRNA
RISC	RNA induced silencing complex
RNA	Ribonucleic acid
ROC	Reciever operating curve
Se	Sensitivity
Sp	Specificity
SVM	Support Vector Machine
T	Thymine
TN	True Negative
TP	True Positive
tRNA	transfer RNA
U	Uracil

Table 1: Accronyms

Contents

List of Figures

List of Tables

Chapter 1

Introduction

MicroRNAs (miRNAs) are small non-coding RNAs of about 22 nucleotides in length [Hammell, 2010]. They have been shown to be key actors during development and other processes in cells [Erik Ladewig, 2012]. In many organisms, disruption of miRNA biogenesis by mutations lead to early embryonic death, and in the short amount of time we have known about miRNAs, the number of diseases linked to misregulation of miRNAs has increased dramatically [Hammell, 2010].

MicroRNA precursors are characterized by forming a hairpin structure. Unfortunately a large amount of similar hairpins can also be found, and being able to separate between miRNAs and other hairpins is crucial to get insight into the nature of miRNAs [Chenghai Xue and Zhang, 2005].

Human miRNAs are processed from primary miRNAs (pri-miRNA) into miRNA precursors (pre-miRNAs) by the Microprocessor complex, consisting of Drosha and DGCR8. The pre-miRNAs are then transported to the cytoplasm where it is processed by Dicer into a double stranded RNA molecule. One of the two strands ends up as the functional mature miRNA while the other is degraded. The functional strand is loaded into an RNA-induced silencing complex (RISC) before it can function as an expression regulator [Hammell, 2010].

Computational methods of prediction provide an effective strategy for identifying miRNAs where conventional genetics techniques are having problems due to the short length of miRNAs, subtle phenotypes, or low expression levels of many miRNAs [Hammell, 2010] [N. D. Mendes and Sagot, 2009]. As a result, several sophisticated tools have been developed using methods such as Random

forest, hidden markov model and Support Vector machines, but none of these have been based around the actual biogenesis of the miRNAs.

In 2007, Snorre A. Helvik et al. presented a classifier that was able to predict the $5'$ Drosha processing site in miRNA candidate hairpins [Snorre A. Helvik, 2007]. The information gathered from this classifier was in turn used to construct a classifier for distinguishing between real and pseudo miRNAs.

This thesis is based on the work done by Helvik et al., and the purpose is to continue the work started in the introduction project. Here the Drosha processing site classifier was re-implemented with a new framework, see [Holst, 2012]. Furthermore the main goal has been to implement a SVM classifier that based on a set of features first used by Helvik et al, is able to distinguish between real and pseudo miRNAs. This classifier will be tested on new data sets constructed based on findings done by Chiang et al. [H. Rosaria Chiang and Bartel, 2010].

In addition, a feature analysis for both the Microprocessor Cut site predictor and the miRNA SVM classifier will be completed.

This thesis starts by giving the reader an introduction to key concepts from biology that will be helpful to understand the remainder of the work. This introduction is found in chapter two. In chapter three, the background for this work will then be presented with a short summary of the work done by Helvik et al. along with the main findings from the introduction project, and a brief summary of existing methods for miRNA prediction. Chapter four will cover the Material and methods used in this thesis, before the results will be presented and discussed in chapter five. Chapter six presents the conclusion of this thesis. Finally chapter seven suggests work that can be undertaken in the future to further expand upon the knowledge obtained.

Chapter 2

Biology

Genes are essential for function of cells, and therefore one of the key components of life. They can be seen as the blueprint of an organism, and understanding the blueprint will help us understand the organism itself.

Some genes contain the recipes on how to make proteins (coding genes) through the protein synthesis (Section 2.7), while others have different functionality that has been harder to identify (non-coding genes). One group of non-coding genes, known as microRNA (miRNA) (Section 2.8), function by regulating protein production in cells. MicroRNAs are the focus of this thesis, and in this chapter, key concepts within the world of biology will be explained to give the reader sufficient knowledge to understand the contents of this thesis.

2.1 Genome

The genome is the complete set of hereditary genetic material of an organism. It can be seen as a cook book containing the recipes for all genes. Almost all cells in the human body contain at least one copy of this book, an exception being the red blood cells [Reece, 2010c].

The alphabet of this book is constructed from nucleotides that can be represented as the letters A (Adenine), T (Thymine), C (Cytosine) and G (Guanine). Each copy of the genome is approximately 3.2 billion nucleotides long [Reece, 2010b]. The building material of the genome is our DNA which is described in Section 2.2.

2.2 DNA

Deoxyribonucleic acid (DNA) is built up of nucleotide (nt) monomers consisting of a sugar (deoxyribose) and a phosphate group comprising the backbone together with one of four nitrogenous bases represented by the aforementioned alphabet [Reece, 2010c].

The DNA structure usually takes shape as a double stranded helix where the nucleotides are connected to the next one with a bond between the phosphate group in the first molecule and the sugar in the next molecule [Reece, 2010c]. The two strands are inversely complimentary to each other by connection between A in strand one and T in strand two, and C in strand one and G in strand two and the other way around. The bases together are held together by hydrogen bonds. The direction of the strand is described based on the orientation of the nucleotides, and is denoted as going from the 5′ to the 3′ end.

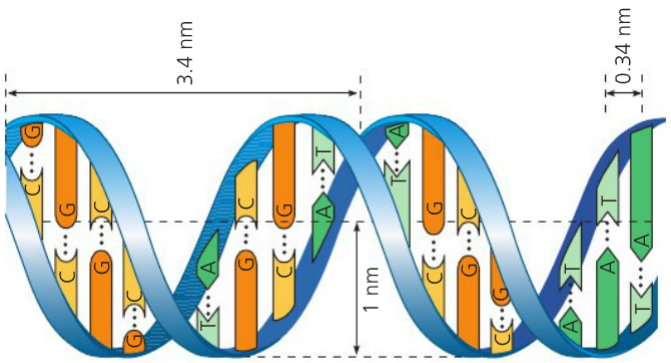
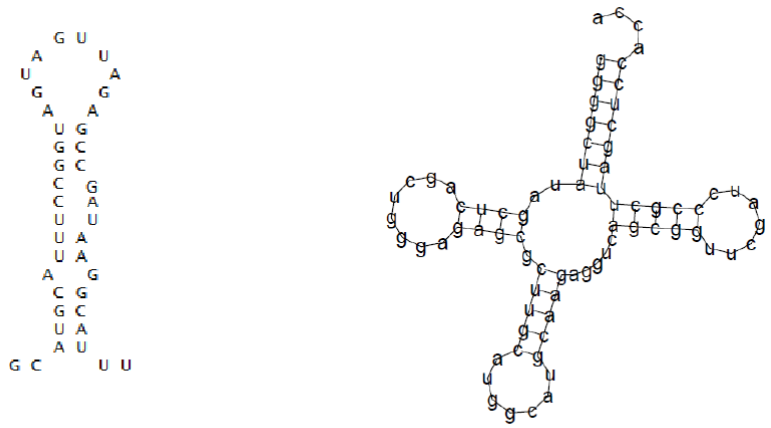


Figure 2.1: DNA helix [Reece, 2010d]

Figure 2.1 is an illustration of the DNA structure. The base complementarity provides us with two strands that are inversely identical to each other and one can be used to reconstruct the other.

2.3 RNA

Ribonucleic acid (RNA) resembles DNA, but the sugar used in RNA is different from the one used in DNA (ribose instead of deoxyribose). In addition, RNA uses a different nitrogenous base than DNA. Here T is replaced with U (Uracil) [Reece, 2010d]. The structure of RNA molecules varies a lot more than the structure of DNA molecules. Where DNA usually is structured as a double helix with two strands that run inversely to each other, RNA can take more shapes by folding and creating bonds between internal nucleotides. This structure is known as RNA secondary structure, and one of the key characteristics of microRNA is based on one of these structures [Reece, 2010d], more specifically the hairpin structure seen in Figure 2.2 a.



(a) A hairpin structure

(b) RNA secondary structure of a tRNA

Figure 2.2: RNA secondary structure examples [Matthias Hochsmann, 2003]

RNA molecules perform a variety of functions, often determined by the secondary structure. Figure 2.2 b display the secondary structure of a transfer RNA (tRNA), a molecule that is vital in the process of translating mRNA to proteins. This will be described in more detail in Section 2.7.2.

To determine the secondary structure of an RNA based on its nucleotide sequence, base pairing configurations are tested and the free energy is calculated based on information such as number of GC versus AU and GU base pairs, number of base pairs in stem regions and number of unpaired bases. The configuration that has the least amount of free energy is called the Minimum Free Energy configuration. This is often considered to be the most probable secondary structure of an RNA molecule [Nelson and Istrail,].

2.4 Gene Expression

Gene expression is the process where the information carried by a gene is used to produce a gene product. These gene products can be proteins or other functional molecules such as microRNAs.

The variation in gene expression from cell to cell is what accounts for the difference in cell function. What this means is that even though every cell with a genome have access to every recipe, only a few of the recipes are actually used. The regulation of which gene and the amount of genetic product that is produced is known as gene expression regulation. This regulation takes place in every step needed for a gene to be expressed.

This thesis is concerned with the post-transcriptional regulation, as this is where microRNAs act as a gene expression control mechanism [Jinju Han and Kim, 2006].

2.5 Amino acids

There are 20 different amino acids. Eight of these cannot be created in the human body (nine are needed by infants), meaning that they have to be ingested through the diet [Reece, 2010d]. Amino acids are linked together by peptide bonds, forming a chain of amino acids known as a polypeptide chain. These chains are then processed to give the protein the correct structure, and in turn function, which is determined by amino acid composition and polypeptide structure [Reece, 2010a].

2.6 Proteins

Proteins are large molecules constructed from chains of amino acids. They are instrumental in almost every process in our cells, and make up nearly 50% of the dry mass of most cells [Reece, 2010a]. Proteins are needed in the correct amounts for a cell to function properly, and different proteins are expressed in different cell types. Proteins act as building blocks, support structure, antibodies, hormones, enzymes, mediators for cell response, and more.

2.7 Protein synthesis

Protein synthesis is the process used to translate a gene into a protein. There are numerous molecules involved in this process. A protein coding gene goes through several steps to be translated into a functional protein, and the steps of transcription and translation will be explained, as they are relevant to this thesis. Figure 2.3 illustrates the different steps of the protein synthesis and molecules involved.

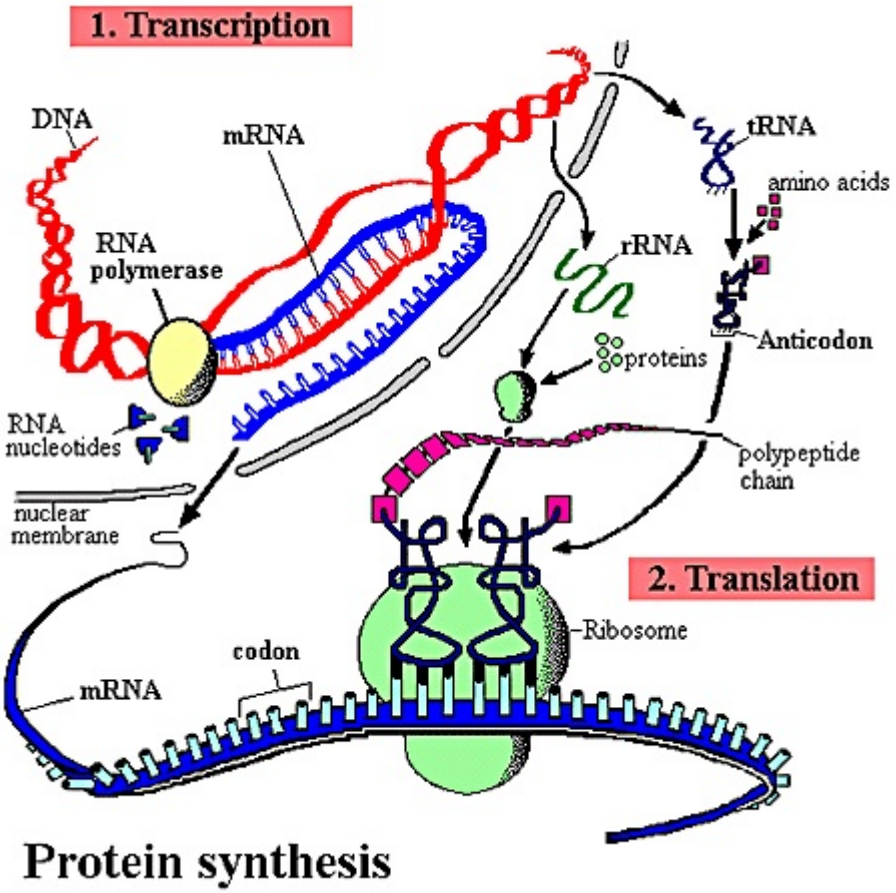


Figure 2.3: Protein synthesis [excellence, 2009]

2.7.1 Transcription

Simply put, the transcription step is the process where a gene is copied into an RNA molecule. To do this, the DNA strands need to be unwound, making the DNA available for transcription. Separated from the nontemplate strand, the template strand can be copied before the DNA is rewound again. This happens through three steps, initiation, elongation, and termination [Reece, 2010c]. Every RNA molecule is processed through these steps, and they all happen in the nucleus. But the RNA processing differs between RNA molecules. Figure 2.4 shows the process of transcription.

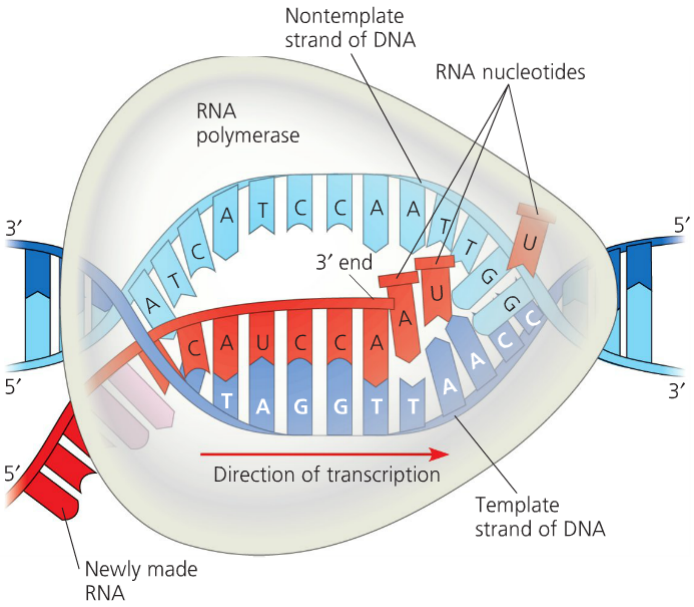


Figure 2.4: Illustration of the transcription process [Reece, 2010c]

2.7.2 Translation

Every amino acid in the transcribed gene can be described by a set of 3 nucleotides. These three nucleotides are known as a codon. The process of translating the mRNA molecule into a polypeptide chain is done by matching each codon with an anti codon that is found at the base of the tRNA. The tRNAs transport amino acids from the pool of amino acids present in the cytoplasm, and when the codon and anti-codons are matched, the correct amino acid is attached onto a growing polypeptide chain (a chain of amino acids linked with peptide bonds) by the ribosomal unit. See Figure 2.5.

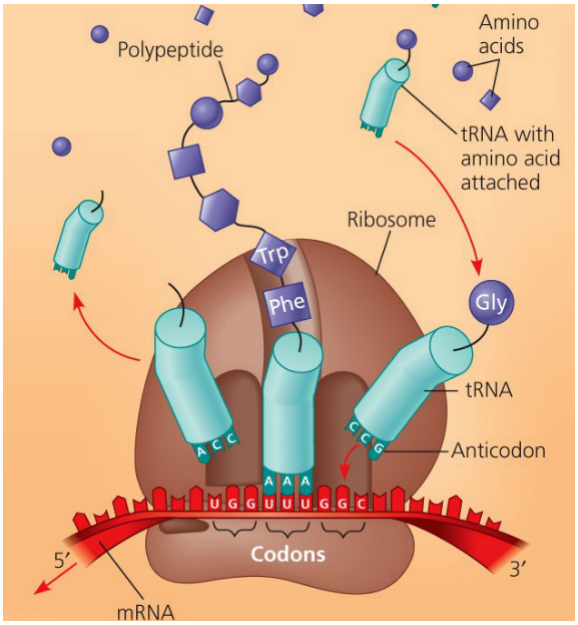


Figure 2.5: Illustration of the translation process [Reece, 2010c]

2.8 MicroRNA

MicroRNAs (miRNA) are short non-coding molecules that act as a part of the regulatory network in animals and plants. They affect the gene expression of target proteins through post transcriptional gene silencing.

More than one-third of our genes have been predicted to be regulated by miRNA, and the combination of miRNAs in each cell therefore affect the use of thousands of mRNAs [V. Narry Kim and Siomi, 2009]. MicroRNA molecules bind to their target mRNAs and either repress translation or destabilize the mRNA molecule [Jinju Han and Kim, 2006]. This interference leads to a down regulation of protein production of the target mRNA.

MicroRNAs go through a process of maturation to yield a final mature single stranded molecule. Figure 2.6 shows the microRNA biogenesis.

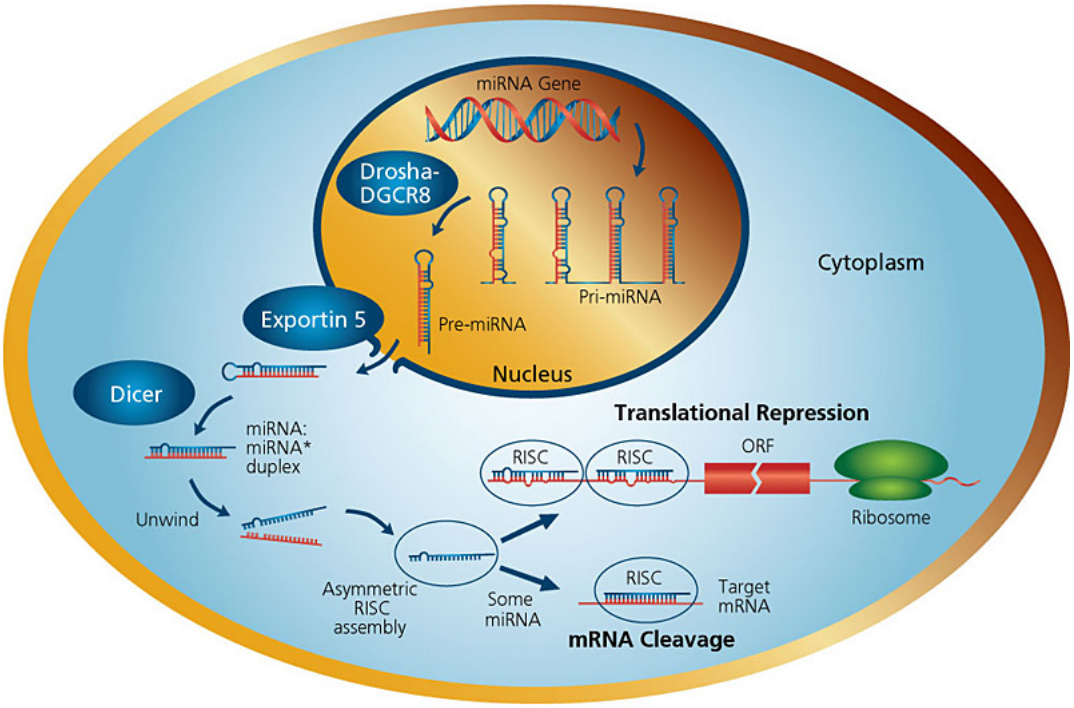


Figure 2.6: Biogenesis of microRNA [Mraz, 2012]

The transcribed molecule that is the precursor for mature miRNAs is known as a primary miRNA (pri-miRNAs). Primary miRNAs are usually several kilobases

long and contain one or multiple local stem loops.

