MACHINE LEARNING METHODS FOR USING NETWORK BASED INFORMATION IN MICRORNA TARGET PREDICTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

MERTER SUALP

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

FEBRUARY 2013

Approval of the thesis:

MACHINE LEARNING METHODS FOR USING NETWORK BASED INFORMATION IN MICRORNA TARGET PREDICTION

submitted by **MERTER SUALP** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Tolga Can
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Mehmet Volkan Atalay
Computer Engineering Dept., METU

Assoc. Prof. Dr. Tolga Can
Computer Engineering Dept., METU

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering Dept., METU

Assoc. Prof. Dr. Çetin Kocaefe
Medical Biology Dept., Hacettepe University

Assoc. Prof. Dr. Hasan Oğul
Computer Engineering Dept., Başkent University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MERTER SUALP

Signature :

ABSTRACT

MACHINE LEARNING METHODS FOR USING NETWORK BASED INFORMATION IN MICRORNA TARGET PREDICTION

Sualp, Merter
Ph.D., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Tolga Can

February 2013, 73 pages

Computational microRNA (miRNA) target identification in animal genomes is a challenging problem due to the imperfect pairing of the miRNA with the target site. Techniques based on sequence alone are prone to produce many false positive interactions. Therefore, integrative techniques have been developed to utilize additional genomic, structural features, and evolutionary conservation information for reducing the high false positive rate. We propose that the context of a putative miRNA target in a protein-protein interaction (PPI) network can be used as an additional filter in a computational miRNA target prediction algorithm. We compute several graph theoretic measures on human PPI network as indicators of network context. We assess the performance of individual and combined contextual measures in increasing the precision of a popular miRNA target prediction tool, TargetScan, using low throughput and high throughput datasets of experimentally verified human miRNA targets. We used classification algorithms for that assessment. Since there exists only miRNA targets as training samples, this problem becomes a One Class Classification (OCC) problem. We devised a novel OCC method, DiVo, based on simple distance metrics and voting. Comparative analysis with the state of the art methods show that, DiVo attains better classification performance. Our eventual results indicate that topological properties of target gene products in PPI networks are valuable sources of information for filtering out false positive miRNA target genes. We show that, for targets of a number of miRNAs, network context correlates better with being a target compared to a sequence based score provided by the prediction tool.

Keywords: microRNA target prediction, protein-protein interaction network, network topology, One Class Classification, Distance Metrics, Voting

ÖZ

MİCRORNA HEDEF TAHMİNLERİNDE AĞ TABANLI BİLGİLERİN KULLLANILMASI İÇİN MAKİNE ÖĞRENİM YÖNTEMLERİ

Sualp, Merter
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Tolga Can

Şubat 2013 , 73 sayfa

Hayvan genlerindeki microRNA (miRNA) hedeflerinin berimsel olarak tanımlanması, miRNA ile hedefi arasındaki kusurlu eşleşmeler sebebiyle zorlu bir problemdir. Yalnızca sıralamaya dayalı teknikler, normalde varolmayan bir çok pozitif etkileşim üretmeye meyillidirler. Bundan ötürü, yüksek pozitif hata oranını azaltmaya yönelik olarak, ek genetik bilgileri, yapısal özellikleri, ve evrimsel muhafaza kurallarını bir araya getiren birleştirici yöntemler oluşturulmaktadır. Biz, olası miRNA hedeflerinin protein-protein etkileşim (PPI) ağlarındaki bağlamlarının, berimsel miRNA hedef tahmini yapan algoritmalarda ilave bir filtre olarak kullanılabileceğini öne sürmekteyiz. Ağ bağlamı göstergesi olarak, insan PPI ağı üzerinde birden fazla kuramsal grafik ölçüsü hesapladık. Deneysel olarak ispatlanmış düşük ve yüksek çıktılı insan miRNA hedef veri kümeleri kullanarak, bağlam ölçülerinin bireysel ve birleştirilmiş performanslarını, popüler bir miRNA hedef tahmin aracı olan TargetScan'in hassasiyetinin artışı üzerinden değerlendirdik. Bu değerlendirmede sınıflandırma algoritmaları kullandık. Eğitim örnekleri olarak yalnızca miRNA hedefleri bulunduğundan, problem Tek Sınıflı Sınıflandırma (OCC) problemi haline geldi. DiVo adından, basit mesafe ölçülerine ve oylamaya dayalı, özgün bir OCC algoritması geliştirdik. Son teknoloji yöntemlerle yaptığımız karşılaştırmalı analizler gösterdi ki, DiVo daha iyi sınıflandırma performansına ulaşmaktadır. Nihai sonuçlarımız, PPI ağlarındaki hedef gen ürünlerinin topolojik özelliklerinin, hatalı pozitif miRNA hedef genlerinin elenmesinde değerli birer bilgi kaynağı olduğunu belirtmektedir. Bir takım miRNA hedefleri için, ağ bağlamının, tahmin aracı tarafından sağlanan sıralama bazlı skora göre, hedef olup olmamayla daha ilintili olduğunu göstermiş bulunmaktayız.

Anahtar Kelimeler: miRNA hedef tahmini, protein-protein etkileşimleri, ağ topolojisi, tek sınıfla sınıflandırma, mesafe ölçütleri, oylama

I dedicate this thesis to Narin.

ACKNOWLEDGMENTS

The best way to make a contribution to the mankind is to leave a small carving in the huge and everlasting tree of science. This work will be my first attempt. I hope it won't be the last.

Eight long years have passed. Frankly, I did not expect it to take that much time. At the very beginning it was really a joyride to read the papers, trying to figure out the basics of bioinformatics. Each passing moment build up the excitement. Through the journey, we had many ups and downs but eventually, the taste of working on bioinformatics is not something that can be forgotten easily. On the contrary, it will always be a part of my life.

I thank my parents. They never oppose my research enthusiasm. I feel myself lucky enough to have a family being beside me.

I want to thank Dr. Tolga Can for his continuous effort and his encouragement. It was really a pleasure for me to work with him. I definitely hope that our cooperation has just begun.

And finally, I would like to present my deepest gratitude to Narin for her support, unending patience and unique personality. Without her, it was impossible for me to reach this far.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS ix

LIST OF TABLES xii

LIST OF FIGURES xiv

CHAPTERS

1 INTRODUCTION 1

2 BACKGROUND 7

2.1 Computational miRNA Target Prediction 7

2.1.1 Basics of miRNA-mRNA Interaction 7

2.1.2 miRNA Target Prediction Tools 9

2.1.2.1 miRanda 9

2.1.2.2 TargetScan and TargetScanS 9

2.1.2.3 PicTar 9

2.1.2.4 DIANA-MicroT 10

2.2 The Protein-Protein Interactions 10

2.2.1 Network Topological Measures 11

2.3 Computational Classification 13

2.3.1 Support Vector Machines 14

2.3.2 Mahalanobis Distance 14

2.3.3 One-Class Classification 15

2.3.3.1 State of the Art Methods 16

3 RELATED WORK 19

3.1 Protein-Protein Interaction Networks and miRNA Targets 19

4 USING NETWORK CONTEXT AS A FILTER FOR MIRNA TARGET PREDICTION 23

4.1 Materials 23

4.1.1 The Positive Ground Truth Dataset 24

4.1.2 miRNA Target Databases 24

4.1.3 Constructing The Negative Ground Truth Dataset 25

4.2 Experimental Evaluation of Topological Properties 27

4.2.1 Performance of Individual Measures as Simple Threshold Filters 27

4.2.2 Integration of Network Context Measures using Support Vector Machines 31

4.2.3 Integration of Network Context Measures using Mahalanobis Distance 35

5 DIVO: A NOVEL DISTANCE BASED VOTING METHOD FOR ONE CLASS CLASSIFICATION 39

5.1 Data Preprocessing 42

5.1.1 Datasets for Parameter Selection 42

5.1.2 Datasets for Performance Comparison 43

5.2 Experimental Evaluation of DiVo Performance 43

5.2.1 Parameter and Data Analysis for DiVo 44

5.2.2 Time Complexity and Analysis of DiVo 49

5.3 Observations 50

6 DIVO ON MIRNA TARGET FILTERING 53

6.1 Correlation Analysis 53

6.2 Experimental Evaluation of DiVo on miRNA Target Prediction 54

6.3 Experimental Evaluation of DiVo on Filtering miRNA Targets 55

6.4 Observations 56

7 DISCUSSION, CONCLUSION AND FUTURE WORK 61

REFERENCES 65

CURRICULUM VITAE 73

LIST OF TABLES

TABLES

Table 2.1 Network topological measures. The seven graph theoretic measures that are used as indicators of a protein's context in a PPI network. 12

Table 4.1 Average sensitivity and precision of network properties on the pSILAC dataset. The average sensitivity and precision of target prediction for five pSILAC miRNAs for each seven graph theoretic measures that are used as indicators of a protein's context in a PPI network. 29

Table 4.2 Average sensitivity and precision of network properties on the TarBase dataset. The average sensitivity and precision of target prediction for eight TarBase miRNAs for each seven graph theoretic measures that are used as indicators of a protein's context in a PPI network. 30

Table 4.3 Fold change of precision in TargetScan using SVM. The ratio of precision of TargetScan predictions filtered by SVM to the precision of original predictions of TargetScan is presented. The threshold level of the SVM is set to 0.5. 0.000 means that SVM filters out all the verified targets. 32

Table 4.4 Effect of different stringency levels in TargetScan using SVM. The ratio of precision of TargetScan predictions filtered by SVM to the precision of original predictions of TargetScan is presented. Five different threshold levels, which are 0.5, 0.4, 0.3, 0.2, and 0.1 of the SVM are compared. 0.000 means that SVM filters out all the verified targets. 33

Table 4.5 Fold change of precision in TargetScan using MD. The ratio of precision of TargetScan predictions filtered by MD to the precision of original predictions of TargetScan is presented. The threshold level of the MD is set to 0. 35

Table 4.6 Effect of different stringency levels in TargetScan using MD. The ratio of precision of TargetScan predictions filtered by MD to the precision of original predictions of TargetScan are given. Six different threshold levels, which are +0.0, +0.2, +0.4, +0.6, +0.8 and +1.0 of the MD are compared. 36

Table 5.1 The datasets for parameter selection and their number of classes, attributes, and instances. 42

Table 5.2 The datasets for performance comparison and their number of classes, attributes, and instances. 43

Table 5.3 The average and standard deviation of f-measures on different datasets. . . . 43

Table 5.4 The multiplication factors required to train each algorithm. 50

Table 6.1 The correlation coefficients between DiVo-Euclidean vote counts and TargetScan context scores. NA indicates that the pSILAC targets of that miRNA are not used or do not exist in the database. Div/0 occurs when all targets of a miRNA have the same number of votes. 53

Table 6.2 The correlation coefficients between DiVo-Mahalanobis vote counts and TargetScan context scores. NA indicates that the pSILAC targets of that miRNA are not used or do not exist in the database. 54

Table 6.3 Effect of different stringency levels in recall using DiVo with Mahalanobis distance on pSILAC dataset. The recall of TargetScan prediction scores and recall of DiVo is presented. Eleven different ratio values, which are 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1 of the DiVo are compared. - means that DiVo filters out all the verified targets. 55

Table 6.4 Effect of different stringency levels in recall using DiVo with Mahalanobis distance on TarBase dataset. The recall of TargetScan prediction scores and recall of DiVo is presented. Eleven different ratio values, which are 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1 of the DiVo are compared. - means that DiVo filters out all the verified targets. 55

LIST OF FIGURES

FIGURES

Figure 1.1 miRNA Production Steps - Artwork originally created for the National Cancer Institute. Reprinted with permission of the artist, Jeanne Kelly. Copyright 2012. 2

Figure 3.1 miR-21 regulated protein-protein interaction subnetwork. The proteins in red are targets of miR-21, taken from the study Hsu et al. [35]. Courtesy of WILEY-VCH with license number 3061890408961. 21

Figure 4.1 Histogram for comparing the 3'UTR lengths of miRNA targets found in a miRNA target database and putative true negative proteins. The comparison of the lengths of 3'UTR parts of mRNAs that produce proteins in putative negative dataset and mRNAs that produce the proteins that are identified as targets by at least one miRNA target database. Here, as an example, the number of mRNAs having a 3'UTR length between 100 and 199 are represented as within the same bin since the bin size of the histogram is 100. The calculated count for each length size 100 is divided by the total number of mRNAs for that dataset and represented as a percentage for an easy comparison. 26

Figure 4.2 pROC curve of each network measure for let-7 using the pSILAC dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold. 28

Figure 4.3 pROC curve of each network measure for miR-1 using the pSILAC dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold. 28

Figure 4.4 pROC curve of each network measure for miR-16 using the pSILAC dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold. 29

Figure 4.5 pROC curve of each network measure for miR-124 using the TarBase dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold. 30

Figure 4.6 pROC curve of each network measure for miR-16 using the TarBase dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold. 31

Figure 4.7 pROC curve comparison of TargetScan total context score and SVM using the pSILAC dataset for miR-16. Results for other miRNAs are given in Supplement 3. Points on the curves correspond to different TargetScan total context scores and SVM probabilities. The point on the TCS curve represented as “tcs” shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the SVM curve show the sensitivity and precision at fixed stringency levels of 0.5, 0.4, 0.3, 0.2, and 0.1. 34

Figure 4.8 pROC curve comparison of TargetScan total context score and SVM using the TarBase dataset for miR-29. Results for other miRNAs are given in Supplement 4. Points on the curves correspond to different TargetScan total context scores and SVM probabilities. The point on the TCS curve represented as “tcs” shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the SVM curve show the sensitivity and precision at fixed stringency levels of 0.5, 0.4, 0.3, 0.2, and 0.1. 34

Figure 4.9 pROC curve comparison of TargetScan total context score and MD using the pSILAC dataset for miR-1. Points on the curves correspond to different TargetScan total context scores and MD distance margins. The point on the TCS curve represented as “tcs” shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the MD curve show the sensitivity and precision at fixed stringent levels of +0.0, +0.2, +0.4, +0.6, +0.8 and +1.0. 36

Figure 4.10 pROC curve comparison of TargetScan total context score and MD using the TarBase dataset for miR-16. Points on the curves correspond to different TargetScan total context scores and MD distance margins. The point on the TCS curve represented as “tcs” shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the MD curve show the sensitivity and precision at fixed stringent levels of +0.0, +0.2, +0.4, +0.6, +0.8 and +1.0. 37

Figure 5.1 The regions that may contain the class members when the Boundary Rule is applied at different ratios. 39

Figure 5.2 The effect of ratio to the outcome of DiVo-E algorithm. 44

Figure 5.3 The effect of ratio to the outcome of DiVo-M algorithm. 45

Figure 5.4 The histogram for Euclidean Distances in Biomed Data. Bin size is 50. . . . 46

Figure 5.5 The histogram for Euclidean Distances in Dermatology Data. Bin size is 50. 46

Figure 5.6 The 2D output of the kPCA for Biomed dataset. 47

Figure 5.7 The 2D output of the kPCA for Dermatology dataset. 47

Figure 5.8 The 2D output of the PCA for Ionosphere dataset. 48

Figure 5.9 The 2D output of the kPCA for Ionosphere dataset. 48

Figure 5.10 Total training times required by each algorithm on different datasets. 49

Figure 6.1 Filtering TargetScan with DiVo using TarBase target database. The x-axis represents stringency and y-axis shows recall values. 57

Figure 6.2 Filtering TargetScan with DiVo using pSILAC target database. The x-axis represents stringency and y-axis shows recall values. 58

CHAPTER 1

INTRODUCTION

Proteins are the main entities that keep the organisms running in an orderly manner. To accomplish this demanding task, they operate either on their own or form complexes with other proteins [37]. By forming complexes, they may increase their level of effectiveness or start to exhibit a different functional behavior other than the functions of their components [65]. Either way, interaction between the proteins in an environment is an undeniable fact and must be taken into account for virtually all processes in a cell, since, assuming that the outputs of isolated proteins will also be observed in a cell environment with many more molecules around, may lead to unrealistic and unexpected results.

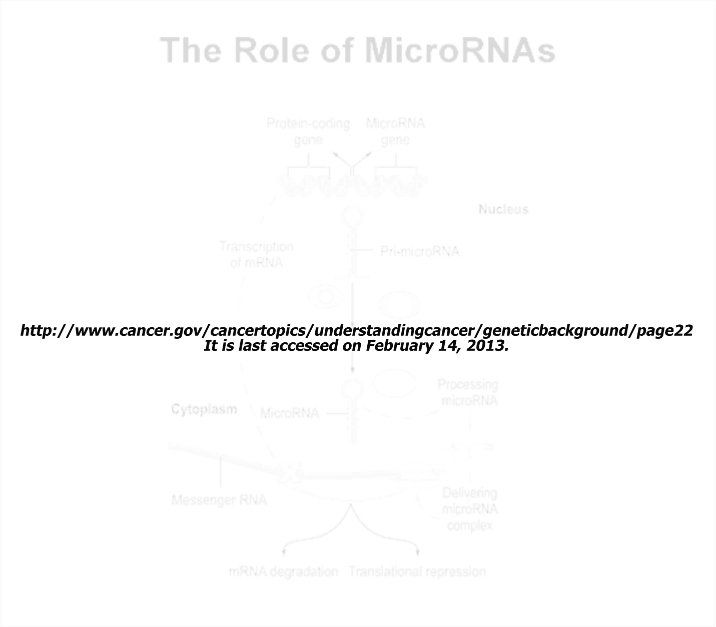
The basic abstraction level of protein-protein interactions is an undirected graph [66]. The nodes of the graph represent proteins and each edge shows the relation between its nodes. Although this simple visualization does not capture all the essential properties [61], it provides important features that are meaningful for both biological and computational purposes. With this abstraction, we can employ algorithms designed for undirected graphs to extract biological information. On the other way around, we can explain some natural facts by translating them into mathematical models. Furthermore, we can make key predictions on the viability of the existence of some new molecules or interactions.

Regulation of metabolism is the ultimate aim of the interactions between the proteins. To preserve the state of a cell, or to adapt for a new state, negative feedback mechanisms are essential. Other than that, in removing an illnesses, inhibiting the protein activity of the pathogen is crucial. Even if there exists no state changes or anomalies, a cell should be able to program itself to death after a certain period, which is called apoptosis [4]. For similar reasons, there must be some inherent mechanisms shared by multicellular organisms that should provide a safe and sound inhibition on protein activity. microRNAs are part of one of those mechanisms.

microRNAs (miRNA) are small nucleotide sequences of approximately 20 bases [7]. They are translated from the non-coding parts of the DNA step by step [50, 67], as can be seen in Figure 1.1. First, every miRNA is a part of a much longer nucleotide sequence inside the nucleus. This form is named as pri-miRNA. It is around 150 nucleotides long, with a cap at the 5' end and a poly-A tail at the 3' terminal. The source for these pri-miRNAs varies. In a study of Rodriguez et al. [67] it has been found that more than half of them are produced from introns and exons. But the main source is introns.

As the second step, the enzyme "Drosha" operates on the pri-miRNA. It cuts down the pri-miRNA to around 80 nucleotides. This form consists of a stem-loop structure, characteristic to each miRNA and known as pre-miRNA. Still inside the nucleus, pre-miRNA waits to be transported out to the cytoplasm. There, the enzyme "Dicer" cleaves the pre-miRNA down to the mature miRNA.

miRNAs primary focus is to interact with gene regulation. Basically, their function is to down



http://www.cancer.gov/cancertopics/understandingcancer/geneticbackground/page22
It is last accessed on February 14, 2013.

Figure 1.1: miRNA Production Steps - Artwork originally created for the National Cancer Institute. Reprinted with permission of the artist, Jeanne Kelly. Copyright 2012.

regulate the production of proteins [4]. However, they do not operate on their own. T hey bind themselves to a RNA - Induced Silencing Complex (RISC) molecule. With that, they aim their specific targets inside the cell, which are most of the time messenger RNAs (mRNAs).

Another feature that miRNAs take part is developmental timing [63]. For various organisms, it has been shown that miRNAs play important roles in different stages of the development of the organism. These can either be in early or late developmental timing. Moreover, miRNAs are also a part of programmed cell death.

There are two ways to prove that an organism contains a certain type of miRNA. The first one is to do experimental work in a biology lab. There are different methods devised for this purpose and forward genetics is one of them. A mutation phenotype is the starting point and scientists try to figure out the mutated gene / protein. The very first miRNAs, which are lin-4 [44] and len-7 [3] , were found by the help of this method. Although this method discovered the very first miRNAs, it is used to identify very few of them because miRNAs are small and tolerant to current programmed mutations. Moreover, it is really hard to generate phenotype mutations that will lead to mutated miRNAs.

Another way is to clone RNA from complementary DNA (cDNA) libraries and decide whether it is a miRNA [55]. The importance of this method lies in constructing a concrete basis of information about the properties of the miRNA. This information is the cornerstone of the main miRNA identification method: the computational miRNA prediction.

Current major challenges in miRNA research are:

1. identification of novel miRNA genes
2. functional annotation of miRNAs
3. identification of genes targeted by miRNAs [6] [1]
4. high throughput experimental validation of miRNA-target interactions

In this study, we focus on the problem of identification of human miRNA target genes.

Computational prediction methods require strict rules or distinctive features to be successful. The biological methods do provide these distinctive features and computer applications use them in such a way that more miRNA can be predicted. For example, the secondary structure of precursor miRNAs have specific stem-loop hairpin and this is an important clue for computational prediction methods [55]. Although this structure is different in animals, plants and viruses, RNA folding programs can be effectively used for exploiting this structure information. Moreover, thermodynamic properties of pre-miRNA are also important. It is proposed that the free energy of a precursor miRNA should be lower than a fixed threshold value. Many other distinctive features are also proposed, including symmetric difference, number of base pairs, GC content, etc. in order to increase the success rate of miRNA prediction algorithms [54, 55, 96].

It is widely believed that miRNAs are evolutionarily conserved in related species [29]. This means that, if a miRNA is identified in an organism, it is highly possible that the relatives of this organism will also code that miRNA. miRNA prediction algorithms almost always keeps use this feature [50], since looking for homologies is a known and widely studied topic. One of the reasons of this is that closely related species have homolog proteins and homolog proteins mean homolog mRNAs, i.e. miRNA targets. Therefore, searching for the homologs of miRNA targets will also discover new miRNAs.

