U N I V E R S I T Y OF T A R T U
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Kaur Alasoo

Combining Support Vector Machines to Predict Novel Angiogenesis Genes

Bachelor's thesis

Supervisors: Hedi Peterson, MSc
Phaedra Agius, PhD

TARTU 2010

Contents

Introduction

Cancer is currently one of the most widespread and deadly diseases accounting for 13% of all deaths worldwide in 2004 [23]. Despite the extensive research efforts over the last decades, it still cannot be neither understood nor cured very effectively. It is known that angiogenesis or the development of blood vessels plays an import role in tumour growth [22]. Therefore, better understanding of this process could potentially help us to find new and more efficient ways to treat cancer.

In the human genome, there are about 22000 genes out of which approximately 350 are known to be involved in angiogenesis. In addition, biologists have measured the expression of all human genes in thousands of different conditions. In this work, our goal is to use these measurements to predict which other of the 22000 genes could also play a role in blood vessel development and prioritise them according to their likelihood. Computational prioritisation of candidate genes is an important question because conducting biological experiments with all genes would not be physically nor economically possible.

Our work is a part of a larger EU FP6 project ENFIN subproject called "Characterize the Human angiogenic sub-network". The experimental labs of Prof. Francesca Sanches in Malaga and Prof. Christine Orego at University College London (UCL) have provided us with a *curated list* of 341 known angiogenesis genes. They have compiled the list based on the information gathered from Gene Ontology [3], literature and commercial microarrays. We will use this list to make our predictions and afterwards we will send the results to our ENFIN collaborators in Spain. There they will combine our results with predictions from two other labs. Finally, they will conduct

biological experiments to determine which genes are really involved in angiogenesis.

The first two chapters of this thesis cover the basic biological and machine learning concepts used throughout this work. In Chapter 3, we compare various state-of-the-art methods and bioinformatics tools that can be used to prioritise candidate genes for a specific biologically process. In the next chapter, we propose a novel machine learning approach that combines results from many separately trained Support Vector Machines (SVM) into one prediction. Finally, we show that the new method statistically outperforms the existing ones and that the new predicted angiogenesis genes are indeed biologically relevant.

Chapter 1

Biological Background

1.1 Biology Behind Angiogenesis

Angiogenesis is the growth and development of blood vessels. Normally it takes place during embryonic development, formation of *corpus luteum*[1], regeneration, and wound healing. Additionally, it has been shown to play a big role in cancer development.

First of all, without angiogenesis tumours would not be able to grow larger than 1-2 millimeters in diameter because of lack of oxygen and nutrients. Experiments with mice [17] have shown that if tumour cells are not able to induce blood vessel development in the neighbouring cells, they cannot grow bigger and become dangerous to the organism. Secondly, angiogenesis makes it possible for cancer to spread and reach other parts of the body using normal blood circulation. This is called metastasis.

A simplified description of the process is following. First, cancer cells produce and release various molecules that start angiogenesis. Some of the most well known molecules are Basic Fibroblast Growth Factor (bFGF) and Vascular Endothelial Growth Factor (VEGF). These molecules in turn activate endothelial cells (blood vessel cells) which causes them to split and form new blood vessels. This is illustrated on Figure 1.1. For further information on mechanism of angiogenesis, please see [7].

[1] A yellow mass of tissue that forms in the ovary after ovulation. It is involved in the production of progestogen, which is needed to sustain a healthy pregnancy.

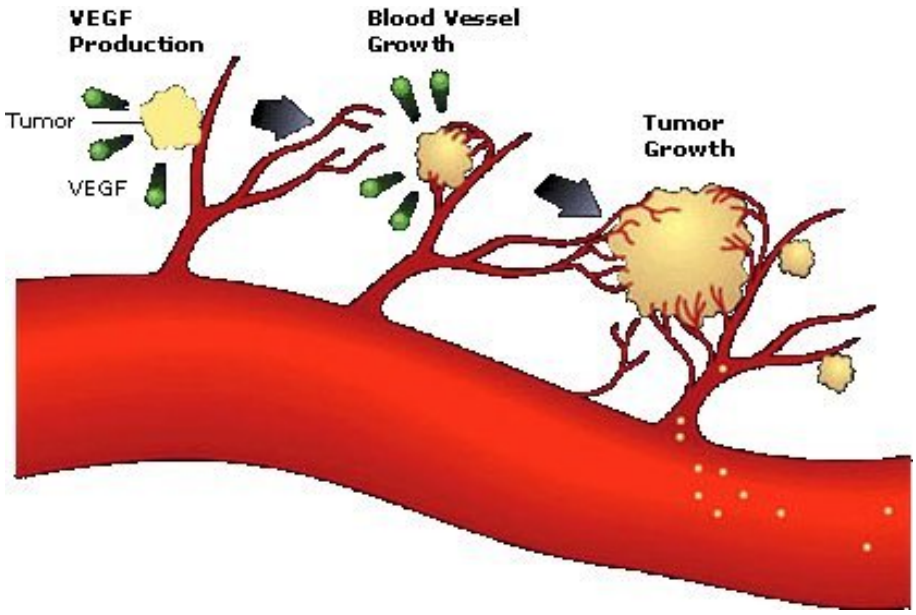


Figure 1.1: The role of angiogenesis in cancer. Image taken from [14].

1.2 Gene Expression Experiments

Molecular functions of a cell, such as initiating the growth of blood vessels, are carried out by proteins. Proteins are translated from messenger RNA (mRNA) which in turn is transcribed from the genes on the DNA. Genes are in essence continuous regions of DNA that are used to encode proteins. While it is very difficult to measure the level of proteins in a cell, it is much easier to measure the amount of mRNA. This is the main idea behind gene expression microarray experiments. Although it is an indirect method, it can still give reasonable estimation of the level of proteins. Microarrays have become popular in the recent years because they are relatively cheap and allow to measure the expression of thousands of genes in one experiment. There are many companies in this field, each having their own slight technological differences but the main idea is the same. Our data is obtained from Affymetrix [10] chips, so we give a short overview of this technology.

A gene expression chip is designed as follows. First, a set of specifically

chosen oligonucleotides (25-mers) called probes[2] are printed to a solid surface (glass, plastic or silicon chips) in an orderly manner. The resulting chips are called microarrays. To measure the amount of mRNA, it is first extracted from the cells of interest. This could be a normal cell, a cancer cell or a cell infected by some other disease. Next, the mRNA is converted to complementary DNA (cDNA) using the reverse transcriptase enzyme. This enzyme works by scanning the mRNA and synthesizing its nucleotides (A, U, G and C) into their DNA complements (T, A, C and G respectively). When this is done, a fluorescent dye is added to the cDNA to make it possible to measure its presence.

Finally, the microarray is hybridized with the cDNA prepared as explained above. In this process the cDNA binds to its complementary oligonucleotides on the chip. A digital image of the microarray is made with laser scanners and analyzed with a computer. Because the oligonucleotides were printed onto the array in a fixed order, it is now possible to measure the amount of mRNA transcribed from different genes by looking at which places on the array the cDNA bound the most i.e which locations had the most colour. Results from different probes are aggregated into probe sets and finally each probe set gets a numerical value. This number does not show the absolute amount of mRNA, but rather how much one specific probe set is over- or under-expressed compared to all other probe sets. The steps of a typical microarray experiment are illustrated on Figure 1.2.

In the perfect world each probe set would correspond to a single gene. However, because some genes on DNA can overlap and the microarrays were often designed before there was much knowledge about the existence and placement of all of the genes, it sometimes occurs that one probe set actually measures the expression of many genes.

1.3 Used Data Set

In our study, we used a single very large Affymetrix microarray gene expression matrix. The data set was combined in European Bioinformatics

[2]short strings of DNA, that in our case are 25 nucleotides long

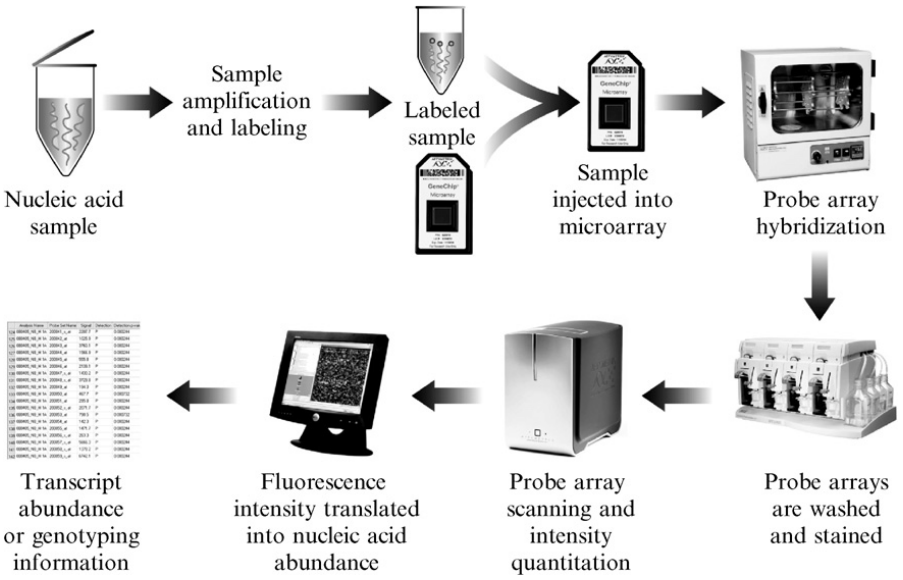


Figure 1.2: The steps of a microarray experiment. Image taken from [10].

Institute (EBI) by Margus Lukk [20]. It consists of 22283 probe sets and 5372 experiments obtained from ArrayExpress gene expression database [4]. From a computer science perspective it is a 22283 by 5372 real valued matrix in which rows correspond to probe sets and columns to experiments. On the intersection of each row and column is the expression value of one probe set in one experiment. The expression values for each experiment were obtained in the manner described in Section 1.2. Because of the reasons already mentioned in Section 1.2 one probe set can correspond to many genes and *vice versa*, one gene might be measured by many probe sets. The data set is available for download from the ArrayExpress database with the id E-MTAB-62.

Before conducting any experiments, we normalized the data row-wise. This means that for each value in each row we subtracted the mean of the row from it and divided it by the standard deviation of the row.

1.4 Pre-processing of the Curated List

There were three reasons why the curated list of genes provided by the experimental labs needed to be pre-processed. First, we wanted to exclude the genes that were added to the curated list based on microarray data. It was necessary, because we planned to use microarray data in our experiments and we did not want the predictions of other methods on the same type of data to influence our results. This left us with 200 genes instead of the initial 341.