The first step of maturation is performed by what is called a Microprocessor complex in animals. This complex consists of a Drosha protein and a DiGeorge syndrome critical region (DGCR8) protein. The pri-miRNA is cropped by the Microprocessor complex to a hairpin-shaped molecule with a 60-70 nucleotide stem loop with a 2 nt $3'$ overhang. This is called a miRNA precursor or pre-miRNA [Snorre A. Helvik, 2007] [Jinju Han and Kim, 2006] [Bartel, 2004]. Research show that the secondary structure of a pri-miRNA is of more importance than the primary sequence for substrate recognition by Drosha [Cullen, 2004].

The introduction to this thesis was to create a classifier that was able to predict where the microprocessor complex cut to yield the pre-miRNA product, and data from this step is further used as features in the final classifier. Read more in Section 4.3.

The pre-miRNA is transported from the nucleus to the cytoplasm by Exportin 5 that recognize the double stranded stem of the hairpin along with the overhang [V. Narry Kim and Siomi, 2009].

In the cytoplasm, the pre-miRNA is processed by a protein called Dicer. Dicer cleaves the pre-miRNA near the terminal loop, producing a double stranded molecule of approximately 22 nucleotides in length.

Finally the RNA duplex is loaded onto an Argonaute protein to produce a RNA induced silencing complex (RISC). Only one of the strands from the duplex is used (the mature miRNA) while the other is degraded [V. Narry Kim and Siomi, 2009].

It is worth mentioning that there are alternate ways to yield mature miRNA molecules. One of which, bypass Drosha processing. This is the most prevalent of these pathways, giving rise to a group referred to as mirtrons. These serve as pre-miRNA mimics and are not processed by the Microprocessor complex [Erik Ladewig, 2012]. There are alternate pathways for other stages of the miRNA biogenesis as well, but the miRNAs obtained through these pathways will not be the focus of this thesis.

MicroRNA precursors are found to have a lower MFE in general than other short non-coding RNAs (ncRNA), and thus MFE can be used to aid us in distinguishing between miRNAs and other ncRNAs [Jiandong Ding, 2010]. There are also variations in the free energy throughout the stem of the miRNA created by internal

loops at specific positions [Jinju Han and Kim, 2006].

Secondary structure features are essential to the recognition of miRNAs as they are recognized by the various components in the biogenesis pathway and therefore are essential for the processing of substrate. Loop size, and the flanking regions of the loop has been shown to be critical in determining the cleavage site of the Microprocessor complex [Jinju Han and Kim, 2006].

Recently V. C. Auyeuyng et al. concluded that secondary structure on its own is inadequate to specify miNRA hairpins [Vincent C. Auyeung and Batel, 2013]. They came up with several primary-sequence features that also contribute to the correct miRNA processing in human cells.

Chapter 3

Background

This chapter will in short introduce the basis for this project with foundation in the work done by Helvik et al [Snorre A. Helvik, 2007]. It will then go through the results obtained during the introduction project to this thesis, which was continued and further used this semester. Finally existing methods of miRNA classification will be discussed.

3.1 Basis for this project

This thesis is based on the work done by Helvik et al. presented in [Snorre A. Helvik, 2007]. In short, a Support Vector Machine (SVM) classifier that is able to predict the position the Microprocessor processing site in microRNAs is described. The results from this classifier was in turn used as input for another classifier that at the time gave more accurate results for finding unconserved miRNAs [Snorre A. Helvik, 2007]. Much of the frame-work from this paper has been reused in the thesis. The main changes are that the data sets and the frame-work used for the SVM classifier has been replaced.

3.1.1 Data Sets

The data sets used by Helvik et al. consisted of the entire set of human miRNA sequences found in miRBase 8.0 (332 miRNAs) and the new human miRNA sequences from 8.1 (130 miRNAs). In addition an algorithm was used to extract hairpins from the human genome.

Processing Site Data Set

The data set used for the microprocessor SVM was constructed by finding candidate processing sites for each miRNA that fulfilled a set of secondary structure requirements. The positive data set consisted of all candidates with the same cut site as listed in miRBase.

miRNA SVM Data Set

The data set used for the miRNA SVM was a set of 3000 random hairpins as negative data set, and the miRNAs from miRBase 8.0 as the positive set.

3.1.2 Features

The features used by Helvik et al. are derived both from structure and sequence, but they will not be describes further here as they can be found in Section 4.3. They are the same as the ones used in this thesis, and the methods for extracting features have also been used to some extent.

3.2 Previous Work

Before Christmas, the introduction project was to reimplement a classifier that was able to predict the Drosha processing site for miRNAs by using a new frame-work for Support Vector Machines. This work was completed using the *PyML* machine learning frame-work instead of the *gist* library which was used by Helvik et al.

More information about this project can be found in [Holst, 2012]. The outcome of this project was a classifier that was able to predict a processing site within 2 nt from the real site in 69,3% for a subset of the verified miRNAs [Holst, 2012].

Various configurations ranging from different types of normalization to feature value area scaling was tested to achieve the best possible performance at this point. Figure 3.1 shows the optimal performance with features scaled to [-1, +1] range and l2 normalization for the data sets.

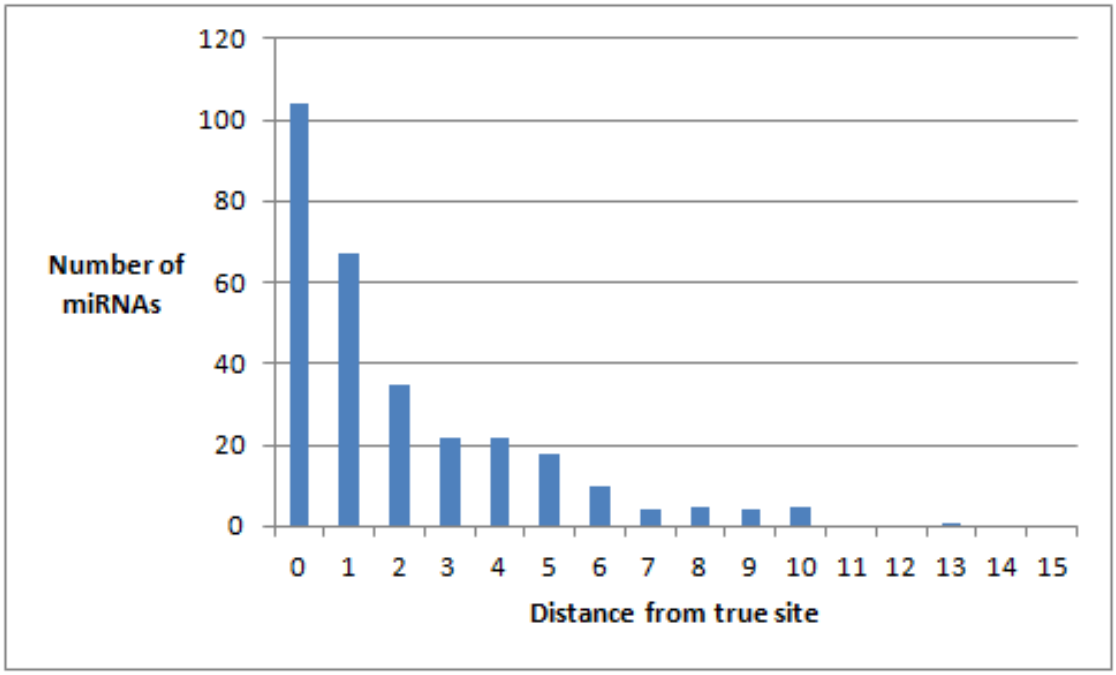


Figure 3.1: The best results gained from cross validation of verified mmu

The data set used for this testing was a set of verified mus musculus (mmu) miRNAs. More information about the data sets used can be found in Section 4.5.

The verified data set was also used to train a SVM that in turn was used to predict the cut sites for a group a miRNAs that were classified as false (see Section 4.5). As can be seen from Figure 3.2, the false data set was harder to classify. This indicate that there are distinguishable differences between the two data sets suggesting that the two data sets will be useful for training a classifier that is able to separate between real and pseudo-miRNAs.

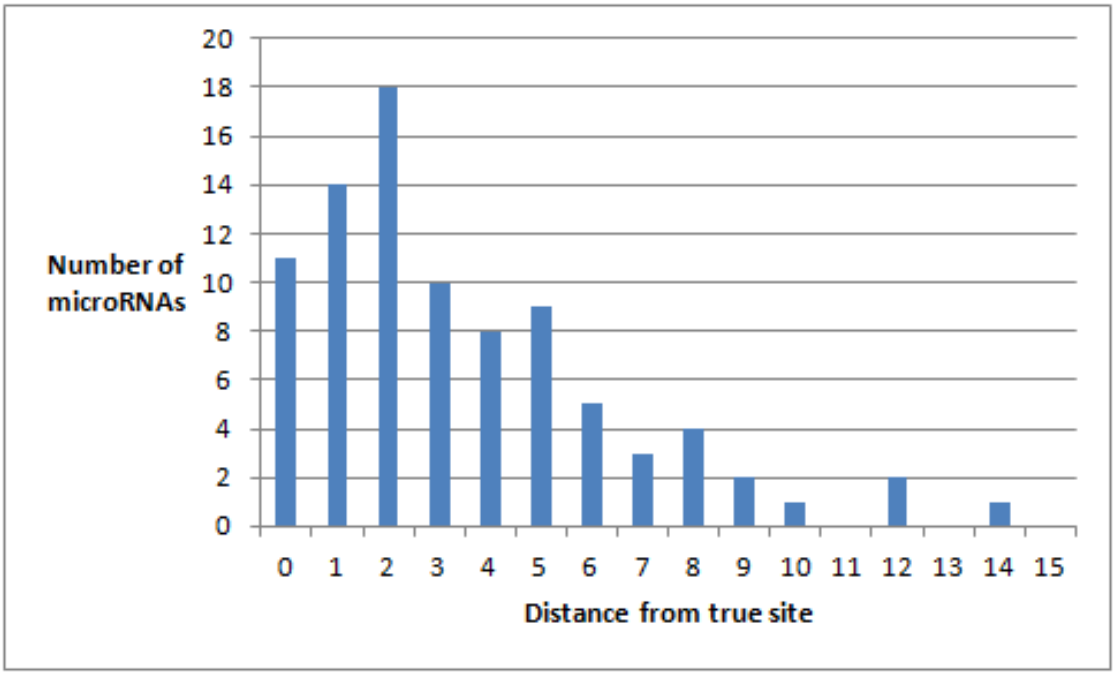


Figure 3.2: Result from training on verified and testing on failed

During this introduction project, PyML components were identified and input formats were constructed. This work has been further improved upon and used as the basis for this master thesis.

3.3 Existing Methods

Since the first discovery of miRNAs, several methods for miRNA gene finding has been developed using various sets of features and tools. Common for most of these is that they rely extensively on the conservation of miRNA between species [N. D. Mendes and Sagot, 2009].

The various tools developed use different approaches to find new miRNAs. N. D. Mendes et al. separate them into the groups *filter-based* methods, *machine learning* methods, *target centered methods*, *homology-based* methods, and finally *Mixed* approaches using a mix of the aforementioned methods [N. D. Mendes and Sagot, 2009].

Filter-based methods use numerous criterions for filtering out good stem-loop

candidates based on conservation and various features.

The *Machine learning* based methods use positive data sets of previously known miRNAs and negative sets of stem-loops that are not miRNAs to attempt to generalize based on the information obtained from the data sets.

Target-centered methods use a set of assumed miRNA targets from conservation studies to search for new miRNAs.

Homology-based methods use known pre-miRNAs and attempt to find similar stem-loops this way.

The SVM method used in this thesis belong to the machine learning group, therefore this group will be described in more depth. I will take a look at a few other methods of predicting miRNAs as this is done using several methods from the machine learning toolbox.

ProMIR attempts to predict homologs of known miRNAs. This is done using a co-learning method based on paired Hidden Markov Model (HMM) [Jin-Wu Nam, 2005]. ProMIR use minimum free energy(MFE) and structural characteristics such as loop size and stem length as features for filtering.

MiPred distinguishes between real and pseudo miRNAs by using random forest (RF). Random forest selects features from the set at random at each node when growing a tree-structured classifier. MiPred uses MFE in addition to nucleotide bonds combined in triplets as structural features [Peng Jiang and Lu, 2007].

MiRenSVM is a SVM-based approach to be used for detection of real miRNA precursors. The features used by miRenSVM are derived from secondary structure and thermodynamic properties based around the difference between miRNA MFE and the MFE of other small non-coding RNAs [Jiandong Ding, 2010].

Chapter 4

Material and Methods

This chapter will go through the tools, data and methods used to obtain the results from this thesis. First I will explain the support vector machine, why it is used and how it works. Then the *ROCR* package is mentioned as it was invaluable to visualize the results obtained. The features used are described in detail to ensure that they can be reproduced. The method of feature elimination is briefly described before the data sets are introduced. Performance measures are explained. Cross validation and normalization is mentioned before a short introduction to *PyML* is given.

4.1 Support Vector Machines

As previously mentioned, the purpose of this project is to create a classifier that can distinguish between true miRNAs and pseudo-miRNAs. The support vector machine has been chosen as a means to achieve this goal.

The Support Vector Machine (SVM) is widely used in the field of computational biology and bioinformatics [Asa Ben-Hur,], and the popularity is due to the ability to deal with high-dimensional and large data sets. SVMs are also flexible for modeling a variety of data sources. Since it is a binary classifier with the positive traits mentioned, it is well suited to deal with the task at hand. This is validated by the fact that SVMs have been a popular framework for use in miRNA classification, used to learn miRNA characteristics [N. D. Mendes and Sagot, 2009].

A support vector machine is a binary classifier that takes data as input in form of numeric vectors. The numeric vector values are plotted in an x-dimensional space

and a prediction is made based on which side of the maximum-margin hyperplane that data point resides. The maximum-margin hyperplane is found by using a training set with labeled input. Kernel functions are used to map the original features to a feature space of higher dimension where an optimal hyperplane separation can be achieved [Peng Jiang and Lu, 2007].

We define the margin as the distance from the hyper plane to the closest feature vectors, known as support vectors. Of the possible hyperplanes, the one with the maximum margin is chosen, hence the name maximum-margin hyper plane.

The hyperplane is described by (**w**,b) where the vector **w** is known as the *weight vector*, and b, is the bias that translate the hyperplane away from the origin [Asa Ben-Hur,]. A data point in form of feature vector is placed on one of the sides of the hyper plane based on the sign of the decision function described by Equation 4.1.

$$\{f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b\} \tag{4.1}$$

The information for all miRNA candidates is contained in $\mathbf{x}$ and where $\mathbf{x}_i$ denote the i^{th} vector. The dot plot of $\mathbf{w}$ and $\mathbf{x}$ is calculated and b is added. Based on the score the points will be separated as the illustration in Figure 4.1 shows.

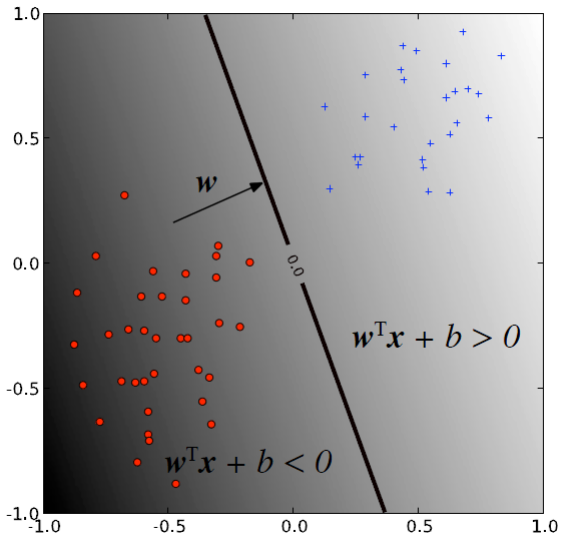


Figure 4.1: A linear SVM example [Asa Ben-Hur, 2008]

It is not always the case that all elements can be correctly predicted to be in the correct group. To account for this, the soft margin is introduced to the SVM. The

soft margin allows a number of the feature vectors to end up on the "wrong" side of the hyper plane.

The SVM requires the information available to be presented in an understandable format as numerical vectors, so instead of feeding the raw information available about each miRNA candidate to the support vector machine we need to transform this data into numerical values. This information layer is referred to as "features". The features used in this project are described in Section 4.3. In general the feature vectors need to be described as numerical values. Chih-Wei Hsu et al. recommends that the feature values are scaled down to the range [-1,+1] or [0,+1] in order to avoid features with large numerical range to become dominating [Chih-Wei Hsu and Lin,].

To use the SVM, we need to train it first. This process finds the maximum-margin hyper plane [Asa Ben-Hur,]. The process of training the SVM is done by supplying it with a data set of feature vectors that are already labeled +1 or -1 depending on which group they belong to.

The trained SVM can in turn be used to classify unknown data sets. In this project a python library called *PyML* that contains an implementation of the SVM has been used. More information on*PyML* can be found in Section 4.9

4.2 R and ROCR

R is a language and environment that is used for statistical computing, data manipulation and graphical display [R-project, 2013]. It is designed to be highly extensible, and produce high quality plots. *R* runs on windows, UNIX platforms and MacOS.

ROCR is a package that is able to evaluate and visualize the performance of classifiers using various scoring schemes. It uses the statistical language *R* to produce more than 25 performance measures that can be combined in two-dimensional performance curves [Tobias Sing, 2005].

In this thesis *R* has been used with the *ROCR* library to plot ROC graphs based on the decision function outputs from various data set tests.

4.3 Features

In this thesis there are two sets of features that have been used (see Table 4.2). For the processing site prediction, feature 1 through 8, and 11 were used. A short description of each feature and representation will now be given in Section 4.3.1. For the miRNA prediction, a set of new features was used. A description of how each of these features were constructed will be given in Section 4.3.2

Figure 4.2 displays two illustrations of miRNA hairpins. Figure 4.2 (A) shows the processing site candidates of a predicted miRNA secondary structure. The true processing site as listed in miRBase is displayed in bold.

Figure 4.2 (B) displays the position of the regions where structural information for feature 3 and 7, in addition to the features that are based on feature 3 and 7 has been obtained. Feature 3 is derived from the 24 nucleotide long region marked with gray font and lines, while feature 7 is derived from the 50 nt long flanking region marked with black font and lines.