This conservation principle may serve well for plants and animals. Yet, it fails in some cases. For instance, all organisms have miRNAs that are specific for their brethren. Most probably their rapid evolvement, viruses show their "sympathy" for conservation the least. Hence, ab initio miRNA predictions should not be underestimated. One way is just using the distinctive features of miRNAs. Another way can be banking on the assumption that most animal miRNAs are clustered [95]. So, for an animal genome, searching for new miRNAs in the vicinity of previously proven miRNAs may extract other miRNAs.

Predicting them in silico is, of course, not enough to prove that they really exist, because computational methods only provide the pre-miRNA. They cannot locate the exact nucleotide positions of miRNA. By means of biological methods, miRNA ends inside the pre-miRNA

---

[1] miRNAs target mRNAs of genes; however, throughout the manuscript we use the corresponding gene products, i.e., proteins as miRNA targets and ignore the differentiation in gene products due to post-translational modifications.

should be found the and their miRNA abilities must be verified. Cloning and sequencing of small RNAs [55] yield the best possible results in candidate validation.

miRNAs are of no use without their targets. They must, interact with their target molecules and accomplish their intended functions. To understand these regulatory functions, the targets of each miRNA should be identified. Almost all of the miRNAs target mRNAs. Since miRNAs mostly target mRNAs, the target prediction methods are being developed accordingly. It should also be known that a miRNA can target more than one mRNA, while an mRNA can be regulated by more that one miRNA. Therefore, there is a many-to-many relation between miRNAs and their targets.

The aim of the computational methods which try to find miRNA targets take advantage of the facts below:

- The complementarity between miRNA and its target,
- The thermodynamics in RNA-RNA complex (This complex tend to have higher negative folding energy), and
- Homology (Within the same kingdom, the conservation of miRNAs, binding sites of mRNA to miRNAs and miRNA-mRNA complexes is quite high)

The tools designed for miRNA target prediction generally take a step-by-step approach [54]. They first check for the complementarity between the miRNA and possible targets. Later, the thermodynamic eligibility of the RNA-RNA complex is investigated. The folding of this complex should have a free energy that has an experimental viability. Of course it is infeasible to search for all possible targets. Instead, using the homology argument, the miRNA targets that exist in the organisms that are in the same kingdom are more likely candidates. As the last step, these predicted miRNA targets should be experimentally verified by the biological methods. The results of both computational and experimental studies should be kept for further access. Many databases are developed for storing the information about miRNA and their targets. The databases may either store manually curated, quality oriented targets, or quantity oriented, high-throughput candidates.

Despite these features, studies show that current computational miRNA target prediction techniques have high false negative rates [54, 55, 96]. Since the sequence complementarity is imperfect and the length of the miRNA is too small, simple sequence comparison produces many tentative targets. Applying thermodynamic rules and searching for homology are two different filters that eliminate infeasible predicted targets.

In this work, our focus is to provide another feature, the topological properties of proteins in the PPIN, as a filtering mechanism in miRNA target prediction. Different studies in the literature investigated whether the known miRNA targets and their topological propertied are related [35, 47, 49]. We used this information and generated a process to leave the predicted targets that have less favorable network properties out, in order to lower the high false positive rate of a widely used miRNA target prediction tool.

Having training data for only one class may lead to an unsuccessful or incomplete training phase for the classification. In our case, we only have the experimentally validated miRNA targets for training. There exist two alternatives [90]. One is to generate a putative dataset which is assumed to represent the *other* class and the second one is to develop an algorithm that only needs training data from only one. The former approaches try to transform available data into a form where readily established multi-class classification techniques can be applied. Negative training data is artificially generated. Here, negative data generation problem has its own challenges. First of all, the process of counter example production is also a classification problem in itself. It is usually difficult to prove or disprove the fidelity of the generated negative dataset in representing the real set of principles that covers all possible real negative data. Especially, if the true positives are not covered well in the training data, a negative data generation method, which aims to cover to whole space of negatives, may lead to a high false negative rate. The success of this artificial generation process definitely affects the estimation accuracy of the classifier.

We followed both alternatives. First, we employed a widely known classification algorithm. The requirement here was that there must be two training sets for both miRNA targets and non-miRNA targets. The miRNA target databases provide the former. However, no dataset exists for the latter. Henceforth, by applying our own heuristic, we populated putative non-miRNA targets. The by-product of this work can be used as a training set for negative data.

Second, we wanted to utilize only miRNA targets as the positive training data throughout the machine learning process. Supervised classification techniques in machine learning depend on gathering training samples and test them on instances that belong to an unspecified class. As in our case, there may be situations where it may not be possible or feasible to gather training data for each class. If samples from only a single class are available, the problem is known as the One-Class Classification (OCC in short) problem [85].

For this, we devised a new algorithm, DiVo, which facilitate a distance based heuristic with a voting mechanism. We investigate two distance metrics, the Euclidean distance and the Mahalanobis distance. We also compare these metrics on different datasets.

To summarize, the contributions of this work can be enumerated as follows:

1. We have constructed a negative gold standard dataset required by supervised learning frameworks for miRNA target prediction. We used many miRNA target databases and PPIN to extract these putative non-miRNA targets. To our knowledge, there is no other dataset that can be used as a training data for non-miRNA targets.
2. We have integrated various graph theoretic measures into a statistical model. We applied this statistical model to a popular miRNA target prediction tool. The implemented model is used as an additional filter in miRNA target prediction. It is not tailored for this specific tool, however. It can be used for other miRNA target prediction tools as well.
3. We devised a novel, intuitive and easy to understand OCC method, DiVo, based on simple distance metrics and voting. We compared its performance with the state of the art methods.

The rest of this dissertation is organized as follows. In Chapter 2, we will present the theoretical background that is necessary for accomplishing this study. Chapter 3 mentions about previous work on miRNA target prediction and One Class Classification. Chapter 4 will present the machine learning mechanisms and their guidance on differentiating less likely miRNA targets from the putative ones. Chapter 5 is dedicated to unveil the thinking behind our novel One-Class Classification algorithm. It will also cover the time and performance analysis of our algorithm and comparison of it with many other known One-Class Classification methods. Chapter 6 is about applying our algorithm to the PPI network properties and explaining the results on miRNA target prediction filtering. Chapter 7 will discuss the overall achievements and conclude with future directions.

CHAPTER 2

BACKGROUND

The computational identification of miRNA targets is a typical classification problem. There are many organic molecules in a cell that may be targeted by miRNAs. Either manually or computationally, these possible targets should be classified. For automatic classification, some well-defined rules should be proposed. These rules are products of the experimental studies, conducted by scientists. The scientists observe different properties that are common in all the interactions between the miRNAs and their targets. These observations can be incomplete for today, but they are useful in predicting more miRNA targets. The computational methods exploit these theoretical and practical information. The first part of this chapter presents these information as the backbone of our study.

The extracted properties simply cannot define a classification pattern by themselves. Using the same set of properties, different approaches can be put forward. Some can achieve successful results, the others cannot. The difference between these results stem from diverse usage models of the same practical observations. The second part of this chapter defines the One-Class Classification problem and specifies the fundamental properties of different techniques solving this problem.

2.1 Computational miRNA Target Prediction

The miRNAs are executive molecules, operating mainly on mRNAs. mRNAs are responsible for transferring genetic information from DNA to ribosome, where proteins are produced. This shows that the regulatory operations of miRNAs depend on their targets. Henceforth, specific purpose of each miRNA cannot be understood without its specific targets. The key point in understanding what a miRNA does is identifying its possibly many targets.

2.1.1 Basics of miRNA-mRNA Interaction

The dynamics of observed miRNA-mRNA interactions provide important evidence for miRNA target discovery. The experimental results indicate that interactions are different for plants and animals. In plants, a miRNA connects itself to translated part of the mRNA, forming a duplex, and degrade it. Though exceptions exist, plant miRNAs are almost always exactly show Watson-Crick sequence complementarity of their target mRNAs. The case is different for animals and viruses. Not only does miRNA bind to the 3' untranslated region (3'UTR) of mRNA but it also does not exactly complement to the part it is connected. In fact, there are a few duplex patterns, as described in a study of Maziere et al. [54]. First, the connection site may have canonical matching. This means that, the 3' UTR matches almost perfectly along the full length of the miRNA with one or no mismatches. Second, the 3' UTR and 5' end of the miRNA matches perfectly but the remaining parts do not show any specific matching pattern. And lastly, there may exist a near perfect matching in the 5' end of the miRNA, followed by a high quality complement in the other end of the miRNA. There is a big mismatching region in between.

Common to the all three types of interactions, there exists a region in the 5' end of the miRNA, the nucleotides in position 2-7. This region almost always perfectly matches with the 3' UTR of its target. This location is called "the seed". The methods employing Watson-Crick complementarity try to match the seed location of the miRNA with its possible target. Later, they may search for other nucleotide sequence based features, including G:U wobbles [78] or enlarge the seed location matching as long as it gets [36].

With this hindsight, we conclude that there should be a set of well-defined 3' UTRs for successfully identifying miRNA targets. This may not be the case for all species. Although the genome wide sequencing of human DNA is a rapid developing research are, roughly 30% of the human 3' UTRs have ill-defined nucleotide sequences [54]. This creates lots of problems for algorithms that depend heavily on the 3' UTR sequence information.

Suppose that DNA sequencing is completed, the seed matching and other miRNA sequence complementary searching over mRNAs is just a starting point in miRNA target discovery. When we realize that the length of the seed is only 7 nucleotides, it becomes clear that there will be many hits throughout all the human 3' UTRs. To be precise, the hit count per 3' UTR is 1 on the average [54]. This means that every miRNA can target every mRNA. However, current studies on Homo Sapiens DNA indicate that there are more than 800 miRNAs [10] and there are reports predicting the miRNA target coverage on genes to be ranging from 30% [6] to 60% [29]. This shows that relying only on seed matching will yield many false positives.

To overcome this, different features have been proposed [96]. The most prominent of them is the viability of the free energy of the miRNA-mRNA complex. The free energy of this complex is generally calculated by Vienna RNA Package [34]. The free energy threshold used in studies is -20kcal/mol [63, 64]. The lower the free energy is, the most likely the miRNA-mRNA duplex viable is.

Another important observation is about the conservation of miRNAs across different organisms. If the predicted connection site also appears in orthologous 3' UTRs in other species, then it means that their common ancestor may also have this miRNA-mRNA interaction. This strengthens the existence of a valid miRNA target. One of the problems with this feature is that, the species in the same family have more or less the same transcript. The distinction is lesser and the miRNA target search will most likely yield a positive result. However, not every miRNA exists for all the members of the same family. Also, this filtering mechanism strictly requires as many sequenced species as possible, which may not be always available.

Combining the sequence information and other features, the miRNA target prediction problem can be formally defined as in the work of Yue et al. [96]. Let the prediction algorithm be a function, $f()$, which takes a set of feature values, $F = (f_1, f_2, \ldots, f_n)$ as an input and maps it to 1, if it predicts the miRNA target as valid, or it maps it to 0, if it decides that it is not a valid target for that miRNA. The feature set, F represents the possible target under consideration. One of the features is sequence information. Let the sequence information be f_i. If the length of the miRNA is l, then $f_i = (f_{i_1}, \ldots, f_{i_l})$, where $f_{i_j} \in \{A, U, G, C\}$ is the nucleotide in the jth position from the miRNA's 5' end. Let the length of the 3'UTR of the mRNA, r be k. Then, $r = r_1, \ldots, r_k$, where r_m is the nucleotide in the mth position from the mRNA's 3' end. The method should decide on whether the interactions between all j's and m's conform the rules imposed by the method.

2.1.2 miRNA Target Prediction Tools

Understanding the basics of miRNA-mRNA interaction is the first and the most important step taken for computational miRNA target prediction. If the features are clear enough, then the automatization process should be seamless. However, the fundamental problem for the time being is that miRNA target detection mechanism is too immature. There are still many unknowns and uncertainties. This leads to many target prediction algorithms taking different approaches. The rule based algorithms try to exploit the rules derived from the features such as complementarity, seed matching, free energy calculations and evolutionary conservation. On the other hand, data driven approaches mainly use the training data to gather information and apply it to the test cases. The following sections will present the widely used miRNA target prediction tools.

2.1.2.1 miRanda

miRanda [36] is one of the first miRNA target prediction algorithms. It is rule based. It applies a three-phase approach. As the initial step, it looks for sequence complementarity at both ends of the miRNA. The algorithm uses a scoring matrix for assigning a value to each nucleotide interaction. It does not search for the seed or any other specific location. Therefore, more than one interaction sites can be discovered. If the Watson-Crick complementarity score is sufficient, it checks whether the duplex composed of miRNA and its target is structurally stable. This is accomplished by RNAfold [34] As the last step, it looks for the conservation of that target among different species. If the default parameters are used, it is reported that the false positive rate fluctuates between 24% and 39% [54].

2.1.2.2 TargetScan and TargetScanS

TargetScan [46] is one of the most popular miRNA target prediction tools. Aiming to reduce false positive rate, it uses a rather different type of target prediction approach. It first scans for perfect complementarity at the seed region. That is the main driving force behind the method. Any target without a seed match is directly eliminated. Successfully seed matched miRNA targets are subjected to the species conservation criteria among a number of selected species. This step also serves for decreasing false positive rate. As the last step, thermodynamic viability is tested. TargetScanS is another variant of the original algorithm. It modifies the seed region matching and removes the free energy calculation step altogether. Both are rule based.

Especially the imposed seed matching criteria helps lowering the false positive rate to 22%-30% range [54, 96] at the expense of rejecting valid miRNA targets that show different secondary structure characteristics in the seed region.

2.1.2.3 PicTar

PicTar [43] starts with a set of miRNAs and their possible targets which are orthologous to each other. Hence, from the very beginning, the number of species under consideration is fixed. The characteristic property of PicTar is that, it takes not only the miRNA as an input, but also its co-expressed miRNAs. This idea is based upon the showings [8] that a miRNAs and its neighbor miRNAs are co-expressed. These factors make PicTar a data driven approach. A program called *nuclMap* searches for best possible target sites. These sites include the seed regions. The conservation of these seed regions in the orthologous mRNAs is very important. Later, these target sites are filtered according to their optimal free energies. If the regions of miRNAs that are filtered by the first two steps appear in all target species, then these regions are called *anchors*. The miRNAs having a minimum of user defined anchors are subject to more filters, which are defined by seed region matching and free energy calculations. These filters resemble the first step in PicTar but this time, no orthologous conservation is required. At the very end, the least likely candidates are eliminated. The remaining ones complete a training set that will be used in training a Hidden Markov Model (HMM). This model assigns scores to each 3' UTR. As the last step, the different alignments in the 3' UTR part of the same target are combined to generate the final score for that miRNA target.

2.1.2.4 DIANA-MicroT

The distinctive property of the rule based DIANA-MicroT algorithm [41] is its usage of a fixed-width screening window on the possible miRNA target in different organisms. The window size is 38 nucleotides. It slides the window one nucleotide at a time. Since there are more than one species, a two dimensional matrix is created. A type of dynamic algorithm is employed on this matrix data structure to calculate the minimum free energies (MFE) of each 38 nucleotide segment. After that, 5 more features are inspected. These are mostly sequence and complementarity based features. A test mRNA must comply with all of these 5 rules to be tagged as a miRNA target. The false positive rate of DIANA-MicroT algorithm is more or less the same as the others.

The performance of all the algorithms here and unmentioned others can be compared as in a study from Alexiou et al. [2]. A group of them has a precision of around 50%. There is a trade-off between specificity and sensitivity. But the main problem is the high false positive rates. The reasons behind the interaction between a miRNA and its targets are not fully understood and this is clear from the results of miRNA target prediction methods.

2.2 The Protein-Protein Interactions

Cellular machinery is a precise automation orchestrated by the DNA. The DNA is not the sole actor, however. RNAs, inorganic molecules and the environmental factors play important roles. All these components help or hinder the clockwork like workings of proteins. The proteins generally cooperate with each other to fulfill their duties. The cooperation mechanism is called Protein-Protein Interaction (PPI). This interaction starts with a physical contact, such as molecular docking. The cell is a crowded place and every molecule can touch and go to others randomly. Moreover, some interactions are generic. These occur in protein production, folding and degradation. So, to define an interaction as a PPI, two facts should be checked. First, the physical docking should not be random. It must be repeatable in the same environment. Second, the interaction must happen for a specific purpose [65]. For this work, we will concentrate on PPI between only two proteins. A comprehensive list of PPI databases can be accessed from `http://www.pathguide.org/` (last access on February 14, 2013)

If we represent each PPI as a two-node, one-undirected-edge graph, then the whole set of proteins of an organism having v proteins and e PPI, originate a larger graph which has v nodes and e edges. This is called Protein-Protein Interaction Network (PPIN). The fundamental model used for PPIN is an undirected, unweighted graph, $G(v, e)$. This way, researchers can apply graph centric algorithms to PPIN.

The computer science field is a rich one on graph algorithms. That said, to employ the proper algorithms, we must understand the real world PPIN examples. In general, they are sparse graphs. There exists many nodes but the number of edges is low. The number of interactions of a protein is called *degree*. Many proteins tend to have small degrees, but some of them have interact with many other proteins, meaning that they act as hubs. Also, the degree of a protein is independent of the number of all proteins. This shows that PPINs can be modeled by $scale - free$ networks. However, some objections have been raised [62]. The problem with this modeling is that, for the time being, the PPIN are noisy and incomplete. This means that we only have access to a proper subset of the real PPIN. It is also known that subsets of scale-free networks are not scale-free [81].

Thinking of PPINs as static entities is misleading. Any PPI depends on cell type, state, developmental stage, environment, modifications, cofactors and other binding factors [65]. This means that all interactions are not available at all times and temporal information is an indispensable part of the PPIN modeling. These imply that the scale-free models may not be the best fit for PPINs.

Another important property of graphs is the number of steps required to convey information from one node to another on the average. This is called *closeness*. Generally PPIN have small closeness values and thus PPINs have $small - world$ property.

The discovery of PPI and construction of PPIN can be obtained in different ways. The small scale validations are called "binary" methods [65]. Here, the two proteins are taken and it is observed whether they interact or not. It yields less but accurate and direct interactions. Yeast-two hybrid [82] (Y2H) is an example of "binary" methods. There also exists large scale discovery approaches. These are called "co-complex" methods [94]. This time, two groups of proteins are taken and their both direct and indirect interactions are recorded. Later, the spoke hub-distribution is used to eliminate irrelevant interactions. Tandem affinity purification coupled to mass spectrometry [11] (TAP-MS) is a widely used co-complex technique.

The human PPI network construction and visualization is a constantly evolving area of research. Database of Interacting Proteins [70], STRING [83], BioGRID [79], Human Protein Reference Database [60] and Ophid I2D [13] are all active protein interaction data sources.

2.2.1 Network Topological Measures

In this section, we give descriptions of the network topological measures we have used (listed in Table 2.1). For all the formulations that are described below, the PPI network is represented as a graph $G = (V, E)$, where V is the set of nodes and E is the set of unweighted, undirected edges.

Betweenness centrality: Betweenness centrality [28] is a topological measure in graphs for determining the bottleneck nodes. Nodes that occur on many shortest paths have higher betweenness centrality values compared to other nodes that do not. This metric is commonly

Table 2.1: Network topological measures. The seven graph theoretic measures that are used as indicators of a protein's context in a PPI network.

Measure	**Locality**	**Short Description**
Degree	local	number of neighbors of a node
Betweenness Centrality	global	ratio of number of shortest paths that pass through a node over number of all shortest paths between all nodes
Closeness Centrality	local	reciprocal of sum of distances to all reachable nodes
Clustering Coefficient	local	describes how connected the neighborhood of a node is
Eigenvector Centrality	local-global	a measure of importance of a node similar to Google's PageRank
Graph Centrality	local	reciprocal of maximum distance to any reachable node
Stress Centrality	global	number of shortest paths that pass through a node

used for identifying vertices which affect the data flow in the network. The effect of a node to the communication between other nodes is required in identifying important yet non-hub proteins [31]. The betweenness centrality of a node i is given by the following equation:

$$B_i = \frac{\sum_{\forall a,b \neq i} d_i(a,b)}{((n-1)(n-2))/2}, \qquad \text{for } n > 2 \tag{2.1}$$

which gives the ratio of the number of shortest geodesic paths between node pairs (not including i) that pass through node i over the number of all node pairs in the graph except node i. Here, n is the total number of nodes in the graph, and $d_i(a,b)$ is the ratio of shortest paths between nodes a and b that pass through node i over the number of all shortest paths between a and b.

Closeness Centrality: Closeness centrality shows how close a node is to the others. A node having a high closeness centrality means that information will spread quickly to the nodes which are reachable from that node [31]. Formally it can be defined as:

$$CL_i = \frac{1}{\sum_{t \in V} d_G(i,t)} \tag{2.2}$$

In this formula, we take the reciprocal of the sum of distances to all reachable nodes. This way, the nodes that are near to many nodes will have smaller total distances, hence larger closeness centrality values.

Clustering coefficient: Clustering coefficient [91] is a measure for determining how the neighbors of a node are interconnected. In other words, it measures the modularity of the neighborhood of a node. The clustering coefficient of C_i of node i in the network is defined with the following equation:

$$C_i = \frac{2m}{k_i(k_i - 1)} \tag{2.3}$$

where k_i is the degree of node i, i.e., the number of links incident to node i, and m is the number of edges in the induced subgraph of the neighbors of i (not including i). The clustering coefficient of a node gives the ratio of the existing edges in the node's neighborhood to all the possible edges that can occur between the neighboring nodes. Note that for a node with degree 0 or 1, the clustering coefficient is undefined. Such cases can be treated specially and the clustering coefficient can be set to 0.

Degree: The degree of a node in an undirected graph is simply the number of links connected to that node. It gives a measure of local connectivity of the node. This measure can be used to identify hub proteins in a PPI network. The hub proteins are essential since their knockdown leads to lethality [31].

Eigenvector Centrality: Eigenvector centrality is closely related to Google's page rank [58]. Relative scores are assigned to each node in the graph proportional to the connectivity and accessibility of nodes. We use the power iteration method to compute eigenvector centrality.