Secondly, we wanted to include some additional previously known angiogenesis genes that are used on a commercially available angiogenesis PCR[3] array. PCR array is yet another technology that can be used to measure the expression of genes. We assumed that if a gene has been put onto a commercially available angiogenesis PCR array then there should be enough proof for that gene to be related to angiogenesis.

As a third step, we had to convert the gene names, because in the curated list the genes were given by their name, but in our data set we had probe set identifiers. The g:Convert tool, part of the g:Profiler [24] web-tool, was used to achieve this. During this step we also removed probe sets that were corresponding to more than one gene, because there was no way to make sure which gene they were actually measuring.

As a result, we obtained a list of 405 probe set identifiers that we used as the training set for all the methods that we studied. We ended up with more probe set identifiers than was the initial number of genes, because some genes corresponded to multiple probe sets.

[3]polymerase chain reaction, a method used to amplify a single or few copies of a piece of DNA across several orders of magnitude

Chapter 2

Machine Learning Background

One of the aims of machine learning (ML) is to generalise the knowledge present in examples and make statistically sound predictions based on that. The examples that we are trying to learn from are usually referred to as the *training set*. Mathematically, we can represent the training set in the following way:

$$S = \{(x_1, y_1), ..., (x_k, y_k), x_i \in X, y_i \in Y\}$$

where X and Y are some properties of the phenomena under study. Usually the values of X are known for all data points, but the values of Y are known for only the samples in the training set. The goal is then to find a function $f : X \rightarrow Y$ which has low *training error* (number of training samples for which $f(x_i) \neq y_i$) and at the same time performs well in predicting the values of Y for new data points.

For example, X could be a set of vectors each measuring the expression of a single gene in many different conditions and Y could consist of labels $\{-1, +1\}$ indicating which of these genes are involved in angiogenesis $(+1)$ and which are not (-1). Machine learning algorithm would then try to learn the relationship between the vectors and their corresponding labels, or more specifically, predict which genes are related to angiogenesis.

In the first part of this chapter, we will give a short introduction to Support Vector Machine (SVM), a popular and well-known machine learning algorithm. In the second part, we will also cover some standard techniques

that are used to compare different ML algorithms.

2.1 Support Vector Machine Classification

SVM is a machine learning algorithm that based on a training set with known labels (classes) can learn a classifier and is thereafter able to predict on new samples to which class they belong. Formally this corresponds to $X = \mathbb{R}^d$ and $Y = \{-1, 1\}$, where $\mathbb{R}^d$ denotes all real-valued vectors of length d. In this chapter we will give only a short overview of this algorithm. For more detailed tutorial please see [5].

2.1.1 The Separable Case

Let us first assume that points in our training set are linearly separable. This means that it is possible to draw an hyperplane between samples that belong to the positive class (label is $+1$) and to the negative class (label is -1). Now, for any new data points we can just look at which side of the hyperplane they are and predict their class membership based on that. Notice also that there might be many such hyperplanes that separate the positive class from the negative class. Suppose that one of those hyperplanes is given by $\mathbf{x} \cdot \mathbf{w} + b = 0$, where $\mathbf{x} \in \mathbb{R}^d$ is a point lying on the hyperplane and $\mathbf{w} \in \mathbb{R}^d$ is normal to the hyperplane. Let d_+ and d_- be the distances from the hyperplane to the closest positive and to the closest negative examples respectively. Let the *margin* of the hyperplane be $d_+ + d_-$. The SVM algorithm simply tries to find the hyperplane with the largest margin.

The idea of finding the maximal margin hyperplane can be formalized in the following way: suppose that all training data satisfy the following constraints:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geqslant +1 \text{ for } y_i = +1 \tag{2.1}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leqslant -1 \text{ for } y_i = -1 \tag{2.2}$$

which can be combined into

$$\forall i \; y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geqslant 0 \tag{2.3}$$

Now, let us look at the points for which the equality in Eq. (2.1) holds. These points lie on the hyperplane $H_1 : \mathbf{x}_i \cdot \mathbf{w} + b = 1$ with normal $\mathbf{w}$ and distance from the origin $|1 - b|/||\mathbf{w}||$. Similarly, the points for which the equality in Eq. (2.2) holds lie on the hyperplane $H_2 : \mathbf{x}_i \cdot \mathbf{w} + b = -1$ with normal $\mathbf{w}$ and distance from the origin $| - 1 - b|/||\mathbf{w}||$. Hence $d_+ = d_- = 1/||\mathbf{w}||$ and the margin is simply

$$\frac{2}{||\mathbf{w}||}.$$

Thus we can find the pair of hyperplanes which gives the maximum margin by minimizing $||\mathbf{w}||^2$, subject to the constraints (2.3). For a typical two dimensional case, this situation is illustrated in Figure 2.1. Training points that lie on the hyperplanes H_1 and H_2 are called *support vectors*.

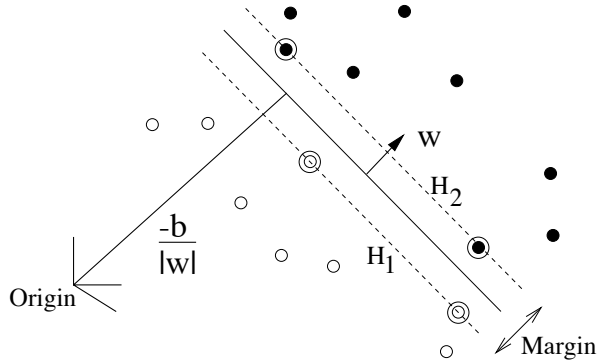


Figure 2.1: Linear separating hyperplanes for the separable case. The support vectors are circled. An illustration taken from [5].

To summarise, we have the following optimization problem:

$$\min_{b,\mathbf{w}} \mathbf{w} \cdot \mathbf{w}, \tag{2.4}$$

subject to

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geqslant 0 \tag{2.5}$$

2.1.2 The Non-Separable Case

Due to noise or other factors the data in hand is often not linearly separable but we would still like to learn the best classifier. One way to solve this is to introduce *slack variables* ξ_i, which allow some points to lie on the wrong side of the hyperplane. This results in the following optimization problem:

$$\min_{b,\mathbf{w},\xi_i} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{l} \xi_i, \tag{2.6}$$

subject to

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geqslant 1 - \xi_i, \xi_i \geqslant 0 \tag{2.7}$$

The parameter C is a free parameter chosen by the user and a larger C corresponds to assigning a higher penalty to errors. In practice, the best C is usually determined by empirical methods such as k-fold cross-validation or bootstrapping and choosing the value that minimizes error.

2.1.3 Finding the Optimal Hyperplane

The separable case is a special case of the non-separable case where there are no slack variables. Therefore, we present the solution only for the latter. Based on optimisation theory, it is possible to show that the optimisation problem of the non-separable case is equivalent to

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j, \tag{2.8}$$

subject to

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \tag{2.9}$$

$$\forall i \; C \geqslant \alpha_i \geqslant 0 \tag{2.10}$$

Given the parameters of the optimal hyperplane α_i^* and b^*, class of the point $\mathbf{x}$ can be determined by

$$y = sign(\sum_{i=1}^{l} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^*)$$

and the SVM *decision value* used many times throughout this thesis is simply the value of

$$dv(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^*.$$

The maximal margin hyperplane can be found using numerical methods that normally converge to global optimum. Fortunately, there are several SVM libraries available that do specifically that. In our experiments, we used *e1071* package for R that is based on LIBSVM [8].

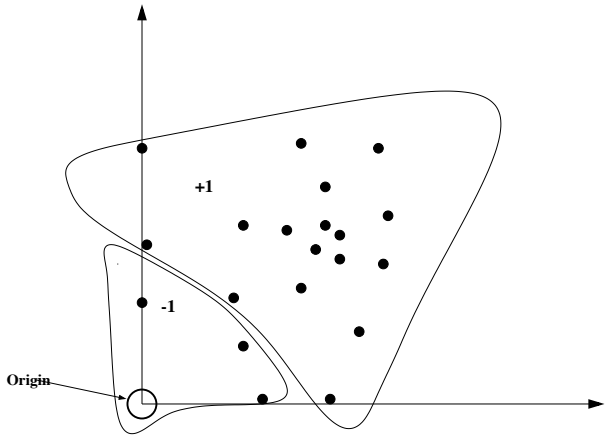


Figure 2.2: One-Class SVM Classifier. The origin is initially the only member of the negative class. An illustration taken from [21].

2.2 One-Class SVM

On of the problems that we had was that the curated list of angiogenesis genes only contained positive examples and no negative ones. Therefore, we also looked for methods that did not require negative training set. One-Class SVM also known as novelty detection SVM was first proposed by Schölkopf *et al* in [25]. The main difference from the regular SVM is that instead of constructing a hyperplane between the two classes it tries to separate the interesting class from the origin. This idea is illustrated on Figure 2.2.

Formally speaking, the familiar optimisation problem now becomes the following:

$$\min_{b,w,\xi_i,\rho} \mathbf{w} \cdot \mathbf{w} + \frac{1}{\nu l} \sum_{i=1}^{l} \xi_i - \rho \tag{2.11}$$

subject to

$$\forall i \ (\mathbf{x}_i \cdot \mathbf{w}) \geqslant \rho - \xi_i, \xi_i \geqslant 0 \tag{2.12}$$

Here $\nu \in \{0, 1\}$ is a parameter specified by the user regulating how far from the origin the hyperplane will be constructed. Finding the optimal solution and the corresponding decision function is analogous to the two-class case. This algorithm is also implemented in the LIBSVM [8] package.

2.3 Comparing Machine Learning Methods

Throughout this work we use 10-fold cross-validation to assess and compare the performance of different algorithms. 10-fold cross-validation means that the training data is divided into ten non-overlapping sets. Depending on the algorithm, the training genes can either consist of only angiogenesis genes or also include the negative set. At each iteration, nine of these sets are used for training and one for testing. The performance measures are computed from the testing results only. This ensures that our classifier works well not only on the training data but also on new, previously unseen data points.

2.3.1 Confusion Matrix

In statistics, a confusion matrix (Table 2.1) is often used to illustrate the performance of a classification algorithm. It shows how many of the actual data points are predicted correctly and what types of mistakes are made with the ones that are not assigned to the appropriate class. One of the merits of the confusion matrix is that it gives very much information about the classifier.

On the other hand, many classification algorithms let the user to specify a threshold at which an instance is considered to be in the positive or

the negative class. For an example, let us look at the SVM that uses the decision value to determine the class. By default, the threshold is set to 0 and instances that have higher or lower decision values are classified into positive and negative classes respectively. Sometimes we might want to raise (lower) the threshold to make out method more (less) stringent. In that case, a separate confusion matrix is needed for each threshold value, which makes it difficult to see how the accuracy of the classifier changes when the classification threshold is modified. This is especially important in the case of ranking, where we essentially have as many different thresholds as there are data points.