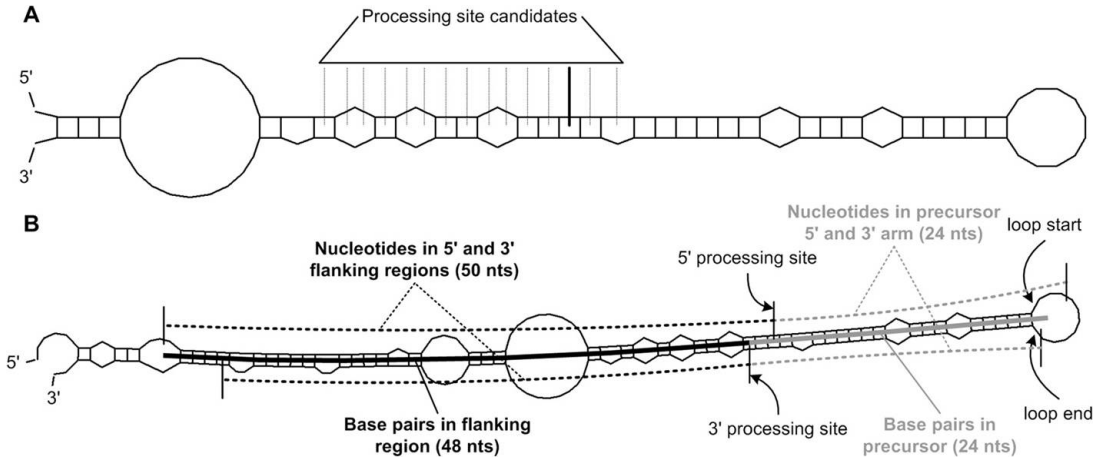


Figure 4.2: MicroRNA feature figure [Snorre A. Helvik, 2007]

4.3.1 MicroRNA Processing Site Features

Feature one consists of two values, one represents the loop size of the precursor, and the other the length of the precursor.

Feature two is the distance from the 5′ processing site to the start of the loop.

Feature 3 contains position specific information about the bases in the 24 nucleotide precursor base. Each possibility is described using binary values as described in Table 4.3.1

- Adenine = [1,0,0,0]
- Cytosine = [0,1,0,0]
- Guanine = [0,0,1,0]
- Uracil = [0,0,0,1]

Table 4.1: Binary representation of the nucleotide bases.

Feature 4 contains base-pair information for each position in the region. This information is represented with the values 0 if neither of the two nucleotides at the specific position contains a bond to another nucleotide, 0.5 if only one of the nucleotides contain a bond, and 1 if both contain a bond. Which nucleotide that the bond is connected to is not being considered.

Feature 5 consists of the base frequencies in feature 3, one value for each base in the 5′ and 3′ section.

Feature 6 contains the total number of base pairs in feature 4.

Feature 7 describes position specific nucleotide information for the 50 nucleotide flanking region, totaling a 100 positions, 50 for each base in the 5′ and 3′ region respectively. The nucleotieds are represented as described in Table 4.3.1.

Feature 8 contains the base pair information for feature 7 presented as described in feature 4.

Feature 11 contains the number of base pairs in the feature 7 region.

To ensure that none of the features are weighted differently, they are scaled down to [0,1] or [-1,1] range. This is done by finding the largest feature value for the specific position, and dividing all values in that position for all the vectors by that

value. Features 1, 2, 5, 6, and 11 have been scaled down.

4.3.2 MicroRNA features

For the miRNA classification, some features constructed from the information gathered during the processing site classification have been added.

Feature 12, the number of potential processing sites, is simply the number of cut site candidates produced for each miRNA.

The score of the best processing site (feature 13) is found by getting the highest scoring candidate based on the decision function. This decision function is set as the best score.

Feature 14, the average processing site score, is calculated from the decision function of all processing site candidates of the given miRNA.

The standard deviation in feature 15 is calculated based on the decision function for every processing site candidate of the miRNA.

Feature 16 is calculated simply by subtracting feature 14 from feature 13.

The distance between the three top scoring processing sites (feature 17) is the distance from top one to top two plus distance from top two to top three.

Feature 18 counts the number of local maximums, where the decision function score is above a threshold (-0,5 in this case) and the decision function score is higher than one of the neighbors.

All features are summarized in Table 4.2

Features used by MicroRNAProcessingSiteSVM	
ID	Explanation
1	Precursor length and loop size
2	Distance from 5' processing site to start of the loop
3	Nucleotide occurrences at each position in the 24 nt at the precursor base
4	Base-pair information of each nucleotide for the 24 nt at the precursor base
5	Nucleotide frequencies in the two regions in 3
6	Total number of base pairs in region 4
7	Nucleotide occurrences at each position in the 50 nt 5' and 3' flanking regions
8	Base-pair information of each nucleotide for the 48 nt in the flanking region outside the precursor
9	Nucleotide frequencies in the two regions in 7
10	Total number of base pairs for the 15 nt immediately flanking the precursor
11	Total number of base pairs in the region in 8
	Additional Features used by MicroRNASVM
12	Number of potential processing sites
13	Score of the best processing site
14	Average score for all potential processing sites
15	Standard deviation for all potential processing sites
16	Difference between feature 13 and 14
17	Distance between the three top-scoring processing sites
18	Number of local maximums in the processing site score distribution

Table 4.2: Features used in this project

4.4 Feature elimination

To gain information on how useful each feature is for the SVM classification, a feature elimination process can be conducted through various methods such as Recursive feature elimination.

In this thesis, the information about the importance of each feature has been studied by removing one feature. For every run a feature is left out, then the classifier has been retrained and the performance has been compared to the classifier where every feature is still in use. This process has been repeated for every feature, and has been completed both for the cut site classifier and the miRNA classifier.

4.5 Data Sets

Every miRNA used in this project has been downloaded from miRBase. miRBase is the repository for miRNA sequences that is most widely used [William Ritchie and Rasko, 2012]. The miRNA SVM needed information from the microprocessor cut site prediction to construct the complete data sets. How the different data sets were constructed is described in the coming sections.

4.5.1 Microprocessor data sets

The Microprocessor classifier was constructed to predict where the Microprocessor complex would cut the pri-miRNA to create the pre-miRNA. For each candidate miRNA, every possible cut site vector with an attached cut site ID was generated. The cut site that is registered by miRBase as the correct cut site was added to the positive data set while the others were added to the negative data sets.

4.5.2 MicroRNA SVM data sets

As described in the feature section, the feature information needed for the vectors in the miRNA classification is based on the decision function values extracted from the cut site prediction. For each miRNA a cut site prediction is extracted, and the decision function values for the candidate cut sites are used to determine the feature vector to be used. This is the feature vector of the cut site with the highest

scoring decision function. The additional features are then added to this feature vector to create the miRNA vector that will be used for miRNA classification.

4.5.3 MicroRNAs used

In this project the miRNAs used are from the species homo sapiens (hsa), and mus musculus (mmu). Table 4.3 describes the various data sets used in the miRNA SVM part of this thesis. Appendix B contains the miRNA id for every miRNA along with the fold it is put into and the label.

Chiang et al. completed an experimental evaluation of novel and previously annotated miRNA genes for mmu. The results of this work can be found in [H. Rosaria Chiang and Bartel, 2010]. One outcome was a list that presented miRNAs that were verified as true miRNAs.

The different data sets are all subsets based on either all mmu miRNAs or hsa miRNAs. *Mmu all* and *hsa all* are the complete sets of miRNAs from miRBase that yielded precursors.

In this thesis the miRNAs from the list of verified mmu miRNAs that were also present in *mmu all* from miRBase comprise the data set known as *verified data set*. Chiang et al. also found several previous annotated miRNAs that failed to fulfill the criteria for a miRNA. The subset of these that were also present in *mmu all* make up the *failed data set*.

To generate a decision function score for the MMU miRNAs that are not covered by the *verified data set* or *failed data set*, another data set was constructed by taking the precursors from *mmu all* that is not present in either set and put these into the *remaining MMU data set*. An equivalent data set was constructed for hsa giving us *remaining HSA data set*.

The training set for the miRNA SVM comprise the *verified data set* and the *failed data set* as positive and negative data vectors respectively.

A hsa test set was generated by looking at the miRNAs from each fold when the training set had been divided based on family. Each miRNA family was identified based on the mmu miRNAs in that specific fold, then in turn the hsa miRNAs from the identified families were collected and put into the corresponding fold in the training set giving us the *family in same fold HSA data set*.

A complete list of data sets used in this project can be found in Table 4.3

Data set	**Size**	**Description**
mmu all	654	Every mmu miRNA from miRBase that yielded a precursor
failed data set	133	mmu miRNAs from mmu all found in list of failed miRNAs
verified data set	475	mmu miRNAs from mmu found in list of verified miRNAs
correctly predicted data set	184	mmu miRNAs from verified data set where the predicted cut site matched the one registered in miRBase
remaining mmu	243	mmu miRNAs from mmu all that are neither in verified data set nor failed data set
hsa all	1594	Every hsa miRNA from miRBase that yielded a precursor
family in same fold HSA data set	353	hsa miRNAs from mmu families found when verified and failed data set were separated into folds with family in same fold
remaining HSA	1241	hsa miRNAs that is not in family in same fold HSA data set, but in hsa all

Table 4.3: Data set size and description

4.6 Performance Measure

The performance of the two classifiers constructed in this project has to be measured in different ways as a result of the input in the two stages.

4.6.1 Performance measure for microRNA processing Site SVM

The input of the average microRNA contains approximately 15 candidate processing sites. Only one of these candidates gets the status of being the correctly predicted one, but since the candidates close to the real cut site can be alternative cut sites we need a performance measure that account for the distance from the real cut site.

The way the performance of the classifier is being displayed is by the use of a bar graph. For each microRNA, the best candidate, meaning the candidate with the highest scoring decision function, is registered. The cut site for this candidate is extracted and compared with the correct cut site. The distance is calculated, and based on this distance each microRNA is plotted in the bar graph as can be seen in Figure 4.3

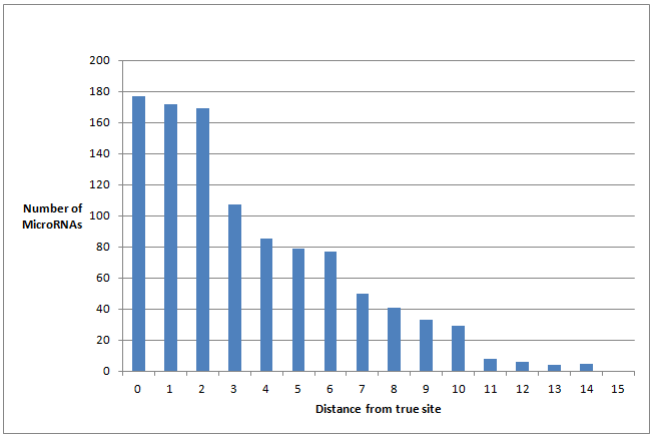


Figure 4.3: Bar graph displaying the performance of a miRNA processing site classifier

4.6.2 Performance Measure Feature Elimination cut site prediction

To evaluate the change in performance when removing features, the distance from the true site has been calculated as described. This distance distribution has been used to generate a cumulative frequency graph.

In a cumulative frequency graph each successive category is equal to the total of the previous categories. Figure 4.4 displays the average distance distribution from running feature evaluation on the verified data set.

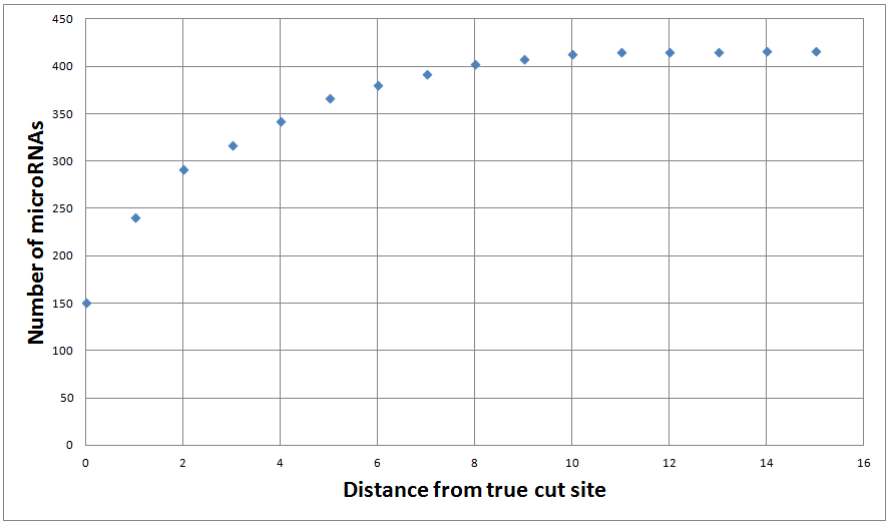


Figure 4.4: Cumulative frequency graph displaying the average performance of a miRNA cut site feature elimination

4.6.3 Performance for the microRNA SVM

Predicting whether a candidate is a miRNA or not, is a classic binary classifier problem where the solution is either true or false. Based on this a miRNA can either be true positive (TP), false positive (FP), true negative (TN), or false negative (FN). The number for each of these four categories have been calculated with a decision function threshold of 0. If a decision function scored above 0, it was considered a positive, otherwise it was considered a negative prediction.

Based on these four categories, Sensitivity (Se) can be calculated using Equation 4.2 and Specificity (Sp) can be calculated using the Equation 4.3.

$$Se = \frac{TP}{TP + FP} \tag{4.2}$$

$$Se = \frac{TN}{TN + FN} \tag{4.3}$$

The performance of miRNA however, has been measured used receiver operating characteristics (ROC) graphs. ROC graphs are used for visualizing and evaluating classifier performance. They provide more details of classification performance than other measures such as accuracy and error rate [Fawcett, 2006]. ROC graphs have seen an increase in use for computational performance, and is seen as well suited for use in machine learning [Fawcett, 2006]. Figure 4.5 shows a ROC graph produced using the ROCR package for R.

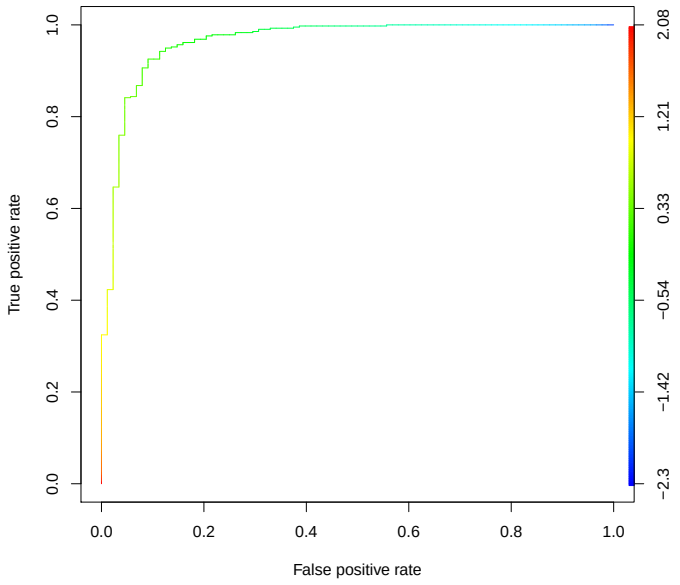


Figure 4.5: A ROC graph created using ROCR

The ROC curve plot the performance of a binary classifier based on the true positive rate (Se) and false positive rate (Sp) when the decision function threshold is varied.

4.7 K-fold Cross Validation

K-fold cross validation is used to measure the predictive performance of a classifier, and can be used to test how the SVM generalizes to unknown data sets. This is done by working with a known data set where the correct predictions are already known. The data set is divided into k sets of fairly equal size. The training set comprise k-1 sets, while the test set is the remaining set that was left out. The classifier is trained on the training set and tested on the test set. Then the solution, and a performance measure for the fold is calculated. This process is repeated until all k sets have been used as the test set. Figure 4.6 illustrate the cross validation process.

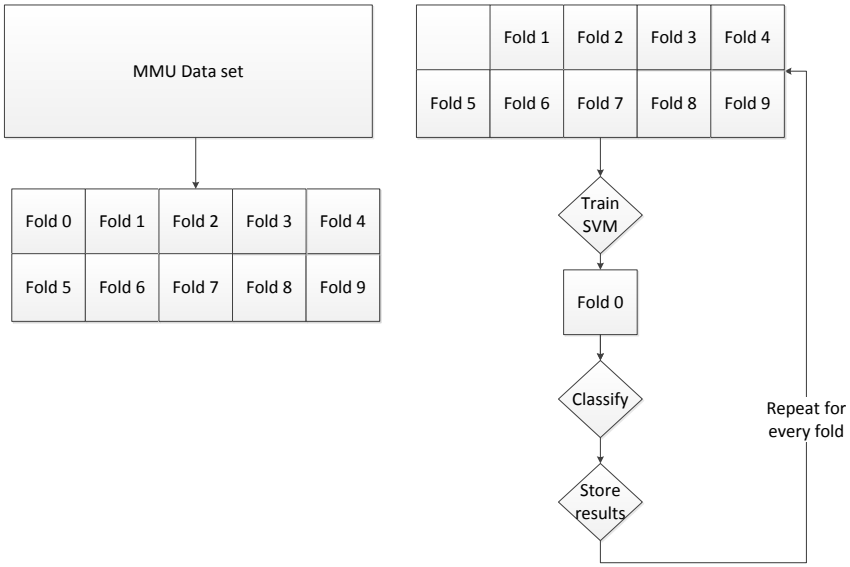


Figure 4.6: Illustration of 10 fold cross validation

For the work with miRNAs, a random division can introduce bias to the results as some miRNAs belong to the same family. What this means is that they have evolved from the same sequence and thus share traits [Snorre A. Helvik, 2007]. The folds have been divided in such a way that miRNAs from the same family end up in the same fold.

4.8 Normalization

The data sets used for the microprocessor cut site classification is unbalanced due to the number of possible cut sites. Since we have 15 possibilities and only one correct cut site, the negative data set is a lot larger than the positive one. To counter this imbalance the data sets have been normalized using built in data set normalization from the PyML package. The normalization is done simply by using *data*.normalize(norm). Where *data* is the data set to be normalized. Norm can be either 1 or 2 resulting in either L1 or L2 normalization [Ben-Hur, b].

It is also possible to normalize the kernel, but this has not been done in this project.

4.9 PyML

PyML is an object oriented machine learning framework supported for Unix/Linux and Mac OS-X [Ben-Hur, b]. It is written in python, and contains an implementation of SVM that is used in this thesis to construct the two classifiers.

The SVM in PyML needs the data to be presented in one of several data containers provided by PyML. In this thesis the *SparseDataSet* container has been used. The input format for each information vector has to follow the format presented in Table 4.4 in order to be accepted by the data containers in PyML.

[id,]	label	feature1:value	feature2:value	feature3:value	feature4:value	....

Table 4.4: Input format for SparseDataSet container labeled

The id tag is optional, but is used in this thesis to validate that the decision function comes from the correct candidate. For known data sets, the label is either +1 or -1 depending on if the vector is positive or not. For unknown data sets the label omitted.

The kernels in PyML are attached with the constructed data set, and by default a linear kernel is used [Ben-Hur, b]. Other kernels can be attached if desired but this has not been done in this thesis.

In this thesis, PyML implementation of the SVM has been used to construct the classifier that generate decision functions and also other useful information

contained in a result object. Table 4.5 summarizes the key elements used, and how they were used.

The outcome	What to use	How it is used
Creating a SVM	SVM()	newSVM = SVM()
Training a SVM	train()	newSVM.train(knownDataSet)
Using created SVM to classify data	test()	result = newSVM.test(unknowndata)
Plot ROC	plotROC()	result.plotROC()

Table 4.5: How to construct, train, and use a SVM

In order to be able to plot the ROC curve, *matplotlib* must be installed on your system. For more information about PyML see [Ben-Hur, a].

Chapter 5

Results and Discussion

This thesis consists of several steps with results for every step. In this chapter the results from each step will be presented and discussed in the order they have been completed, starting with the additional results gained from the processing site prediction. The results from feature elimination for the processing site classifier will then be presented, after which the miRNA classifier will be discussed. Finally the feature elimination results for the miRNA classifier will be presented.

5.1 MicroRNA processing site SVM results

As described in Section 3.2 a *PyML* classifier was implemented to predict the cut site of a miRNA. During the spring semester, this work has been further adapted to accommodate for more post processing of the results. This has made it possible to create ROC curves for the two data set runs described.