Graph Centrality: Graph centrality is another type of closeness. It gives information about how far the nodes are from each other. The important factor in this measure is the farthest node to the node in consideration. Only that distance adds up to graph centrality measure.

$$G_i = \frac{1}{\max_{t \in V} d_G(i,t)} \tag{2.4}$$

For node i, the distance to any reachable node t is calculated as $d_G(i,t)$. Then, maximum of these distances represents the graph centrality measure for node i.

Stress Centrality: Stress centrality is very similar to betweenness centrality in that, while betweenness centrality takes the number of all shortest paths passing through the start and end nodes into account, stress centrality does not. It only deals with the number of all shortest paths which include the node i. It provides another centrality measure which is based on shortest paths.

$$S_i = \sum_{s \neq i \neq t \in V} \sigma_{st}(i) \tag{2.5}$$

Here, $\sigma_{st}(i)$ represents the number of shortest paths starting from node s, passing through node i and ending at node t.

2.3 Computational Classification

In the broader sense, labeling an input value is called Pattern Recognition. When an input is given, the authority, who has the right to assign the label, searches for discriminating patterns. Prior to the search action, there must be some training step. Without a training, the authority has no knowledge about patterns and labels. If we know the training data and its real label, then the learning is known as supervised learning. If the training is supervised and the labels are classes, the assignment process becomes Classification. Computational Classification is the class assignment accomplished by an automata to test samples after a period of supervised learning.

Generally, the number of classes under consideration is two or more. When there are two classes to be assigned, it is called binary classification. If it is more than two, it means that we have a multi-class classification problem. The algorithms devised for binary classification, either inherently support multi-class problems, or can be converted to solve them. Therefore the main focus in this field is on binary classification. In this section we will present two methods designed for binary classification.

2.3.1 Support Vector Machines

Support Vector Machine, a technique proposed by Vapnik in [88], is a supervised learning algorithm that provides discriminative statistical models for classification problems. SVM technique has been applied in various domains from computer vision to web mining. SVMs have also provided solutions for a number of bioinformatics problems. An extensive review of these solutions can be found in the work of Noble [57]. Given a set of objects $x_1, \ldots, x_n \in \mathcal{X}$ (an object is a protein in this context), and a series of labels $y_1, \ldots, y_n \in \mathcal{Y}$ associated with the objects, support vector machines can be used to learn a function $g : \mathcal{X} \to \mathcal{Y}$, which can be used to predict the label of any new object $x \in X$. SVM is a kernel based technique in which the objects are represented by their similarities to other objects using a kernel function. Given a set of training objects (a combination of gold-standard positives and gold-standard negatives), SVM learns a hyperplane (possibly in a higher dimensional space) to separate the two classes of objects. This hyperplane defines two half-spaces of objects classified positively and negatively, and the goal of SVM is to maximize the distance between these two half-spaces. A more comprehensive description of SVMs can be found in a study of Schoelkoph et al. [72]. We used the libsvm [18] implementation to construct the SVMs used for this work.

2.3.2 Mahalanobis Distance

Euclidean distance is a metric distance which treats each dimension separately and does not account for scale differences or correlations between dimensions. In 1936, P. C. Mahalanobis introduced a distance measure for computing the distance of a sample to a multivariate distribution. Mahalanobis Distance takes into account different scales of dimensions and also the correlation between different dimensions. Mahalanobis Distance is computed using the equation

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1} (x-\mu)} \tag{2.6}$$

where x is a multidimensional sample point, μ is a vector representing the means of each variable, i.e., dimension, and S is the covariance matrix. S captures the correlations between different dimensions and the variance within a single dimension.

Mahalanobis Distance is used in classification problems and cluster analysis. In this study, the Mahalanobis Distance of predictions of TargetScan are calculated to both gold-standard positive and gold-standard negative data sets. Then, by comparing the two distances, we identify possible false positive targets.

2.3.3 One-Class Classification

For a classification, when the number of classes under consideration is two or more, many alternatives are ready to be chosen from. There is always the possibility of transforming a binary classifier to multi-class classifiers. The problem arises if the case has only one class. This may happen surprisingly frequent. Especially in application domains such as molecular biology, proving membership to a class may be easy but disproving may be too time-consuming, expensive, or even impossible. Therefore, in such cases, classification algorithms have to work with insufficient training data. The research area dealing with these problems is One-Class Classification.

When there exist sufficiently many training examples, the estimation error of the model tends to decrease [77]. Otherwise, there may be many false positives and false negatives. Although, it is convenient to have as many training data as possible for each class under consideration, it may not be possible or feasible to collect sufficient training data. There are various techniques to remedy for this problem. One is to generate a putative dataset which is assumed to represent the *other* class [51] and the second one is to develop an algorithm that only needs training data from only one class [22]. The former approaches try to transform available data into a form where readily established multi-class classification techniques can be applied. Negative training data is artificially generated. Here, negative data generation problem has its own challenges. First of all, the process of counter example production is also a classification problem in itself. It is usually difficult to prove or disprove the fidelity of the generated negative dataset in representing the real set of principles that covers all possible real negative data. Especially, if the true positives are not covered well in the training data, a negative data generation method, which aims to cover to whole space of negatives, may lead to a high false negative rate. The success of this artificial generation process definitely affects the estimation accuracy of the classifier.

Methods which are specifically developed to work with one class training datasets bypass the artificial data generation stage. There are many one class classification algorithms [24, 39, 53, 85]. Some of them use probabilistic estimations [85] and some form a boundary around the training data [15]. The others construct models that fit the training samples [53]. Even, some of the algorithms designed for binary classification are modified for one class classification [73].

The One-Class Classification problem can be formalized as in the study of Mazhelis [53]. Let's assume that the class under consideration is C and any sample is represented as a vector $x = (x_1, x_2, \ldots, x_n)$, where the number of features for each vector is n. The training set is composed of T samples, $DS_T = \{(x_1, x_2, \ldots, x_n)_j | j = 1, 2, \ldots, T\}$, Test set is composed of X samples, $DS_X = \{(x_1, x_2, \ldots, x_n)_j | j = 1, 2, \ldots, X\}$ Using the training set, DS_T , the one-class classification algorithm will construct a model, Θ, such that, it can decide whether a test sample from DS_X belongs to the class C by applying $u(x, \Theta)$.

The model Θ described in the problem formalization can be produced in many different ways. Generally, the employed mathematical and statistical methods produce the main diversity. The first approach is estimating a probability density function by means of the training data and comparing the output of this function on the test sample with a threshold value. These are Density Methods [53, 85]. The Θ is the probability density function and the threshold value.

Another approach is assuming a *data generator* produced the training data. Because of this, methods using this paradigm are known Reconstruction Methods [53, 85]. The parameters of that *data generator* are found during the training phase. After the parameters are set, the *data generator* and a threshold value can be used to calculate the reconstruction error for any test sample and deciding whether that error is lower than the threshold. So the *data generator* and the threshold value become the model Θ.

The last technique tries to encapsulate the training data by estimating the boundary around it. These are named Boundary Methods [53, 85]. They are specifically designed for One-Class Classification problems. Finding the boundary around the training data is accomplished by calculating the distances between the training data. Different metrics can be used for this purpose. The testing step depends on the same distance calculations, but this time, the distance between the test sample and the training data is important.

In the subsequent sections, we will present some of the methods used extensively for solving One-Class classification problems.

2.3.3.1 State of the Art Methods

To measure the success level of the proposed algorithm, we compare it with some well-established one class classifiers. There are many one class classification algorithms. In this study, we decided to implement five algorithms, which are Global Gaussian Distribution [24], Local Gaussian Distribution [24], k-Nearest Neighbor Data Description [85], k-Nearest Neighbor Data Description with Structural Risk Minimization [15] and Support Vector Machines (SVM) for One Class Classification [73]. We used Gaussian Kernel for OCC SVM, therefore it is theoretically the same as Support Vector Data Descriptor (SVDD) [84]. So the results and comparisons with OCC SVM also apply to SVDD.

Global Gaussian Algorithm simply calculates the Mahalanobis Distance between the test object, x and the training set, T. If the distance is less than a *threshold* value, then x is said to be a member of the class. Otherwise, it is rejected. *threshold* is the only parameter of the algorithm to be adjusted.

In **k-Nearest Data Description**, we first calculate the nearest other training data for every training sample. Later, we find the nearest k training data to the test point, x. This is a proper subset of training set, called set_{xk}. For each element i of this set, we calculate the following:

$$f(i) = \frac{\|dist(i, NN_{tr}(i)\|}{\|dist(NN_{tr}(i), NN_{tr}(NN_{tr}(i)))\|} \tag{2.7}$$

If the result of this equation is less than one, the training data i is said to vote in favor of x. Otherwise, i rejects x. The main idea here is that, if a test sample is closer to a training data than the all other training data, then the membership of that test sample to the class under consideration is more likely. After each training data votes for x, the number of votes is calculated. If it is more than a *ratio* of all training data, then the test data is accepted. k

and $ratio$are the parameters of the algorithm to be adjusted.

Local Gaussian Algorithm blends Global Gaussian method with k-Nearest Data Description. We first find the nearest training data, t, to the test point, x. This time, we find the nearest k other training data to t. These k+1 samples form a proper subset of training set, called set_{xk+1}. Up until now, k-Nearest Data Description is applied. From here, Gaussian approach is implemented. The Mahalanobis Distance from the test data x to this proper subset set_{xk+1} is calculated. If the distance is less than a $threshold$ value, then x is said to be a member of the class. Otherwise, it is rejected. $threshold$ and k are the parameters of the algorithm to be adjusted.

k-Nearest Data Description with Structural Risk Minimization is a modified version of the standard k-Nearest Data Description and Nearest Neighborhood with Structural Risk Minimization [38]. Structural Risk Minimization [87] is a technique developed to overcome the problem of over-fitting. While training machine learning methods, there exists a risk such that the method focuses completely on training data. It produces great results in training but the testing becomes more error-prone. This is called over-fitting.

k-Nearest Data Description with Structural Risk Minimization is first proposed with using only 1 member of the training data [14] and later expanded to use k-nearest training data [15]. The main idea is to lower the number of comparisons required by k-Nearest Data Description while maintaining its performance. The important balance here is that the decrease in number of comparisons should only eliminate comparisons which do not contribute to the classification of test samples.

Assuming that there are n training data and m dimensions, the process of eliminating irrelevant comparisons in training data starts with calculating the center of mass of the training data using the following formula:

$$cmass = \left(\frac{\sum_{i=1}^{n} a_{i1}}{n}, \frac{\sum_{i=1}^{n} a_{i2}}{n}, \ldots, \frac{\sum_{i=1}^{n} a_{im}}{n}, \right) \tag{2.8}$$

There will be two data sets at the end of the training phase. The first one, $rejectedset(RS)$ will contain the outliers. The assumption here is that a certain percent of the training data is noise. They do not belong to the class. The center of mass is used in calculating the distances from each training data to the center of mass. Think of it as a sphere. A $fracrej$ portion of the training data that are furthest to the center of mass form a shell and these are though to be the outliers. They are directly put into the RS.

At the very end of the training phase, a proper subset of the remaining training data will form the second one, which is the $prototypeset(PS)$. The algorithm tries to put as little training data as possible into PS while achieving the zero training error. The minimum number of elements is essential, since it will be used in the test phase. The training data that are not in PS and RS will not be used in the test phase. They are the samples which do not add any useful information for testing.

After training, the shortest euclidean distance, d_1 from the test sample, x to the RS is calculated. The same process is repeated for PS and d_2 is produced. If d_1/d_2 is less than a $threshold$, then it is a reject. Otherwise, it is an accept. This acceptance check is done for k

nearest PS and RS members. At the end, if the accept votes are more, than the test sample is labeled as a member of the class.

SVM for One-Class Classification tries to draw a hypersphere boundary around the training data while occupying the minimum volume. Since the boundary is a type of sphere, it has a center, a and a radius, R. Testing the membership of the sample is checking whether the sample is inside the hypersphere or not, as in the following formula:

$$\|x - a\|^2 \leq R^2 \tag{2.9}$$

SVM for One-Class Classification produces the better results in many experiments [71, 84, 89].

All these algorithms try to exploit the distance information between the data points. Our method also presents a new heuristic on the same distance information. Henceforth, we find it adequate to directly compare the classification power of the aforementioned algorithms with ours.

We implemented our algorithm and also the other methods in Java as a new Weka package [32] except the SVM for One Class Classification. We decided to use the Python scikit OneClassSVM function [59].

CHAPTER 3

RELATED WORK

The discovery of miRNAs has a very short history. Although they are identified in a study by Lee et al. [44], they are classified as a separate RNA and their functions are exposed only within a decade. Despite their brief existence, the work on miRNAs is immense, due to their roles in many important disease, such as cancer and their activities in stem cells for cell differentiation. The proliferation of studies on cancer and stem cells make miRNAs one of the hottest topics in biology.

The constant interest about miRNAs also affected the bioinformatics research field. The computational identification of both the miRNAs and their targets are growing areas of interest. There are many algorithms, tools and databases that contain information about miRNA targets. Each database may contain different miRNA targets for different organisms. For this work, we examined the most widely used miRNA target databases for humans.

Different biological phenomena can be mathematically modeled by different abstractions. PPI networks represent the interactions between proteins in an organism. Some interactions may be temporary and others can be permanent. Although the graph representation cannot capture this, at least it shows the existence of known interactions. We used human PPIN to extract topological properties of human miRNA targets. Also, we benefited the PPIN in generating putative human non-miRNA target set.

In this chapter, we will present the studies and contributions previously made on the computational miRNA target predictions using Protein-Protein Interaction Networks.

3.1 Protein-Protein Interaction Networks and miRNA Targets

Recent discoveries in genetics show that gene expression in higher eukaryotes is regulated in part by small non-coding RNA molecules named miRNAs [4, 7, 48]. In plants, miRNAs regulate protein expression by binding to the coding region of the corresponding mRNA. Plant miRNAs exhibit near-perfect complementarity with the target sites. In animals, miRNAs usually bind to the 3' untranslated region (UTR) of the target mRNA and have only limited complementarity to their target sites [7].

miRNAs are involved in many important cellular processes [23] and their dysregulation is related to anomalies such as cancer [33] and heart diseases [17, 97]. In recent years, several thousand miRNAs were identified in animals and plants.

Computational prediction of miRNA target genes is difficult in animal genomes due to the imperfect pairing of the miRNA with the corresponding target site. Techniques based only on sequence comparison are prone to produce many false positive interactions. Therefore, recent computational target prediction methods [36, 43, 46, 63] integrate filters such as 'stability of the RNA-RNA duplex' and 'target site conservation across species' to reduce the large number of false positive target sites identified by sequence matching only. Some others [40, 93] try to devise different structural features to achieve lower false positive rates. Despite these improvements, studies show that current miRNA target prediction techniques have high false positive and false negative rates. For example, miRanda [36] is estimated to have a false positive rate of 24-39% [9]. Similarly, TargetScan [46] and PicTar [43] are estimated to have false positive rates around 30% [45, 76].

Recently, two studies by Selbach et al. [74] and Baek et al. [5] analyzed proteome-wide effects of miRNA expression. Both studies provide important insights toward high throughput experimental validation of miRNA-target interaction. Baek et al. compared down-regulated miRNA targets to computational predictions by TargetScan. TargetScan provides a total context score, which is a sequence based measure, as an indicator for efficiency of targeting of an mRNA by a miRNA. Based on this score, Baek et al. showed that top third targets predicted by TargetScan correlate better with protein down-regulation [5]. According to their results, about two-thirds of TargetScan and PicTar predictions for miR-223 appeared to be false positive targets. In order to work with a more trusted set of miRNA targets some studies apply different filtering strategies based on the total context score. Cheng et al. [20] use a fixed threshold of -0.20 while Yehya et al. [92] filter out three quarters of the predictions made by TargetScan. These studies indicate that there is a need for additional filters to reduce false positive rates of computational miRNA target predictions. Our study shows that, for targets of some miRNAs, the network context based filter we propose is more effective in reducing false positive rate compared to the total context score, provided by TargetScan.

PPI networks provide systems level view of the complex organization of biological processes in a cell. Genome-scale PPI networks have become available in recent years due to high throughput methods for detecting protein-protein interactions [68]. In addition, accurate computational approaches have been developed which provide increased genome coverage [13, 52].

The information hidden inside biological networks is constantly being investigated. Although signaling pathways can also be used [42], especially PPI networks are under the radar for many studies. In the study of Zhu et al. [98], it has been shown that drug-targets have distinguishable properties. These are PPI network related features. Degree, betweenness and clustering coefficient of nodes in PPIN are calculated. It is reported that drug-targets have higher degree and betweenness and lower clustering coefficient than the non-drug-targets. These hypothesis are backed by another study [56]. There, it is shown that using only PPI network topological properties, only cancer-related drugs can be identified but the same principle does not lead to a successful general drug-target classifier. The integration of features should be the ultimate aim. Although these revolve around human PPI networks, similar approaches are conducted for other organisms as well [26].

Interestingly, in the work of Liang et al. [47], it is presented that having higher degree, betweenness and lower clustering coefficient values is highly correlated with being a miRNA target. The overlap in the results of these papers suggests that, the topological properties derived from PPI networks have undeniable biological meanings and contain valuable knowledge [69].

The reasons for the existence of such biological meanings can be explained by the basics of miRNA operations. The miRNAs either "switch on and off" protein expression, or "tune them down" to required levels [7]. The proteins which interact with many other proteins are more likely to be regulated by miRNAs since, if some interactions of that protein may lead to problematic situations, it must be completely shut down. For other cases, the complete shut

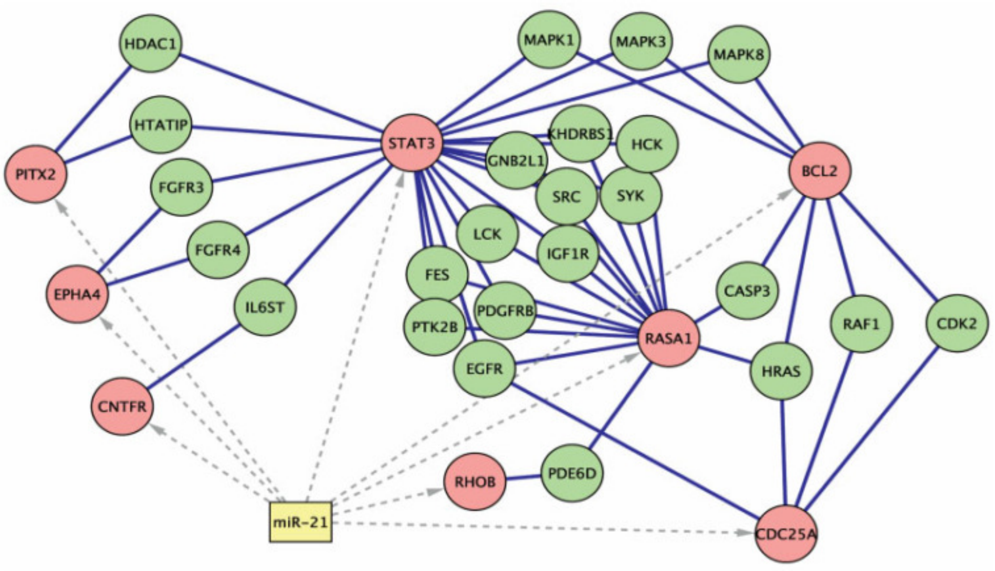


Figure 3.1: miR-21 regulated protein-protein interaction subnetwork. The proteins in red are targets of miR-21, taken from the study Hsu et al. [35]. Courtesy of WILEY-VCH with license number 3061890408961.

down of that protein is not necessary but some of its interactions should proceed while others are to be stopped. Therefore, the protein expression levels must be sustained or lowered, and the protein production output should be diverted to the required interactions [47]. Because of the operational behaviors explained above, the miRNAs can handle both situations.

Two recent studies by Liang et al. [47] and Hsu et al. [35] investigate the relationship between miRNAs and PPI networks, as depicted in Figure 3.1. In their work, they show that miRNAs regulate PPI networks by targeting the mRNAs of hub and bottleneck proteins. In other words, genes targeted by miRNAs tend to have more important roles in the cell compared to *non-target* genes and this difference is reflected in the interactome. Based on this observation, we investigate how the network context of a protein can be utilized for increasing the precision of miRNA target prediction.

It is important to note the main differences of our work compared to the studies from Liang et al. [47] and Hsu et al. [35]. Both Liang and Li and Hsu et al. aim to understand the mechanisms of miRNA regulation of PPI networks by analyzing known targets of miRNAs. They show that various network topological features are correlated with being a miRNA target gene. However, they do not use PPI data for improving miRNA target prediction. In the paper of Linde et al. [49], simple ranking approaches are introduced to order the putative targets for miRNAs that are related specifically to breast cancer. The ranking mechanisms only compare mean values of some network properties from different datasets and apply greater or smaller operators to putative targets. Our goal is to use network context as a filter in a computational miRNA target prediction algorithm and increase the precision of miRNA target prediction by using binary classification methods. When using the One-Class Classification techniques, our aim is to increase the recall (sensitivity) of miRNA target prediction.

CHAPTER 4

USING NETWORK CONTEXT AS A FILTER FOR MIRNA TARGET PREDICTION

In this chapter, we specify the process of applying PPI network context information with computational methods to filter the miRNA target predictions of TargetScan. A major requirement for this method is a gold standard, i.e., ground truth, dataset which is needed for training the classifier. We build our gold positive dataset with experimentally validated miRNA targets. However, a negative gold standard is not easy to obtain as there exists no *non-target* database. As a solution, we construct our putative negative training dataset by using genes which do not appear in any of the major miRNA target databases.

For the genes in our ground truth dataset, we compute various graph topological measures (listed in Table 2.1) on a PPI network. We train our SVM and MD classifiers on these computed measures and construct discriminative models to separate target genes from non-target genes. Given a putative target protein identified by a miRNA target prediction tool, we compute the same network context indicators for this protein and use the learned discriminative models to filter out that protein if it is classified as a non-target gene.

Our preliminary approach is to employ traditional machine learning and statistical methods to miRNA target data. We analyze different ways of integrating network context information as a filter to an existing miRNA target prediction method. The Support Vector Machine (SVM) and Mahalanobis Distance (MD) methods that we implemented allows integration of various contextual measures. These methods are used extensively in many areas of both computer science and bioinformatics. This way, we have the opportunity to evaluate discriminative power of the PPI network information without worrying about the performance and success levels of new methods.