Table 2.1: Confusion matrix.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

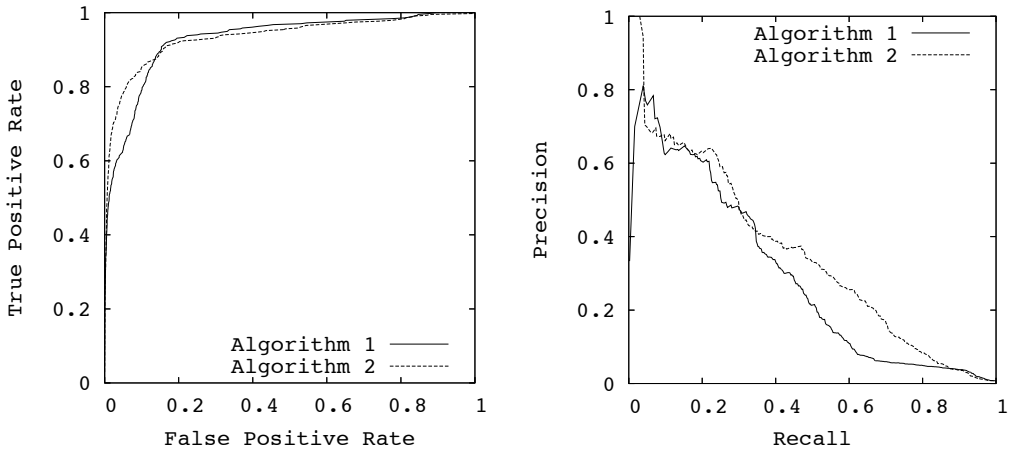
2.3.2 Precision-Recall and ROC Curves

Two most popular ways to compare classifiers and to see how they perform at different cut-offs are the Receiver Operator Characteristic (ROC) curve and the Precision-Recall curve.

On the ROC curve the True Positive Rate (Eq. 2.13) is plotted against the False Positive Rate (Eq. 2.14) calculated at each cut-off. To compare two different classifiers usually areas under these curves (AUC) are computed. A random classifier would have an area equal to 0.5 and a perfect classifier would have an area equal to 1.

$$\text{True Positive Rate} = \frac{TP}{TP + FN} \tag{2.13}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{2.14}$$



(a) ROC Curve

(b) Precision-Recall Curve

Figure 2.3: Comparing the performance of two algorithms using ROC and Precision-Recall curves.

On the Precision-Recall curve, as the name indicates, Precision (Eq. 2.15) is plotted against Recall (Eq. 2.16). Although the ROC curve is more popular in the Machine Learning community, Precision-Recall curve might be more suitable [11] in our case, because we have a very small positive set and a very large unlabeled set. Although, often larger area under the ROC curve also means a larger area under the Precision-Recall curve, it has been shown that it is not always so [11]. For an illustration of these curves, see Figure 2.3.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.16}$$

Chapter 3

Application of Standard Methods

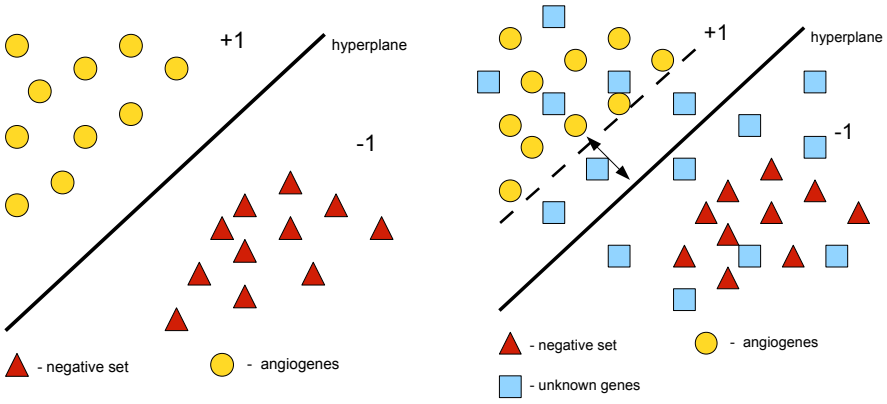
In this chapter, we will give a short overview of some of the existing methods that we applied to our data. We divide them into two groups based on whether they are approaching this task as a classification problem or as a ranking problem.

3.1 Classification Approach

One way to approach the problem of predicting new candidate genes is to consider it as a classification problem. In that case we have positive examples and negative examples and the goal is to find a classifier that can separate them into two classes most accurately. The genes in the curated list are positive examples. Defining the negative examples, however, is more difficult, because we only have a large set of genes about which we essentially do not know anything (*unlabeled genes*). One option is to use prior knowledge and try to fix some genes as negative examples. The problem with this approach is that it is often almost impossible to know, which genes are definitely not involved in the process of interest. To overcome it, many methods have been developed that either do not require negative examples at all (One-Class SVM [21]) or learn negative examples from the data (Roc-SVM [19], PEBL [30], PSoL [29]).

3.1.1 Binary SVM

The first approach that we tried was a simple binary SVM. As the negative set we decided to use known housekeeping genes. Housekeeping genes are genes involved in basic functions needed for the sustenance of the cell and should therefore be constantly expressed over various conditions. Intuitively, this property makes them potentially good candidates to be considered as the negative set. The list of housekeeping genes was obtained from [12]. In the end, after pre-processing the list similarly as described in Section 1.4, we had 1232 probe set identifiers. We trained the SVM on angiogenesis and housekeeping genes and then used it to classify all unlabeled genes. These steps are illustrated on Figure 3.1. Results are presented in Chapter 6.



(a) Training the SVM (b) Predicting new genes

Figure 3.1: Training and prediction phases of the SVM.

3.1.2 One-Class SVM

The second method we applied was One-Class SVM that did not require a pre-defined strong negative set which we did not have. In our experiments, we first determined the best value for ν parameter, which controls how far from the origin the hyperplane will be constructed. This was done by varying ν from 0.05 to 0.95 in 0.05 steps and using 10-fold cross-validation at each

step to check the performance. Finally, we used the value for ν that had the lowest cross-validation error and classified all other genes in our data set. The results are presented in Chapter 6.

3.1.3 Roc-SVM

This section is inspired by a method developed by Li and Liu [19] and modified for gene expression data. It consists of two steps. In the first step, a simple and computationally cheap method is used to extract strong negative samples from the unlabeled data. In the second part, the strong negative genes and known angiogenesis genes are used to train the SVM, which is then in turn used to predict new candidates.

The Rocchio's classifier was used as the initial weak classifier, because it was shown in [19] to perform well on predicting strong negative examples without too many false negatives. The main idea is to first create two separate prototype vectors for positive and unlabeled examples. After that, all data points are assigned to the class to which prototype vector they are most similar to. Rocchio's classifier cannot be used as the final classifier, because there will be too many false positives. Longer analysis of the suitability of this approach for this particular task can be found in [19].

The modified algorithm is the following. Strong negative set is denoted by N.

1. Assign the unlabeled genes (U) to the negative class and the angiogenesis genes (A) to the positive class.

2. Find the prototype vector $\overrightarrow{\beta}_u$ for the unlabeled class from $\overrightarrow{\beta}_u = \overrightarrow{u} - b\overrightarrow{a}$ where $\overrightarrow{u}$ and $\overrightarrow{a}$ are the centroids of unlabeled and positive class and b is the weight of the positive centroid relative to the unlabeled centroid.

3. Find the prototype vector $\overrightarrow{\beta}_p$ for the positive class analogically from $\overrightarrow{\beta}_p = \overrightarrow{a} - b\overrightarrow{u}$.

4. **for** each vector $\overrightarrow{v}$ in U **do**

5. **If** $\text{sim}(\overrightarrow{\beta}_p, \overrightarrow{v}) \leq \text{sim}(\overrightarrow{\beta}_u, \overrightarrow{v})$

6. $N = N \cup \{\overrightarrow{v}\}$

Here *sim* stands for any reasonable similarity measure. In our work the Pearson's correlation was used. The results obtained are presented in Chapter 6.

3.2 Ranking Approach

Another way to approach candidate gene finding is to see it as a ranking or prioritisation problem. In this section we will describe one SVM based algorithm and two publicly available tools that can be used to rank candidate genes.

3.2.1 Multi Experiment Matrix

Multi Experiment Matrix (MEM) [1] is a web-based gene expression analysis and visualisation tool that gathers hundreds of publicly available data sets from ArrayExpress[4]. Given a gene as an input, MEM first ranks all other genes according to their similarity in each individual data set. These different rankings are then combined into one ranked list using BetaMEM [18] algorithm described in Section 4.4.3.

To predict novel angiogenesis genes, we first used MEM to rank all other genes according to their similarity to each angiogenesis gene. To obtain one candidate list for all angiogenesis genes we applied BetaMEM once again to the ranked lists created in the first step. For results, please see Chapter 6.

3.2.2 Endeavour

Endeavour[2] is a web-based tool for candidate gene prioritisation that uses a set of training genes (genes known to play a role in the process of interest). The main idea is the following. First, information about training genes is collected from various data sources including functional annotations, protein-protein interactions, regulatory information, expression data, sequence based data and literature mining data. Next, models are built based on the training genes and all different types of data. Finally, the models are used to score candidate genes and rank them according to their similarity to

training genes. In the very last step, all different rankings from distinct data types are aggregated into one ranking using order statistics. In the tool, it is possible to either specify a limited set of candidate genes to prioritise or use the whole genome.

So far it has been successfully used for example to identify genes related to osteporosis [16], ataxia (the loss of full control of bodily movements) [27] and bone mineral density variation [9]. For application details and results with angiogenesis genes, please see Chapter 6.

3.2.3 Binary SVM With Unlabeled Data

Yet another approach is to train a binary SVM using angiogenesis genes as positive examples and all other genes (which also contain some unknown angiogenesis genes) as negative examples. From here onward, we refer to this method as `All negative`. In the article [13], the authors prove that if the known positive examples are selected randomly from the all true positive examples that are there in the data set, then the classifier trained in this manner predicts probabilities that differ by only a constant factor from the true conditional probabilities of being positive. This means that instead of using only true negative examples we can also incorporate unlabeled genes into the negative training set for the SVM. As a result, the absolute values of predicted decision values for new positive examples do change, but the ranking of the genes stays the same. By ignoring the absolute decision values and looking only at their ranking we essentially get a ranking algorithm. Comparison of this method to the others is presented in Chapter 6.

Chapter 4

The Comb-SVM Algorithm

After getting poor prediction accuracy from many pre-existing algorithms, we devised a novel algorithm that we will describe in detail in this chapter.