Figure 5.1 displays the ROC curve for a 10-fold cross validation run on the *verified data set*. The performance here may turn out to be better than the ROC curve

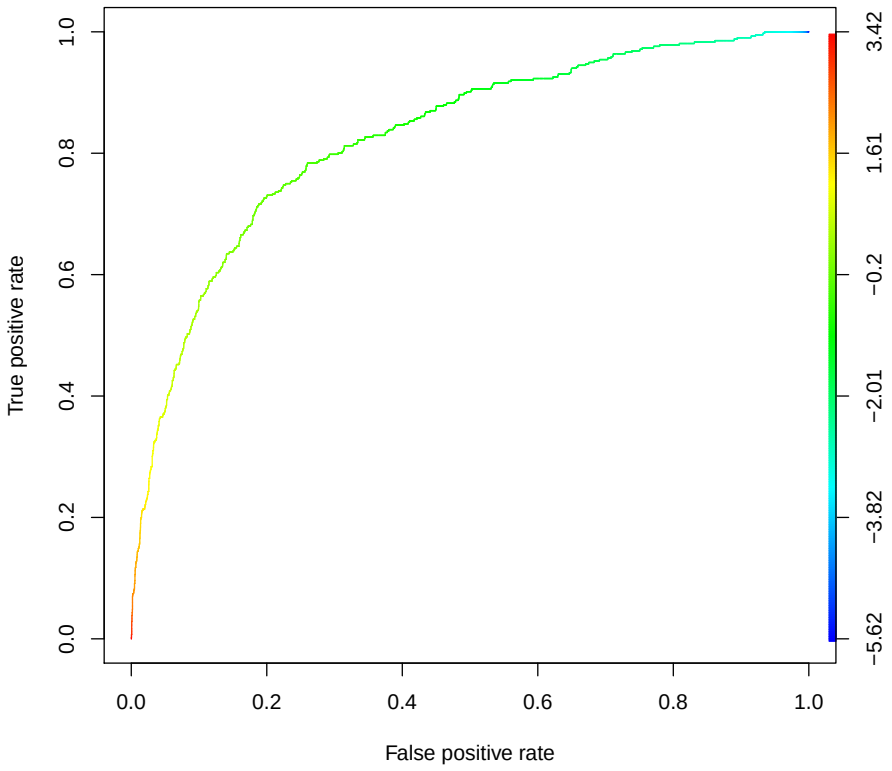


Figure 5.1: ROC plot for cross validation on the verified data set.

show. This is due to the theory that there may be more than one possible cut site. These alternative cut sites are in that case positioned close to the real cut site listed by miRBase. This is not accounted for when using this way of displaying the results, but it does a good job displaying the performance in finding the correct site which is useful when comparing the data sets.

Figure A.5 shows the ROC plot for an SVM trained on the verified data set and used to classify the *failed data set*.

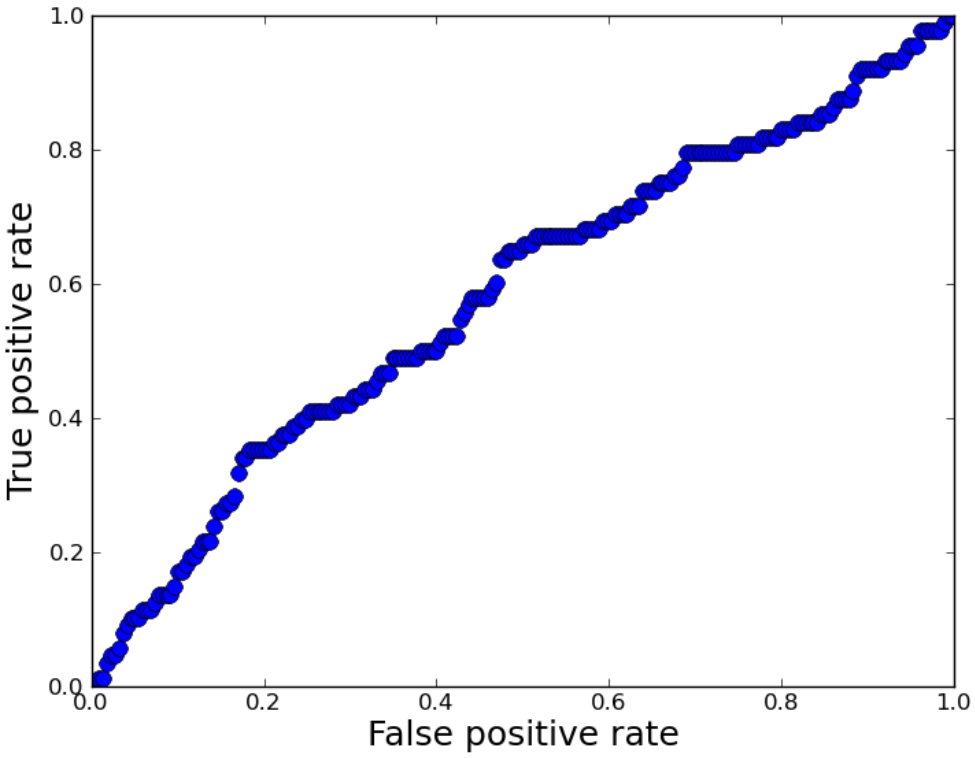


Figure 5.2: ROC plot for a SVM trained on the verified data set and tested on the failed data set.

These two ROC plots further support the theory that there are distinct differences between the miRNAs in the *verified data set* and the *failed data set*. And that these differences can be identified through the features generated for each candidate. We can see this because the classifier is able to predict the correct Microprocessor cut site with some certainty for the the *verified data set*, but is picking the cut site at random for the *failed data set*. This is the same pattern that was observed in the last project.

The output from the cut site prediction is a PyML result object and a result map containing the miRNA id, the decision function score, and a label that is either 0 or 1 depending on which data set the miRNA resides from. In addition the id map contains the miRNA vector of the candidate with the highest scoring

decision function. The number of candidates is added so that all the additional features can be calculated to be used by the miRNA SVM.

5.2 Feature Elimination

To obtain information about the importance of each feature, every feature has been analyzed by leaving one out as described in Section 4.4. The distance between true cut site and the best cut site was estimated and used to create a cumulative distribution graph combining the performance of every feature.

Figure 5.3 presents the performance of the classifier as one feature is removed at a time. The labels on the right refer to the feature that has been removed by number. The label *all* displays the results from running the classifier with all features included in the candidate vectors.

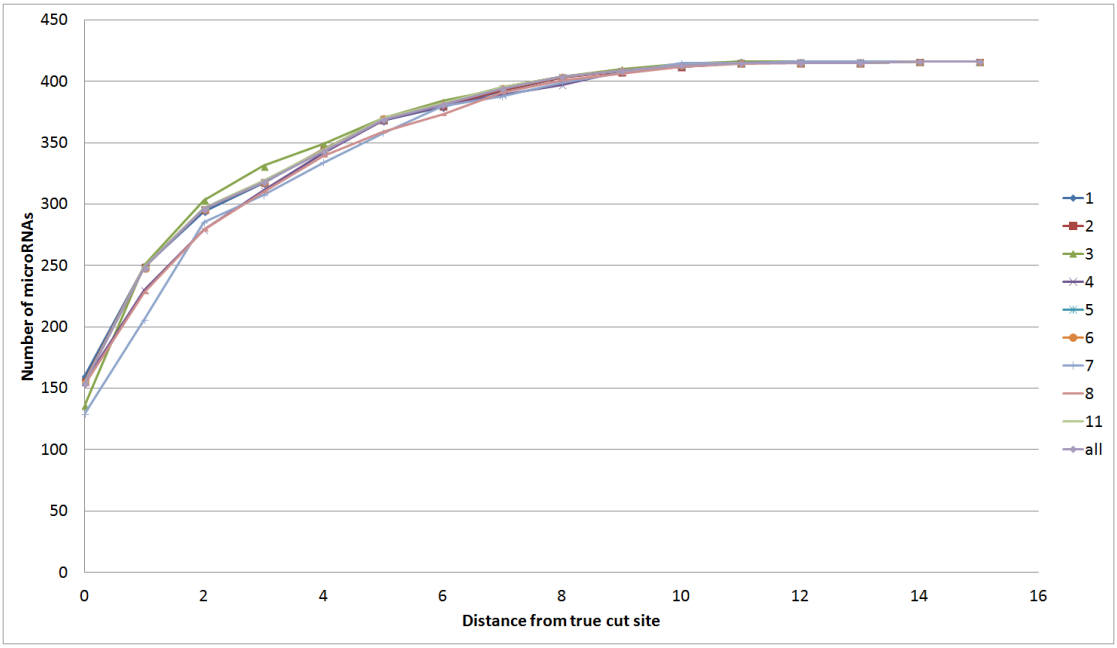


Figure 5.3: Diagram displaying the cumulative performance of each feature.

The main differences for each of the features can be found when we look at the amount of miRNAs that had their best candidate scoring within the 2 nt window, but the total amount distributed between these three distances are fairly equal for the various features. With the exceptions being features 3,4,7, and 8. This can be

seen more clearly when looking at Figure 5.4.

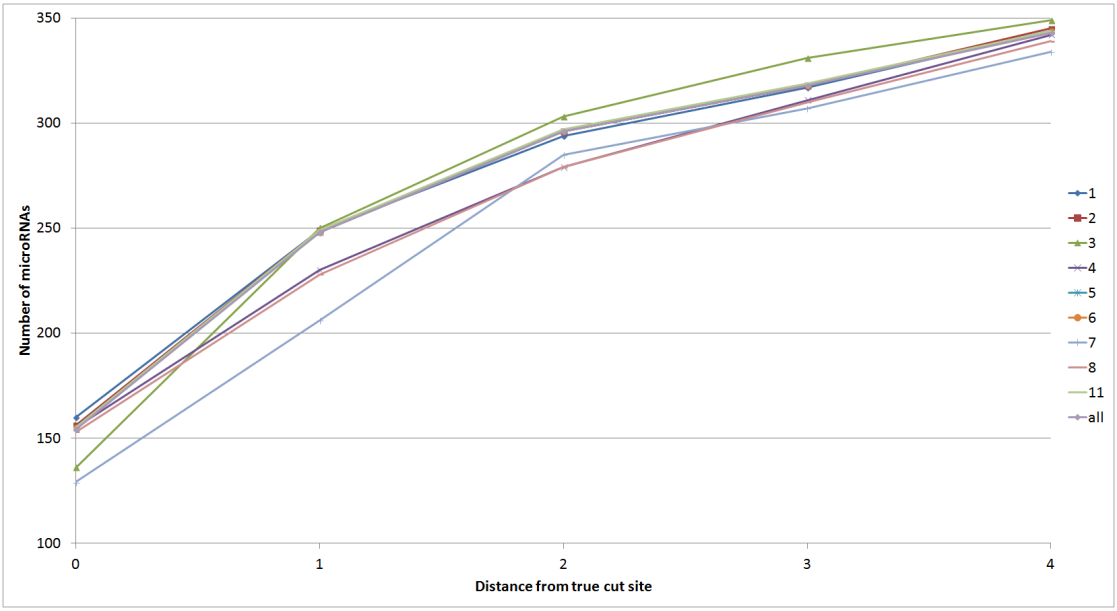


Figure 5.4: Cumulative distribution for the first 4 distances when removing each feature one at a time.

The classifier that is trained without feature 3, which contain nucleotide information for each position in the 24 nt at the precursor base, has the second worst performance when looking at correctly predicted cut sites, but outperfroms the other features when looking at the number of candidates predicted to be correct that are actually within 2 nt from the real cut site.

The classifier that is without Feature 7, which contains the nucleotide occurrence at each position in the 50 nt $3'$ and $5'$ flanking regions, underperforms all the other classifiers up to a distance of 2 nt from the real site, but starts catching up with the others at this point.

When removing feature 4 and feature 8, base-pair information for the region in feature 3 and feature 7 respectively, little change in performance is observed for the correctly predicted cut sites but the performance declines for the next two distances. At a distance of 2 nt from the real cut site, the classifiers without feature 4 and 8 have the worst performance. This may indicate that the base pairing is important for the task of finding alternative cut sites for the Microprocessor. To

further investigate this, more information about the possibility of alternative cut sites must be obtained.

By looking at these results, it seems that the information contained in feature 7 is the most critical for correctly predicting the Microprocessor cut site, and that a slight performance improvement may be obtained by removing feature 3 for classification of cut sites within 2 nt from the real site. Overall however the performance is not greatly affected by removing one feature or the other when looking at the classification within this distance. The largest difference is for the correctly predicted cut sites where the classifier without feature 1 predicted 160 miRNA cut sites correctly while the classifier without featur 7 only managed to correctly predict 129. This difference deminishes rapidly the further away from the correct cut site we get, and at a distance of 2 nt the difference is only 9. 294 for the classifier without feature 3 and 285 for the classifier without feature 8. The difference obtained by removeing one feature or the other did not present a result that is conclusive as to whether a feature should be removed or not. As a result, all features are used for the remainder of the work presented in this thesis.

A new feature was constructed in order to replace feature 7 and feature 3. Instead of looking at the nucleotide at each specific position, the new feature calculated the base composition of nucleotide windows of length 6 or 12. When replacing the information in feature 3 and 7 with the base pair composition feature the performance dropped. It gave a better performance than removing all information of the two features entirely, but quite a bit worse performance than when replacing feature 3 and 7 with the base pair information. The largest difference was found when replacing feature 7 with the new feature.

This indicates that there are nucleotide positions within the two regions that have specific bases connected to them. Identifying these bases can further assist the classifier performance in the future. Combining this information with the base composition of the regions may turn out to be a better feature than the existing ones.

5.3 MicroRNA SVM results

A miRNA classifier, that based on the output from the miRNA processing site SVM makes a prediction on whether or not a miRNA candidate is a real miRNA or not, has been constructed. To test how this classifier generalizes to unknown data, a 10-fold cross validation has been run on this classifier. The data sets consisting of the *verified data set* and the *failed data set* and the *family in same fold HSA data set* has been used for this purpose. These data sets were constructed as described in Section 4.5. The *verified data set* and the *failed data set* were combined into one set where the *verified data set* was labeled as positive while the *failed data set* was labeled negative. The 10-fold cross validation run on the combined data set resulted in the ROC curve plotted in Figure 5.5

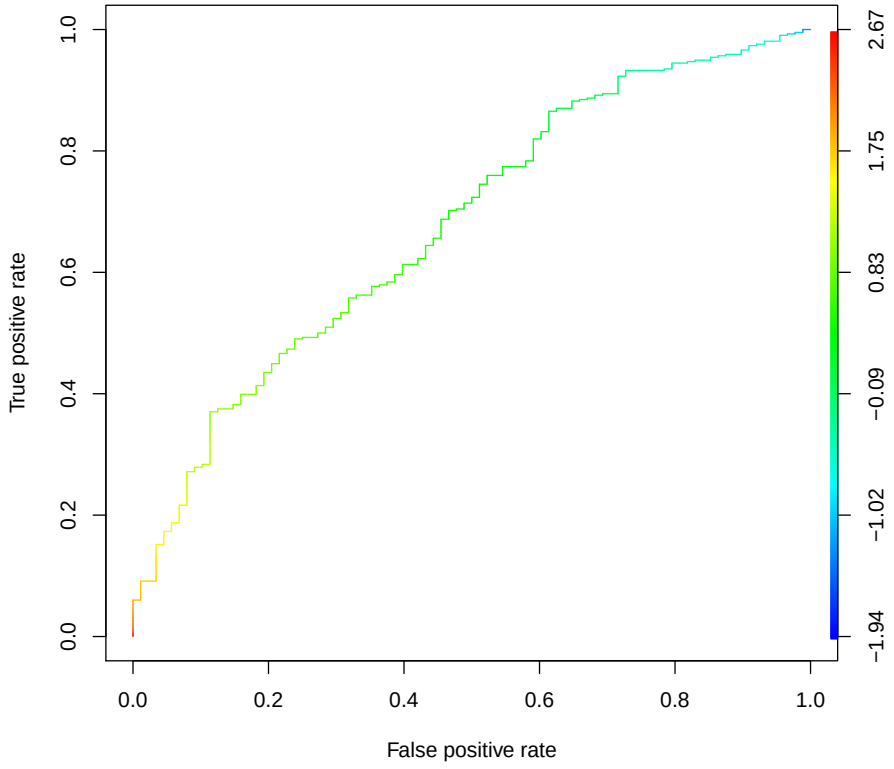


Figure 5.5: ROC plot for 10-fold cross validation on mmu verified and failed data set.

Compared to the performance of a random ROC curve, the result is better. But the performance is not good by any means. When looking at the number of true positives and true negatives with a threshold of 0, one can see that the classifier performs ok when picking out the real miRNAs with regards to true positive versus false positive rate. The sensitivity is 86,4% at this point, but the specificity is only 34 %. It is worth mentioning that the number of positive miRNAs is quite a bit larger than the negative data set. If they were of the same size, the performance would most likely drop even further as one could reasonably expect the number of false positives to increase, which in turn would reduce the sensitivity.

The data set referred to as *family in same fold HSA data set* consists of two lists of lists. One list contains the folds with the hsa miRNAs from families based on the folds generated for the *verified data set*, while the other contains the equivalent list for the *failed data set*. To obtain the miRNA vectors for the *family in same fold HSA data set*, the process illustrated in Figure 5.6 was completed.

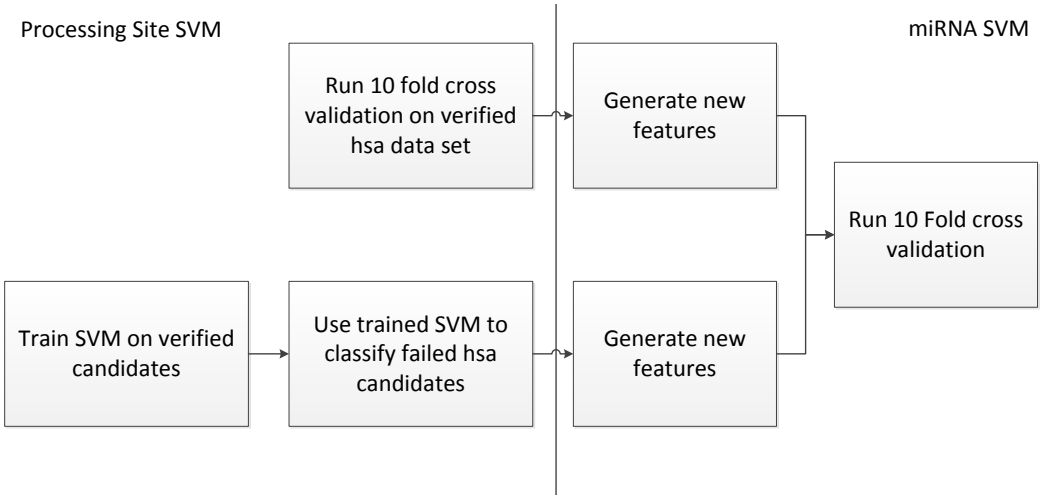


Figure 5.6: Flow chart for the process of generating HSA vectors.

The decision functions are obtained in a way that mimic the way the mmu miRNA decision functions were generated in order to make the outcome as comparable as possible. The process for the mmu data sets is identical except for how the 10-fold cross validation was completed. Rather than training on the hsa data set, the cross validation used two data sets as described in Figure 5.7.