We share the results of applying our approach of filtering the predictions of TargetScan. The first set of evaluation is about the discriminative power of network properties by themselves. Later, we compare these results with the results of two different machine learning algorithms. We investigate whether integrative techniques trained on a dataset of known miRNA targets and putative *non-targets* perform better as a filter compared to simple threshold filters.

4.1 Materials

Application of graph theoretic measures for miRNA target prediction requires the composition of PPI network data and miRNA target databases. The main entity in all these datasets is the *protein* or the *gene* that encodes that specific protein. In the field of bioinformatics, there is an abundance of naming policies for proteins. Different tools and datasets may choose different naming conventions. Widely used protein identifiers are Protein Information Resource (PIR) accession codes (`http://pir.georgetown.edu/`, last access on Febuary 14, 2013), Ensemble Protein Identifiers (`http://www.ensembl.org/index.html`, last access on February 14, 2013)

Universal Protein Resource (UniProt) [21] and protein names. Since we employed more than one data source, it was necessary to use one common universal naming scheme for all proteins. We chose UniProt identifiers for this purpose. We replaced all protein names that exist either in the PPI network or miRNA databases with their respective UniProt primary accession numbers. This way, we were able to identify each protein uniquely and accurately. However, a number of proteins in some databases do not have UniProt identifiers. We ignored such proteins in our analyses.

We use the I2D human PPI network [13] in our results reported in this work. The network is maintained by the Jurisica Lab, Ontario Cancer Institute and University of Toronto. It is a combination of different human PPI networks and is regularly updated. We downloaded it from `http://ophid.utoronto.ca/ophidv2.201/downloads.jsp`. The version is last accessed and downloaded on May 1, 2010. We ignored self-interactions and duplicate entries in the I2D PPI network. The resultant PPI network we used in our experiments contains 13,504 nodes and 91,264 edges, in which each node represents a unique protein and an edge represents the interaction between two proteins.

4.1.1 The Positive Ground Truth Dataset

Verification of miRNA targets can be done in different ways. High quality results can be achieved by testing the interaction between a miRNA and its putative target in vivo. This is time consuming but produces much more reliable validations. For attaining high quantity results in a relatively short period of time, high-throughput screening techniques are essential. The results may yield lower accuracy.

For evaluation of the filtering power of graph theoretic measures and training our discriminative statistical models, we need a gold-standard true positive and a gold-standard true negative miRNA target dataset. We use two different sources as positive gold-standard datasets of miRNA targets. The first one is TarBase v5 [75]. TarBase contains manually curated targets that are verified by high quality, low throughput experiments. TarBase dataset is also used in training the classifiers; however, in our tests we use cross validation by excluding the targets of the tested miRNA from the training set. 663 miRNA targets for 65 miRNAs in TarBase v5 exist in the I2D human PPI network. The second positive ground truth dataset we used is based on a high throughput experimental technique called pSILAC [5]. The pSILAC dataset also contains experimentally validated miRNA targets but the difference is that it is a high throughput validation. Instead of direct miRNA-target interaction, it measures the change in protein expression levels in the presence and absence of a miRNA. The difference of production levels is represented as fold changes. We chose a threshold of -0.2 for fold change as in the study of Alexiou et al. [2]. Therefore, the proteins having a fold change strictly lower than -0.2 are considered as miRNA targets. This way, there exists a total of 1,460 different miRNA target genes for 5 miRNAs used in the study: let-7, miR-1, miR-16, miR-30, and miR-155. TarBase and pSILAC datasets are also used as ground truth set of miRNA targets in a recent study which provides a comparison of several miRNA target prediction tools [2].

4.1.2 miRNA Target Databases

In order to generate a putative true negative dataset, we decided to select genes which exists in PPI network but do not appear in any of the major miRNA target databases. We collected miRNA target information from five different miRNA target databases: miRBase [30], microRNA.org [12], TargetScan [46], TarBase [75], and PicTar [43]. The motivation behind using as many miRNA target databases as possible is to construct a negative dataset that is as accurate as possible.

miRBase [30] uses miRanda [36] miRNA target prediction algorithm. As of June 12, 2009, miRBase v5 contains 463,100 target genes. The data includes the miRNA families, the target genes, and their possible interaction sites. Each interaction site is assigned a score by applying the miRanda algorithm [36]. There were 21,200 distinct human proteins in the data we downloaded and 19,306 of them had UniProt identifiers.

microRNA.org [12] hosts a miRNA target database that is constructed in a similar way as in miRBase. miRanda algorithm is used as the target prediction method. microRNA.org explicitly provides the miRNA families, their target genes, and the sequence alignments between them. miRanda assigns a score to each of these alignments. As of January 2008, microRNA.org contains 18,506 distinct human miRNA target genes in 1,791,961 miRNA-target interaction records. We were able to find UniProt identifiers for 13,352 miRNA targets in the data we downloaded.

We used TargetScan version 5. It has 218,298 miRNA-target interaction records. We identified 7,927 distinct human miRNA target genes in TargetScan. We found UniProt information for 7,695 of these targets.

The version of PicTar we have accessed is composed of four vertebrates, which are human, mouse, rat, and dog. We downloaded the data from UCSC Genome Bioinformatics Site [99], listed under assembly of May 2004. It consists of 205,263 records for 9,152 distinct human miRNA target genes. 8,313 of these genes have UniProt identifiers.

4.1.3 Constructing The Negative Ground Truth Dataset

In addition to the true positive miRNA target set, we also need a true negative dataset for training the SVM and MD classifiers. To the best of our knowledge, there is no database that provides a list of genes that are not targeted by miRNAs. We extracted the proteins that are in the PPI network but do not occur in any miRNA target databases we have used. This approach is reasonable because miRNA target prediction programs scan all known 3' UTR sequences against all known miRNAs and we aim to reduce the overall false negative rate by using the union of major miRNA target databases. Also, we believe that the probability of these proteins being a miRNA target, within the rules applied by current miRNA target prediction tools, is much less compared to a randomly selected protein in the PPI network; because, a majority (10,198/13,504) of the proteins in the PPI network is included in at least one of the miRNA target databases we accessed. With this approach we bypass the necessity of artificially generating a negative training dataset as in other studies such as in the works of Kim et al. [40], and Showe et al. [93]. At the end of this process, putative negative miRNA target dataset became a collection of 1,896 proteins.

We analyzed the putative negative dataset in depth, to see whether there is a specific biological reason that these genes are not listed in any of the major miRNA target databases. We first performed a Gene Ontology (GO) term enrichment analysis of the negative dataset. We used the GOrilla [25] tool which gives significantly represented GO terms in a ranked list of proteins. We used all the three main GO hierarchies in this analysis: molecular function, biological process, and cellular compartment. We set the p-value threshold to 10^{-9}. In that case, GOrilla did not find any significantly represented GO term in the 1,896 proteins in our negative dataset. This shows that a gene that does not appear in any miRNA target database cannot be identified by a specific molecular function, biological process, or cellular compartment.

As a second analysis, we investigated whether the 1,896 proteins in our putative negative dataset have distinctive features in their 3' UTR regions. Since the miRNA and its target usually have a near perfect match in the 3' end of the target, the 3' UTR region is important in miRNA target identification. From the UCSC Genome Bioinformatics Site [99], we downloaded 3' UTR sequences of mRNAs encoding the proteins in our dataset. We downloaded 3' UTR sequences for both our putative negative dataset and for the 10,198 proteins that are identified as targets by at least one miRNA target database. We compared the total length and base composition of 3' UTR sequences in the negative dataset to the length and base composition of the rest of the proteins by histogram analysis. Figure 4.1 shows the distribution of the lengths of the 3' UTR regions. The negative dataset exhibits a similar 3' UTR length distribution compared to the proteins that are identified as targets in at least one of the major miRNA target databases. The difference between the two histograms is statistically not significant (p-value = 0.7805). Our conclusions in base composition show that the genes in our putative negative dataset cannot be easily identified by their 3' UTR sequences.

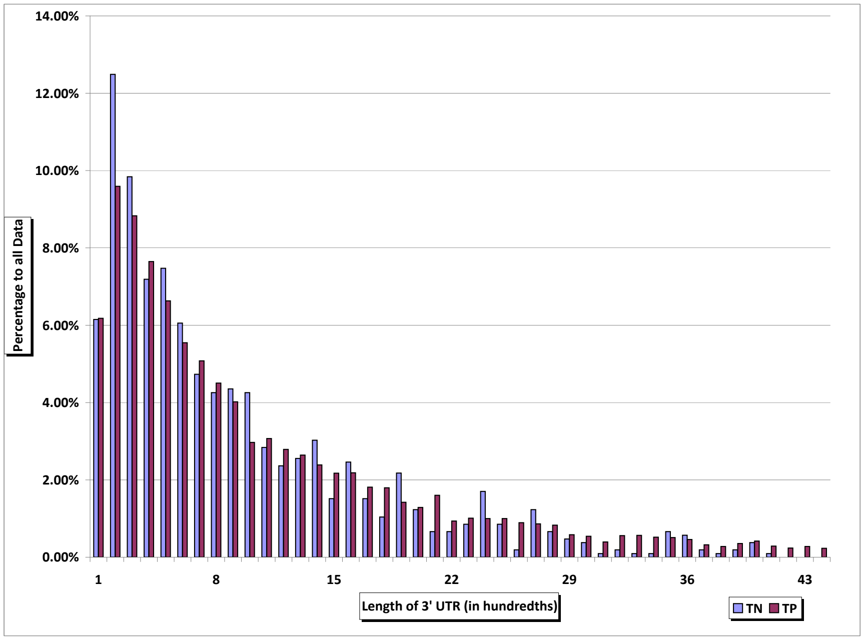


Figure 4.1: Histogram for comparing the 3'UTR lengths of miRNA targets found in a miRNA target database and putative true negative proteins. The comparison of the lengths of 3'UTR parts of mRNAs that produce proteins in putative negative dataset and mRNAs that produce the proteins that are identified as targets by at least one miRNA target database. Here, as an example, the number of mRNAs having a 3'UTR length between 100 and 199 are represented as within the same bin since the bin size of the histogram is 100. The calculated count for each length size 100 is divided by the total number of mRNAs for that dataset and represented as a percentage for an easy comparison.

4.2 Experimental Evaluation of Topological Properties

In this section, we investigate various ways of integrating network context of a protein as a filter to a miRNA target prediction algorithm. We apply our filter on TargetScan [46], which is one of the most widely used miRNA target prediction methods [19]. First, before employing a rigorous information integration approach, we assess the performance of individual topological measures as simple threshold filters. Then, we integrate these measures using two separate supervised learning approaches and analyze their performance when applied as an additional filter on TargetScan.

4.2.1 Performance of Individual Measures as Simple Threshold Filters

We use seven graph theoretic measures as indicators of network context of a protein in a PPI network: degree, betweenness centrality, closeness centrality, clustering coefficient, eigenvector centrality, graph centrality, and stress centrality (Table 1). For a given network context measure, we define a simple threshold filter which classifies a given query protein p as a *miRNA target* or a *non-target* based on a simple comparison of the calculated measure for p on the PPI network against a threshold value, τ, and filters out non-target proteins.

In order to measure the performance of a simple threshold filter, we applied precision Receiver Operating Characteristic (pROC) curve analysis. A pROC curve shows the classification performance of a classifier on a ground truth dataset as a two-dimensional plot of precision versus recall, or *sensitivity.* The main reason for using a pROC curve instead of a regular ROC curve is that a miRNA target prediction tool tries to identify *targets* and the data to validate such a tool is a validated *target* dataset. Currently, there is no validated *non-target* dataset that can be used for validation purposes. Therefore, specificity, i.e., true negative rate, cannot be computed. Instead, it is more reasonable to compute precision for this specific problem as in a study of Alexiou et al. [2]. We apply pROC analysis on TargetScan predictions using two different sets of experimentally validated miRNA targets, TarBase [75] and pSILAC [74] datasets, as true positives.

For each topological measure, we sort all the values for that measure in descending order and assign each value as a threshold τ. With this approach, we vary threshold values used in the simple threshold filters and plot the pROC curves. Sensitivity is computed as the ratio of the number of true positives that pass the filter to the total number of experimentally verified targets. Precision is the ratio of the number of true positives that pass the filter to the total number of predictions that pass the filter. The same procedure is applied to the context score provided by TargetScan. Henceforth, eight different curves are produced for comparison (Figure 4.2).

Figure 4.2 shows the performance of threshold filters for let-7 in the pSILAC dataset. The sensitivity values are low, since the number of TargetScan predictions are much more than the validated number of miRNA targets. In general, we see that eigenvector and closeness centrality measures perform better than other measures, while graph centrality and clustering coefficient perform worse. However, except miR-1, in Figure 4.3 and miR-16, in Figure 4.4, all measures perform worse than the total context score (TCS) provided by TargetScan. Here, the sensitivity values are also low, for the same reasons.

With the help of pROC analysis, one can determine the best threshold as the threshold

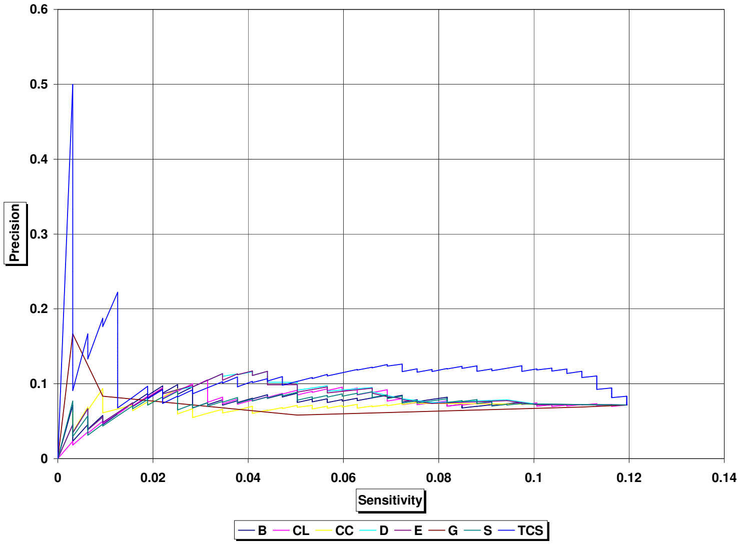


Figure 4.2: pROC curve of each network measure for let-7 using the pSILAC dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold.

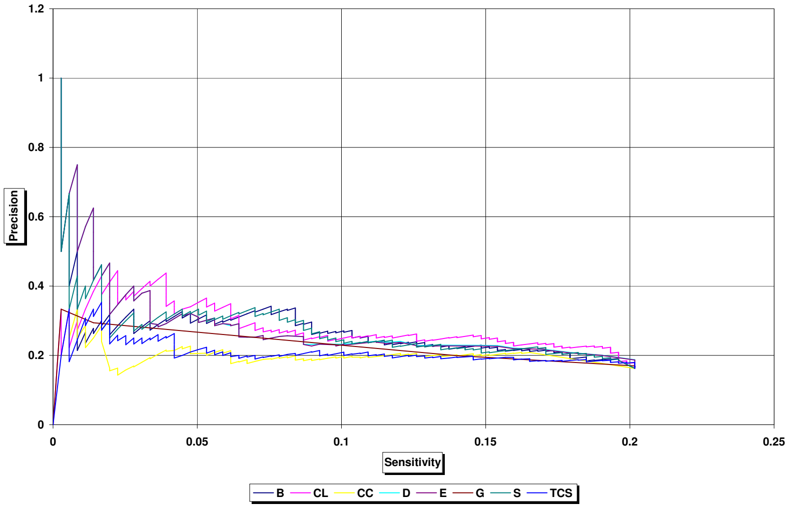


Figure 4.3: pROC curve of each network measure for miR-1 using the pSILAC dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold.

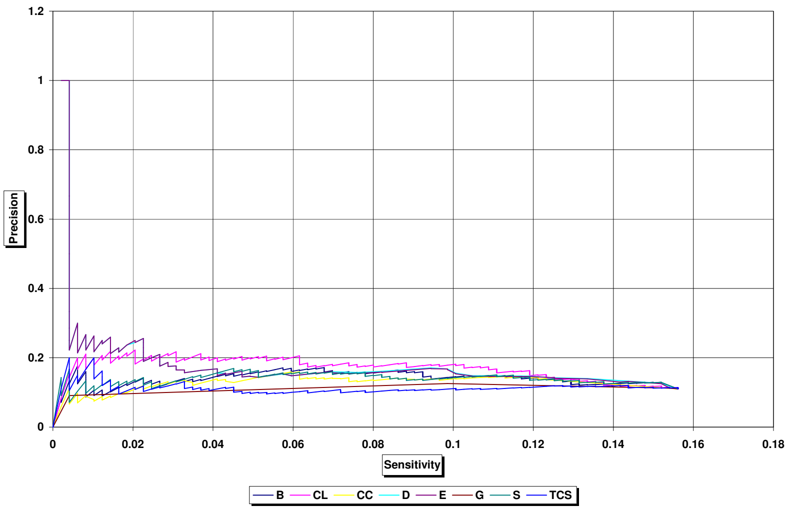


Figure 4.4: pROC curve of each network measure for miR-16 using the pSILAC dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold.

which maximizes the F-measure. Table 4.1 shows the average precision and sensitivity of each individual filter on target lists of each of the five miRNAs when the best threshold is found using cross validation, i.e., by excluding the pROC curve of the miRNA for which the accuracy is measured. The best threshold for TargetScan total context score is also derived by the same cross validation process. We see that, for sensitivity, only graph centrality narrowly outperformed the total context score based filter. However, none of the other individual

Table 4.1: Average sensitivity and precision of network properties on the pSILAC dataset. The average sensitivity and precision of target prediction for five pSILAC miRNAs for each seven graph theoretic measures that are used as indicators of a protein's context in a PPI network.

Network Property	**Average Sensitivity**	**Average Precision**
Betweenness	0.122	0.129
Closeness	0.119	0.130
Clustering Coefficient	0.122	0.123
Degree	0.127	0.128
Eigenvector	0.122	0.132
Graph	0.143	0.112
Stress	0.122	0.129
TCS	0.141	0.113

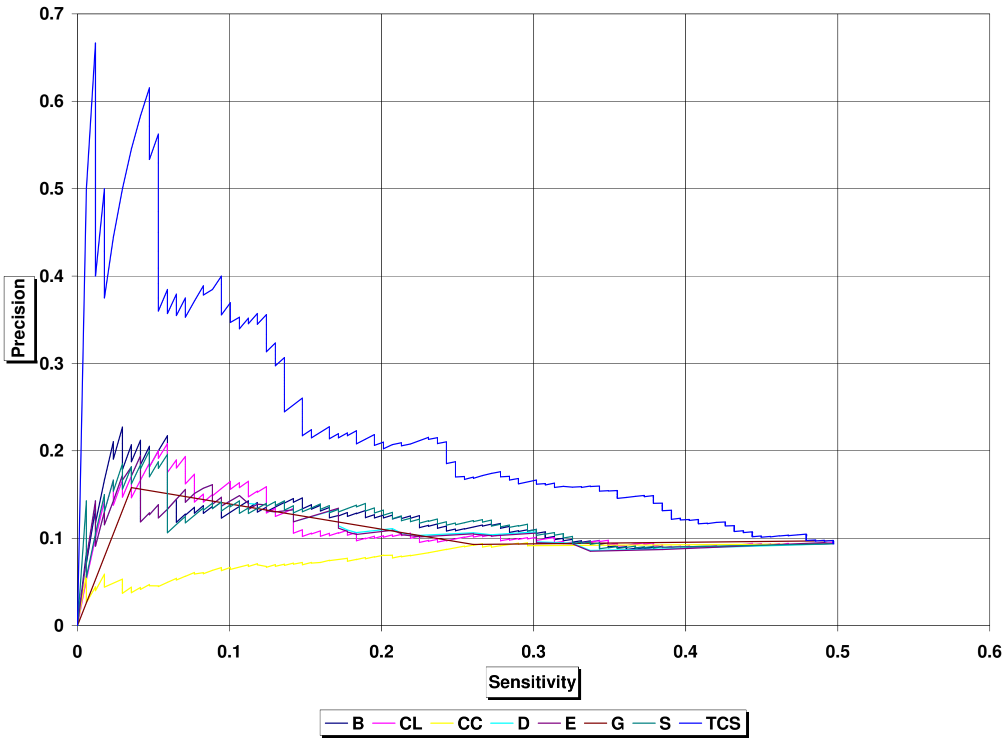


Figure 4.5: pROC curve of each network measure for miR-124 using the TarBase dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold.

Table 4.2: Average sensitivity and precision of network properties on the TarBase dataset. The average sensitivity and precision of target prediction for eight TarBase miRNAs for each seven graph theoretic measures that are used as indicators of a protein's context in a PPI network.

Network Property	**Average Sensitivity**	**Average Precision**
Betweenness	0.255	0.069
Closeness	0.261	0.065
Clustering Coefficient	0.307	0.059
Degree	0.249	0.073
Eigenvector	0.249	0.072
Graph	0.340	0.060
Stress	0.254	0.071
TCS	0.230	0.087

measures is able to provide better sensitivity.

Figure 4.5 and Table 4.2 show the results of a similar analysis on the TarBase dataset. There are eight miRNAs in TarBase for which TargetScan is able to identify more than 10 verified

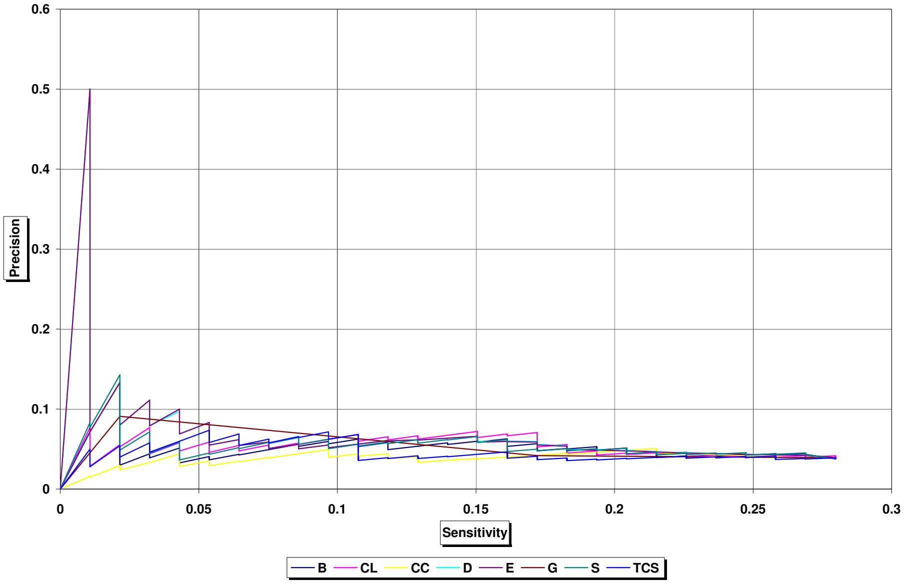


Figure 4.6: pROC curve of each network measure for miR-16 using the TarBase dataset. Here, the network properties and the total context score (TCS) of TargetScan are used independently as simple threshold filters. Every possible value of a given measure is considered as a threshold. Each curve is constructed by calculating the sensitivity and precision for that threshold.

target genes. Figure 4.5 shows the pROC curve for miR-124. We observe similar results on TarBase dataset as eigenvector and closeness centrality measures perform better than other measures. Especially clustering coefficient again has the worst performance.