The Comb-SVM algorithm is based on SVM classification and consists of four steps. First, we choose 100 random negative sets. Then, we train 100 SVM classifiers using the training sets consisting of angiogenesis genes and each negative set generated in the previous step. Next, we classify all genes in our data set using these 100 classifiers. Finally, we aggregate the classification results into one ranked list. These steps are also illustrated on Figure 4.1.

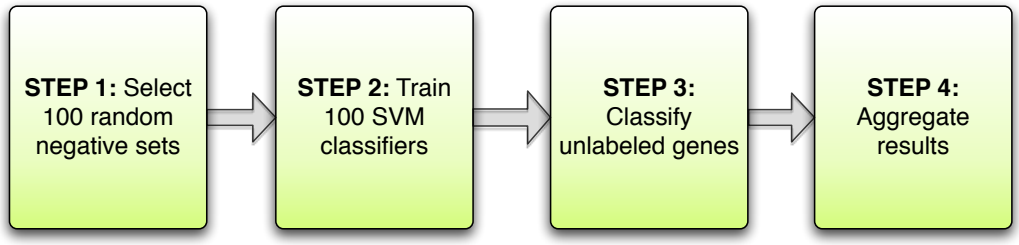


Figure 4.1: Four steps of the Comb-SVM algorithm.

4.1 Generating Negative Sets

We used randomly chosen negative sets because it is biologically very difficult to identify a reliable set of genes that are certainly not related to specified biological process e.g. angiogenesis. In addition, if one is successful at finding such a negative set, this set might be "too" negative and as a result too many genes will be predicted to belong to the positive class. This could have been the case when we used binary SVM classification and housekeeping genes as the negative set. In addition, using only one fixed negative set makes it more difficult to rank the predictions. Finally, it is almost impossible to identify whether a gene was assigned to the positive class because it was actually similar to the positive examples or just very dissimilar from the negative examples that were used to train the model.

We generated the negative sets by randomly selecting genes from the pool of all genes excluding our list of known angiogenesis genes. The size of each negative set was the same as the size of the positive training set, i.e. in the case of angiogenesis it was 405 probe sets. We tried increasing it by up to 6 times but the results did not change significantly, so we decided to keep the size fixed. We repeated the process 100 times, because our experiments showed that this was enough for the list of predicted genes to remain stable ,i.e. two different selections of random negative sets resulted in essentially the same predictions. It might well be that a smaller number of randomly chosen negative sets would have also been sufficient, but exploring this was out of the scope of this work.

4.2 Training 100 SVM Classifiers

To conduct the experiments, we used LIBSVM [8] and its R interface that is part of the *e1071* package. We trained 100 SVM classifiers using known angiogenesis genes and each negative set generated in the previous step. We used linear kernel with parameter C fixed to 1. We also tried Gaussian kernel and various values for C, put the differences were not significant. This conforms with the article from Zhang et al [31], were they concluded that data is usually far from sufficient for reliably estimating nonlinear relations

for microarray data.

4.3 Classifying Unlabeled Genes

In order to classify all genes in the data set we used SVM classifiers trained in the previous step. Additionally to noting class assignments (whether a gene was classified as belonging to angiogenesis or to the negative set) of each classifier, we also stored the SVM decision values and class probabilities (probability of a gene belonging to one class or to the other). We used this information in the next step to aggregate the results and create one ranked list.

4.4 Aggregating Classification Results

We tried various methods to aggregate the classification results obtained from 100 randomly chosen negative sets. In this section we give a short overview of all of these methods and compare their performance. Finally, we choose the best one to make the final predictions in the Results section.

4.4.1 Naïve Approach: Sum of SVM Decision Values

The first approach that we tried was to take the SVM decision values from all experiments and simply find the sum of those for each gene. In the end, we would rank all genes according to this sum and the genes having the highest score would be the strongest candidates. The SVM decision values usually ranged from -3 to +3. Because many of the training angiogenesis genes were also support vectors, then most of them had scores around +1.

To take this into account, we made two modifications to this algorithm. In the first case, we ignored the negative decision values. This meant that always when a gene got a negative decision value we set it to be equal to 0. The result of this was that we stopped penalizing if the gene was assigned to the wrong class. In the second case, we decided to also limit very high decision values. To do this, we set all decision values that were greater than 1 equal to 1. The intuition behind this step is that when a gene receives a

decision value of +3, then we do not necessarily want to say that this gene is 3 times more likely to be a good candidate than the one with a decision value of +1. Especially, because we noticed that most of the training angiogenesis genes had decision values around +1.

4.4.2 DCDiv Algorithm

The *e1071* SVM package for R has the ability to output probabilities of a gene belonging to one class or the other instead of just decision values. To make use of this information, we tried a DCDiv [6] algorithm designed to aggregate probabilities. The idea of the algorithm is the following. For each gene we calculate

$$S = \prod_{i=1}^{100} \frac{P(X = -1)}{P(X = 1)}$$

where $P(X = -1)$ is the probability that the gene belongs to the negative class and $P(X = 1)$ is the probability that the gene X belongs to the positive class. S will converge to 0 when the gene belongs to the positive class and will diverge to infinity when the gene belongs to the negative class. Finally, we can identify strongest candidates by ranking all genes according to the value of S.

4.4.3 BetaMEM Rank Aggregation

BetaMEM [18] is a rank aggregation method developed by Raivo Kolde for MEM web-tool [1]. The easiest way to explain it is to look at an example. Suppose we have decision values for all genes from experiments with 20 different negative sets. Ordering the genes according to the decision values gives us 20 different rankings. By looking at the rankings more closely, we can determine for each gene all the different positions it got. This is illustrated on Figure 4.2.

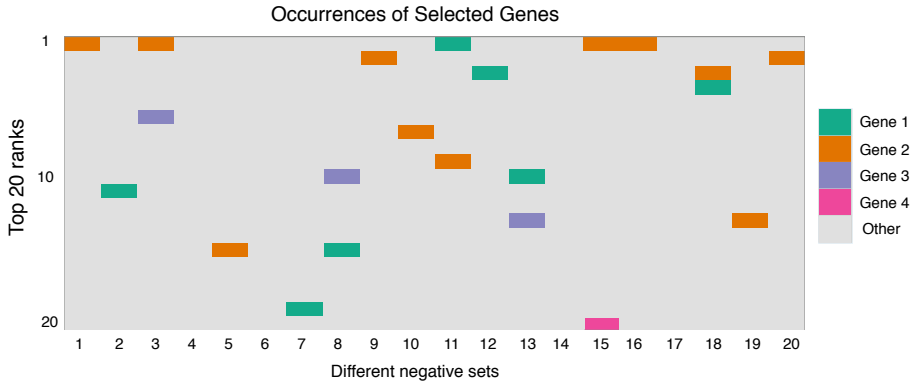


Figure 4.2: Top 20 ranks from 20 different experiments.

To make this information more comprehensible, we can look at how the ranks of a gene are distributed (Figure 4.3).

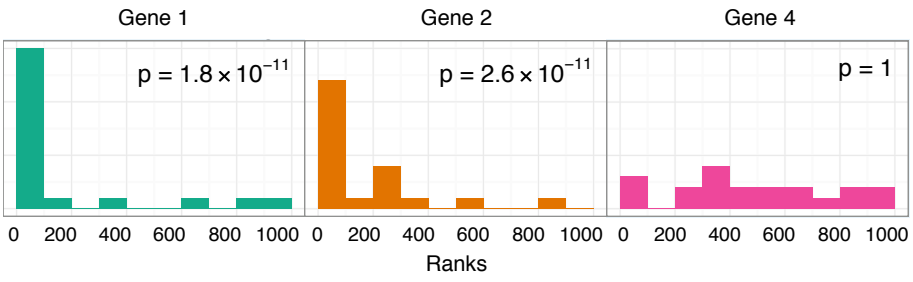


Figure 4.3: Distribution of ranks for three different genes.

Finally, we just have to notice that if the ranks were random then they would be uniformly distributed. Taking this into account, we can at each rank calculate the probability for that gene to have so many that low or lower ranks. This is illustrated on Figure 4.4. To get the final ranking, we just have to find the minimal p-value for each gene and sort the genes according to that value.

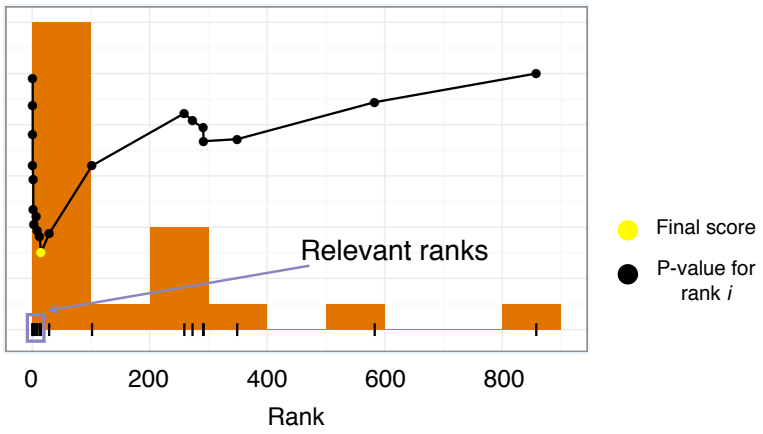


Figure 4.4: Calculating p-value for single distribution. Only the lowest one is taken into account.

The advantage of this method is that it should be more robust towards outliers than simply summing the decision values.

All images in this section were kindly provided by Raivo Kolde.

4.4.4 Comparison of Different Methods

To rate the goodness of our proposed approach and to compare different aggregation methods we have performed 10-fold cross-validation and calculated areas under Receiver Operator Characteristic curve (ROC) and Precision Recall Curve (PRC) as described in Chapter 2.3.

The results are presented in Table 4.1 and on Figures 4.5 and 4.6. `Sum` is the naïve approach, `ignore negative` and `limit1` are respectively the modifications where just negative values are ignored or higher decision values

Table 4.1: Areas under Precison-Recall curve (auPRC) and Receiver Operator Characteristic curve (auROC) of different aggregation methods.

	Sum			DCDiv	BetaMEM
	`sum`	`ignore negative`	`limit1`		
auROC	0.8436	0.8364	0.8411	0.8437	**0.8535**
auPRC	0.0235	0.0226	**0.0464**	0.0234	**0.0441**

are also limited to 1. Other names are the same as used in the text. The curves of `sum` and `ignore negative` are not shown on the ROC plot (Figure 4.6), because they were not significantly different from the others and would have therefore made the plot more difficult to understand.

Based on the ROC curve, BetaMEM is the best aggregation method with the highest area under the curve. The reason for this might be that BetaMEM is more robust to noise and will not rank highly the genes that had good decision values only in a few experiments. Looking at the Precision-Recall plot reveals that overall `limit1` has a slightly higher AUC than BetaMEM, but it is caused by a small number of top positions. As can be seen from the Figure 4.5, after recalling 20% of the angiogenesis genes BetaMEM starts to outperform `limit1`. Ignoring just the top 6 positions would already cause BetaMEM to have higher AUC score.