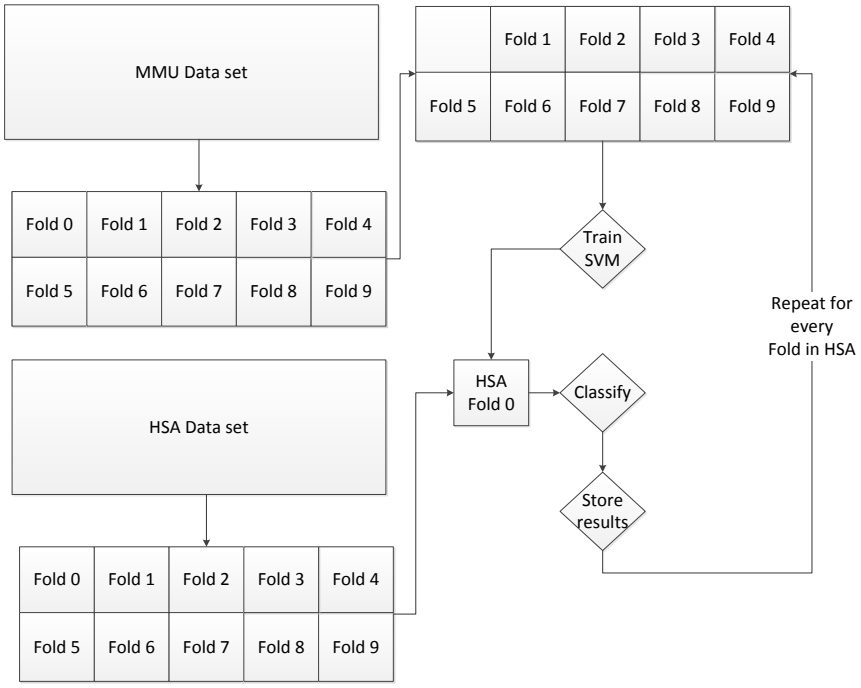


Figure 5.7: 10-fold cross validation where the SVM is trained on mmu, and used to classify the hsa folds.

The training set in this cross validation is constructed from the positive and negative folds in the same way as normal. The difference now is that instead of classifying the last fold from the mmu set, the set to be classified is replaced with the corresponding fold from the hsa set.

The outcome of this test can be seen in Figure 5.8

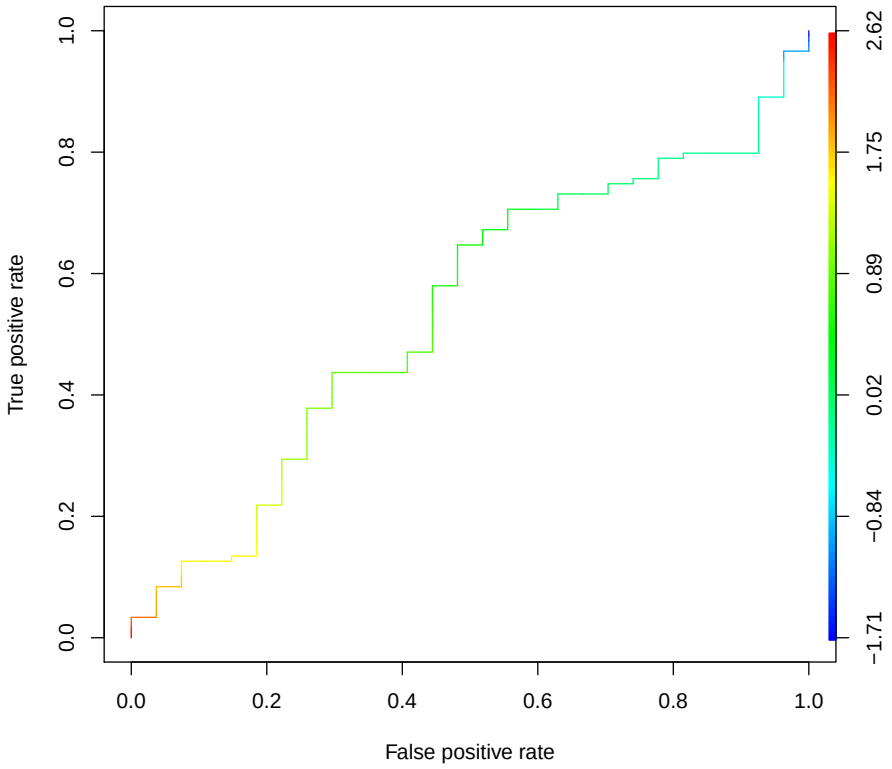


Figure 5.8: ROC plot for 10-fold cross validation on HSA data set.

The performance of the classifier when run on the *family in same fold HSA data set* is close to the performance of a random ROC curve, so it appears as if the classifier is unable to distinguish between the miRNAs from the positive and negative data sets. The performance is comparable to that of the performance obtained when training the cut site SVM on the verified data set and classifying the failed set.

An assumption was made when constructing the failed and verified data sets. This assumption was that if a mmu miRNA was from the verified data set, the hsa miRNAs from that family were true miRNAs as well.

In addition, the hsa data set was constructed without regard to the fact that

the number of miRNAs found in a miRNA family varied. For instance, one family contained 1 hsa miRNA but 9 mmu miRNAs. Other families did not contain hsa miRNAs at all. This variation did in turn affect the variation of fold sizes. Since the fold each hsa miRNA was to be put in was decided by which fold the miRNA family came from regardless of the number in that fold, the fold size varied a great deal as well. The smallest fold ended up being only two positive miRNAs. These variations may result in a worse performance when classifying the hsa miRNAs this way.

5.4 SVM feature elimination

Removing one feature at a time did not present any large changes in performance. The classifier did slightly worse without feature 3 and feature 7 (results not shown), but the change was not of any conclusive scale. What did have a large impact however was training the classifier on a vector set containing the new features based on the decision function, but not the original features used by the Microprocessor classifier. The result of this new feature set can be seen in Figure 5.9.

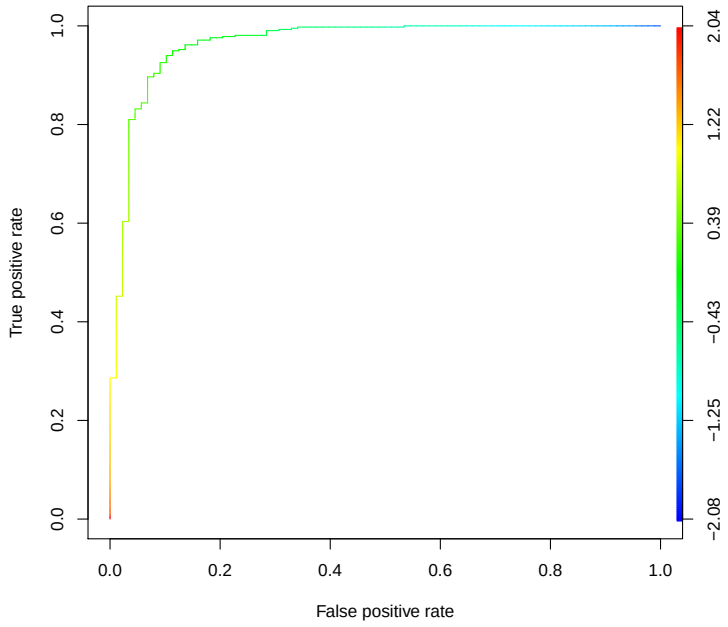


Figure 5.9: ROC plot for 10-fold cross validation of mmu with the new features.

The increased performance of the classifier when removing features may be explained by the fact that past a certain point, additional features can lead to worse performance [Trunk, 1979]. Even though the number of features might not be that high, each feature contains a varying number of numeric values. This number of values range from 1 for several features up to 400 values for feature 7. The total number of values in a vector is 684 when every feature is included. And only 9 for the vector constructed only from the new features.

There is a monumental difference between the performance of the classifier using all features, and the classifier that only uses the new features added. This become apparent when looking at the ROC curves plotted in Figure 5.12.

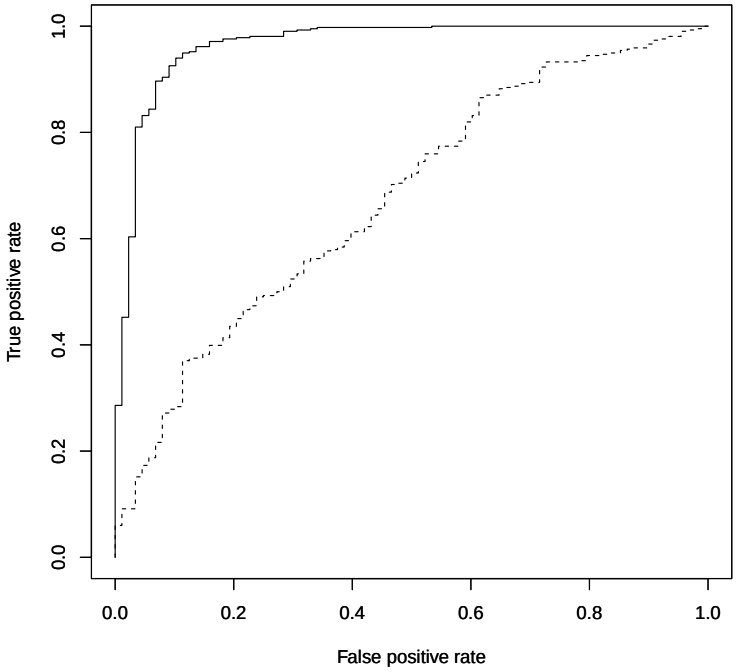


Figure 5.10: ROC plot for 10-fold cross validation of mmu with all features and new features. The ROC curve for the classifier using all features is drawn with a dotted line, while the ROC curve of the classifier using only the new features is drawn using a solid line.

To see if the change in performance held true for the data set constructed from hsa miRNAs, a 10-fold cross validation was run in the same manner as described in Section 5.3. The outcome of this cross validation can be seen in Figure 5.11.

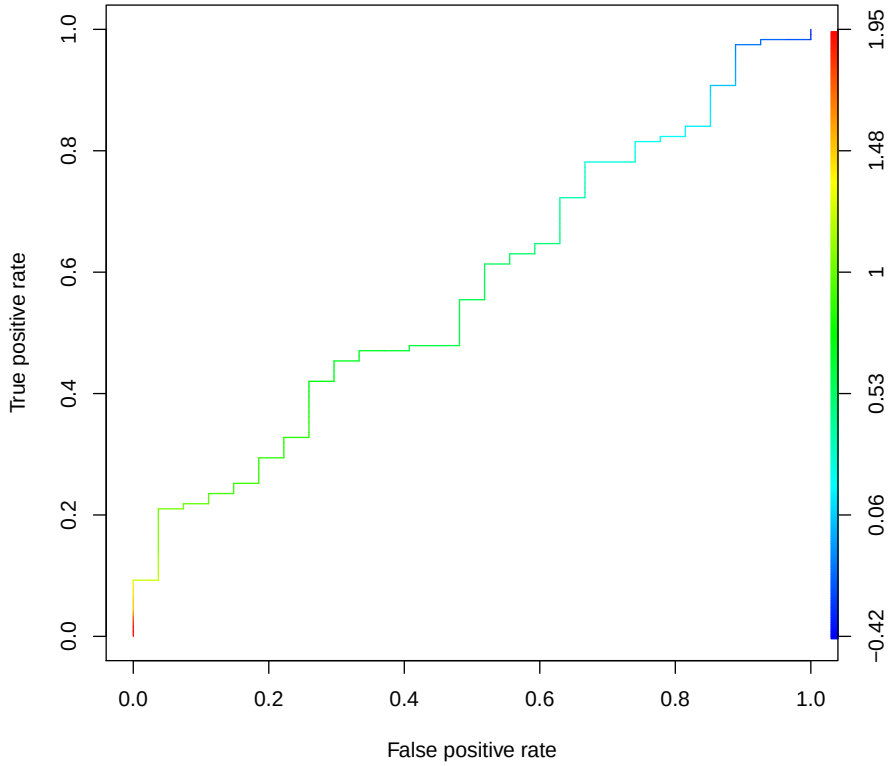


Figure 5.11: ROC plot for 10-fold cross validation of hsa with new features.

The performance is still close to that of a random classifier, and it did not adhere to the expectation of results closer to that of the ones seen with the miRNA data set from mmu. The expectation of a better performance was due to the concervation of miRNAs observed between human an mouse miRNAs.

When comparing this ROC curve with the ROC curve plotted for the classifier with all features one can see that the difference is negligible. This can be seen clearly in Figure 5.12.

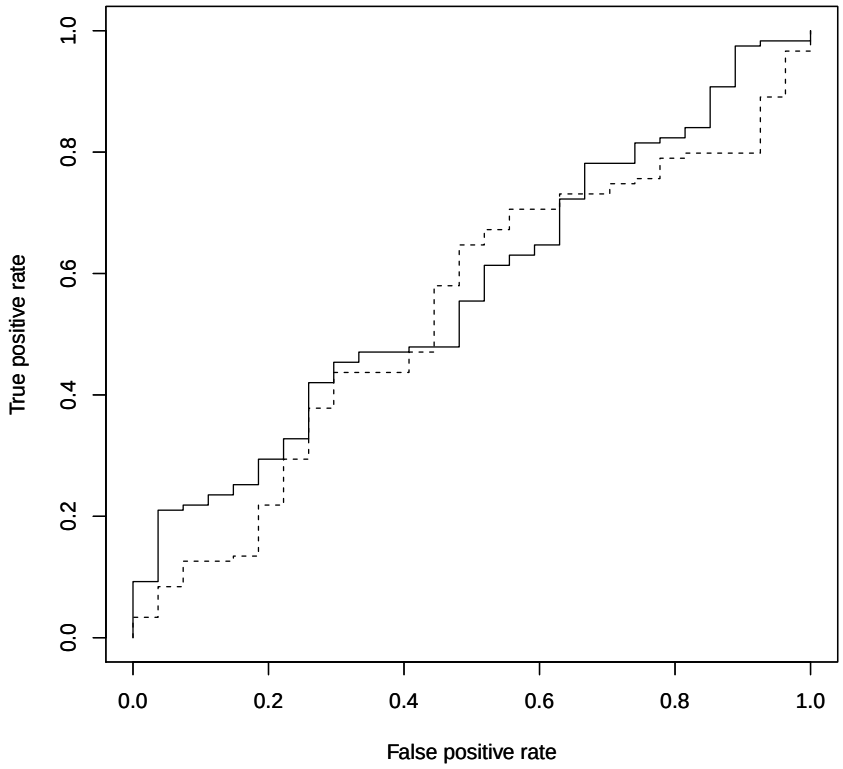


Figure 5.12: ROC plot for 10-fold cross validation of mmu with all features and new features. The classifier with new features only is drawn with a solid line while the classifier with all features is drawn with a dotted line.

Figure 5.13 compare the ROC plot for the results of the new features on the mmu data set and the ROC plot for the results of the new features on the data set that consist of hsa miRNAs.

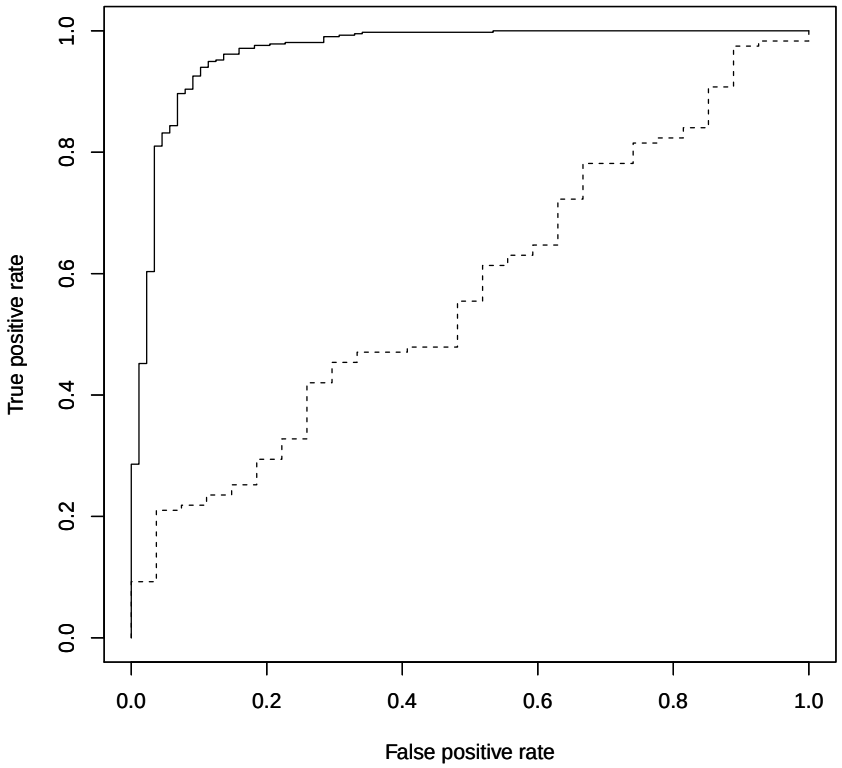


Figure 5.13: ROC plot for 10-fold cross validation of mmu with the new features only drawn in a solid line, and for the hsa data set drawn in a dotted line

There maybe many reasons why the HSA data set does not perform as well as the MMU set. None of the HSA miRNAs are actually verified. Instead the level of conservation is assumed to be high enough that a verified miRNA from the mmu family should be verified for the hsa family as well. This might very well be the case, but since the data set was constructed with the possible sources of error described in Section 5.3, the positive miRNAs in the hsa data set might not be verified at all and vice versa. Further tests on other data sets is recommended to make sure that the performance of the difference in performance is caused by

varying quality of the data sets rather than a lack in ability to generalize to new data sets from the classifier.

After the substantial change in performance while using only the new features as described, the feature analysis focus shifted. Now the whole process needed to be repeated for the new features only in order to be rid of the noise the other features may be causing.

Feature 12 proved to be the least useful feature, and the classifier that was not using this feature even managed to improve upon the performance obtained with the new features only. This was to be expected as the number of processing site candidates appeared not to be of importance for the processing of miRNAs.

The performance increase can be seen in Figure 5.14, where to ROC curve is getting close to the performance of a perfect ROC.

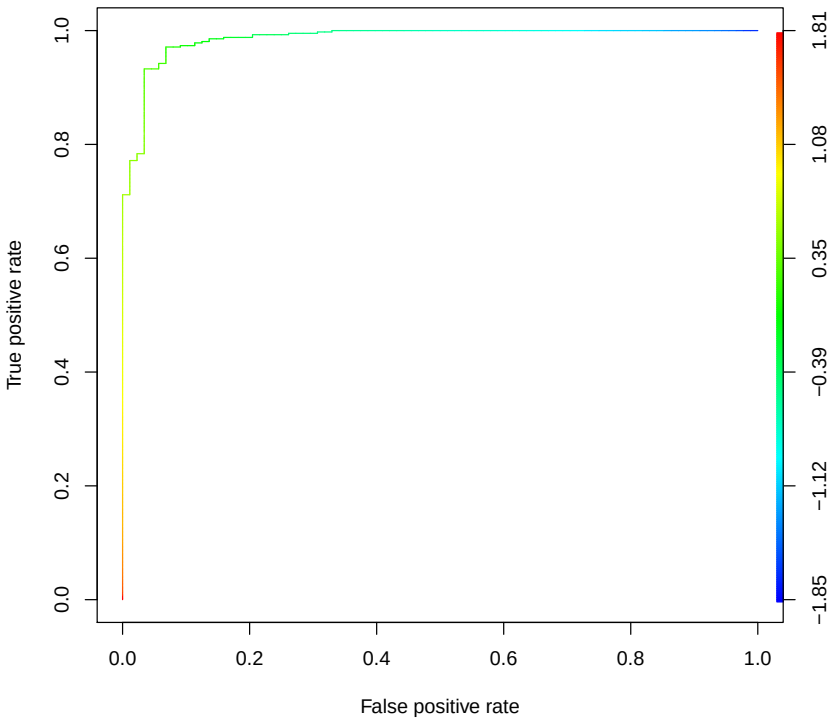


Figure 5.14: ROC plot for 10-fold cross validation without feature 12.