The difference between the total context score (TCS) provided by TargetScan and other measures is much more apparent, except again for miR-16, which can be seen in Figure 4.6. The number of TargetScan predictions outnumbers the validated number of miRNA targets, making the sensitivity values low. In Table 4.2, for the TarBase dataset, it is clear that using TCS as a filter provides a better precision while individual measures provide better sensitivity. Although individual topological measures do not perform well as threshold filters, we investigate whether a combination of network context indicators can be used as a filter in the next subsections.

4.2.2 Integration of Network Context Measures using Support Vector Machines

Support Vector Machines are supervised discriminative models which optimize a separating hyperplane between two classes of high dimensional data. We used SVMs to learn discriminative models between combined network context features of miRNA targets and non-miRNA

Table 4.3: Fold change of precision in TargetScan using SVM. The ratio of precision of TargetScan predictions filtered by SVM to the precision of original predictions of TargetScan is presented. The threshold level of the SVM is set to 0.5. 0.000 means that SVM filters out all the verified targets.

	miRNA	**Fold Change**
TarBase	let-7	1.815
	miR-1	0.347
	miR-155	0.000
	miR-16	0.914
	miR-30	1.439
	miR-124	1.433
	miR-29	3.874
	miR-373	0.767
pSILAC	let-7	1.100
	miR-1	0.000
	miR-155	0.000
	miR-16	1.361
	miR-30	1.258

targets. SVMs are originally developed to make binary boolean classifications; however, several extensions allow for multi-class and probabilistic classifications. We make use of the probability estimates provided by such an extension of the SVM library we used in this study [18].

We represent each gene as a seven dimensional vector based on the topological measures, i.e., betweenness, closeness centralities, clustering coefficient, degree, eigenvector, graph and stress centralities computed for that gene. We trained an SVM model for each miRNA by using verified targets of *other* miRNAs[1] in TarBase and pSILAC targets separately and the putative non-target dataset using five-fold cross validation. Several kernel functions such as radial basis function (RBF), linear, polynomial, or sigmoid kernels can be used when learning an SVM model. However, in our initial experiments (results not shown), the RBF kernel produced the best results among available standard kernel functions; hence, we used the RBF kernel in the results reported in this article. We compared the probability estimates of SVM predictions against a threshold to label a protein as a miRNA target or a non-target. We first set the probability threshold to 0.5, which provides the same binary classification as the original SVM. This threshold can be increased or decreased in order to obtain a more stringent or lenient classifier, respectively.

Table 4.3 shows the results of the fold change of precision in TargetScan predictions when this SVM classifier is used as a filter for the targets of five miRNAs in the pSILAC dataset and eight miRNAs in the TarBase dataset. We see that the precision can be increased for some of the miRNAs if SVM predictions are used as filters. The average fold change of precision is 0.744 for pSILAC and 1.324 for TarBase datasets. This is an indication of the difference between using a quality training dataset over a quantity dataset. It seems that, at least for the SVM, training with quality dataset produces better results than training with quantity validated miRNA targets. Next, we used different probability thresholds and see how this effects the efficiency of the filter.

---

[1] in order to ensure separation of training and test data

Table 4.4 shows the results of filtering at different stringency levels. We chose to analyze decreasing stringency to see the effect of filtering less number of miRNA targets for higher sensitivity. The precision fold change decreases especially after the probability threshold level of 0.3. For the same miRNA, the fold changes in TarBase are generally better than the pSILAC. The exception is miR-16. The other miRNAs take advantage of training the SVM with quality validated targets. Using TarBase is much more beneficial in SVM than using pSILAC for training.

Table 4.4: Effect of different stringency levels in TargetScan using SVM. The ratio of precision of TargetScan predictions filtered by SVM to the precision of original predictions of TargetScan is presented. Five different threshold levels, which are 0.5, 0.4, 0.3, 0.2, and 0.1 of the SVM are compared. 0.000 means that SVM filters out all the verified targets.

	miRNA	**Fold Chng. @0.5**	**Fold Chng. @0.4**	**Fold Chng. @0.3**	**Fold Chng. @0.2**	**Fold Chng. @0.1**
TarBase	let-7	1.815	1.644	1.863	1.505	1.063
	miR-1	0.347	0.273	0.636	0.711	0.873
	miR-155	0.000	0.000	0.721	0.721	0.631
	miR-16	0.914	1.052	1.256	1.432	1.018
	miR-30	1.439	1.849	0.984	0.980	1.039
	miR-124	1.433	1.283	1.258	1.283	1.019
	miR-29	3.874	3.818	3.182	2.422	1.083
	miR-373	0.767	0.739	0.703	0.580	1.036
pSILAC	let-7	1.110	1.075	1.126	1.069	1.000
	miR-1	0.000	0.000	0.961	1.609	0.999
	miR-155	0.000	0.000	0.000	0.000	0.536
	miR-16	1.361	1.350	1.318	1.269	0.991
	miR-30	1.258	1.222	1.138	1.008	1.007

As another analysis, we compare TargetScan total context score (TCS) with SVM filter using pROC curve analysis. The results can be seen in Figures 4.7-4.8. Each point on the curve corresponds to a different TCS and SVM probability threshold. The sensitivity and precision point for the TCS threshold which keeps the top third of the targets is marked as “tcs”. For SVM, points for different probability values are marked as “p50”, “p40”, “p30”, “p20”, and “p10”.

In the case of miR-29, we see that SVM has a comparable performance to TCS; however, for other miRNAs, TCS is a better indicator of being a miRNA target, overall for both pSILAC and TarBase datasets. Next, we use an alternative statistical filter and analyze its performance.

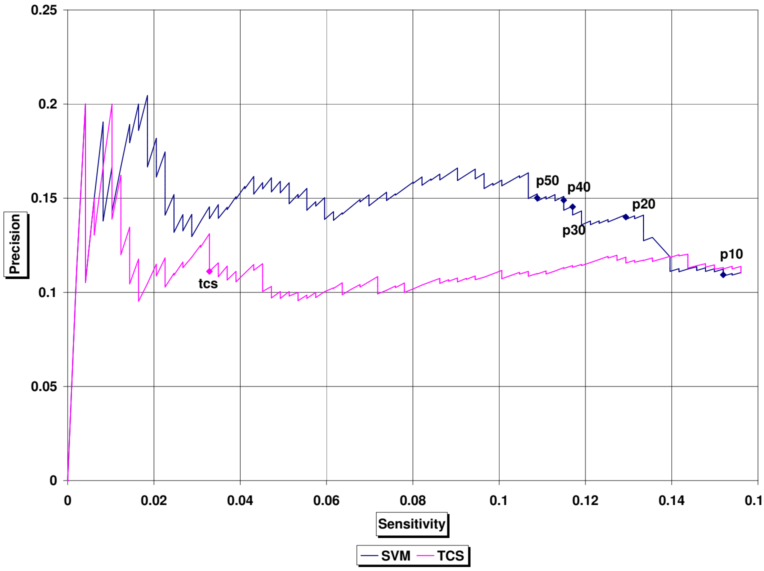


Figure 4.7: pROC curve comparison of TargetScan total context score and SVM using the pSILAC dataset for miR-16. Results for other miRNAs are given in Supplement 3. Points on the curves correspond to different TargetScan total context scores and SVM probabilities. The point on the TCS curve represented as "tcs" shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the SVM curve show the sensitivity and precision at fixed stringency levels of 0.5, 0.4, 0.3, 0.2, and 0.1.

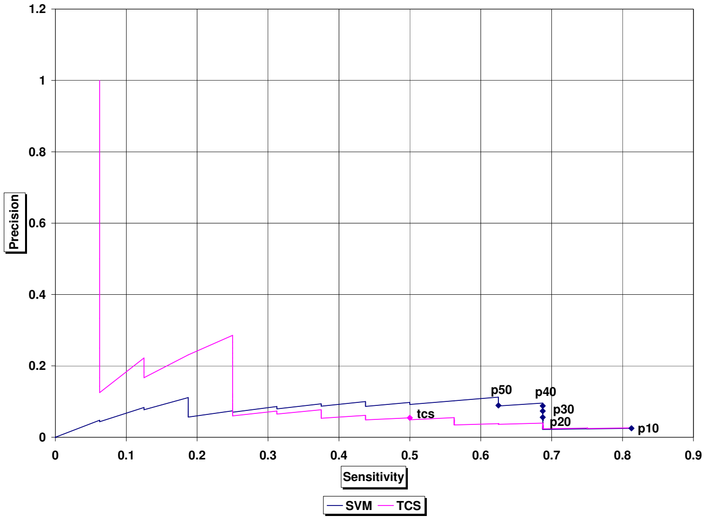


Figure 4.8: pROC curve comparison of TargetScan total context score and SVM using the TarBase dataset for miR-29. Results for other miRNAs are given in Supplement 4. Points on the curves correspond to different TargetScan total context scores and SVM probabilities. The point on the TCS curve represented as "tcs" shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the SVM curve show the sensitivity and precision at fixed stringency levels of 0.5, 0.4, 0.3, 0.2, and 0.1.

4.2.3 Integration of Network Context Measures using Mahalanobis Distance

Mahalanobis Distance is used in computing the distance of a multi-variate sample to a multi-variate distribution. This distance measure can be used as a classifier by comparing the distance of a sample to multiple multi-variate distributions. We employed this statistical metric as a filter to the predictions of TargetScan. Here, the two multi-variate distributions correspond to the network context measures of the true miRNA target genes and putative non-miRNA target genes and are the same datasets used in training the SVM models. We again used five-fold cross validation. We computed the MD of each TargetScan target to both of the distributions. The decision process involves a simple comparison of whether the distance to the positive training data is smaller than the distance to the negative training data.

The MD filter outperformed the SVM filter and attained better precision for all of the miRNAs in the pSILAC dataset and for six out of eight miRNAs in the TarBase dataset (Table 4.5). The average fold change is 1.282 for pSILAC and 1.147 for TarBase. The better performance on the pSILAC dataset suggests that a more complete experimentally validated database promises more effective filtering of false positive predictions.

Similar to SVM predictions, the stringency level of MD filter can be varied by using a distance margin when labeling a non-target gene. Table 4.6 shows the results of MD filters at increasing distance margins, i.e., decreasing stringency levels. Each time the distance margin increases by adding 0.2, the set of miRNA targets generated by the previous margin becomes the proper subset of the recently constructed set of miRNA targets, increasing sensitivity. The fixed amount of 0.2 represents a 10% addition to distance difference, because for all miRNA targets, more than 70% of them have an MD difference between -1 and +1. Table 4.6 shows that the precision fold change is insensitive to the changes in the distance margin.

We performed a similar comparative analysis with TargetScan total context scores. Figures 4.9 and 4.10 show the pROC curves of MD and TCS filters on miRNAs in the pSILAC and TarBase

Table 4.5: Fold change of precision in TargetScan using MD. The ratio of precision of TargetScan predictions filtered by MD to the precision of original predictions of TargetScan is presented. The threshold level of the MD is set to 0.

	miRNA	**Fold Change**
TarBase	let-7	1.487
	miR-1	1.091
	miR-155	1.299
	miR-16	1.244
	miR-30	0.941
	miR-124	1.172
	miR-29	1.282
	miR-373	0.658
pSILAC	let-7	1.124
	miR-1	1.578
	miR-155	1.088
	miR-16	1.335
	miR-30	1.283

Table 4.6: Effect of different stringency levels in TargetScan using MD. The ratio of precision of TargetScan predictions filtered by MD to the precision of original predictions of TargetScan are given. Six different threshold levels, which are +0.0, +0.2, +0.4, +0.6, +0.8 and +1.0 of the MD are compared.

	miRNA	**Fold Chng. +0.0**	**Fold Chng. +0.2**	**Fold Chng. +0.4**	**Fold Chng. +0.6**	**Fold Chng. +0.8**	**Fold Chng. +1.0**
TarBase	let-7	1.487	1.184	1.182	1.086	0.910	0.894
	miR-1	1.091	1.243	1.195	1.278	1.224	1.094
	miR-155	1.299	1.045	1.224	1.172	1.289	1.165
	miR-16	1.244	1.107	1.226	1.163	1.053	1.077
	miR-30	0.941	1.078	1.071	1.101	1.076	1.029
	miR-124	1.172	1.067	1.036	1.051	1.085	1.070
	miR-29	1.282	1.625	1.713	1.545	1.146	1.093
	miR-373	0.658	0.531	0.486	0.589	0.862	0.929
pSILAC	let-7	1.124	0.998	0.921	0.883	0.934	0.973
	miR-1	1.578	1.465	1.514	1.435	1.213	1.134
	miR-155	1.088	1.065	1.096	1.290	1.309	1.215
	miR-16	1.335	1.266	1.295	1.198	1.004	1.020
	miR-30	1.283	1.229	1.199	1.161	1.082	1.056

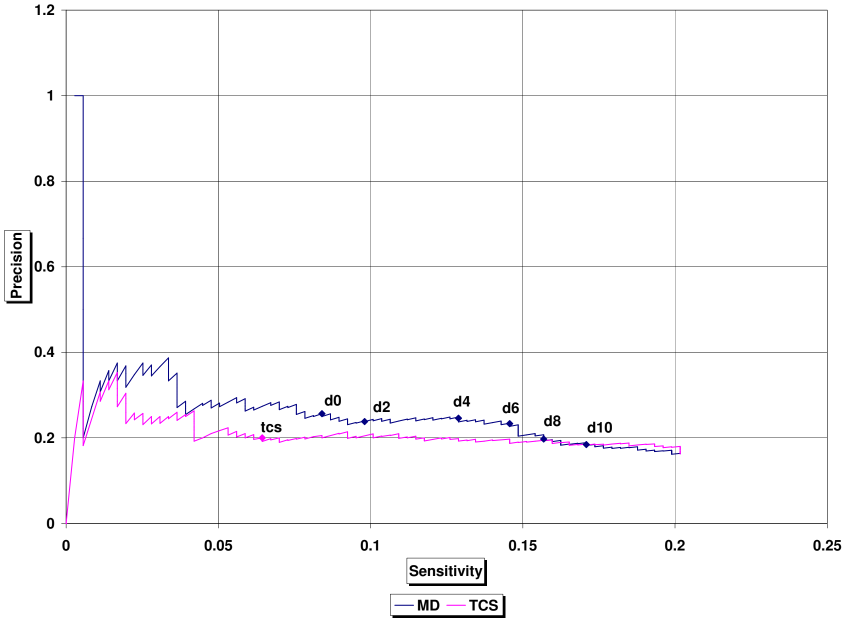


Figure 4.9: pROC curve comparison of TargetScan total context score and MD using the pSILAC dataset for miR-1. Points on the curves correspond to different TargetScan total context scores and MD distance margins. The point on the TCS curve represented as "tcs" shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the MD curve show the sensitivity and precision at fixed stringent levels of +0.0, +0.2, +0.4, +0.6, +0.8 and +1.0.

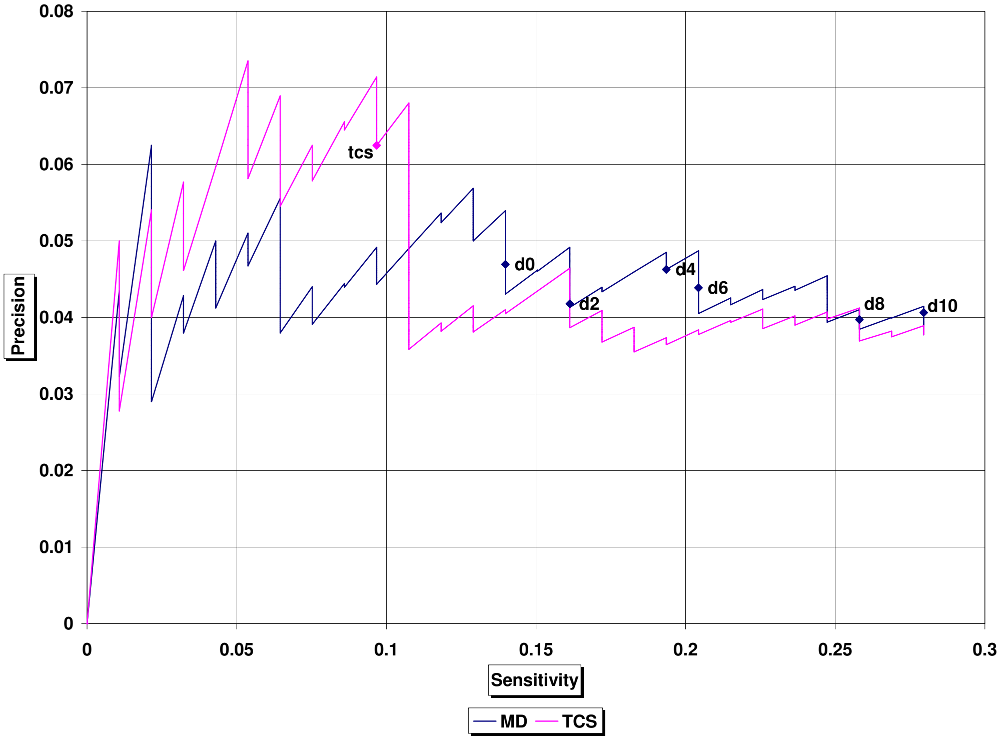


Figure 4.10: pROC curve comparison of TargetScan total context score and MD using the TarBase dataset for miR-16. Points on the curves correspond to different TargetScan total context scores and MD distance margins. The point on the TCS curve represented as "tcs" shows the sensitivity and precision where top one third of the TargetScan predictions are retained. The points on the MD curve show the sensitivity and precision at fixed stringent levels of +0.0, +0.2, +0.4, +0.6, +0.8 and +1.0.

datasets for miR-1 and miR-16, respectively. Again, each point on the curve corresponds to a different TCS threshold and MD distance margin. The sensitivity and precision point for the TCS threshold which keeps the top third of the targets is marked as "tcs". For MD, the same points for different distance margins are marked as "d0", "d2", "d4", "d6", "d8", and "d10". Figures 4.9 and 4.10 show that MD outperforms TCS. Similar pROC curves are observed for all of the miRNAs in the pSILAC data. The sensitivity values are low, since the number of TargetScan predictions are much more than the validated number of miRNA targets. The Mahalanobis Distance based filter produced the best results and increased the precision of target predictions for all of the miRNAs in the pSILAC dataset and six out of eight miRNAs in the TarBase dataset. This strong result shows that the network context of a protein in a PPI network may be as important as sequence related features of the target site.

CHAPTER 5

DIVO: A NOVEL DISTANCE BASED VOTING METHOD FOR ONE CLASS CLASSIFICATION

The classification problems in bioinformatics have their own characteristics. Some of them have very small datasets, some have very large but inaccurate. For miRNA target prediction, there exists only validated miRNA target data since validation of a gene product to be a non-target is impossible. Since deciding whether a gene product is a miRNA target or not is a classification problem, we either form a putative non-miRNA target data set or use only validated miRNA targets as training. In our study, we followed both of them. For the latter, the problem become One-Class Classification. This is a rather established field of research. One-Class Classification algorithms try to find similarities among training samples and generalize these similarities while avoiding over-fitting. In this study, we devised a more easy to understand and intuitive approach that exploit the distances between data that belong to the class. The DiVo algorithm we propose assumes that training samples cluster in a closed and connected boundary in the feature space.

The class boundary is established by the following rule:

- Boundary Rule: The distance from a class member q to a training sample t, is less than or equal to the farthest distance from t to any of the other training samples.

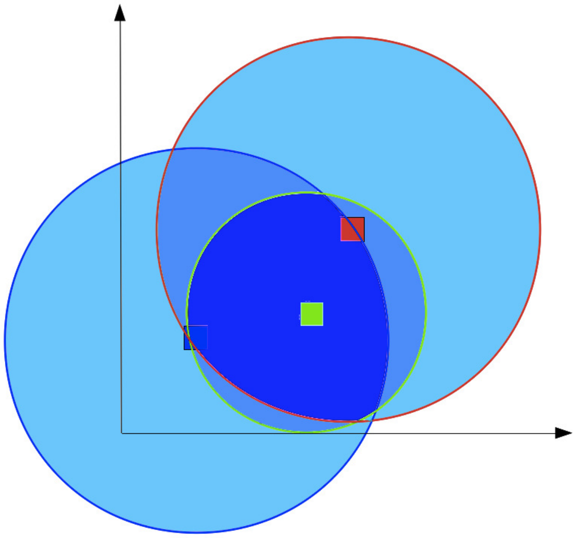


Figure 5.1: The regions that may contain the class members when the Boundary Rule is applied at different ratios.

For a test instance, x, to be accepted as a class member, at least a certain percentage of the training samples should approve that x complies the Boundary Rule. This percentage, which is named *ratio* for the rest of the paper, is the only parameter of our method. We demonstrate the intuition behind this heuristic with a visual example. Suppose that we have three two-dimensional training samples as given in Figure 5.1.

In Figure 5.1, the darkest regions show the space of class members where all three training samples agree upon, i.e., $ratio > 2/3$. The lighter zones are where two training samples agree, i.e., $1/3 < ratio \leq 2/3$. The lightest parts are where only one training sample votes positively for the test instance, i.e., $ratio \leq 1/3$.