Another interesting result is that according to areas under Precision-Recall curves, `limit1` is more successful than other naïve approaches. This indicates that ignoring genes with few very high decision values is a good idea. This in turn might explain why BetaMEM is performing better than other methods, because it does this type of outlier elimination inherently (by not giving low p-values to genes that are ranked highly in only a few cases). Somewhat surprising was the poor performance of the DCDiv algorithm, because probabilities from different SVM experiments should be more comparable than decision values, which can have different ranges depending on the data.

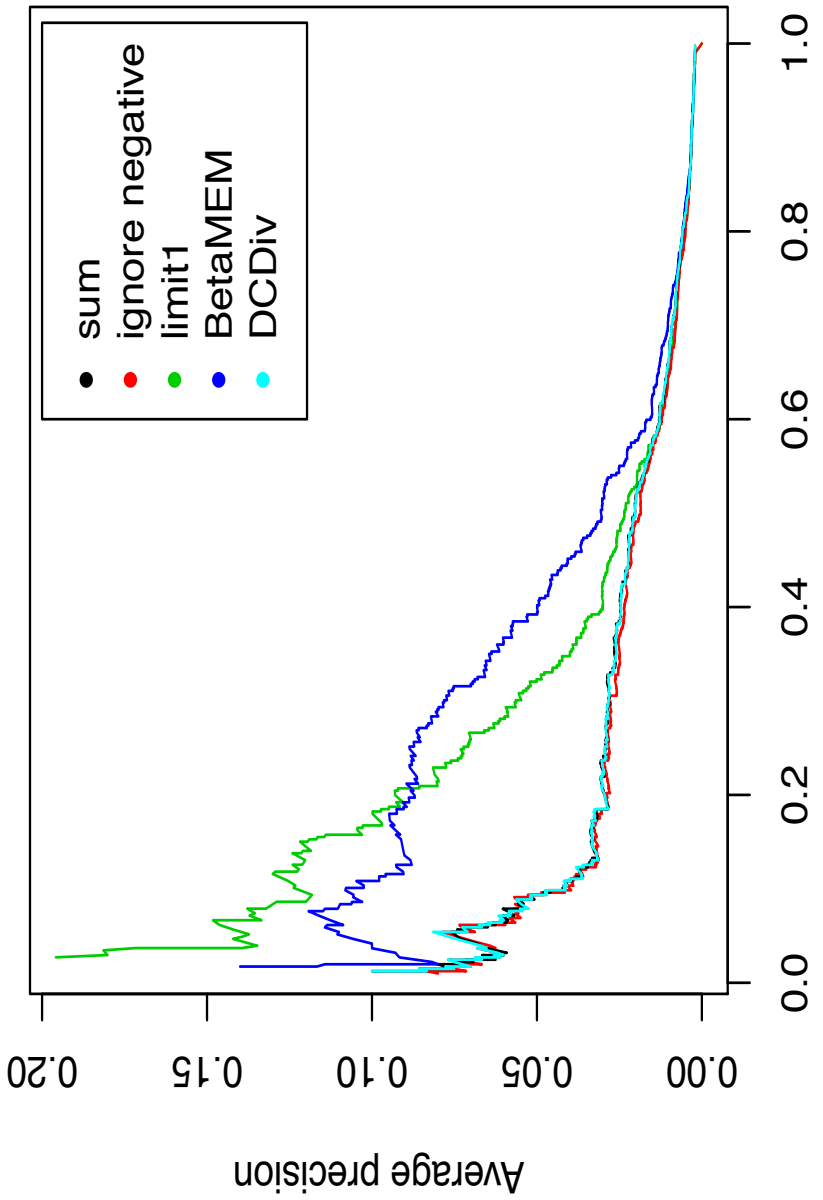


Figure 4.5: Precision-Recall Curve

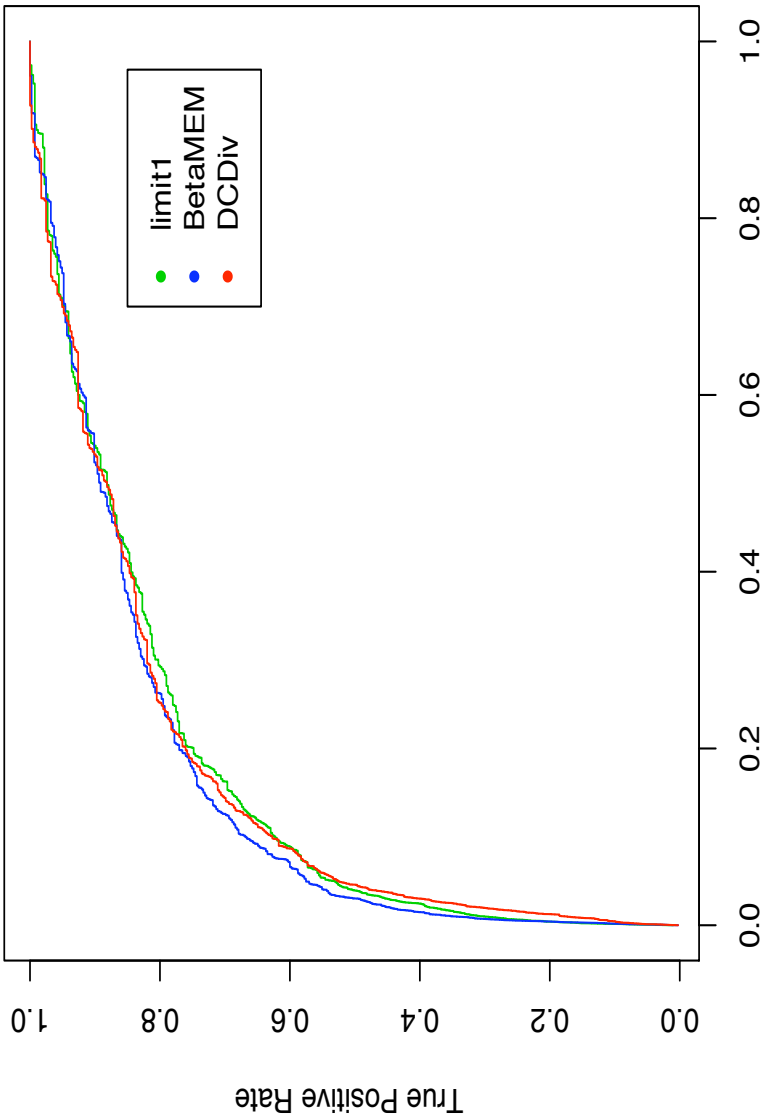


Figure 4.6: ROC curve.

Chapter 5

Improving the Classifier

We tried a few additional approaches to improve our methods accuracy. In this chapter, we will give a short overview of two of the methods that were explored and present the results that we obtained. First, we will start with feature selection and then we will cover an experiment in which we tried to give different weights to different randomly generated negative sets.

5.1 SVM-based Feature Selection

"Curse of dimensionality" is a term used in machine learning, when there are many more features than there are training samples in the data set. It can cause overfitting[1] and therefore poorer performance of the classifier. This is something we have to pay attention in our situation also, because our positive training set consists of only 405 probe sets and they are measured in 5372 conditions.

One way to overcome this problem is to use feature selection for which many different methods have been developed over time. In classification problems, the main idea of all feature selection approaches is to find a small subset of features that could best help to separate the two classes from each other. In this section we will compare two methods that try to find the optimal features by recursively removing the least relevant ones at each iteration.

[1] finding random relations from the data that are specific to the training set, but do not help to make predictions on new examples.

These methods are Support Vector Machine Recursive Feature Elimination (SVM-RFE) [15] and Recursive Support Vector Machine (R-SVM) [31].

5.1.1 Methods

Both of these methods are similar in a sense that they take advantage of SVM vector of weights of the features

$$w = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

and use exactly the same recursive procedure to remove unimportant features. The difference lies in how they assess the importance of the features. For SVM-RFE, the

$$S_j^{RFE} = w_j^2,$$

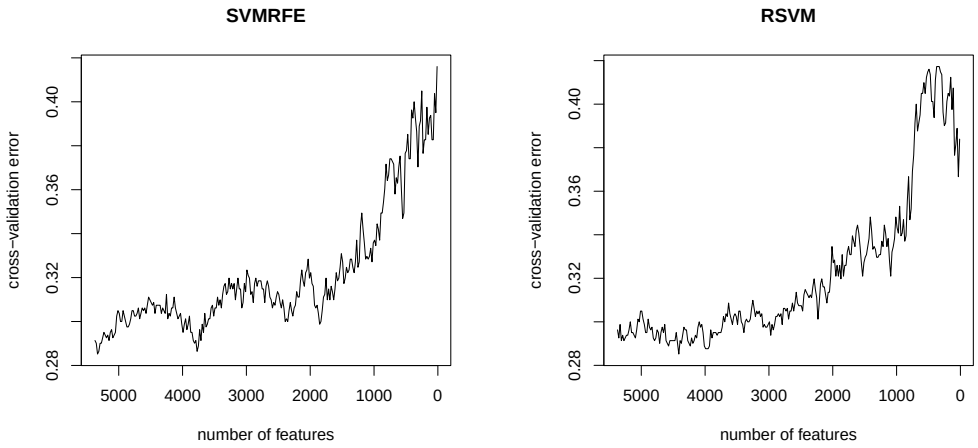
measure is used, where w_j is the weight of the j-th feature. In the newer R-SVM method, a slightly modified

$$S_j^R = w_j(m_j^+ - m_j^-),$$

measure is used, where w_j is the weight of the j-th feature and m_j^+ and m_j^- are the means of the values of the positive and negative examples in the same feature. The authors claim that this makes their method less sensitive to noise and possible outliers [31].

5.1.2 Results

To compare these two methods and to see if we could use one of them to improve our Comb-SVM approach we used the R code available from [31]. For the positive training set, we took the 405 known angiogenesis genes and for the negative examples, we took a list of 405 randomly chosen other genes. This is the same approach that we used in Comb-SVM to generate negative sets. We started with 5372 features. At each iteration left out 20 features with the lowest scores until we had only 12 features left. The results are presented on Figure 5.1.



(a) SVM-RFE

(b) R-SVM

Figure 5.1: Recursive feature elimination with two different methods.

With SVM-RFE we observed no improvement in cross-validation accuracy. When the number of features decreased the cross-validation error started to increase instead. The overall pattern was the same for R-SVM. Although there was some improvement at around 4400 features, it was hardly stable. Without feature selection, the cross-validation error was 29.6%. With R-SVM, the lowest cross-validation error achieved was 28.5% with 4412 features. With 4452 it was still 29.5% and at 4392 it was already 29.1%.