Feature 13, 14, 17, and 18, namely Score of the best processing site, Average score for all potential processing sites, distance between the top scoring sites, and information about local maximums, had close to similar ROC curves to the one produced with all features available. The ROC plot for these features can be found in Appendix A.

Removing Feature 15 and 16 on the other hand proved to reduce performance drastically. These two features are the standard deviation of the decision functions for all processing sites, and the difference between the best processing site and the average processing site. This indicates that it does not matter how good the processing site is, just how much better it is than the other candidates.

This result does in turn supports the theory that a miRNA has one clear cut site candidate whereas other hairpin shaped non-coding RNAs might have several cut site candidates that score well. Figure 5.15 shows the ROC plot for the classifier where feature 15 was removed. Feature 16 presented the same curve.

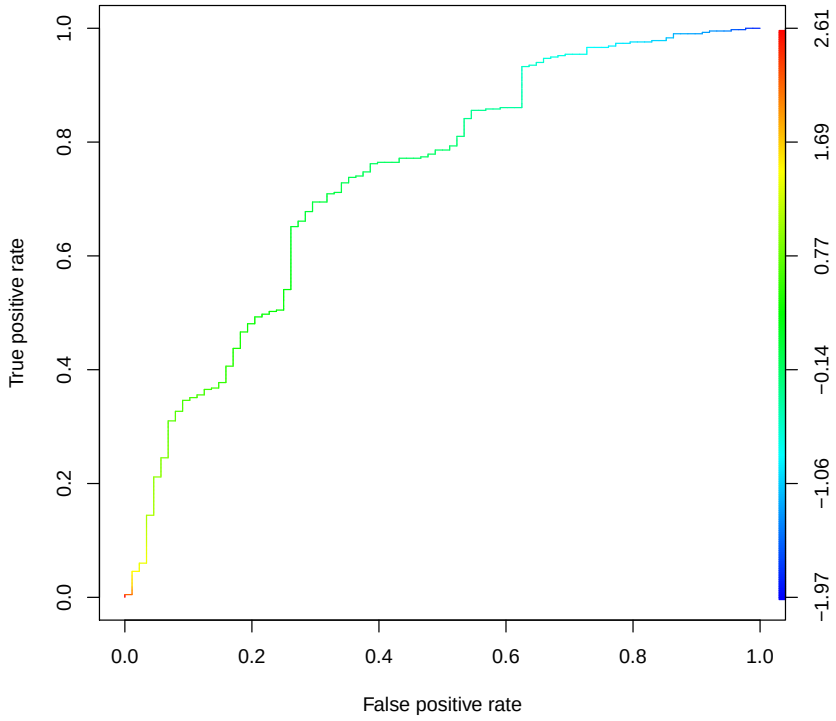


Figure 5.15: ROC plot for 10-fold cross validation without feature 15.

These findings can be used to further investigate reasons for the difference in performance between the data sets based on mmu miRNAs and hsa miRNAs. More on this in Section 7.

All miRNAs and pseudo-miRNAs from mmu in this thesis are as mentioned collected from the miRNA data set found in miRBase, making the data set well suited to test the performance of the miRNA created as they at some point have been annotated as a real miRNA due to similarities with a real miRNA. However, as the classifier here shows, there are clear differences between the miRNAs in the data set, and all of the mmu miRNAs from the *failed data sets* are in fact treated as pseudo-miRNAs because they were unable to fulfill one or more of the criteria established for annotating a candidate as a real miRNA. I therefore extend the recommendation made by Helvik et al. that a distinction between verified miRNAs and candidate miRNAs should be made.

Chapter 6

Conclusion

This thesis has been a continuation of the work done by Helvik et al. to further explore the approach for miRNA classification presented there [Snorre A. Helvik, 2007]. I have investigated the possibility to reimplement a miRNA classifier that based on information obtained by predicting the Microprocessor processing site can distinguish between real and pseudo miRNAs. This has been done through the construction of new data sets on which the classifier has been tested, and by using a new framework for classification. The novelty of this research is attributed mainly to the new data sets that have been introduced and subsequent feature analysis.

I have presented the PyML implementation of a support vector machine classifier. The outcome of this implementation is a classifier that is able to distinguish between real and pseudo-miRNAs with high accuracy for mouse miRNAs.

As a part of the reimplementation process, the effect of removing features from this classifier has been investigated. This has been done both for the Microprocessor cut site predictor and the miRNA classifier. This effort has shown that there were no miRNA features that stood out as either clear performance enhancers or performance inhibitors for the purpose of predicting the true cut site. However, some variations were present. Through the construction of a new feature based on information from feature 7, and by comparing it with the performance without feature 7, the results indicated that positions within feature 7 should be examined in more detail. In addition the removal of feature 3 showed that performance improvement might be obtained. These differences were however not of a scale that called for a removal of the mentioned features.

When eliminating the features for the miRNA classifier, the findings were of a different nature. The changes were minor when removing one feature at a time. At the same time, the performance of the classifier was poor when all features were in use. When all but the new features were removed however, the performance improved greatly. This performance was further improved by removing feature 12. With the new insight the feature elimination was repeated for the new features only, and feature 15 and 16, clearly distinguished themselves as the most important features for the classifier performance. These features contained information about the standard deviation of the decision function (feature 15), and the difference between the best cut site candidate and the average of all cut site candidates for the given miRNA. This knowledge can in turn be used to investigate the difference in performance between the mouse data sets and human data sets.

It is important to remember that there are other pathways for a mature miRNA, and that these are not processed by Drosha [Erik Ladewig, 2012]. This makes the computational approaches described here unfit for finding these other miRNAs as it relies on the Drosha processing. In addition, more tests of the classifier on new data sets would be ideal to make sure that the difference in performance between the human miRNAs and mouse miRNAs is in fact caused by varying quality of the data sets rather than the miRNA classifier having problems with generalizing to new data.

Even when considering the limitations of the research, the results obtained in this thesis further support the theory that the information obtained about the Drosha cut site can be used to further classify miRNAs as real or pseudo-miRNAs. This can in turn help improving the set of known miRNAs. A good set of real miRNAs will be of great importance to understand the nature of miRNAs, as this among other things will help remove unnecessary noise from further studies.

Chapter 7

Further work

This thesis has provided new information that can be useful in the endeavour to get a better understanding of miRNAs, but there are also several questions that arises. The results obtained indicated that there may be specific positions in the flanking region that are important for correct processing of a miRNA substrate by the Microprocessor complex. Identifying these positions could prove valuable to get a better understanding of how Drosha recognize substrates.

In addition, the information obtained during the feature analysis can be used to reanalyze why the results were so different between the data sets based on mmu miRNAs and the data sets based hsa miRNAs. One can for instance take a look at the hsa miRNAs that were assumed false due to conservation and calculate the average scores for feature 15 and 16, this can in turn be compared to the same calculation for the false mmu miRNAs. This can hopefully reveal information about the similarities between these two data sets to see if the false hsa data set is in fact false.

Finally, to compare the performance of the classifier constructed here to other methods of classification, the other methods could be tested on the data sets used here. This would give comparative results, and also yield information about how the features used by other methods fare at the task of distinguishing between these data sets.

Bibliography

[Asa Ben-Hur,] Asa Ben-Hur, J. W. A user's guide to support vector machines. -.

[Asa Ben-Hur, 2008] Asa Ben-Hur, Cheng Soon Ong, S. S.-B. S. G. R. (2008). Support vector machines and kernels for computational biology. *PLoS Computational Biology*, 4:1–10.

[Bartel, 2004] Bartel, D. P. (2004). Microrna: Genomics, biogenesis, mechanisms, and function. *Cell*, 116:281–297.

[Ben-Hur, a] Ben-Hur, A. Pyml - machine learning in python. Available: `http://pyml.sourceforge.net/`. Last accessed 3.6.2013.

[Ben-Hur, b] Ben-Hur, A. Pyml tutorial. Available at: `http://pyml.sourceforge.net/tutorial.html`. Last Accessed: 20.5.2013.

[Chenghai Xue and Zhang, 2005] Chenghai Xue, Fei, L. T. H. G.-P. L. Y. L. and Zhang, X. (2005). Classification of real and pseudo microrna precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6:–.

[Chih-Wei Hsu and Lin,] Chih-Wei Hsu, C.-C. C. and Lin, C.-J. A practical guide to support vector classification. -.

[Cullen, 2004] Cullen, B. R. (2004). Transcription and processing of human microrna precursors. *Molecular Cell*, 16:861–865.

[Erik Ladewig, 2012] Erik Ladewig, Katsutomo Okamura, A. S. F.-e. a. (2012). Discovery of hundreds of mirtrons in mouse and human small rna data. *Genome Research*, 22:1634–1645.

[excellence, 2009] excellence, A. (2009). Protein syntheis. available at: `http://www.accessexcellence.org/RC/VL/GG/protein_synthesis.php`. last accessed 12.6.2013.

[Fawcett, 2006] Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874.

[H. Rosaria Chiang and Bartel, 2010] H. Rosaria Chiang, Lori W. Schoenfeld, J. G. R. V. C. A. N. S.-D. B. W. K. J. C. R.-S. L. J. E. B. R. B. G. P. S. C. N. and Bartel, D. P. (2010). Mammalian microrna: experimental evaluation of novel and previously annotated genes. *Genes and development*, 24:992–1009.

[Hammell, 2010] Hammell, M. (2010). Computational methods to identify mirna targets. *Seminars in Cell and Developmental Biology*, 21:738–744.

[Holst, 2012] Holst, F. K. (2012). A pyml implementation of the support vector machine for use in microrna classification. Intro project for this master thesis, 2012.

[Jiandong Ding, 2010] Jiandong Ding, Shuigen Zhou, J. G. (2010). Mirensvm: towards better prediction of microrna precursors using and ensemble svm classifier with multi-loop features. *BMC Bioinformatics*, 11:–.

[Jin-Wu Nam, 2005] Jin-Wu Nam, Ki-Roo Shin, J. H. Y. L. V. N. K. B.-T. Z. (2005). Human microrna prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acid Research*, 33:3570–3581.

[Jinju Han and Kim, 2006] Jinju Han, Yoontae Lee, K.-H. Y. J.-W. N. I. H. J.-K. T. S. Y. S. Y. C. B.-T. Z. and Kim, V. N. (2006). Molecular basis for the recognition of primary microrna by the drosha-dgcr8 complex. *Cell*, 125:887–901.

[Matthias Hochsmann, 2003] Matthias Hochsmann, Toller, R. G.-S. K. (2003). Local similarity in rna secondary structures. *Proceedings of the Computational Systems Bioinformatics*, -:–.

[Mraz, 2012] Mraz, M. (2012). Web book: Biological role of microrna in animal cells, development and cancer. Available at: `http://www.microrna.ic.cz/mirna4.html`. Last accessed: 13.6.2013.

[N. D. Mendes and Sagot, 2009] N. D. Mendes, A. T. F. and Sagot, M.-F. (2009). Current tools for the identification of mirna genes and their targets. *Nucleic Acid Research*, 37:2419–2433.

[Nelson and Istrail,] Nelson, M. and Istrail, S. Rna structure and prediction computational molecular biology (bio502). Available at: `http://tuvalu.santafe.edu/~pth/rna.html`. Last accessed 12.6.2013.

[Peng Jiang and Lu, 2007] Peng Jiang, Haonan Wu, W. W. W. M.-X. S. and Lu, Z. (2007). Mipred: classification of real and pseudo microrna precursors using

random forest prediction model with combined features. *Nucleic Acids Research*, 35:339–344.

[R-project, 2013] R-project (2013). What is r? Available at: `http://www.r-project.org/`. Last accessed: 11.6.2013.

[Reece, 2010a] Reece, Urry, C. W. M. J. (2010a). *Essential Amino Acids*, page 922. Pearson, 9th edition.

[Reece, 2010b] Reece, Urry, C. W. M. J. (2010b). *From Gene to Protein*, pages 371–394. Pearson, 9th edition.

[Reece, 2010c] Reece, Urry, C. W. M. J. (2010c). *The Molecular basis of Inheritance*, chapter 16, pages 351–368. Pearson, 9th edition.

[Reece, 2010d] Reece, Urry, C. W. M. J. (2010d). *Structure and function of large biological molecules*, pages 123–132. Pearson, 9th edition.

[Snorre A. Helvik, 2007] Snorre A. Helvik, Olav Snoeve Jr, P. S. (2007). Reliable prediction of drosha processing sites improves microrna gene prediction. *Bioinformatics*, 23:142–149.

[Tobias Sing, 2005] Tobias Sing, Oliver Sander, N. B. T. L. (2005). Rocr: visualizing classifier performance in r. *Bioinformatics*, 21:3940–3941.

[Trunk, 1979] Trunk, G. (1979). A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAM1-1:306–307.

[V. Narry Kim and Siomi, 2009] V. Narry Kim, J. H. and Siomi, M. C. (2009). Biogenesis of small rnas in animals. *Nature*, 10:126–139.

[Vincent C. Auyeung and Batel, 2013] Vincent C. Auyeung, Igor Ulitsky, S. M. and Batel, D. (2013). Beyond secondary structure: Primary-sequence determinants license pri-mirna hairpins for processing. *Cell*, 152:844–858.

[William Ritchie and Rasko, 2012] William Ritchie, D. G. and Rasko, J. E. J. (2012). Defining and providing robust controls for microrna prediction. *Bioinformatics*, 28:1058–1061.