Applying the Boundary Rule require distance calculations. In this study, we use two different distance metrics. These are the Euclidean (DiVo-E) and the Mahalanobis (DiVo-M) distances. The major difference of the Mahalanobis distance is that it computes the distance of a single sample to the rest of the training dataset as a single scalar value. Therefore, instead of the maximum distance, we use this distance in the Boundary Rule. For the Mahalanobis distance, we used the following equation:

$$D_M(p, \mu_{set}) = \sqrt{(p - \mu_{set})^T S^{-1} (p - \mu_{set})} \tag{5.1}$$

where p is a multidimensional sample point, μ_{set} is a vector representing the means of each variable in the *set*, and S is the covariance matrix. S captures the correlations between different dimensions and the variance within a single dimension. When calculating the Mahalanobis distance, for certain data distributions, calculating the inverse of a covariance matrix may not be possible. This occurs when an attribute of all data samples has the same value. In these cases, we simply add a regularization matrix to the covariance matrix [85]. Regularization matrix is a square matrix where only the diagonal elements are non-zero and all these values are equal to a fixed value, which is called the *regularization parameter*. In our experiments, we used 0.01 as the *regularization parameter*.

In the DiVo algorithm, we used the following step function in Equation 5.2 as the Boundary Rule.

$$\varphi_b(x) = \begin{cases} 1 & \text{if } x \leq b \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

Algorithm 1 Training Phase

```
Input 1: A set T of $k$ positive samples,
$T = \{t_1, t_2, \ldots, t_k\} \subset \Re^n$
Input 2: Distance metric, $D$
Output: A set B of k boundary distances,
$B = \{b_1, b_2, \ldots, b_k\} \subset \Re$
for $\forall t_i \in T$, do
  if $D = Mahalanobis$ then
    $b_i \leftarrow D(t_i, T \setminus \{t_i\})$
  else
    $b_i \leftarrow max_{j\in[1..k], j\neq i} D(t_i, t_j)$
  end if
end for
```

Algorithm 2 Testing Phase

```
Input 1: A set R of k positive samples with corresponding boundary distances,
$R = \{(t_1, b_1), \ldots, (t_k, b_k)\} \subset \Re^n \times \Re$
Input 2: Distance metric, $D$
Input 3: Input sample,
$x \in \Re^n$ whose label is to be predicted
Input 4: The vote ratio threshold,
$ratio \in [0..1]$
Output: The label $y \in \{0, 1\}$ of sample $x$,
i.e., 0 for negative and 1 for positive
if $D = Mahalanobis$ then
  $T = \emptyset$
  for $\forall (t_i, b_i) \in R$, do
    $T \leftarrow T \cup \{t_i\}$
  end for
  $y \leftarrow$
  $(\sum_{j\in[1..k]} \varphi_{b_j}(D(x, T \setminus \{t_j\})) >= (k * ratio))$
else
  $y \leftarrow (\sum_{j\in[1..k]} \varphi_{b_j}(D(x, t_j)) >= (k * ratio))$
end if
```

For the DiVo algorithm, the basic principal is that, every training sample used in the testing phase has a single and equal vote. Using the Boundary Rule, the total number of positive votes is compared to a *ratio* of the training samples. Hence, the only parameter for the DiVo algorithm is the *ratio*.

The time complexity of the training phase of our algorithm depends on the number of the training samples, k, and the number of the attributes, n, of the training samples. The distance between each pair of training samples is computed once; hence, the total running time is $O(k^2)O(t)$, where t is the time it takes to compute the distance between a pair of samples. For the Euclidean distance, the complexity is linear in the number of attributes; therefore, the total complexity is $O(k^2n)$. For the Mahalanobis distance, it is $O(k^2n^2)$.

It is important to note that the training phase of DiVo does not have an optimization of an error function as in other boundary OCC algorithms described in the work of Mazhelis [53]. Hence, we do not require complex optimization methods in the training phase. The training phase is simply the computation of all-to-all distances of the training samples.

5.1 Data Preprocessing

Since the methods we compare are sensitive to scaling, we normalized all attribute values between 0 and 1. For 3-fold cross-validation, we divided each class of each dataset into three randomly and evenly distributed subsets. In every iteration, one of these subsets become the training samples while the rest of the dataset becomes the test samples.

Some of the datasets we use have missing values. To overcome this obstacle, we get all the instances which share the same class with the instance having the missing value. We calculated the median of that attribute for these instances and assign the calculated value to the missing attribute [1].

Before evaluating all the algorithms, we need to set their parameters with the best possible values. This process requires iterating the algorithm over a predefined set of parameter values using a dedicated dataset. Then, we can use the selected parameters and evaluate the results on another dataset. The two datasets should be mutually exclusive. Otherwise, the performance measurement will be unsound. Here, we will give the details of the datasets we used for the parameter selection and performance evaluation.

5.1.1 Datasets for Parameter Selection

The classification algorithms compared in this study have a number of parameters that affect classification accuracy. Therefore a mechanism for setting them to the best available values is crucial. For this, we used three different datasets. These are Iris, Wine and Ecoli datasets from UCI [27]. A summary of these datasets is given in Table 5.1. Optimum values of the *ratio* parameter of DiVo and the compared algorithms are discovered by 3-fold cross validation on these datasets.

Table 5.1: The datasets for parameter selection and their number of classes, attributes, and instances.

Dataset Name	Class Count	Attribute Count	Instance Count
Iris	3	4	150
Wine	3	13	178
Ecoli	2	7	336

5.1.2 Datasets for Performance Comparison

After the optimal parameters are determined, we compare the algorithms on five independent datasets. These are Biomed from StatLib [16], Breast Cancer [80], Dermatology, Diabetes and Ionosphere from UCI [27]. A summary of these datasets are given in Table 5.2. We used a group of UCI datasets for both parameter selection and performance comparison since they are appropriate for comparing one class classification algorithms, as can be seen in a study of Tax and Duin [86]. Originally, all of these datasets are multi-class datasets. We simulate the one class classification problem by selecting each class as the target class and the rest of them as the non-targets and using a subset of the target class samples during the training phase.

Table 5.2: The datasets for performance comparison and their number of classes, attributes, and instances.

Dataset Name	Class Count	Attribute Count	Instance Count
Biomed	2	4	209
Breast Cancer	2	9	463
Dermatology	3	34	366
Diabetes	2	8	768
Ionosphere	2	34	350

5.2 Experimental Evaluation of DiVo Performance

In this section, we present the performance of DiVo compared to different state of the art one class classification algorithms on five different datasets. The results are summarized in Table 5.3. We computed the average and standard deviation of f-measures on five independent datasets.

For the Biomed dataset, the DiVo-M variant performs on par with OCC SVM, which returns the best classifications. However, the standard deviation of DiVo-E is the smallest one. This shows that the results provided by DiVo-E for Biomed dataset are much more precise than the others. For instance, Local Gaussian has a higher accuracy than the DiVo-E, but the standard deviation across its classifications is larger.

On the Breast Cancer dataset, the DiVo-E variant outperforms the rest of the methods, except

Table 5.3: The average and standard deviation of f-measures on different datasets.

Algorithm Name	Biomed		Breast Cancer		Dermatology		Diabetes		Ionosphere	
	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
DiVo-E	0.508	0.126	0.776	0.064	0.759	0.254	0.409	0.162	0.433	0.236
DiVo-M	0.530	0.272	0.685	0.183	**0.773**	0.182	**0.421**	0.192	**0.538**	0.340
GG	0.472	0.228	0.634	0.243	0.669	0.121	0.414	0.169	0.503	0.411
LG	0.510	0.253	0.571	0.259	0.210	0.111	0.303	0.065	0.446	0.431
κNNDD	0.447	0.188	0.448	0.131	0.625	0.116	0.372	0.149	0.378	0.296
κNNDD-SRM	0.423	0.241	0.759	0.156	0.177	0.062	0.415	0.181	0.400	0.144
OCC SVM	**0.538**	0.137	**0.791**	0.057	0.529	0.113	0.415	0.182	0.450	0.398

OCC SVM. But again, the small standard deviation in DiVo-E shows that this algorithm performs more or less the same for different classes of the Breast Cancer data.

With the Dermatology dataset, we see that both DiVo-M and DiVo-E achieve the best results. They are the only ones to get better than 0.750 f-measure values. The third best, which is Global Gaussian, can only get an f-measure value of 0.669. But this time, the standard deviations of DiVo are higher compared to the other algorithms.

The DiVo-M variant produces the best f-measure average for the Diabetes dataset. However, the differences here are not as significant as in the Dermatology results. All the methods, except Local Gaussian and k-Nearest Neighbor Data Description, achieve more or less the same f-measure values.

On the Ionosphere dataset, again DiVo-M produces the best results. The overall results are an indication that DiVo algorithm can perform on par with all the other well established methods in the field. On some occasions, DiVo-M produces the best f-measure averages.

5.2.1 Parameter and Data Analysis for DiVo

For each dataset, we also investigated the effect of the *ratio* parameter on the performance of DiVo-E and DiVo-M. In Figures 5.2 and 5.3 there exists two patterns for each distance metric. For the Euclidean distance, datasets other than the Breast Cancer and Dermatology, the algorithms perform more or less the same. They seem to be independent of the *ratio* threshold. But for the Breast Cancer and Dermatology, they perform better for larger *ratio* threshold values. In fact, these two datasets are the ones that DiVo-E returns better f-measures.

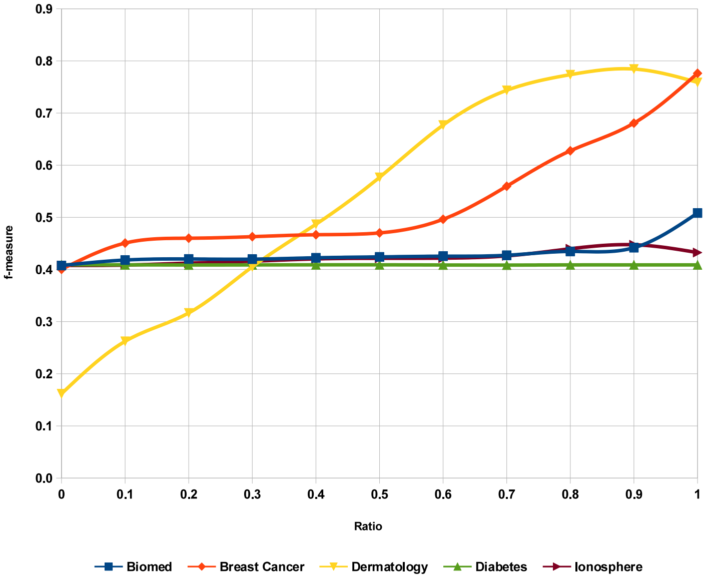


Figure 5.2: The effect of ratio to the outcome of DiVo-E algorithm.

For the Mahalanobis distance, the *ratio* affects the outcome in a different way than using the Euclidean distance. The changes in the *ratio*, making a peek at around 0.1 and 0.2. For the increasing values of *ratio*, the effectiveness of DiVo-M decreases. Here, all five datasets show this trend.

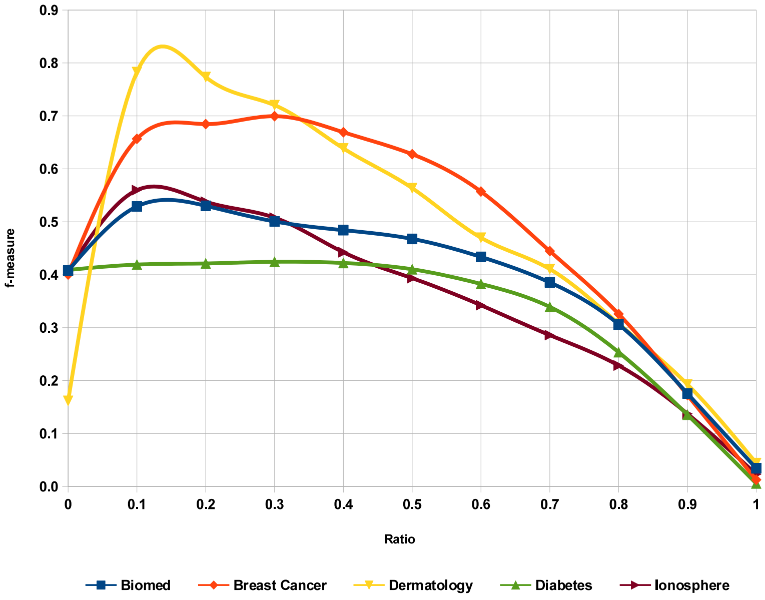


Figure 5.3: The effect of ratio to the outcome of DiVo-M algorithm.

We conducted two different data analyses to answer what makes a dataset suitable for the DiVo algorithm. The first one is about the statistical properties of Euclidean distance distributions.

For each dataset, we analyzed the distance distributions using histograms. Our observation is that, when the distribution is left or right skewed, the success of DiVo-E decreases. In Figures 5.4 and 5.5, we see two different distributions. DiVo-E performs better than the other non-DiVo algorithms when applied on the Dermatology data. The average of the minimum and maximum distances is close to the average of all distances. However, the performance of DiVo-E is not that successful when run on the Biomed dataset. The average of the minimum and maximum distances is quite different than the average of all distances. The same is also true for the Diabetes data, although it is not shown.

We performed another analysis to see if the dimensionality reduction by PCA and kernel PCA (kPCA) can bring some hidden patterns out. We analyzed only the first two principal components. We chose Gaussian kernel for kernel PCA with sigma is set to the reciprocal of median distance for each dataset. There are two classes clearly visible in the 2D kPCA plot of the Biomed data as shown in Figure 5.6. The graph shows that the outliers affect the performance of algorithms detrimentally.

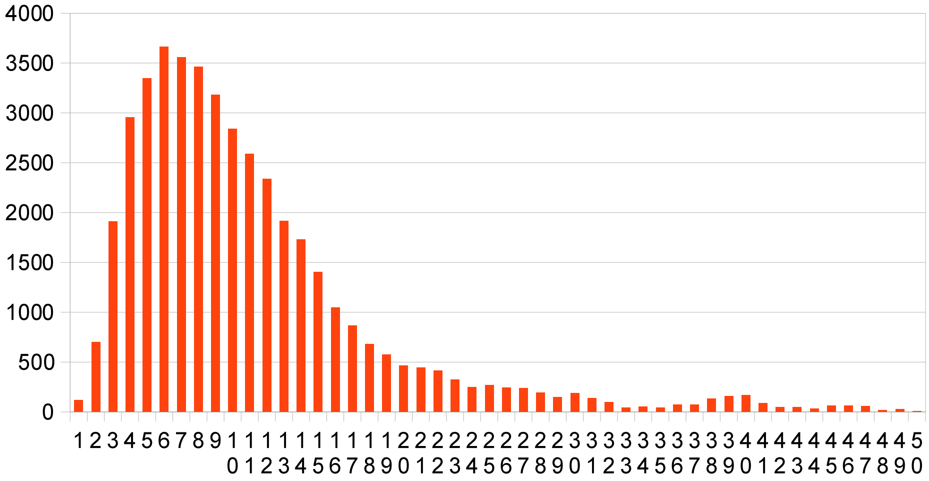


Figure 5.4: The histogram for Euclidean Distances in Biomed Data. Bin size is 50.

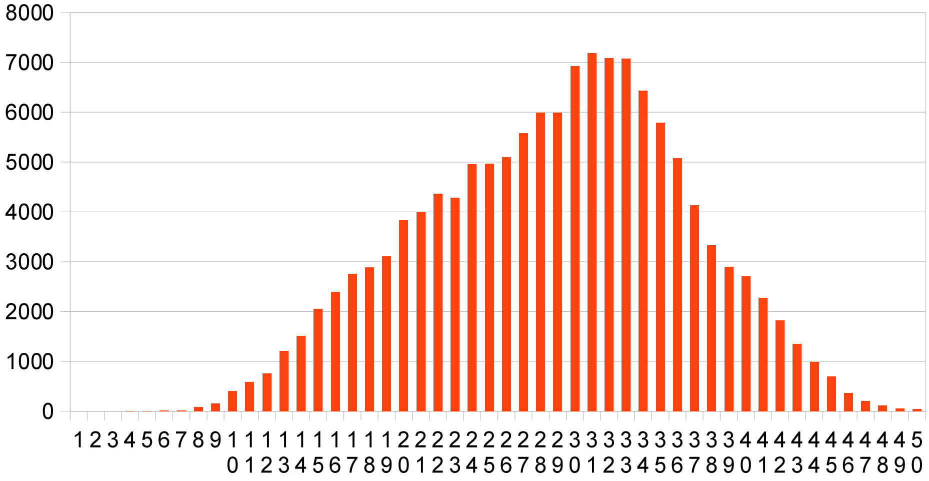


Figure 5.5: The histogram for Euclidean Distances in Dermatology Data. Bin size is 50.

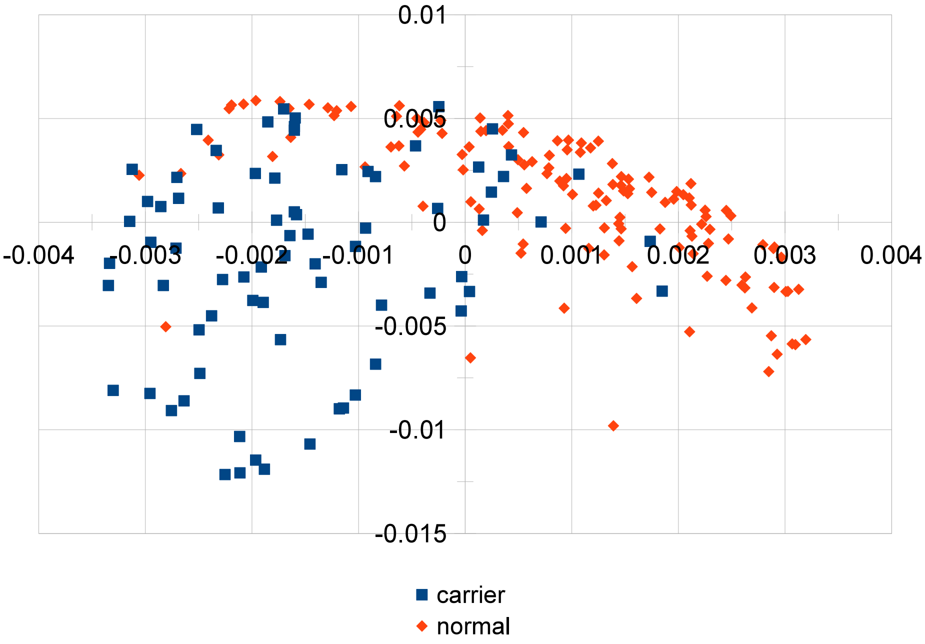


Figure 5.6: The 2D output of the kPCA for Biomed dataset.

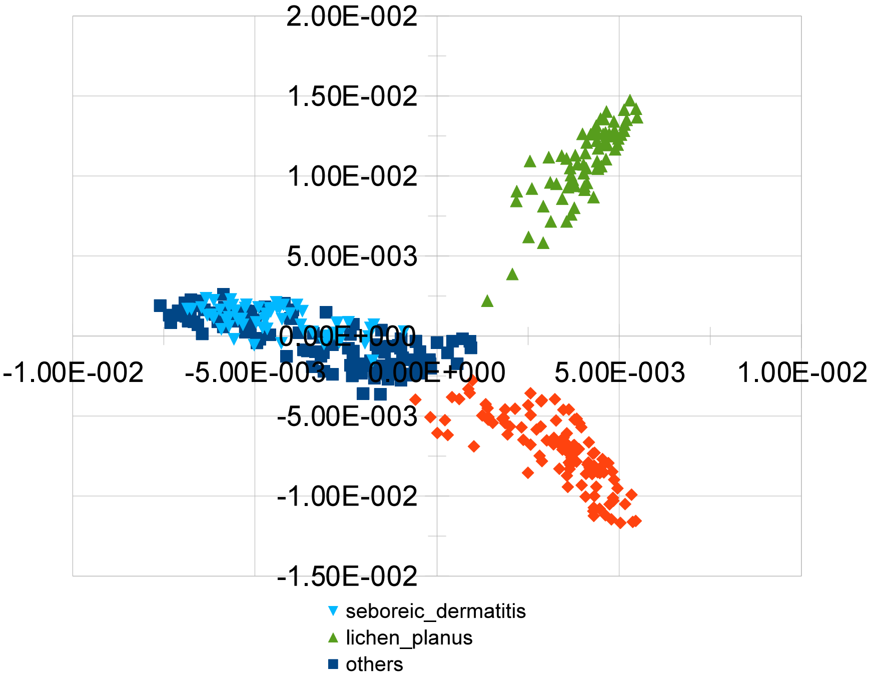


Figure 5.7: The 2D output of the kPCA for Dermatology dataset.

Figure 5.7 shows that, in the Dermatology dataset, three classes are so apart from each other that they can be separated linearly. Although two classes occupy nearly the same space, DiVo achieves better results.

The 2D PCA and kPCA plots of the Diabetes dataset do not reveal any specific clusters (data not shown). They are so interleaved that a successful classification cannot be achieved. The worst f-measures are observed for the Diabetes dataset.

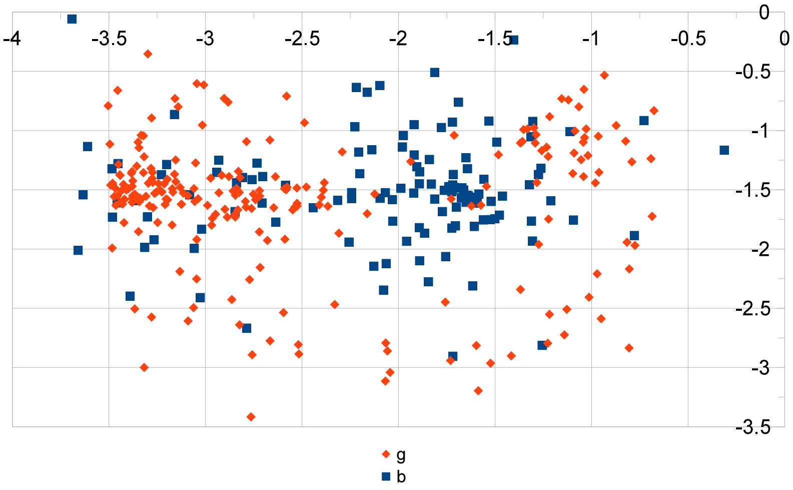


Figure 5.8: The 2D output of the PCA for Ionosphere dataset.