Because there were only slight improvements in cross-validation accuracy on our data and because in our Comb-SVM approach we would have had to perform separate feature selection for each 100 negative sets increasing the running time of the algorithm more than 10 fold, we decided not to incorporate it into our approach. In addition, the slight improvements that we witnessed in cross-validation accuracy do not necessarily mean improved ranking for angiogenesis genes among other unlabeled genes. Furthermore, feature selection with different negative sets could yield different results, but this requires some further research.

5.2 Weighted Aggregation

We constructed negative sets randomly, so it is quite natural to assume that some of these sets could be better than others in helping to separate angiogenesis genes from the other genes. As a result, we tried to alter our method so that it would give higher weights to negative sets that better help to separate angiogenesis genes from all other genes.

First, we identified which of the negative sets were better suited than others for finding angiogenesis genes. We did this by taking each negative set and corresponding training set mentioned above and used 10-fold cross-validation once again. We divided both the initial training set and the negative set into 10 subsets. Next, we performed conventional 10-fold cross-validation and finally got an accuracy score from 0 to 1 that indicated, how large part of the genes left for testing were correctly classified. For different negative sets these scores varied from 0.68 to 0.77. We assumed that negative sets that had higher cross-validation accuracy were also better.

We tried three different ways to give more importance to better negative sets. First, we just multiplied the decision values given by each negative set with their cross-validation accuracy so that negative sets with higher cross-validation accuracy would also have higher weight. This approach is represented under title *weight* in Table 5.1. Because the differences in the cross-validation accuracy scores were small, we also tried two other approaches. In the first case we calculated a modified weight value denoted by $weight^3$ in the following way:

$$weight^3 = (weight + 0.30)^3 \tag{5.1}$$

Adding 0.30 to the initial score shifts the mean closer to 1 and taking the third power amplifies the differences between stronger and weaker negative sets. The third approach was essentially the same, but instead of third power we took the fifth power.

$$weight^5 = (weight + 0.30)^5 \tag{5.2}$$

As can be seen from the table, the auROC scores did not change at all or maybe even slightly decreased when compared to baseline method of not giving weights at all (*limit1*). One of the reasons for this could be that

Table 5.1: Comparison of areas under ROC curves using different weighting methods.

	limit1	weight	weight3	weight5
auROC	**0.8411**	0.8373	0.8351	0.8364

cross-validation accuracy is not the correct way to assess the goodness of the negative set. Negative sets with high cross-validation accuracy could be well separable from the training genes but they might not help to rank unknown angiogenesis genes more highly than other genes. The problem could be the same that we had with housekeeping genes. Trying to find strong negative sets could potentially undermine the strength of our method, because we are excluding negative sets that are very close to angiogenesis genes and therefore can help to distinguish them better from all other genes.

Chapter 6

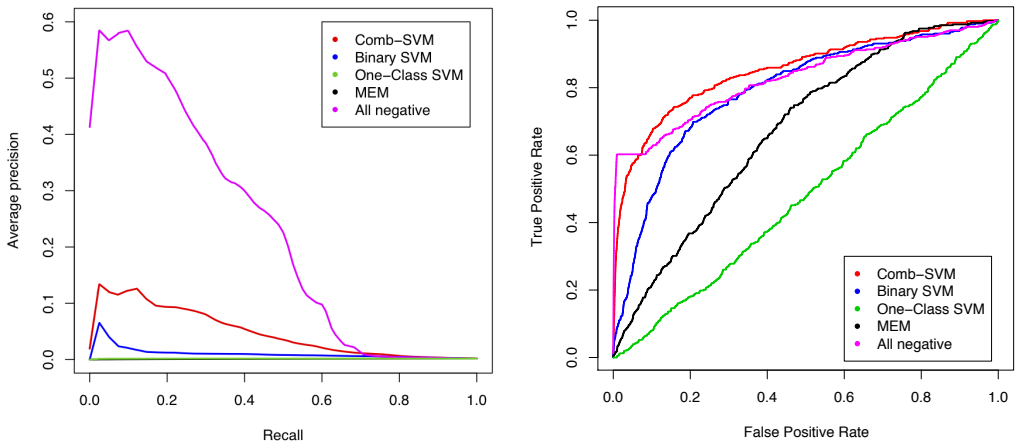
Results

In this chapter, we will present the results of all of the experiments. First, we will compare our method to the existing ones introduced in Chapter 3. Secondly, we will analyze the predictions made by our algorithm.

6.1 Performance of Different Methods

We compared Comb-SVM to existing methods described in Chapter 3 by using 10-fold cross-validation and drawing Precision-Recall and ROC curves. More details on comparison measures can be found in Section 2.3. To make results more comparable, we used the same division into training and test sets in all experiments. In Comb-SVM, we used BetaMEM for aggregation, because it proved to be the best algorithm (see Section 4.4.4).

The results are presented on Figure 6.1 and in Table 6.1. Our proposed method is denoted by Comb-SVM and `All negative` represents binary SVM with unlabeled data. Other methods are the same as described in Chapter 3. It can be clearly seen that based on 10-fold cross-validation our algorithm outperforms all other compared methods except `All negative`, which has higher area under Precision-Recall curve. The differences based on ROC curves are not that drastic (2 per cent difference between our method and `All negative`), because the number of known angiogenesis genes is very small compared to the number of unlabeled genes. This in turn means that small changes in the top rankings do not have a strong effect on the overall



(a) Precision-Recall Curve (b) ROC curve

Figure 6.1: Precision-Recall and ROC curves.

score.

Precision-Recall curve does not depend so much on the size of the negative set and therefore may reflect the differences more accurately. On the other hand, we are drawing our Precision-Recall plots based on positive and unlabeled examples and we do not know which of the unlabeled genes are truly negative and which of them are actually related to angiogenesis. This means that any method that ranks new strong candidates more highly than training angiogenesis genes is bound to have lower precision. This might explain why `All negative` method has so high precision compared to Comb-SVM, but gives less stable and biologically less valid predictions as will be shown later in this chapter.

6.1.1 Roc-SVM

We excluded the Roc-SVM from our final comparison, because it performed badly in the first step of trying to find a strong negative set. Namely, the set of genes classified as negative by the Rocchio's classifier was too big (more than 16000 genes) and contained many of the angiogenesis genes that were used to find this negative set. It was clear that continuing with this list

Table 6.1: Areas under ROC and Precision-Recall curves obtained with different algorithms.

	Comb-SVM	Binary SVM	One-Class SVM	MEM	All negative
auROC	**0.8535**	0.7921	0.4818	0.6742	0.8302
auPRC	0.0433	0.0096	0.0017	0.0034	**0.2179**

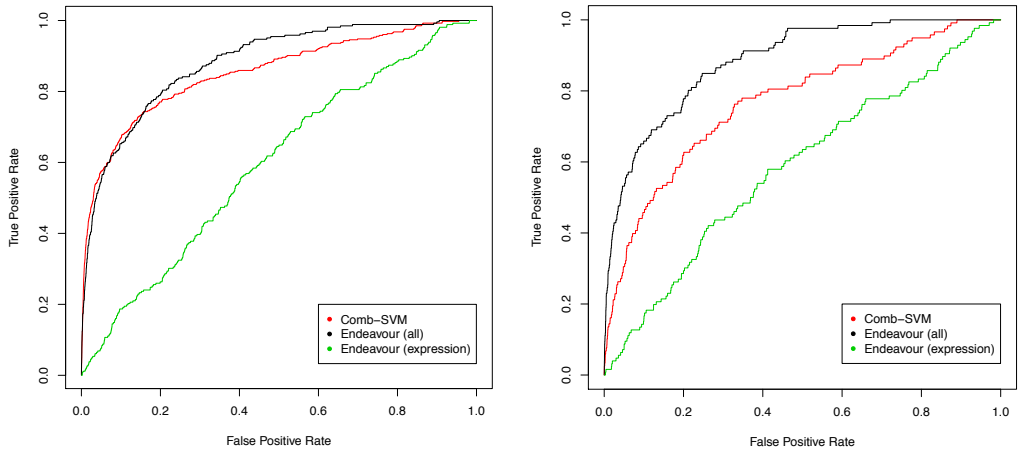
of genes was not meaningful, because it was too inaccurate and failed to help us reduce the size of the negative set. The reason why the Rocchio's classifier failed may be that angiogenesis genes in total are quite diverse. This means that the prototype vector created based on these genes might not have been very different from the prototype vector created from all other genes which of course resulted in a poor classifier. This coincides with our notions that on Principal Component Analysis (PCA) and NeRV [28] plots angiogenesis genes do not form a clearly distinguishable group.

6.1.2 Endeavour

The problem with the Endeavour is that we are essentially trying to compare the incomparable. Firstly, while all other methods except MEM use exactly the same gene expression data set, Endeavour uses many different types of data including gene expression, protein-protein interaction and functional annotation data. In addition, it requires the input genes to be in a different format. In all other methods we can use Affymetrix probe set identifiers that are also present int the main data set described in Section 1.3. The Endevour, on the other hand, requires Ensembl identifiers of the same genes. There are tools like g:Profiler [24] to convert gene identifiers from one system to another but the problem is that one Ensembl ID can have many matching probe set identifiers which in turn could affect cross-validation results.

To take this into account, we created two lists of genes. In the first case, we used all 274 Ensembl IDs corresponding to our 405 known probe set identifiers. In the second case, we filtered out only those probe sets and

Ensembl IDs that had one-to-one matching. Additionally, with these two lists of genes we conducted two different experiments. In the first experiment, we used all 24 data sources available in Endeavour. In the second case, we only took the six different gene expression data sets.



(a) ROC curve (all angiogenesis genes) (b) ROC curve (unique probe sets only)

Figure 6.2: ROC curves.

We found that, based on gene expression data only, the Comb-SVM clearly outperforms Endeavour. On the other hand, incorporating all other types of data makes the Endeavour to be better. The results are presented on Figure 6.2 and in Table 6.2. We must not forget that even the gene expression data used in Endeavour is different from ours, and therefore it is not possible to make any strong conclusions based on these results. The strength of Endeavour seems to be its ability to gather different types of data and use it to make one prediction. The goal of this research is to predict candidate genes based on only gene expression data and in this case Endeavour seems to perform worse than other approaches.

Table 6.2: Areas under ROC curves obtained with different algorithms.

	Comb-SVM	Endeavour (all)	Endeavour (expression)
all probe sets	0.8535	0.8815	0.5950
unique probe sets	0.7690	0.8844	0.5830

6.2 Analyzing the Stability of Predictions

In the previous section, we looked at how well different methods can bring back known training genes in 10-fold cross-validation experiments. In this section, we will concentrate more on the predictions made by two best methods and determine how stable they are to the changes in the training samples.