Appendix A

Feature elimination ROC curves

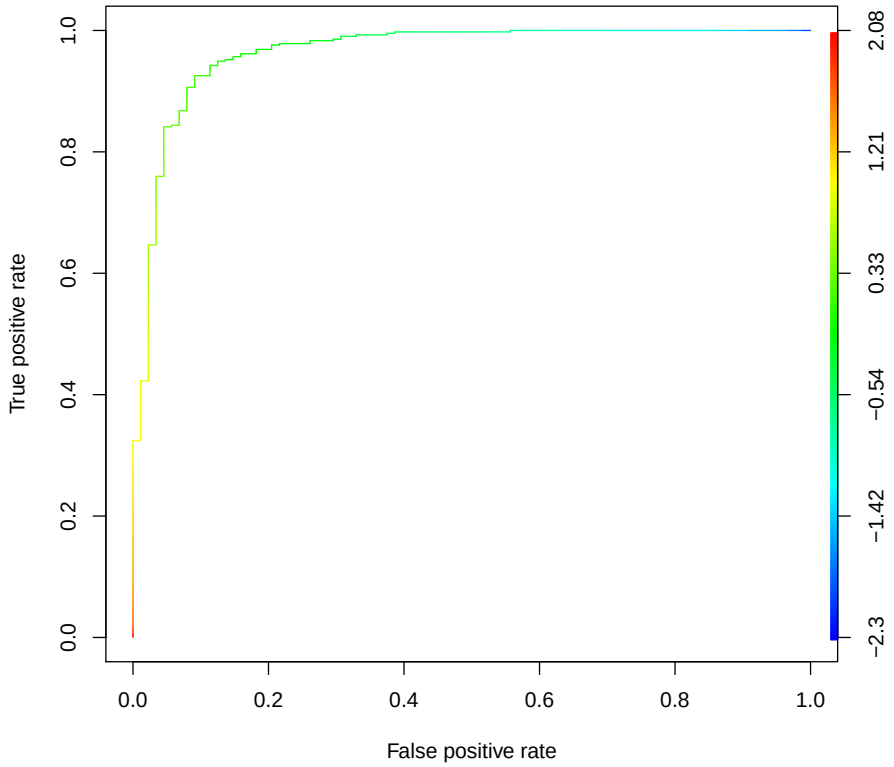


Figure A.1: ROC curve, feature 13 removed

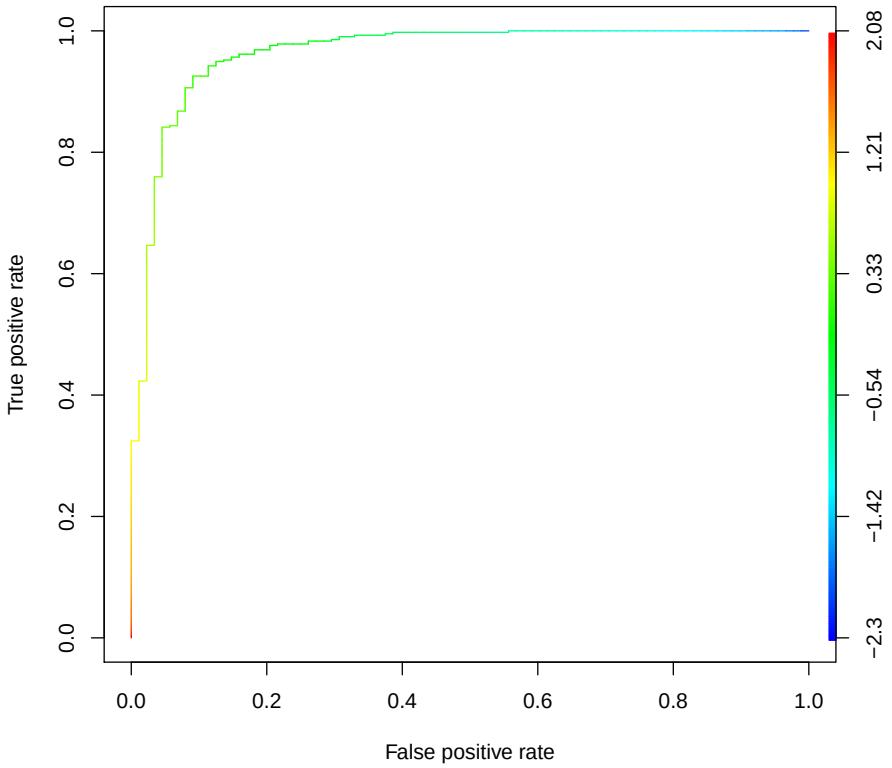


Figure A.2: ROC curve, feature 14 removed

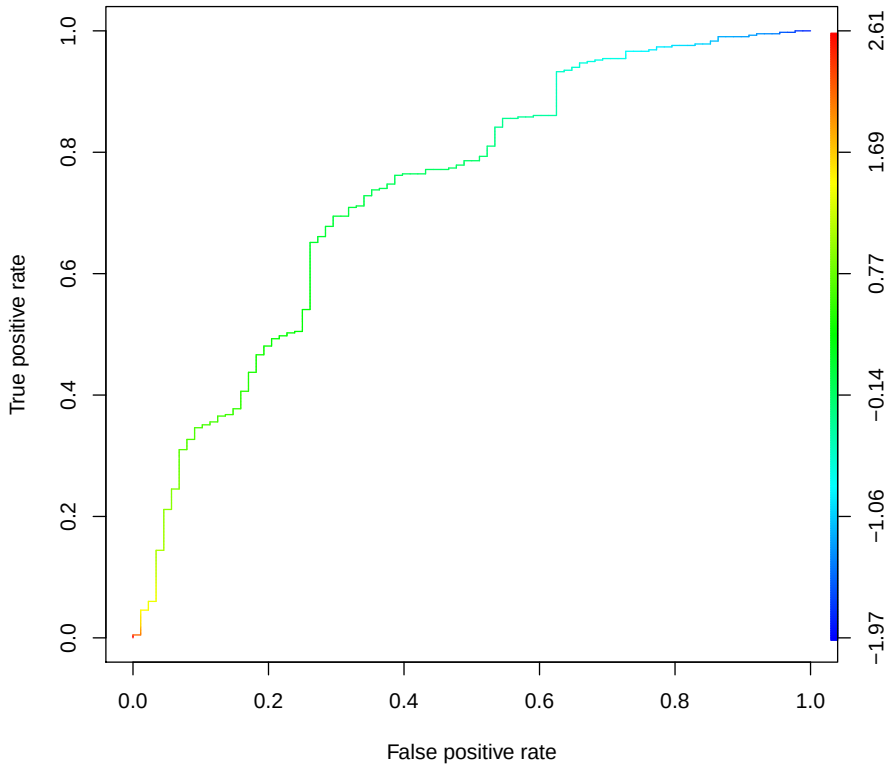


Figure A.3: ROC curve, feature 16 removed

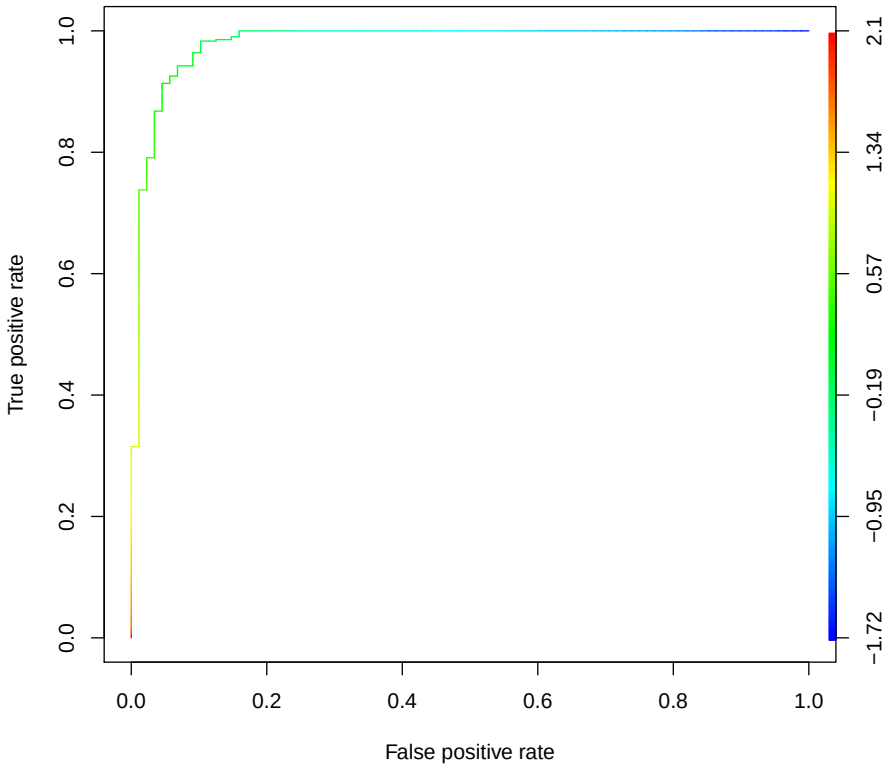


Figure A.4: ROC curve, feature 17 removed

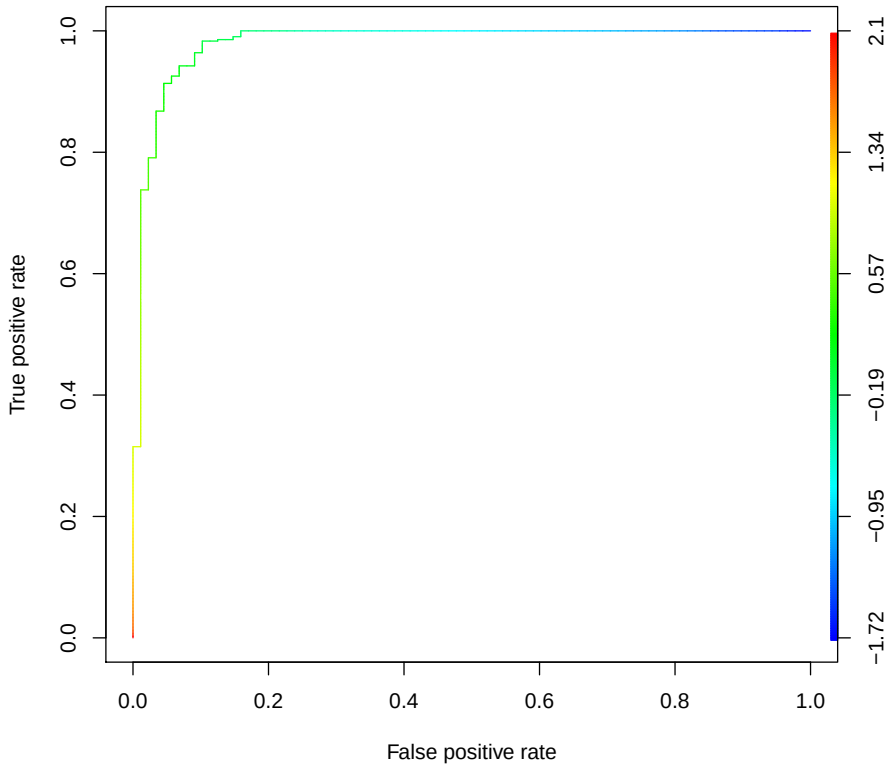


Figure A.5: ROC curve, feature 18 removed

Appendix B

MMU and HSA folds

B.1 MMU Folds

Fold 0
miRNA@mmu-mir-467a-1, Label: 1
miRNA@mmu-mir-669a-1, Label: 1
miRNA@mmu-mir-669b, Label: 1
miRNA@mmu-mir-669a-2, Label: 1
miRNA@mmu-mir-669a-3, Label: 1
miRNA@mmu-mir-467b, Label: 1
miRNA@mmu-mir-669c, Label: 1
miRNA@mmu-mir-466f-1, Label: 1
miRNA@mmu-mir-466h, Label: 1
miRNA@mmu-mir-467c, Label: 1
miRNA@mmu-mir-467d, Label: 1
miRNA@mmu-mir-466d, Label: 1
miRNA@mmu-mir-467e, Label: 1
miRNA@mmu-mir-466l, Label: 1
miRNA@mmu-mir-669d, Label: 1
miRNA@mmu-mir-669f, Label: 1
miRNA@mmu-mir-669h, Label: 1
miRNA@mmu-mir-669l, Label: 1
miRNA@mmu-mir-669m-1, Label: 1
miRNA@mmu-mir-669m-2, Label: 1
miRNA@mmu-mir-669o, Label: 1
miRNA@mmu-mir-466m, Label: 1

miRNA@mmu-mir-669d-2, Label: 1
miRNA@mmu-mir-466o, Label: 1
miRNA@mmu-mir-467a-2, Label: 1
miRNA@mmu-mir-669a-4, Label: 1
miRNA@mmu-mir-669a-5, Label: 1
miRNA@mmu-mir-467a-3, Label: 1
miRNA@mmu-mir-669a-6, Label: 1
miRNA@mmu-mir-467a-4, Label: 1
miRNA@mmu-mir-669a-7, Label: 1
miRNA@mmu-mir-467a-5, Label: 1
miRNA@mmu-mir-669p-1, Label: 1
miRNA@mmu-mir-467a-6, Label: 1
miRNA@mmu-mir-669a-8, Label: 1
miRNA@mmu-mir-669a-9, Label: 1
miRNA@mmu-mir-467a-7, Label: 1
miRNA@mmu-mir-669p-2, Label: 1
miRNA@mmu-mir-467a-8, Label: 1
miRNA@mmu-mir-669a-10, Label: 1
miRNA@mmu-mir-467a-9, Label: 1
miRNA@mmu-mir-669a-11, Label: 1
miRNA@mmu-mir-467a-10, Label: 1
miRNA@mmu-mir-669a-12, Label: 1
miRNA@mmu-mir-466n, Label: 1
miRNA@mmu-mir-466g, Label: 0
miRNA@mmu-mir-568, Label: 0
miRNA@mmu-mir-466i, Label: 0
miRNA@mmu-mir-669j, Label: 0
miRNA@mmu-mir-669i, Label: 0
miRNA@mmu-mir-467g, Label: 0

Fold 1
miRNA@mmu-mir-466a, Label: 1
miRNA@mmu-mir-466b-1, Label: 1
miRNA@mmu-mir-466b-2, Label: 1
miRNA@mmu-mir-466b-3, Label: 1
miRNA@mmu-mir-466c-1, Label: 1
miRNA@mmu-mir-466e, Label: 1
miRNA@mmu-mir-466c-2, Label: 1
miRNA@mmu-mir-466b-4, Label: 1
miRNA@mmu-mir-466b-6, Label: 1

miRNA@mmu-mir-466p, Label: 1
miRNA@mmu-mir-466b-8, Label: 1
miRNA@mmu-mir-103-1, Label: 1
miRNA@mmu-mir-103-2, Label: 1
miRNA@mmu-mir-107, Label: 1
miRNA@mmu-mir-450a-1, Label: 1
miRNA@mmu-mir-450a-2, Label: 1
miRNA@mmu-mir-450b, Label: 1
miRNA@mmu-mir-128-2, Label: 1
miRNA@mmu-mir-365-1, Label: 1
miRNA@mmu-mir-365-2, Label: 1
miRNA@mmu-mir-674, Label: 1
miRNA@mmu-mir-1930, Label: 1
miRNA@mmu-mir-3094, Label: 1
miRNA@mmu-mir-3062, Label: 1
miRNA@mmu-mir-488, Label: 1
miRNA@mmu-mir-3109, Label: 1
miRNA@mmu-mir-3086, Label: 1
miRNA@mmu-mir-3090, Label: 1
miRNA@mmu-mir-363, Label: 1
miRNA@mmu-mir-328, Label: 1
miRNA@mmu-mir-874, Label: 1
miRNA@mmu-mir-1912, Label: 1
miRNA@mmu-mir-1943, Label: 1
miRNA@mmu-mir-551b, Label: 1
miRNA@mmu-mir-3098, Label: 1
miRNA@mmu-mir-342, Label: 1
miRNA@mmu-mir-190b, Label: 1
miRNA@mmu-mir-483, Label: 1
miRNA@mmu-mir-3099, Label: 1
miRNA@mmu-mir-670, Label: 1
miRNA@mmu-mir-880, Label: 1
miRNA@mmu-mir-680-3, Label: 0
miRNA@mmu-mir-697, Label: 0
miRNA@mmu-mir-343, Label: 0
miRNA@mmu-mir-1190, Label: 0
miRNA@mmu-mir-1949, Label: 0
miRNA@mmu-mir-1950, Label: 0
miRNA@mmu-mir-1954, Label: 0
miRNA@mmu-mir-2136, Label: 0

Fold 2
miRNA@mmu-mir-300, Label: 1
miRNA@mmu-mir-323, Label: 1
miRNA@mmu-mir-377, Label: 1
miRNA@mmu-mir-381, Label: 1
miRNA@mmu-mir-382, Label: 1
miRNA@mmu-mir-409, Label: 1
miRNA@mmu-mir-410, Label: 1
miRNA@mmu-mir-539, Label: 1
miRNA@mmu-mir-494, Label: 1
miRNA@mmu-mir-487b, Label: 1
miRNA@mmu-mir-369, Label: 1
miRNA@mmu-mir-15b, Label: 1
miRNA@mmu-mir-15a, Label: 1
miRNA@mmu-mir-16-1, Label: 1
miRNA@mmu-mir-16-2, Label: 1
miRNA@mmu-mir-26a-1, Label: 1
miRNA@mmu-mir-26b, Label: 1
miRNA@mmu-mir-26a-2, Label: 1
miRNA@mmu-mir-486, Label: 1
miRNA@mmu-mir-3107, Label: 1
miRNA@mmu-mir-194-2, Label: 1
miRNA@mmu-mir-129-2, Label: 1
miRNA@mmu-mir-3063, Label: 1
miRNA@mmu-mir-431, Label: 1
miRNA@mmu-mir-3108, Label: 1
miRNA@mmu-mir-3083, Label: 1
miRNA@mmu-mir-3089, Label: 1
miRNA@mmu-mir-3105, Label: 1
miRNA@mmu-mir-1198, Label: 1
miRNA@mmu-mir-423, Label: 1
miRNA@mmu-mir-1941, Label: 1
miRNA@mmu-mir-22, Label: 1
miRNA@mmu-mir-331, Label: 1
miRNA@mmu-mir-339, Label: 1
miRNA@mmu-mir-31, Label: 1
miRNA@mmu-mir-491, Label: 1
miRNA@mmu-mir-592, Label: 1
miRNA@mmu-mir-881, Label: 1

miRNA@mmu-mir-367, Label: 1
miRNA@mmu-mir-210, Label: 1
miRNA@mmu-mir-207, Label: 0
miRNA@mmu-mir-449c, Label: 0
miRNA@mmu-mir-703, Label: 0
miRNA@mmu-mir-453, Label: 0
miRNA@mmu-mir-449b, Label: 0
miRNA@mmu-mir-1191, Label: 0
miRNA@mmu-mir-1192, Label: 0
miRNA@mmu-mir-1902, Label: 0
miRNA@mmu-mir-1929, Label: 0
miRNA@mmu-mir-1948, Label: 0
miRNA@mmu-mir-1969, Label: 0

Fold 3
miRNA@mmu-let-7d, Label: 1
miRNA@mmu-let-7a-1, Label: 1
miRNA@mmu-let-7a-2, Label: 1
miRNA@mmu-let-7b, Label: 1
miRNA@mmu-let-7c-1, Label: 1
miRNA@mmu-let-7c-2, Label: 1
miRNA@mmu-let-7e, Label: 1
miRNA@mmu-let-7f-1, Label: 1
miRNA@mmu-let-7f-2, Label: 1
miRNA@mmu-mir-98, Label: 1
miRNA@mmu-mir-29a, Label: 1
miRNA@mmu-mir-29c, Label: 1
miRNA@mmu-mir-29b-2, Label: 1
miRNA@mmu-mir-24-2, Label: 1
miRNA@mmu-mir-3074-2, Label: 1
miRNA@mmu-mir-27a, Label: 1
miRNA@mmu-mir-221, Label: 1
miRNA@mmu-mir-222, Label: 1
miRNA@mmu-mir-883a, Label: 1
miRNA@mmu-mir-883b, Label: 1
miRNA@mmu-mir-193b, Label: 1
miRNA@mmu-mir-143, Label: 1
miRNA@mmu-mir-3060, Label: 1
miRNA@mmu-mir-326, Label: 1
miRNA@mmu-mir-493, Label: 1

miRNA@mmu-mir-504, Label: 1
miRNA@mmu-mir-3085, Label: 1
miRNA@mmu-mir-3104, Label: 1
miRNA@mmu-mir-700, Label: 1
miRNA@mmu-mir-455, Label: 1
miRNA@mmu-mir-3057, Label: 1
miRNA@mmu-mir-1934, Label: 1
miRNA@mmu-mir-876, Label: 1
miRNA@mmu-mir-412, Label: 1
miRNA@mmu-mir-223, Label: 1
miRNA@mmu-mir-667, Label: 1
miRNA@mmu-mir-542, Label: 1
miRNA@mmu-mir-217, Label: 1
miRNA@mmu-mir-666, Label: 1
miRNA@mmu-mir-511, Label: 1
miRNA@mmu-mir-320, Label: 1
miRNA@mmu-mir-297a-1, Label: 0
miRNA@mmu-mir-297a-2, Label: 0
miRNA@mmu-mir-105, Label: 0
miRNA@mmu-mir-574, Label: 0
miRNA@mmu-mir-1187, Label: 0
miRNA@mmu-mir-1905, Label: 0
miRNA@mmu-mir-1895, Label: 0
miRNA@mmu-mir-669n, Label: 0

Fold 4
miRNA@mmu-mir-344-1, Label: 1
miRNA@mmu-mir-344d-3, Label: 1
miRNA@mmu-mir-344d-1, Label: 1
miRNA@mmu-mir-344d-2, Label: 1
miRNA@mmu-mir-344-2, Label: 1
miRNA@mmu-mir-344e, Label: 1
miRNA@mmu-mir-344b, Label: 1
miRNA@mmu-mir-344c, Label: 1
miRNA@mmu-mir-344g, Label: 1
miRNA@mmu-mir-344f, Label: 1
miRNA@mmu-mir-301a, Label: 1
miRNA@mmu-mir-130b, Label: 1
miRNA@mmu-mir-301b, Label: 1
miRNA@mmu-mir-124-1, Label: 1

miRNA@mmu-mir-124-2, Label: 1
miRNA@mmu-mir-148a, Label: 1
miRNA@mmu-mir-148b, Label: 1
miRNA@mmu-mir-101b, Label: 1
miRNA@mmu-mir-212, Label: 1
miRNA@mmu-mir-672, Label: 1
miRNA@mmu-mir-136, Label: 1
miRNA@mmu-mir-122, Label: 1
miRNA@mmu-mir-3061, Label: 1
miRNA@mmu-mir-490, Label: 1
miRNA@mmu-mir-3113, Label: 1
miRNA@mmu-mir-3087, Label: 1
miRNA@mmu-mir-206, Label: 1
miRNA@mmu-mir-653, Label: 1
miRNA@mmu-mir-3106, Label: 1
miRNA@mmu-mir-3064, Label: 1
miRNA@mmu-mir-1982, Label: 1
miRNA@mmu-mir-505, Label: 1
miRNA@mmu-mir-375, Label: 1
miRNA@mmu-mir-744, Label: 1
miRNA@mmu-mir-764, Label: 1
miRNA@mmu-mir-346, Label: 1
miRNA@mmu-mir-1251, Label: 1
miRNA@mmu-mir-664, Label: 1
miRNA@mmu-mir-3077, Label: 1
miRNA@mmu-mir-350, Label: 1
miRNA@mmu-mir-762, Label: 0
miRNA@mmu-mir-687, Label: 0
miRNA@mmu-mir-693, Label: 0
miRNA@mmu-mir-804, Label: 0
miRNA@mmu-mir-1195, Label: 0
miRNA@mmu-mir-1897, Label: 0
miRNA@mmu-mir-1892, Label: 0
miRNA@mmu-mir-1932, Label: 0
miRNA@mmu-mir-1961, Label: 0