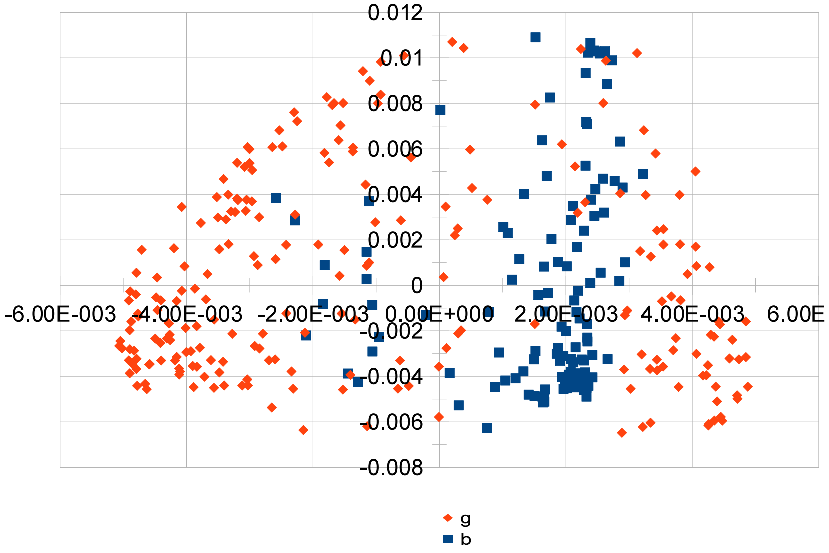


Figure 5.9: The 2D output of the kPCA for Ionosphere dataset.

Figures 5.8 and 5.9 show interesting patterns in the Ionosphere dataset. Although the 2D PCA plot is fairly cluster-free, the 2D kPCA reveals two clusters: one of them surrounds the other. The characteristic distribution of this dataset is the main reason that DiVo-M variant perform best on this specific dataset.

5.2.2 Time Complexity and Analysis of DiVo

We extracted the temporal properties of all the methods that we have selected for performance comparison. We run all implementations on the same data-class-fold set for 150 times and saved the minimum running time for training. Then we calculated the average of execution times of each class-fold set. As a result, we have a training time for each algorithm on each dataset. These results can be seen in Figure 5.10. Since there is no specific training phase for Global Gaussian, Local Gaussian and k-Nearest Neighbor Data Description, they are omitted. The performance of kNNDD-SRM on Diabetes took 120 million nanoseconds, but the time axis is capped at 30 million nanoseconds to visualize small numbers more clearly.

As can be seen in Figure 5.10, DiVo-E spends the minimum training time. Biomed is the dataset that has the least number of data and attributes. It can be concluded that for smaller data sizes, DiVo-E will be both effective and efficient.

The total time required to train kNNDD-SRM is much more higher than even the sum of all other algorithms. The increase in the training dataset also increases the training times. The

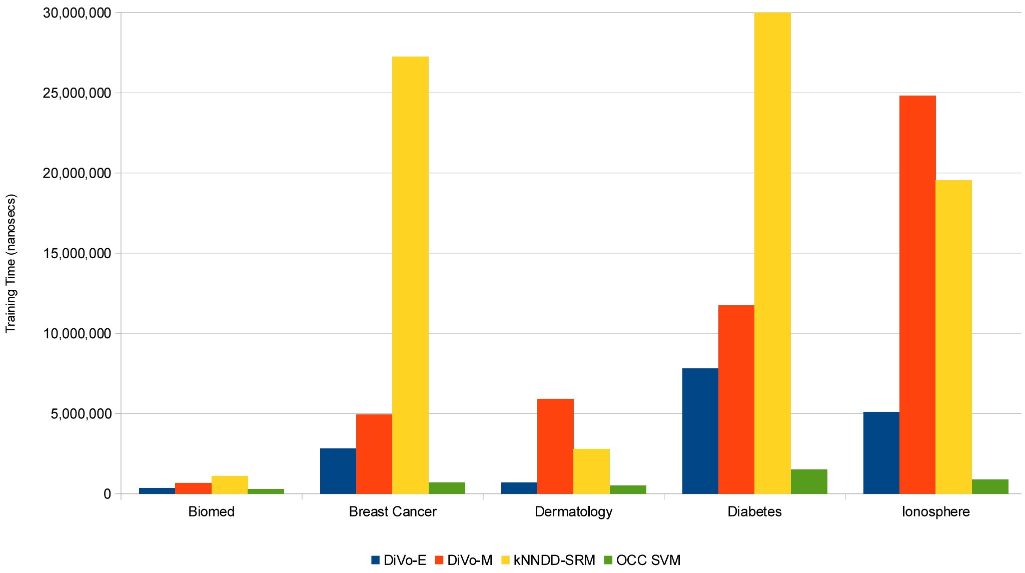


Figure 5.10: Total training times required by each algorithm on different datasets.

OCC SVM algorithm is the only one that is not affected considerably.

Here, the training time for DiVo-M is bigger than the training time of kNNDD-SRM. That is because of the dramatic increase in number of attributes. Dermatology is one of the two datasets having 34 attributes.

The training time is the highest for both OCC SVM and DiVo-E on Diabetes dataset. The reason behind these results is that one of the Diabetes dataset class has 330 training data. That is the largest class in our study. The execution time required to train kNNDD-SRM is 120 million nanoseconds on the average.

The trend seen in the results of Dermatology repeats itself, since Ionosphere dataset also has a high number of attributes. The increase in attribute size imposes a time penalty to DiVo-M algorithm. This is due to the increase in the size of matrix calculations in Mahalanobis distance.

We have implemented all algorithms except OCC SVM in Java. The OCC SVM is a scikit function coded in Python. The programming languages and implementation details may affect the execution times. However, their behavior against changes in data and attribute size are independent of these two facts. This is presented in Table 5.4. Here, we assume that for a dataset having k data and n attributes, training time takes 1 unit of time. By means of our running time calculations, we extracted the multiplication factor that is required to train a dataset which has $2k$ data and n attributes. Moreover, we carried the same process to find the training time for another dataset which has $3k$ data and $2n$ attributes. These are in fact different classes in different datasets. The first one is "Biomed-carrier" class. The second one is "Biomed-normal" class. The third one is "Breast Cancer-benign" class. This way, we have the opportunity to investigate the effect of changes in the number of both training data and attributes.

Table 5.4: The multiplication factors required to train each algorithm.

Algorithm	k data n attr.	2k data n attr.	3k data 2n attr.
DiVo-E	1.00	2.93	14.55
DiVo-M	1.00	2.49	12.43
kNNDD-SRM	1.00	2.40	31.82
OCC SVM	1.00	1.32	2.58

First observation is that OCC SVM is affected by a much more lesser extent than the others. Second, the number of training data plays a bigger role than the number of attributes for DiVo algorithm. Also, the increase in training data size and number of attributes adds up a greater complexity for kNNDD-SRM.

5.3 Observations

The DiVo-M variant achieves the best results on three datasets, while coming a very close second on the Biomed. It is the best and most consistent algorithm for the datasets used in our experiments.

The average f-measures and performance rank of DiVo-E varies from dataset to dataset, but its comparative classification power do not fluctuate. For example, kNNDD-SRM generated one of the better results in the Breast Cancer dataset but for the Dermatology dataset, its accuracy was the worst. Local Gaussian also performed better on the Biomed dataset, however it performed poorly on the Dermatology dataset. DiVo-E sustains its performance level for all of the datasets. Moreover, the standard deviations of DiVo-E within a dataset are also worth attention. The different random training data distributions for each fold and different classes affect it to a lesser extent than the other algorithms. In general, our results show that the proposed algorithm, DiVo, achieves better success levels on some UCI datasets than the state of the art one class classification methods.

CHAPTER 6

DIVO ON MIRNA TARGET FILTERING

After examining the performance results of DiVo algorithm, we decided to use it on filtering the miRNA target predictions of TargetScan. In this study, we do not need the ground truth negative dataset that we have produced. Instead, we train the DiVo algorithm with only the ground truth positive dataset. First, we analyzed if there exists a correlation between TargetScan predictions and positively labeling targets by DiVo. Following this analysis, we compare both of them with the experimentally validated miRNA targets.

The important assumption here is that, the first one third of the TargetScan predictions having the best context scores are the most likely candidates of being a real target. Therefore, we count the number of experimentally validated targets within that set. To make a comparison, we also count the number of experimentally validated targets within the positively labeled targets by DiVo.

As the last study, we tried to enhance the TargetScan predictions by filtering them with DiVo. The distinction between two studies is that, the last one shows the effect of adding the topological network properties as a feature to existing miRNA target prediction tools while the previous one measures the miRNA target prediction power of topologic network properties on their own.

6.1 Correlation Analysis

For the correlation analysis, we used the same validated targets in TarBase and pSILAC. For each miRNA, we saved the targets of others as positive training sets. We trained DiVo by

Table 6.1: The correlation coefficients between DiVo-Euclidean vote counts and TargetScan context scores. NA indicates that the pSILAC targets of that miRNA are not used or do not exist in the database. Div/0 occurs when all targets of a miRNA have the same number of votes.

miRNA Name	pSILAC	TarBase
let-7	Div/0	Div/0
mir-1	-0.02454	-0.025167
mir-16	Div/0	Div/0
mir-29	NA	Div/0
mir-30	0.00244	0.001608
mir-124	NA	-0.016943
mir-155	-0.00712	-0.007122
mir-373	NA	0.045897

using these. From the TargetScan predictions, we removed the ones that are in the training data. Then, for each of the remaining predictions, we find the number of votes assigned by DiVo. We run the DiVo method with both Euclidean and Mahalanobis distances. We tried to see if there exists a correlation between the votes they get from DiVo and context scores they get from TargetScan. For each miRNA, we accumulated the following tables for Euclidean distance (Table 6.1) and Mahalanobis distance (Table 6.2):

Table 6.2: The correlation coefficients between DiVo-Mahalanobis vote counts and TargetScan context scores. NA indicates that the pSILAC targets of that miRNA are not used or do not exist in the database.

miRNA Name	pSILAC	TarBase
let-7	0.03562	0.029660
miR-1	-0.00577	-0.035094
miR-16	-0.00657	0.014043
miR-29	NA	-0.006864
miR-30	-0.00061	0.034123
miR-124	NA	-0.036311
miR-155	-0.00712	0.006299
miR-373	NA	0.030477

The results do indicate that there is no correlation between the TargetScan context scores and DiVo vote counts (Tables 6.1 and 6.2). All values are around zero. This shows that the positively labeled targets by DiVo and TargetScan predictions depending on the context score will produce different miRNA target prediction sets. To verify this claim, we held another experiment which is described in the next section.

6.2 Experimental Evaluation of DiVo on miRNA Target Prediction

Using the same setup as in correlation analysis, we run the DiVo algorithm with Euclidean and Mahalanobis distance. The aim here is to measure the discrimination power of topological network properties. For each miRNA, the TargetScan targets are gathered. These targets are supplied to DiVo and their votes are calculated. Applying different stringency levels generated different miRNA target prediction sets. For each set, we count the number of experimentally validated targets within that set. This way, the voting results of DiVo directly translated into miRNA target prediction. To make a comparison, we also count the number of experimentally validated targets in the best one third predictions of TargetScan that have the highest context scores. All these mean that, we position DiVo as a miRNA target prediction tool by itself.

Here we only present the Mahalanobis results since for the Euclidean case, all the training almost always gave full votes to TargetScan predictions. This means that the stringency has no effect while using Euclidean Distance. DiVo with Euclidean distance cannot differentiate the miRNA targets with miRNA non-targets. It almost always classify a test data as target.

It is clear that DiVo with Mahalanobis Distance generates better results than the TargetScan scores for miR-16 and miR-30 (Table 6.4). That is the case for both pSILAC and TarBase datasets. When using TarBase, DiVo, again, produces a higher recall value for let-7 and

Table 6.3: Effect of different stringency levels in recall using DiVo with Mahalanobis distance on pSILAC dataset. The recall of TargetScan prediction scores and recall of DiVo is presented. Eleven different ratio values, which are 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1 of the DiVo are compared. - means that DiVo filters out all the verified targets.

miRNA	Recalls											
Name	TargetScan	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
let-7	**0.12360**	0.07064	0.07031	0.05392	0.06849	0.06944	0.07042	0.07246	0.06250	0.05952	0.03704	-
miR-1	**0.20134**	0.18321	0.17101	0.17822	0.17365	0.17073	0.17610	0.17308	0.16779	0.18627	0.15294	0.17647
miR-16	0.09787	**0.12693**	0.12195	0.11141	0.11864	0.12083	0.12169	0.11950	0.11215	0.11688	0.09091	1.00000
miR-30	0.10931	0.07544	0.07679	0.08061	0.08553	0.08311	0.08202	0.08494	0.09375	0.09649	**0.16981**	-
miR-155	**0.14706**	0.14428	-	-	-	-	-	-	-	-	-	-

Table 6.4: Effect of different stringency levels in recall using DiVo with Mahalanobis distance on TarBase dataset. The recall of TargetScan prediction scores and recall of DiVo is presented. Eleven different ratio values, which are 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1 of the DiVo are compared. - means that DiVo filters out all the verified targets.

miRNA	Recalls											
Name	TargetScan	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
let-7	0.06742	0.03612	0.03485	0.03647	0.03409	0.03543	0.04000	0.05344	0.05941	**0.09091**	0.08333	-
miR-1	0.16107	0.15065	0.15134	0.14887	0.14981	0.16667	0.17000	0.17089	0.18919	0.17333	**0.21951**	-
miR-16	0.04255	0.04341	0.03769	0.03916	**0.04848**	**0.04848**	0.04430	0.04605	0.04054	0.01887	0.02941	0.00000
miR-29	**0.04787**	0.02748	0.02948	0.01429	0.01053	0.01111	0.01604	0.02128	0.02703	0.01351	0.02273	-
miR-30	0.06073	0.03577	0.03770	0.03883	0.04615	0.04663	0.04688	0.04839	0.04420	0.04167	0.05769	**0.07143**
miR-124	**0.16225**	0.08772	0.08488	0.08518	0.07927	0.08243	0.08108	0.08550	0.10270	0.09836	0.12281	-
miR-155	**0.13235**	0.09950	-	-	-	-	-	-	-	-	-	-
miR-373	**0.03676**	0.01955	0.02778	0.03101	0.02542	0.02564	0.02586	0.01316	0.00000	0.00000	0.00000	-

miR-1. However, the results produced by DiVo using only topological network properties for miRNA target prediction show that, these features are not enough on their own for a successful miRNA target prediction. Independent of the prediction method, the recall values are low, since the number of TargetScan predictions are much more than the validated number of miRNA targets.

6.3 Experimental Evaluation of DiVo on Filtering miRNA Targets

Using only the DiVo votes as a miRNA target prediction mechanism does not produce much better results than the TargetScan scores. Instead of putting DiVo as a direct competitor to the established miRNA target prediction tools, we add it as a feature to filter the TargetScan predictions. This way, DiVo become a collaborative algorithm rather than a competitor one.

We used the same setup as in correlation analysis. For each TargetScan prediction, we gathered the TargetScan context score and DiVo vote. We only investigated DiVo with Mahalanobis Distance since for the Euclidean case, we have almost all target predictions of TargetScan set to be a possible miRNA target. We named the first one third of the TargetScan predictions having the best context scores as the set B and named the remaining predictions as the set W.

We employed two different filtering methods. The first one filters the elements of only B by DiVo votes using stringency levels. If the ratio of DiVo votes to all possible training data (vote ratio) is above or equal to the stringency level s, then we retained that prediction as a possible target. Else, we left out that prediction. We called the proper subsets that have only filtered best predictions as "filterBest". The stringency s is between 0 and 1. For each iteration, it is incremented by 0.1. The second method does the same filtering as the first one. Moreover, it also adds some predicted targets which are low on context scoring but above the stringency level $(1-s)$ on voting. These come from the set W. We called these miRNA target prediction subsets as "filterAll". Here, The stringency s is between 0 and 0.5. For each iteration, it is incremented by 0.05.

In Figure 6.1 we can see the effect of the stringency to the recall of TargetScan predictions when using TarBase dataset. The orange plots represent the recalls for "filterBest" sets while the blue plots represent "filterAll" sets. For example, when the stringency level $s = 0.1$, the predictions in B should have a vote ratio higher than 0.1 to be positively labeled as a target. On the other side, for the same stringency level $s = 0.1$, the predictions in W should have a vote ratio higher than 0.9 to be positively labeled as a target. Since the incrementation steps for the second filter is smaller than the first one, the last stringency will be 0.5. This means that, as the last case, the "filterAll" subset will contain all targets in both B and W which are approved by half of the training data. The dashed lines show where the original TargetScan recall lies and added for comparison purposes. The same explanations also hold for figure 6.2, where we can see the effect of the stringency on the recall of TargetScan predictions when using pSILAC dataset.

The pictures depicted by using TarBase or pSILAC are different. For TarBase, we cannot report definitive results for miR-29. For the other miRNAs, "filterBest" recalls are almost always better than the "filterAll" ones (Figures in 6.1). let-7, miR-1 and miR-16 shows a trend where increase in stringency causes an increase the recall of "filterBest", while decreasing the recall of "filterAll". "filterBest" is always better. Another trend can be seen for miRNAs miR-30, miR-155 and miR-373. Here, "filterBest" is better than the TargetScan score until stringency reaches to 0.6 or 0.7. Increasing the stringency makes "filterBest" worse.

Looking at specific stringency values, we see that for let-7, miR-1, miR-16, miR-30 and miR-155, 0.7 is consistently successful when "filterBest" sets are used. They produce better results than the unmodified TargetScan predictions. When the stringency is 0.6, "filterBest" achieves good results for miR-373, but for 0.7, it is far worse than the original TargetScan predictions.

For pSILAC, we cannot report definitive results for miR-16. For let-7 and miR-155, "filterBest" recalls are better than the "filterAll" ones and TargetScan predictions (Figures in 6.2). The remaining miR-1 and miR-30 results indicate that "filterAll" is better than "filterBest". In fact, for miR-30, "filterAll" is better even than original TargetScan predictions.

There are no clear trends as in pSILAC. But we can say that, when the stringency is 0.6, "filterBest" is better than TargetScan for let-7 and miR-155. For stringencies 0.05 and 0.15, "filterAll" is better than TargetScan for miR-30.

6.4 Observations

The first observation here is that, the topological network properties on their own is an insufficient feature set for effectively identifying miRNA targets. Hence, these should be added as a complementarity properties to the currently used features. However, the type of the integration also matters. Except in one case (for miR-30 with pSILAC), "filterBest" achieves better recall values than "filterAll". The problem with "filterAll" is that, increasing stringency removes predictions with higher context scores and adds the ones having lower context scores. This almost always leads to decreasing recall value. On the other hand, increasing stringency

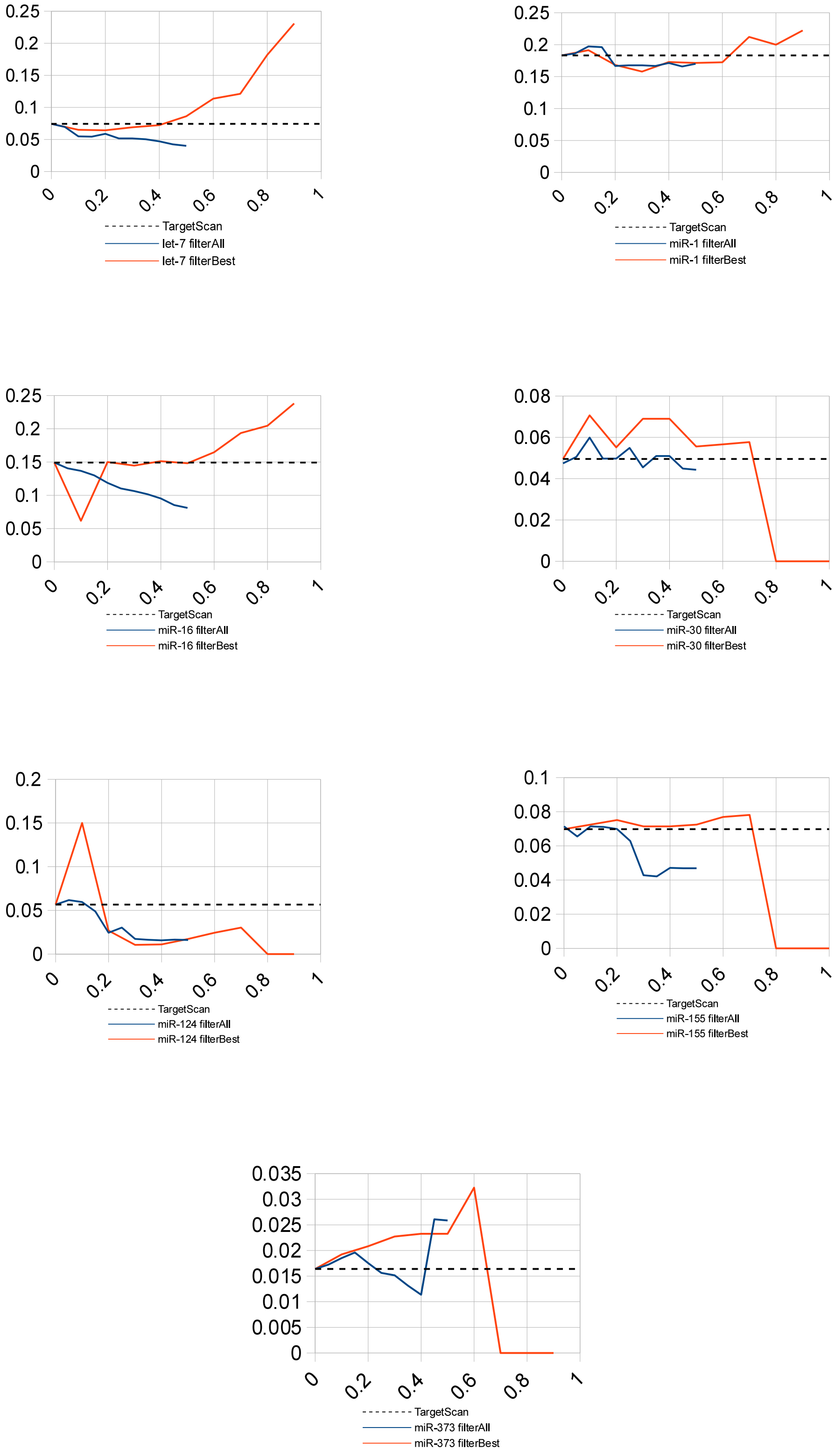


Figure 6.1: Filtering TargetScan with DiVo using TarBase target database. The x-axis represents stringency and y-axis shows recall values.

makes the "filterBest" approach more and more elitist. Only the very best predictions are retained and the number of predictions gets smaller in size, while preserving the experimentally validated targets within the filtered set. The only miRNA, where "filterAll" consistently surpassed "filterBest" was miR-30 when using pSILAC dataset. Other than that, there is no example where "filterAll" should be used.