6.2.1 Experimental Setup

We decided to compare our best method (Comb-SVM with BetaMEM rank aggregation) to `All negative`, which was the method that had highest area under the Precision-Recall curve. Two experiments were conducted to assess the stability of the predictions. In the first case, the training set of 405 probe sets was randomly split into two sets of equal size and then both of them were given as input to both of the methods in question. In the second case, the initial training set was first separated into 3 non-overlapping parts $\{A, B, C\}$ and then all pairwise combinations $\{A \cup B, A \cup C, B \cup C\}$ of these parts were used as input. Finally, in both cases we looked at how many of the predicted candidate genes were overlapping in top 500 positions.

6.2.2 Results

The results are presented on Figure 6.3. In the first case, we looked at the overlap of predictions after separating the training genes into two random halves *(2 sets)*. In the second case we observed the overlap of predictions from the first two of the three intersecting subsets *(3 sets)*. As can be seen from the

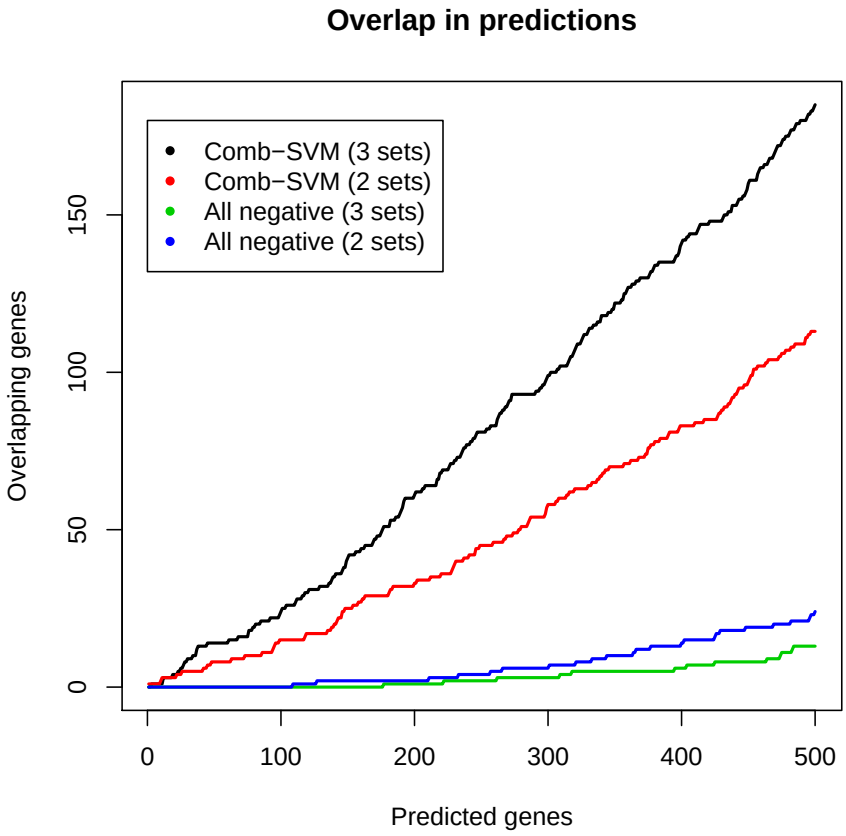


Figure 6.3: Overlap in predictions using different training sets.

plot, Comb-SVM constantly outperforms `All negative`. The most striking differences are in the biologically most interesting top 100 predictions. With the first 2 from the 3 intersecting sets, our method results in 24 overlapping genes while `All negative` gives predictions that do not overlap at all.

The stability of predictions is very important in the biological sense since there is some uncertainty in the way the curated list of the angiogenesis genes was composed. It is important to know that if the lists had been compiled in a slightly different way, then our predictions would not have changed too much. In addition, biologists have to validate our results experimentally and

can therefore only look at a relatively small number of top predictions. If there are very large variations in top 100 predictions, then they cannot trust our results. For these reasons, we decided to use Comb-SVM instead of `All negative` to make our final predictions.

6.3 Newly Predicted Angiogenesis Genes

In addition to judging the performance of our method based on training data only, we also tried to verify our newly predicted genes. At this point we performed a simple verification based on literature and public databases. In the future, our collaborators will also conduct biological experiments.

6.3.1 Literature Verification

We performed initial verification by looking at the top 10 predictions of our algorithm and seeing if we can relate these genes to angiogenesis based on literature only. The results are presented in Table 6.3.

Table 6.3: 8 out of top 10 newly predicted genes could be easily verified based on literature.

Pos.	**Gene**	**PubMed ID**
1	CXCL2	16207631
2	FMO1	
3	GNG11	
4	CCL20	19340288
5	ELTD1	18483404
6	ESM1	11866539
7	MMRN1	19924294
8	CSF2	18691491
9	IER3	19690192
10	PTPRB	17360632

PubMed ID is the id of the article in PubMed in which we found a connection between the given gene and angiogenesis. As can be seen from the table,

8 out of top 10 predictions could be easily related to angiogenesis. This shows that the predictions of our method are indeed meaningful. We conducted similar analysis with predictions obtained from `All negative` method and found only 1 gene in top 10 to be related to angiogenesis. This further validates the strength of our approach.

One could argue that maybe some of these highly ranked genes should also be in the curated set, but solving this question is out of the scope of this work. For the time being we trust the biologists from the experimental lab in their decisions and use their supplied list of genes.

6.3.2 Gene Ontology Annotations

To further validate the biological relevance of the predictions we looked at the Gene Ontology (GO) annotations of top 50 new candidate genes. GO is an online curated database that organizes knowledge about the function and role of genes in many species. The g:Profiler found 71 significant annotations out of which a selection of 15 most related to angiogenesis are presented in Table 6.4. Some of them, like wound healing, blood vessel development, vasculature development and regulation of endothelial cell proliferation are obviously related to angiogenesis. For others, it has been shown that extracellular matrix plays an important role in angiogenesis [26]. Full table with results containing some additional information is given in the Appendix.

We also performed a similar analysis on the predictions of `All negative` method. In that case, the top 50 predictions were not significantly enriched with any GO terms. Looking at the top 100 predictions revealed some significant results (e.g. BP cellular process, BP regulation of developmental process, BP regulation of cell migration) but none of them were directly related to angiogenesis.

6.3.3 Biological Experiments

Ultimately, we want to discover novel genes involved in angiogenesis. To this end, we sent 147 top predictions to our ENFIN collaborators in Spain. There they will combine our results with predictions form two other labs into one sub-network model. Biological experiments will be conducted in

Table 6.4: A sample of significant GO annotations found for top 50 newly predicted genes. (CC - cellular component, MF - molecular function, BP - biological process, re - REACTOME pathway)

P-value	**Term ID**	**Name**
5.51E-011	GO:0005576	CC extracellular region
9.74E-011	GO:0009611	BP response to wounding
1.27E-010	GO:0044421	CC extracellular region part
5.64E-009	GO:0005615	CC extracellular space
1.46E-007	GO:0008083	MF growth factor activity
1.08E-006	GO:0031012	CC extracellular matrix
1.27E-006	GO:0042060	BP wound healing
1.70E-006	GO:0048513	BP organ development
3.72E-006	GO:0001568	BP blood vessel development
4.29E-006	GO:0001944	BP vasculature development
1.05E-005	GO:0001936	BP regulation of endothelial cell proliferation
1.93E-005	GO:0043062	BP extracellular structure organization
2.51E-005	GO:0048514	BP blood vessel morphogenesis
2.95E-005	GO:0001525	BP angiogenesis
2.97E-005	REAC:109582	re Hemostasis

the following way. First, drugs will be prioritized by their effect on specific angiogenic targets and by their reported anti-tumour action. Next, proteins from the sub-network model will be prioritized based on their specific angiogenic role and their potential as drug target. Top ranked potential drugs and targets will be selected for experimental validation using a set of assays. Unfortunately, we do not have any results to report from this analysis yet.

Summary

The aim of this thesis was to study the possibility of using gene expression data and machine learning methods to predict new candidate genes for angiogenesis based on a list of known genes.

We tested many standard machine learning methods and bioinformatics tools that could be used to solve this particular task. We compared the methods by training them with the same data and seeing how successful they were at retrieving known angiogenesis genes. Afterwards, we proposed a novel Comb-SVM approach that is based on the idea of training multiple Support Vector Machines and aggregating their predictions.

The experiments showed that Comb-SVM outperformed most of the other methods in 10-fold cross-validation experiments when looking areas under Receiver Operator Characteristic and Precision-Recall curves. We also determined that our method gave significantly more stable results than the second best approach proposed by Elkan and Noto [13].

Finally, we verified the biological relevance of the predicted genes by searching the literature and determining the enriched Gene Ontology terms of top 50 identified genes.

Tugivektormasinate kombineerimine angiogeneesiga seotud geenide ennustamiseks

Bakalaureusetöö (6 EAP)

Kaur Alasoo

Resümee

Vähk on tänapäeval üks levinumaid ja ohtlikumaid haigusi põhjustades igal aastal 13% kõigist surmajuhtumitest üle maailma [23]. Hoolimata aastatepikkustest jõupingutustest ei ole seni ikka veel efektiivset ravi selle haiguse vastu leitud. Küll on aga teada, et vähi arengus on olulisel kohal angiogenees, mille käigus vähk paneb enda ümber asuvad veresooned hargnema ja kasvama. Parem arusaamine sellest protsessist võimaldaks potentsiaalselt luua uusi ja efektiivsemaid ravimeetodeid.

Aastate jooksul tehtud eksperimentide käigus on mõõdetud enamiku inimese geenide ekpressiooni rohkem kui 5000 tingimuses. Lisaks on meie koostööpartnerid koostanud nimekirja 341-st veresoonte loomega seotud geenist. Käesoleva töö eesmärgiks ongi uurida, kuidas geenidekspressiooni andmete ja väikese hulga tuntud angiogeneesi geenide põhjal on võimalik ennustada uusi angiogeneesiga seotud geene.

Selleks võrreldakse kõigepealt mitmeid olemasolevaid masinõppe meetodeid ja avalikult kättesaadavaid bioinformaatika tööriistu, mida saaks kasutada kandidaatgeenide ennustamiseks. Kõigi nende meetodite puhul kasuta-

takse sisendiks võimalikult sarnaseid andmeid ning mõõdetakse siis 10-kordse ristvalideerimise abil, kui edukad need on juba tuntud angiogeneesi geenide ülesleidmisel.