Fold 5
miRNA@mmu-mir-10a, Label: 1
miRNA@mmu-mir-100, Label: 1
miRNA@mmu-mir-125b-1, Label: 1

miRNA@mmu-mir-201, Label: 1
miRNA@mmu-mir-470, Label: 1
miRNA@mmu-mir-743a, Label: 1
miRNA@mmu-mir-743b, Label: 1
miRNA@mmu-mir-871, Label: 1
miRNA@mmu-mir-376a, Label: 1
miRNA@mmu-mir-376b, Label: 1
miRNA@mmu-mir-376c, Label: 1
miRNA@mmu-mir-199a-2, Label: 1
miRNA@mmu-mir-199b, Label: 1
miRNA@mmu-mir-500, Label: 1
miRNA@mmu-mir-501, Label: 1
miRNA@mmu-mir-532, Label: 1
miRNA@mmu-mir-192, Label: 1
miRNA@mmu-mir-215, Label: 1
miRNA@mmu-mir-879, Label: 1
miRNA@mmu-mir-338, Label: 1
miRNA@mmu-mir-1298, Label: 1
miRNA@mmu-mir-802, Label: 1
miRNA@mmu-mir-3112, Label: 1
miRNA@mmu-mir-3080, Label: 1
miRNA@mmu-mir-497, Label: 1
miRNA@mmu-mir-1188, Label: 1
miRNA@mmu-mir-3058, Label: 1
miRNA@mmu-mir-741, Label: 1
miRNA@mmu-mir-3068, Label: 1
miRNA@mmu-mir-139, Label: 1
miRNA@mmu-mir-337, Label: 1
miRNA@mmu-mir-32, Label: 1
miRNA@mmu-mir-742, Label: 1
miRNA@mmu-mir-345, Label: 1
miRNA@mmu-mir-96, Label: 1
miRNA@mmu-mir-499, Label: 1
miRNA@mmu-mir-434, Label: 1
miRNA@mmu-mir-425, Label: 1
miRNA@mmu-mir-678, Label: 0
miRNA@mmu-mir-683-1, Label: 0
miRNA@mmu-mir-875, Label: 0
miRNA@mmu-mir-1907, Label: 0
miRNA@mmu-mir-1945, Label: 0

miRNA@mmu-mir-1956, Label: 0
miRNA@mmu-mir-1962, Label: 0
miRNA@mmu-mir-683-2, Label: 0
miRNA@mmu-mir-767, Label: 0

Fold 6
miRNA@mmu-mir-106a, Label: 1
miRNA@mmu-mir-106b, Label: 1
miRNA@mmu-mir-18a, Label: 1
miRNA@mmu-mir-20a, Label: 1
miRNA@mmu-mir-93, Label: 1
miRNA@mmu-mir-17, Label: 1
miRNA@mmu-mir-20b, Label: 1
miRNA@mmu-mir-18b, Label: 1
miRNA@mmu-mir-92a-2, Label: 1
miRNA@mmu-mir-92a-1, Label: 1
miRNA@mmu-mir-92b, Label: 1
miRNA@mmu-mir-19b-2, Label: 1
miRNA@mmu-mir-19a, Label: 1
miRNA@mmu-mir-19b-1, Label: 1
miRNA@mmu-mir-7a-1, Label: 1
miRNA@mmu-mir-7a-2, Label: 1
miRNA@mmu-mir-7b, Label: 1
miRNA@mmu-mir-329, Label: 1
miRNA@mmu-mir-543, Label: 1
miRNA@mmu-mir-495, Label: 1
miRNA@mmu-mir-3070a, Label: 1
miRNA@mmu-mir-3070b, Label: 1
miRNA@mmu-mir-760, Label: 1
miRNA@mmu-mir-3088, Label: 1
miRNA@mmu-mir-3100, Label: 1
miRNA@mmu-mir-3071, Label: 1
miRNA@mmu-mir-770, Label: 1
miRNA@mmu-mir-3110, Label: 1
miRNA@mmu-mir-3079, Label: 1
miRNA@mmu-mir-449a, Label: 1
miRNA@mmu-mir-679, Label: 1
miRNA@mmu-mir-362, Label: 1
miRNA@mmu-mir-452, Label: 1
miRNA@mmu-mir-1968, Label: 1

miRNA@mmu-mir-205, Label: 1
miRNA@mmu-mir-191, Label: 1
miRNA@mmu-mir-351, Label: 1
miRNA@mmu-mir-872, Label: 1
miRNA@mmu-mir-3101, Label: 1
miRNA@mmu-mir-1199, Label: 1
miRNA@mmu-mir-433, Label: 1
miRNA@mmu-mir-3082, Label: 1
miRNA@mmu-mir-322, Label: 1
miRNA@mmu-mir-652, Label: 1
miRNA@mmu-mir-684-1, Label: 0
miRNA@mmu-mir-684-2, Label: 0
miRNA@mmu-mir-686, Label: 0
miRNA@mmu-mir-717, Label: 0
miRNA@mmu-mir-1899, Label: 0
miRNA@mmu-mir-1904, Label: 0
miRNA@mmu-mir-1893, Label: 0
miRNA@mmu-mir-1958, Label: 0
miRNA@mmu-mir-432, Label: 0

Fold 7
miRNA@mmu-mir-290, Label: 1
miRNA@mmu-mir-291a, Label: 1
miRNA@mmu-mir-292, Label: 1
miRNA@mmu-mir-293, Label: 1
miRNA@mmu-mir-294, Label: 1
miRNA@mmu-mir-295, Label: 1
miRNA@mmu-mir-291b, Label: 1
miRNA@mmu-mir-465a, Label: 1
miRNA@mmu-mir-465b-1, Label: 1
miRNA@mmu-mir-465b-2, Label: 1
miRNA@mmu-mir-465c-1, Label: 1
miRNA@mmu-mir-465c-2, Label: 1
miRNA@mmu-mir-9-2, Label: 1
miRNA@mmu-mir-9-1, Label: 1
miRNA@mmu-mir-9-3, Label: 1
miRNA@mmu-mir-135b, Label: 1
miRNA@mmu-mir-135a-2, Label: 1
miRNA@mmu-mir-1264, Label: 1
miRNA@mmu-mir-421, Label: 1

miRNA@mmu-mir-219-1, Label: 1
miRNA@mmu-mir-219-2, Label: 1
miRNA@mmu-mir-146b, Label: 1
miRNA@mmu-mir-3097, Label: 1
miRNA@mmu-mir-3091, Label: 1
miRNA@mmu-mir-3076, Label: 1
miRNA@mmu-mir-3059, Label: 1
miRNA@mmu-mir-668, Label: 1
miRNA@mmu-mir-540, Label: 1
miRNA@mmu-mir-1247, Label: 1
miRNA@mmu-mir-675, Label: 1
miRNA@mmu-mir-673, Label: 1
miRNA@mmu-mir-384, Label: 1
miRNA@mmu-mir-544, Label: 1
miRNA@mmu-mir-1964, Label: 1
miRNA@mmu-mir-489, Label: 1
miRNA@mmu-mir-1193, Label: 1
miRNA@mmu-mir-3072, Label: 1
miRNA@mmu-mir-341, Label: 1
miRNA@mmu-mir-33, Label: 1
miRNA@mmu-mir-471, Label: 1
miRNA@mmu-mir-875, Label: 1
miRNA@mmu-mir-324, Label: 1
miRNA@mmu-mir-214, Label: 1
miRNA@mmu-mir-484, Label: 0
miRNA@mmu-mir-692-1, Label: 0
miRNA@mmu-mir-707, Label: 0
miRNA@mmu-mir-711, Label: 0
miRNA@mmu-mir-721, Label: 0
miRNA@mmu-mir-1903, Label: 0
miRNA@mmu-mir-1898, Label: 0
miRNA@mmu-mir-1901, Label: 0
miRNA@mmu-mir-1927, Label: 0
miRNA@mmu-mir-1940, Label: 0

Fold 8
miRNA@mmu-mir-181a-1, Label: 1
miRNA@mmu-mir-181b-1, Label: 1
miRNA@mmu-mir-181c, Label: 1
miRNA@mmu-mir-181b-2, Label: 1

miRNA@mmu-mir-181d, Label: 1
miRNA@mmu-mir-200b, Label: 1
miRNA@mmu-mir-200a, Label: 1
miRNA@mmu-mir-200c, Label: 1
miRNA@mmu-mir-429, Label: 1
miRNA@mmu-mir-297b, Label: 1
miRNA@mmu-mir-297a-3, Label: 1
miRNA@mmu-mir-297a-4, Label: 1
miRNA@mmu-mir-297c, Label: 1
miRNA@mmu-mir-34c, Label: 1
miRNA@mmu-mir-34b, Label: 1
miRNA@mmu-mir-34a, Label: 1
miRNA@mmu-mir-23a, Label: 1
miRNA@mmu-mir-204, Label: 1
miRNA@mmu-mir-211, Label: 1
miRNA@mmu-mir-208a, Label: 1
miRNA@mmu-mir-208b, Label: 1
miRNA@mmu-mir-3092, Label: 1
miRNA@mmu-mir-3066, Label: 1
miRNA@mmu-mir-3078, Label: 1
miRNA@mmu-mir-133b, Label: 1
miRNA@mmu-mir-3075, Label: 1
miRNA@mmu-mir-3081, Label: 1
miRNA@mmu-mir-383, Label: 1
miRNA@mmu-mir-485, Label: 1
miRNA@mmu-mir-202, Label: 1
miRNA@mmu-mir-701, Label: 1
miRNA@mmu-mir-298, Label: 1
miRNA@mmu-mir-335, Label: 1
miRNA@mmu-mir-3103, Label: 1
miRNA@mmu-mir-1933, Label: 1
miRNA@mmu-mir-463, Label: 1
miRNA@mmu-mir-224, Label: 1
miRNA@mmu-mir-878, Label: 1
miRNA@mmu-mir-3095, Label: 1
miRNA@mmu-mir-489, Label: 0
miRNA@mmu-mir-705, Label: 0
miRNA@mmu-mir-710, Label: 0
miRNA@mmu-mir-1896, Label: 0
miRNA@mmu-mir-1952, Label: 0

miRNA@mmu-mir-1960, Label: 0
miRNA@mmu-mir-1970, Label: 0
miRNA@mmu-mir-599, Label: 0

Fold 9
miRNA@mmu-mir-30b, Label: 1
miRNA@mmu-mir-30e, Label: 1
miRNA@mmu-mir-30c-1, Label: 1
miRNA@mmu-mir-30c-2, Label: 1
miRNA@mmu-mir-30d, Label: 1
miRNA@mmu-mir-379, Label: 1
miRNA@mmu-mir-380, Label: 1
miRNA@mmu-mir-411, Label: 1
miRNA@mmu-mir-758, Label: 1
miRNA@mmu-mir-1197, Label: 1
miRNA@mmu-mir-302a, Label: 1
miRNA@mmu-mir-302b, Label: 1
miRNA@mmu-mir-302c, Label: 1
miRNA@mmu-mir-302d, Label: 1
miRNA@mmu-mir-196a-1, Label: 1
miRNA@mmu-mir-196a-2, Label: 1
miRNA@mmu-mir-196b, Label: 1
miRNA@mmu-mir-218-1, Label: 1
miRNA@mmu-mir-218-2, Label: 1
miRNA@mmu-mir-216a, Label: 1
miRNA@mmu-mir-216b, Label: 1
miRNA@mmu-mir-138-2, Label: 1
miRNA@mmu-mir-138-1, Label: 1
miRNA@mmu-mir-296, Label: 1
miRNA@mmu-mir-3065, Label: 1
miRNA@mmu-mir-1955, Label: 1
miRNA@mmu-mir-325, Label: 1
miRNA@mmu-mir-671, Label: 1
miRNA@mmu-mir-541, Label: 1
miRNA@mmu-mir-582, Label: 1
miRNA@mmu-mir-615, Label: 1
miRNA@mmu-mir-330, Label: 1
miRNA@mmu-mir-676, Label: 1
miRNA@mmu-mir-665, Label: 1
miRNA@mmu-mir-1947, Label: 1

miRNA@mmu-mir-547, Label: 1
miRNA@mmu-mir-340, Label: 1
miRNA@mmu-mir-203, Label: 1
miRNA@mmu-mir-147, Label: 1
miRNA@mmu-mir-503, Label: 1
miRNA@mmu-mir-1839, Label: 1
miRNA@mmu-mir-361, Label: 1
miRNA@mmu-mir-3067, Label: 1
miRNA@mmu-mir-370, Label: 1
miRNA@mmu-mir-708, Label: 1
miRNA@mmu-mir-704, Label: 0
miRNA@mmu-mir-713, Label: 0
miRNA@mmu-mir-718, Label: 0
miRNA@mmu-mir-509, Label: 0
miRNA@mmu-mir-654, Label: 0
miRNA@mmu-mir-1900, Label: 0
miRNA@mmu-mir-1936, Label: 0
miRNA@mmu-mir-1951, Label: 0
miRNA@mmu-mir-1963, Label: 0
miRNA@mmu-mir-1967, Label: 0

B.2 HSA Folds

Fold 0
miRNA@hsa-mir-1277, Label: 1
miRNA@hsa-mir-466, Label: 1
miRNA@hsa-mir-568, Label: 0
miRNA@hsa-mir-1277, Label: 0
miRNA@hsa-mir-466, Label: 0

Fold 1
miRNA@hsa-mir-103a-2, Label: 1
miRNA@hsa-mir-103a-1, Label: 1
miRNA@hsa-mir-128-1, Label: 1
miRNA@hsa-mir-190a, Label: 1
miRNA@hsa-mir-128-2, Label: 1
miRNA@hsa-mir-365a, Label: 1
miRNA@hsa-mir-365b, Label: 1

miRNA@hsa-mir-450a-1, Label: 1
miRNA@hsa-mir-450a-2, Label: 1
miRNA@hsa-mir-551a, Label: 1
miRNA@hsa-mir-551b, Label: 1
miRNA@hsa-mir-670, Label: 1
miRNA@hsa-mir-450b, Label: 1
miRNA@hsa-mir-874, Label: 1
miRNA@hsa-mir-190b, Label: 1
miRNA@hsa-mir-103b-1, Label: 1
miRNA@hsa-mir-103b-2, Label: 1
miRNA@hsa-mir-1912, Label: 1

Fold 2
miRNA@hsa-mir-323a, Label: 1
miRNA@hsa-mir-592, Label: 1
miRNA@hsa-mir-655, Label: 1
miRNA@hsa-mir-656, Label: 1
miRNA@hsa-mir-1185-2, Label: 1
miRNA@hsa-mir-1185-1, Label: 1
miRNA@hsa-mir-300, Label: 1
miRNA@hsa-mir-323b, Label: 1
miRNA@hsa-mir-449a, Label: 0
miRNA@hsa-mir-449b, Label: 0
miRNA@hsa-mir-449c, Label: 0

Fold 3
miRNA@hsa-mir-320a, Label: 1
miRNA@hsa-mir-513a-1, Label: 1
miRNA@hsa-mir-513a-2, Label: 1
miRNA@hsa-mir-509-1, Label: 1
miRNA@hsa-mir-514a-1, Label: 1
miRNA@hsa-mir-514a-2, Label: 1
miRNA@hsa-mir-514a-3, Label: 1
miRNA@hsa-mir-320b-1, Label: 1
miRNA@hsa-mir-320c-1, Label: 1
miRNA@hsa-mir-320b-2, Label: 1
miRNA@hsa-mir-509-2, Label: 1
miRNA@hsa-mir-876, Label: 1
miRNA@hsa-mir-509-3, Label: 1
miRNA@hsa-mir-513b, Label: 1

miRNA@hsa-mir-513c, Label: 1
miRNA@hsa-mir-320c-2, Label: 1
miRNA@hsa-mir-3074, Label: 1
miRNA@hsa-mir-514b, Label: 1
miRNA@hsa-mir-574, Label: 0
miRNA@hsa-mir-297, Label: 0
miRNA@hsa-mir-3149, Label: 0

Fold 4
miRNA@hsa-mir-122, Label: 1
miRNA@hsa-mir-124-1, Label: 1
miRNA@hsa-mir-124-2, Label: 1
miRNA@hsa-mir-124-3, Label: 1
miRNA@hsa-mir-301a, Label: 1
miRNA@hsa-mir-653, Label: 1
miRNA@hsa-mir-764, Label: 1
miRNA@hsa-mir-744, Label: 1
miRNA@hsa-mir-301b, Label: 1
miRNA@hsa-mir-1251, Label: 1
miRNA@hsa-mir-664a, Label: 1
miRNA@hsa-mir-3064, Label: 1
miRNA@hsa-mir-3591, Label: 1
miRNA@hsa-mir-664b, Label: 1
miRNA@hsa-mir-762, Label: 0

Fold 5
miRNA@hsa-mir-376c, Label: 1
miRNA@hsa-mir-499a, Label: 1
miRNA@hsa-mir-500a, Label: 1
miRNA@hsa-mir-532, Label: 1
miRNA@hsa-mir-660, Label: 1
miRNA@hsa-mir-802, Label: 1
miRNA@hsa-mir-1298, Label: 1
miRNA@hsa-mir-892a, Label: 1
miRNA@hsa-mir-890, Label: 1
miRNA@hsa-mir-888, Label: 1
miRNA@hsa-mir-892b, Label: 1
miRNA@hsa-mir-500b, Label: 1
miRNA@hsa-mir-892c, Label: 1
miRNA@hsa-mir-767, Label: 0

miRNA@hsa-mir-875, Label: 0

Fold 6
miRNA@hsa-mir-92a-1, Label: 1
miRNA@hsa-mir-92a-2, Label: 1
miRNA@hsa-mir-449a, Label: 1
miRNA@hsa-mir-92b, Label: 1
miRNA@hsa-mir-652, Label: 1
miRNA@hsa-mir-449b, Label: 1
miRNA@hsa-mir-449c, Label: 1
miRNA@hsa-mir-770, Label: 1
miRNA@hsa-mir-543, Label: 1
miRNA@hsa-mir-760, Label: 1

Fold 7
miRNA@hsa-mir-33a, Label: 1
miRNA@hsa-mir-371a, Label: 1
miRNA@hsa-mir-544a, Label: 1
miRNA@hsa-mir-33b, Label: 1
miRNA@hsa-mir-421, Label: 1
miRNA@hsa-mir-1264, Label: 1
miRNA@hsa-mir-668, Label: 1
miRNA@hsa-mir-675, Label: 1
miRNA@hsa-mir-875, Label: 1
miRNA@hsa-mir-1247, Label: 1
miRNA@hsa-mir-3120, Label: 1
miRNA@hsa-mir-544b, Label: 1
miRNA@hsa-mir-1193, Label: 1
miRNA@hsa-mir-371b, Label: 1
miRNA@hsa-mir-711, Label: 0

Fold 8
miRNA@hsa-mir-208a, Label: 1
miRNA@hsa-mir-298, Label: 1
miRNA@hsa-mir-208b, Label: 1
miRNA@hsa-mir-297, Label: 1
miRNA@hsa-mir-3149, Label: 1
miRNA@hsa-mir-599, Label: 0

Fold 9

miRNA@hsa-mir-147a, Label: 1
miRNA@hsa-mir-203a, Label: 1
miRNA@hsa-mir-216a, Label: 1
miRNA@hsa-mir-582, Label: 1
miRNA@hsa-mir-615, Label: 1
miRNA@hsa-mir-411, Label: 1
miRNA@hsa-mir-758, Label: 1
miRNA@hsa-mir-671, Label: 1
miRNA@hsa-mir-541, Label: 1
miRNA@hsa-mir-708, Label: 1
miRNA@hsa-mir-147b, Label: 1
miRNA@hsa-mir-665, Label: 1
miRNA@hsa-mir-216b, Label: 1
miRNA@hsa-mir-1197, Label: 1
miRNA@hsa-mir-3065, Label: 1
miRNA@hsa-mir-676, Label: 1
miRNA@hsa-mir-203b, Label: 1
miRNA@hsa-mir-513a-1, Label: 0
miRNA@hsa-mir-513a-2, Label: 0
miRNA@hsa-mir-509-1, Label: 0
miRNA@hsa-mir-514a-1, Label: 0
miRNA@hsa-mir-514a-2, Label: 0
miRNA@hsa-mir-514a-3, Label: 0
miRNA@hsa-mir-654, Label: 0
miRNA@hsa-mir-509-2, Label: 0
miRNA@hsa-mir-509-3, Label: 0
miRNA@hsa-mir-513b, Label: 0
miRNA@hsa-mir-513c, Label: 0
miRNA@hsa-mir-718, Label: 0
miRNA@hsa-mir-514b, Label: 0