Töö teises osas pakutakse välja uudne *Comb-SVM* meetod kandidaatgeenide ennustamiseks. Selle põhiidee baseerub kolmel sammul. Kõigepealt kasutatakse juba tuntud angiogeneesi geene ning juhuslikult valitud negatiivseid geene, et treenida paralleelselt mitu tugivektormasinal (ingl k *Support Vector Machine)* põhinevat klassifitseerijat. Järgnevalt kasutakse neid klassifitseerijaid uute angiogeneesi geenide ennustamiseks. Viimaks agregeeritakse kõigi klassifitseerijate tulemused kokku üheks ennustuseks.

Töö lõpus näidatakse, et 10-kordse ristvalideerimise põhjal on *Comb-SVM* täpsem kui enamik olemasolevaid meetodeid. Lisaks näidatakse, et *Comb-SVM* ennustused on oluliselt stabiilsemad väikeste muudatuste suhtes treeningandmetes kui paremuselt teise algoritmi tulemused. Kõige lõpuks kasutatakse teaduskirjandust ning *Gene Ontology* [3] andmebaasi veendumaks, et uued ennustatud geenid on tõpoolest seotud angiogeneesiga.

Bibliography

[1] P. Adler, R. Kolde, M. Kull, A. Tkachenko, H. Peterson, J. Reimand, and J. Vilo. Mining for coexpression across hundreds of datasets using novel rank aggregation and visualisation methods. *Genome Biology*, 10(12):R139, 2009.

[2] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.C. Tranchevent, B. De Moor, P. Marynen, B. Hassan, et al. Gene prioritization through genomic data fusion. *Nature biotechnology*, 24(5):537–544, 2006.

[3] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[4] A. Brazma, H. Parkinson, U. Sarkans, M. Shojatalab, J. Vilo, N. Abeygunawardena, E. Holloway, M. Kapushesky, P. Kemmeren, G.G. Lara, et al. ArrayExpress–a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 31(1):68, 2003.

[5] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[6] B. Calvo, N. López-Bigas, S.J. Furney, P. Larrañaga, and J.A. Lozano. A partially supervised classification approach to dominant and recessive human disease gene prediction. *Computer methods and programs in biomedicine*, 85(3):229–237, 2007.

[7] P. Carmeliet. Mechanisms of angiogenesis and arteriogenesis. *Nature Medicine*, 6:389–395, 2000.

[8] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] C.L. Cheung, P.C. Sham, V. Chan, A.D. Paterson, K.D.K. Luk, and A.W.C. Kung. Identification of LTBP2 on chromosome 14q as a novel candidate gene for bone mineral density variation and fracture risk association. *Journal of Clinical Endocrinology & Metabolism*, 93(11):4448, 2008.

[10] DD Dalma-Weiszhausz, J. Warrington, EY Tanimoto, and CG Miyada. The affymetrix GeneChip platform: an overview. *Methods in enzymology*, 410:3, 2006.

[11] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM New York, NY, USA, 2006.

[12] E. Eisenberg and E.Y. Levanon. Human housekeeping genes are compact. *TRENDS in Genetics*, 19(7):362–365, 2003.

[13] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220. ACM, 2008.

[14] Napoleone Ferrara. Genentech: Research: Angiogenesis. `http://www.gene.com/gene/research/focusareas/oncology/angiogenesis.html` (last visited 30.05.2010).

[15] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[16] Q.Y. Huang, G.H.Y. Li, W.M.W. Cheung, Y.Q. Song, and A.W.C. Kung. Prediction of osteoporosis candidate genes by computational disease-gene identification strategy. *Journal of Human Genetics*, 53(7):644–655, 2008.

[17] National Cancer Institute. `http://www.cancer.gov/cancertopics/understandingcancer/angiogenesis/allpages` (last visited 30.05.2010).

[18] R Kolde. Co-expression queries across multiple experiments. Master's thesis, University of Tartu, 2008.

[19] X. Li and B. Liu. Learning to classify texts using positive and unlabeled data. In *International joint Conference on Artificial Intelligence*, volume 18, pages 587–594. Citeseer, 2003.

[20] M. Lukk, M. Kapushesky, J. Nikkilä, P. Parkinson, A. Goncalves, W. Huber, E. Ukkonen, and A. Brazma. A global map of human gene expression. *Nature Biotechnology*, 2010.

[21] L.M. Manevitz and M. Yousef. One-class svms for document classification. *The Journal of Machine Learning Research*, 2:154, 2002.

[22] N. Nishida, H. Yano, T. Nishida, T. Kamura, and M. Kojiro. Angiogenesis in cancer. *Vascular Health and Risk Management*, 2(3):213, 2006.

[23] World Health Organization. Fact sheet: cancer. `http://www.who.int/mediacentre/factsheets/fs297/en/index.html` (last visited 30.05.2010).

[24] J. Reimand, M. Kull, H. Peterson, J. Hansen, and J. Vilo. g: Profiler–a web-based toolset for functional profiling of gene lists from large-scale experiments. *Nucleic acids research*, 35(Web Server issue):W193, 2007.

[25] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12:582–588, 2000.

[26] J. Sottile. Regulation of angiogenesis by extracellular matrix. *BBA-Reviews on Cancer*, 1654(1):13–22, 2004.

[27] E. Storey, M. Bahlo, M. Fahey, O. Sisson, CJ Lueck, and RJM Gardner. A new dominantly inherited pure cerebellar ataxia, SCA 30. *British Medical Journal*, 80(4):408, 2009.

[28] J. Venna and S. Kaski. Nonlinear dimensionality reduction as information retrieval. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, pages 568–575. Citeseer, 2007.

[29] C. Wang, C. Ding, R.F. Meraz, and S.R. Holbrook. PSoL: a positive sample only learning algorithm for finding non-coding RNA genes. *Bioinformatics*, 22(21):2590, 2006.

[30] H. Yu, J. Han, and K.C.C. Chang. PEBL: Positive example based learning for Web page classification using SVM. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–248. ACM New York, NY, USA, 2002.

[31] X. Zhang, X. Lu, Q. Shi, X. Xu, H.E. Leung, L.N. Harris, J.D. Iglehart, A. Miron, J.S. Liu, and W.H. Wong. Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC bioinformatics*, 7(1):197, 2006. Code available from `http://www.hsph.harvard.edu/bioinfocore/RSVMhome/R-SVM.html`.

Appendix

Table 6.5: All significant GO annotations of top 50 newly predicted genes using Comb-SVM. (CC - cellular component, MF - molecular function, BP - biological process, re - REACTOME pathway, ke - KEGG pathway)

P-value	**Term ID**	**Name**
5.51E-011	GO:0005576	CC extracellular region
9.74E-011	GO:0009611	BP response to wounding
1.27E-010	GO:0044421	CC extracellular region part
3.81E-010	GO:0005125	MF cytokine activity
1.76E-009	GO:0005102	MF receptor binding
2.66E-009	GO:0009605	BP response to external stimulus
5.64E-009	GO:0005615	CC extracellular space
7.50E-009	GO:0005126	MF cytokine receptor binding
3.14E-008	KEGG:04060	ke Cytokine-cytokine receptor interaction
7.77E-008	GO:0042127	BP regulation of cell proliferation
1.17E-007	GO:0006954	BP inflammatory response
1.46E-007	GO:0008083	MF growth factor activity
4.20E-007	GO:0030334	BP regulation of cell migration
4.87E-007	GO:0050896	BP response to stimulus
5.83E-007	GO:0006950	BP response to stress
9.11E-007	GO:0051270	BP regulation of cell motion
9.70E-007	GO:0040012	BP regulation of locomotion
1.08E-006	GO:0031012	CC extracellular matrix

P-value	Term ID	Name
1.14E-006	GO:0032879	BP regulation of localization
1.27E-006	GO:0042060	BP wound healing
1.35E-006	GO:0006952	BP defense response
1.70E-006	GO:0048513	BP organ development
2.78E-006	REAC:76002	re Platelet Activation
3.72E-006	GO:0001568	BP blood vessel development
4.13E-006	GO:0048522	BP positive regulation of cellular process
4.29E-006	GO:0001944	BP vasculature development
4.34E-006	GO:0008285	BP negative regulation of cell proliferation
4.74E-006	GO:0065007	BP biological regulation
4.79E-006	GO:0005520	MF insulin-like growth factor binding
5.79E-006	GO:0030335	BP positive regulation of cell migration
5.81E-006	GO:0051239	BP regulation of multicellular organismal process
5.87E-006	GO:0048523	BP negative regulation of cellular process
6.87E-006	REAC:75178	re Formation of Platelet plug
7.20E-006	GO:0005515	MF protein binding
7.53E-006	GO:0010557	BP positive regulation of macromolecule biosynthetic process
8.40E-006	GO:0005578	CC proteinaceous extracellular matrix
8.52E-006	GO:0030193	BP regulation of blood coagulation
8.61E-006	GO:0051272	BP positive regulation of cell motion
1.01E-005	GO:0042325	BP regulation of phosphorylation
1.05E-005	GO:0001936	BP regulation of endothelial cell proliferation
1.11E-005	GO:0007165	BP signal transduction
1.13E-005	GO:0031328	BP positive regulation of cellular biosynthetic process
1.22E-005	GO:0009891	BP positive regulation of biosynthetic process
1.25E-005	GO:0019220	BP regulation of phosphate metabolic process
1.25E-005	GO:0051174	BP regulation of phosphorus metabolic process
1.26E-005	GO:0006955	BP immune response
1.28E-005	GO:0048731	BP system development
1.35E-005	GO:0050794	BP regulation of cellular process

P-value	Term ID	Name
1.37E-005	GO:0048518	BP positive regulation of biological process
1.54E-005	GO:0048519	BP negative regulation of biological process
1.93E-005	GO:0043062	BP extracellular structure organization
2.06E-005	GO:0060205	CC cytoplasmic membrane-bounded vesicle lumen
2.28E-005	GO:0005604	CC basement membrane
2.32E-005	GO:0019838	MF growth factor binding
2.42E-005	GO:0051240	BP positive regulation of multicellular organismal process
2.43E-005	GO:0031983	CC vesicle lumen
2.51E-005	GO:0048514	BP blood vessel morphogenesis
2.78E-005	GO:0030194	BP positive regulation of blood coagulation
2.84E-005	GO:0050818	BP regulation of coagulation
2.89E-005	GO:0050789	BP regulation of biological process
2.94E-005	GO:0048856	BP anatomical structure development
2.95E-005	GO:0001525	BP angiogenesis
2.97E-005	REAC:109582	re Hemostasis
2.97E-005	GO:0002376	BP immune system process
3.54E-005	GO:0010604	BP positive regulation of macromolecule metabolic process
3.55E-005	GO:0070851	MF growth factor receptor binding
3.79E-005	GO:0031091	CC platelet alpha granule
3.99E-005	GO:0007154	BP cell communication
9.63E-005	REAC:114611	re Exocytosis of Alpha granule
1.10E-004	REAC:114608	re Platelet degranulation