The positive training data sets also affects the outcome of DiVo. Using pSILAC for training leads marginal gains over original TargetScan predictions. The DiVo "filterBest" for miR-155 generates the only significant increase in recall. Other than that, we can see a comparable recall value for let-7 from the "filterBest", and for miR-30 from the "filterAll" prediction sets. When the TarBase database used for training DiVo, five of the possible eight miRNAs' recall values prospers with "filterBest". For miR-1 and miR-155, again "filterBest" also generates comparable results. We can conclude that the manually curated, high quality miRNA target sets add much more power to the classification power of DiVo.

As for the stringency levels, $s = 0.6$ and $s = 0.7$ gains considerable success and attention for "filterBest". Almost in all experiments, these two stringency levels produces either better or comparable recall values. In a few cases, $s = 0.9$ attained the highest recalls, but then, the number of predictions was too low. Moreover, for the cases where $s = 0.9$ didn't show up well, it failed significantly. It was too inconsistent to be recommended. For the "filterAll"

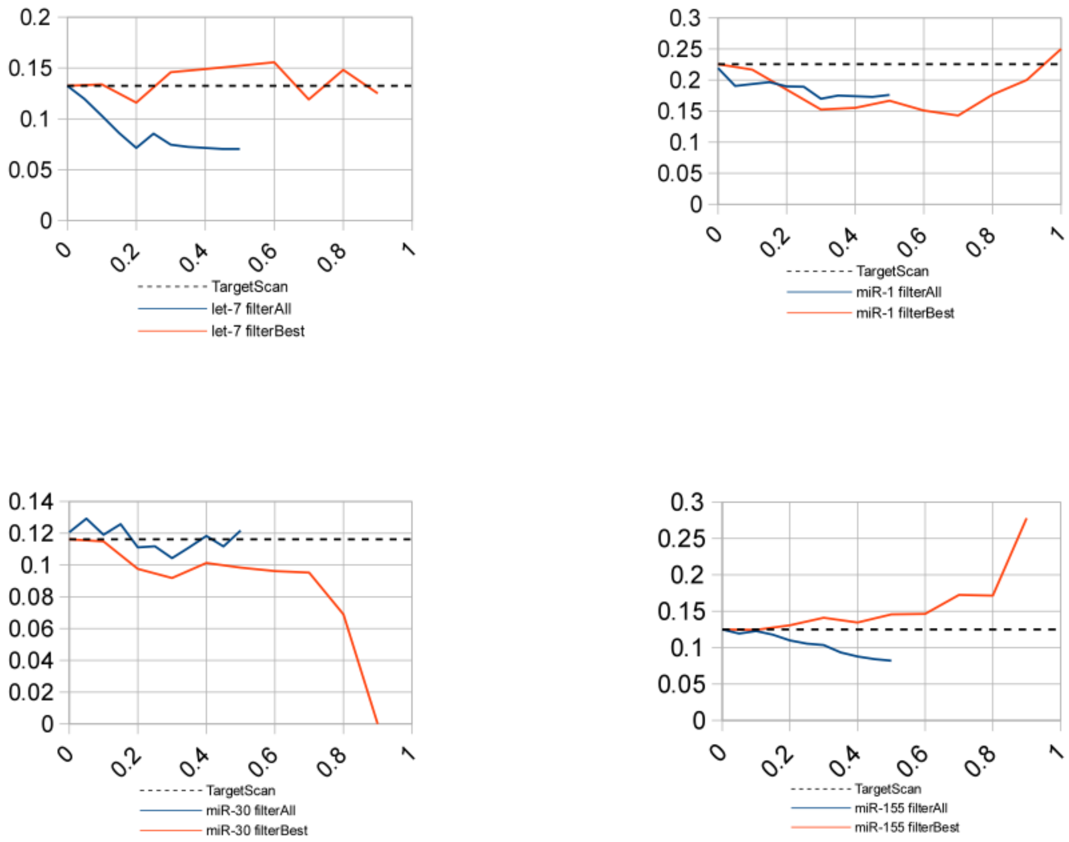


Figure 6.2: Filtering TargetScan with DiVo using pSILAC target database. The x-axis represents stringency and y-axis shows recall values.

stringencies, around $s = 0.1$ can be seen as an important step but this does not change the fact that "filterAll" is not a good mechanism as "filterBest". Overall, we can say that the filtering mechanism backed by a One-Class Classification technique yields overlapping success levels with the results of our binary classification study. Therefore, we conclude that integration of proposed features, as PPI network properties, is beneficial to existing target prediction methods.

CHAPTER 7

DISCUSSION, CONCLUSION AND FUTURE WORK

To the best of our knowledge, this is the first study that uses network context as a filter in miRNA target prediction. We implemented the Support Vector Machines and Mahalanobis Distance methods, which are both supervised learning techniques, to integrate seven different context measures. We evaluated the results of our models on two datasets of experimentally verified human miRNA targets, TarBase and pSILAC datasetsThe success of a supervised learning approach depends highly on the quality of the datasets used. Both the gold-standard miRNA target dataset used for training and the PPI network used for extracting topological features are incomplete datasets and may contain false positive targets or interactions. However, we did not conduct robustness analyses in this study due to a widely accepted assumption in the biological network analysis field that node-related statistics should not deviate too much from the ones in a complete and correct network as long as the PPI network is neither strongly biased nor too small [47].

Due to the nature of the implemented SVM and MD, we are also in need of a negative training data set. As far as we know, there is no database that stores organic molecules that are not targeted by any miRNA. In other words, there is no verified miRNA non-targets. To overcome this obstacle, we devised a heuristic and generate a set of putative miRNA non-targets. It is an important information source, especially for miRNA target prediction techniques using binary classifiers.

There are many types of biological networks, such as gene regulatory networks, metabolic pathways, and PPI networks. Although PPI networks are incomplete and noisy, they provide better genome coverage compared to other networks. Combination of these networks and applying our approach to other networks may be an important future direction.

As we explained in previous sections, the performance of individual measures varied significantly. Closeness and eigenvector centrality measures perform better while others, such as clustering coefficient and graph centrality measures have poor performance. Moreover, there is a correlation between some measures. For instance, for betweenness and stress centrality, the order of miRNA targets when sorted with respect to these measures is the same. Also, for degree and eigenvector centrality, the order of miRNA targets when sorted with respect to these measures is, again, the same. As a follow-up study, a feature selection method may be applied to select the most informative subset of topological measures for miRNA target prediction, before integration of these measures using SVM or MD.

Another performance difference can be observed between miRNAs. Our methods have shown varied results for each miRNA. For example, the average betweenness and stress centrality measures of let-7, miR-124, and miR-155 targets are relatively higher than the others. On the contrary, the average closeness and graph centrality values for the same miRNA targets are smaller than the rest. Interestingly, these are the miRNAs that SVM and MD perform poorly. On the other hand, the same network properties show just the opposite trend for miR-29 and miR-16 targets. These are the ones for which network context performs better.

The underlying biological reasons should be investigated for discovering a common pattern.

TargetScan is a popular miRNA target prediction tool among many others [96]. For this work, we used only TargetScan to apply our method. However, this filter is easily applicable to other computational miRNA target prediction algorithms. Because of the inherent problems within the miRNA target prediction tools, and lack of rapid experimental validation of miRNA targets, the number of predictions are very much higher than the current miRNA targets. This is reflected throughout the performance analysis of our studies. The sensitivity values are too low. The increase in the size of real miRNA targets will yield better recall values. Nonetheless, our results show that network context is a valuable information source that can be used in miRNA target prediction.

Using the network context of a protein in a PPI network and employing this information with binary classification techniques, we were able to increase the precision of miRNA target predictions. Note that we do not propose a new standalone miRNA target prediction tool. As we presented in Chapter 6.4, using only topological network properties for miRNA target prediction is not enough on its own for a successful miRNA target prediction process. The biology of miRNA-target interaction for many species is well studied and understood; and, we strongly believe that a miRNA target prediction method should be based on sequence, structure, and conservation analysis. But integration of additional features, such as the ones proposed in this study, to existing target prediction methods promises increased precision.

Our results show that the network context of a protein in a PPI network may be as important as sequence related features of the target site, which implies that topological properties of proteins in PPI networks can be used as an additional information source for filtering out false positive miRNA target predictions. As a follow-up study, integrating sequence based scoring measures with network context using a Decision Tree Learning model and testing it with all miRNA target prediction tools can be done.

The field of One Class Classification is an established research area. In this study, we also introduce a different heuristic that is based on a voting scheme using simple distance information between the training data samples which belong to the class under consideration. For this work, we propose a more intuitive and easy to understand one class classification method based on voting on the distances of the training samples. DiVo-M, which utilizes the Mahalanobis Distance, returns the best results for three different datasets and comes a very close second for a fourth one. The DiVo-E, which uses the Euclidean distance, produces comparable results throughout all datasets.

The main advantage of the proposed approach over existing techniques is its consistency. Both distance metrics used in conjunction with the DiVo algorithm sustain their performance level for all of the datasets. The intuitive nature of our approach makes it easy to understand and its consistency makes it applicable on different types of datasets without a decrease in performance.

As a last study, we applied our novel One-Class Classification algorithm, DiVo, to miRNA target prediction problem. The features of the classification are seven topological network properties that we used in the binary classification study. The first application was using only these topological features for miRNA target prediction and compare the output with that of TargetScan. The recall values have shown that, the topological network properties on their own is an insufficient feature set for effectively identifying miRNA targets. This agrees with the conclusion we have come in our binary classification study. As a second experiment, we integrated the DiVo algorithm utilizing the same set of features. This time, we did not predict the targets but only filter the ones having less votes. Moreover, we wanted to see the effect of different filtering mechanisms by applying two different methods. Although these methods provide diverse recall values, the main outcome aligns with our previous conclusion, where binary classifiers are used. Both of these experiments conclude that adding topological network properties have a beneficial effect on the recall of miRNA target prediction tools.

REFERENCES

[1] E. Acuna and C. Rodriguez. The Treatment of Missing Values and its Effect on Classifier Accuracy. *Classification, Clustering, and Data Mining Applications*, (1995):639–647, 2004.

[2] Panagiotis Alexiou, Manolis Maragkakis, Giorgos L. Papadopoulos, Martin Reczko, and Artemis G. Hatzigeorgiou. Lost in translation: an assessment and perspective for computational microRNA target identification. *Bioinformatics*, 25(23):3049–3055, 2009.

[3] V. Ambros. microRNAs: tiny regulators with great potential. *Cell*, 107:823–826, 2001.

[4] V. Ambros. The functions of animal microRNAs. *Nature*, 431:350–355, 2004.

[5] Daehyun Baek, Judit Villen, Chanseok Shin, Fernando D. Camargo, Steven P. Gygi, and David P. Bartel. The impact of micrornas on protein output. *Nature*, 455:64–71, 2008.

[6] Christian Barbato, Ivan Arisi, Marcos E. Frizzo, Rossella Brandi, Letizia Da Sacco, and Andrea Masotti. Computational challenges in miRNA target predictions: To be or not to be a true target? 2009. ISSN 1110-7243. doi: 10.1155/2009/803069.

[7] David P. Bartel. MicroRNAs: Genomics, Biogenesis, Mechanism, and Function. *Cell*, 116(2):281–297, 2004.

[8] Scott Baskerville and David P. Bartel. Microarray profiling of microRNAs reveals frequent coexpression with neighboring miRNAs and host genes. *RNA (New York, N.Y.)*, 11(3): 241–247, March 2005. ISSN 1355-8382. doi: 10.1261/rna.7240905.

[9] Isaac Bentwich. Prediction and validation of micrornas and their targets. *FEBS Letters*, 579(26):5904–5910, 2005.

[10] Isaac Bentwich. Identification of hundreds of conserved and nonconserved human micrornas, 2005.

[11] Tord Berggård, Sara Linse, and Peter James. Methods for the detection and analysis of protein-protein interactions. *Proteomics*, 7(16):2833–2842, August 2007. ISSN 1615-9853. doi: 10.1002/pmic.200700131.

[12] Doron Betel, Manda Wilson, Aaron Gabow, Debora S. Marks, and Chris Sander. The microRNA.org resource: targets and expression. *Nucl. Acids Res.*, 36(suppl_1):D149–153, 2008. doi: 10.1093/nar/gkm995.

[13] Kevin R. Brown and Igor Jurisica. Online Predicted Human Interaction Database. *Bioinformatics*, 21(9):2076–2082, 2005. doi: 10.1093/bioinformatics/bti273.

[14] George G. Cabral, Adriano L.I. Oliveira, and Carlos B.G. Cahú. A Novel Method for One-Class Classification Based on the Nearest Neighbor Data Description and Structural Risk Minimization. *2007 International Joint Conference on Neural Networks*, pages 1976–1981, 2007. ISSN 10987576. doi: 10.1109/IJCNN.2007.4371261.

[15] George G. Cabral, Adriano L.I. Oliveira, and Carlos B.G. Cahú. Combining nearest neighbor data description and structural risk minimization for one-class classification. *Neural Computing and Applications*, 18(2):175–183, 2008. ISSN 09410643. doi: 10.1007/s00521-007-0169-8.

[16] Colin Campbell and Kristin P. Bennett. A Linear Programming Approach to Novelty Detection. In Todd K Leen, Thomas G Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, volume 13, pages 395–401. MIT Press, 2001.

[17] A. Carè, D. Catalucci, and F. Felicetti et al. MicroRNA-133 controls cardiac hypertrophy. *Nature Medicine*, 13(5):613–618, 2007.

[18] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (last access on February 15, 2013).

[19] Chao Cheng, Xuping Fu, Pedro Alves, and Mark Gerstein. mrna expression profiles show differential regulatory effects of micrornas between estrogen receptor-positive and estrogen receptor-negative breast cancer. *Genome Biol.*, 10(9):R90, 2009.

[20] Chao Cheng, Xuping Fu, Pedro Alves, and Mark Gerstein. mRNA expression profiles show differential regulatory effects of microRNAs between estrogen receptor-positive and estrogen receptor-negative breast cancer. *Genome Biology*, page R90, 2009.

[21] The UniProt Consortium. The Universal Protein Resource (UniProt). *Nucl. Acids Res.*, 36(suppl_1):D190–195, 2008. doi: 10.1093/nar/gkm895.

[22] Koby Crammer. A Rate-Distortion One-Class Model and its Applications to Clustering. *ICML Proceedings*, 2008.

[23] Qinghua Cui, Zhenbao Yu, Enrico O Purisima, and Edwin Wang. Principles of microRNA regulation of a human cellular signaling network. *Molecular Systems Biology*, 2:46, 2006.

[24] Duin R. de Ridder D., Tax D.M.J. An Experimental Comparison of One-class Classification Methods. 1998.

[25] Eran Eden, Roy Navon, Israel Steinfeld, Doron Lipson, and Zohar Yakhini. GOrilla: a tool for discovery and visualization of enriched go terms in ranked gene lists. *BMC Bioinformatics*, 10(1):48, 2009. ISSN 1471-2105. doi: 10.1186/1471-2105-10-48.

[26] A.F. Florez, D. Park, J. Bhak, B.C. Kim, A. Kuchinsky, J.H. Morris, J. Espinosa, and C. Muskus. Protein network prediction and topological analysis in leishmania major as a tool for drug target selection. *BMC Bioinformatics*, 11:484, 2010. doi: 10.1186/1471-2105-11-484.

[27] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[28] L.C. Freeman. Centered graphs and the construction of ego networks. *Mathematical Social Sciences*, 3:291–304, 1982.

[29] Robin C. Friedman, Kyle K. Farh, Christopher B. Burge, and David P. Bartel. Most mammalian mRNAs are conserved targets of microRNAs. *Genome Research*, 19(1):92–105, January 2009. ISSN 1549-5469. doi: 10.1101/gr.082701.108.

[30] Sam Griffiths-Jones, Harpreet Kaur Saini, Stijn van Dongen, and Anton J. Enright. miRBase: tools for microRNA genomics. *Nucl. Acids Res.*, 36(suppl_1):D154–158, 2008. doi: 10.1093/nar/gkm952.

[31] Attila Gursoy, Ozlem Keskin, and Ruth Nussinov. Topological properties of protein interaction networks from a structural perspective. *Biochem Soc Trans*, 36(Pt 6):1398–403, 2008. ISSN 1470-8752.

[32] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11 (1):10–18, 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278.

[33] L. He, J.M. Thomson, and M.T. Hemann et al. A microRNA polycistron as a potential human oncogene. *Nature*, 435(7043):828–833, 2005.

[34] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, L. Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of rna secondary structures (the vienna rna package), 1994.

[35] Chun-Wei Hsu, Hsueh-Fen Juan, and Hsuan-Cheng Huang. Characterization of microRNA-regulated protein-protein interaction network. *Proteomics*, 8(10):1975–1979, 2008.

[36] B. John, A.J. Enright, A. Aravin, T. Tuschl, and C. Sander et al. Human microrna targets. *PLoS Biology*, 2(11):e363, 2004.

[37] S. Jones and J.M. Thornton. Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences*, 93(1):13–20, January 1996. ISSN 1091-6490.

[38] B. Karacali and H. Krim. Fast minimization of structural risk by nearest neighbor rule. *Neural Networks, IEEE Transactions on*, 14(1):127 – 137, jan 2003. ISSN 1045-9227. doi: 10.1109/TNN.2002.804315.

[39] Shehroz S. Khan and Michael G. Madden. A Survey of Recent Trends in One Class Classification. pages 188–197, 2010.

[40] Sung-Kyu Kim, Jin-Wu Nam, Wha-Jin Lee, and Byoung-Tak Zhang. A kernel method for microrna target prediction using sensible data and position-based features. In *CIBCB*, pages 46–52. IEEE, 2005. ISBN 0-7803-9388-0.

[41] Marianthi Kiriakidou, Peter T. Nelson, Andrei Kouranov, Petko Fitziev, Costas Bouyioukos, Zissimos Mourelatos, and Artemis Hatzigeorgiou. A combined computational-experimental approach predicts human microRNA targets. *Genes Dev*, 18(10):1165–78, May 2004. doi: 10.1101/gad.1184704.

[42] Andreas Kowarsch, Carsten Marr, Daniel Schmidl, Andreas Ruepp, and Fabian J. Theis. Tissue-Specific target analysis of Disease-Associated MicroRNAs in human signaling pathways. *PLoS ONE*, 5(6):e11154+, June 2010. doi: 10.1371/journal.pone.0011154.

[43] Azra Krek, Dominic Grun, Matthew N. Poy, Rachel Wolf, Lauren Rosenberg, Eric J. Epstein, Philip MacMenamin, Isabelle da Piedade, Kristin C Gunsalus, Markus Stoffel, and Nikolaus Rajewsky. Combinatorial microRNA target predictions. *Nature Genetics*, 37:495–500, 2005.

[44] R.C. Lee, R.L. Feinbaum, and V. Ambros. The c. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. *Cell*, 75(5):843–54, 1993.

[45] Benjamin P. Lewis, I-Hung Shih, Matthew W. Jones-Rhoades, David P. Bartel, and Christopher B. Burge. Prediction of mammalian microrna targets. *Cell*, 115:787–798, 2003.

[46] Benjamin P. Lewis, Christopher B. Burge, and David P. Bartel. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, 120:15–20, 2005.

[47] Han Liang and Wen-Hsiung Li. MicroRNA regulation of human protein protein interaction network. *RNA*, 13(9):1402–1408, 2007. doi: 10.1261/rna.634607.

[48] Lee P. Lim, Margaret E. Glasner, Soraya Yekta, Christopher B. Burge, and David P. Bartel. Vertebrate MicroRNA Genes. *Science*, 299(5612):1540–, 2003. doi: 10.1126/science.1080372.

[49] Jorg Linde, Bjorn Olsson, and Zelmina Lubovac. Network properties for ranking predicted mirna targets in breast cancer. *Adv. Bioinformatics*, 2009, 2009.

[50] Morten Lindow and Jan Gorodkin. Principles and limitations of computational microrna gene and target finding, 2007.

[51] L.M. Manevitz and M Yousef. One-Class SVMs for Document Classification. *The Journal of Machine Learning Research*, 2:139–154, 2002. ISSN 15324435.

[52] Suresh Mathivanan, Balamurugan Periaswamy, TKB Gandhi, Kumaran Kandasamy, Shubha Suresh, Riaz Mohmood, YL Ramachandra, and Akhilesh Pandey. An evaluation of human protein-protein interaction data in the public domain. *BMC Bioinformatics*, 7 (Suppl 5):S19, 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-S5-S19.

[53] Oleksiy Mazhelis. One-class classifiers : a review and analysis of suitability in the context of mobile-masquerader detection. *South African Computer Journal*, 36:29–48, 2006.

[54] Pierre Mazière and Anton J. Enright. Prediction of microRNA targets. *Drug discovery today*, 12(11-12):452–458, June 2007. ISSN 1359-6446. doi: 10.1016/j.drudis.2007.04.002.

[55] Nuno Mendes, Ana T. Freitas, and Marie-France Sagot. Current tools for the identification of mirna genes and their targets, May 2009.

[56] Antonio Mora and Ian Donaldson. Effects of protein interaction data integration, representation and reliability on the use of network properties for drug target prediction. *BMC Bioinformatics*, 13(1):294+, 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-294.

[57] W.S. Noble. *Kernel Methods in Computational Biology.* MIT Press, 2004.

[58] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.

[59] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, October 2011.

[60] T. S. Keshava Prasad, Renu Goel, Kumaran Kandasamy, Shivakumar Keerthikumar, Sameer Kumar 0002, Suresh Mathivanan, Deepthi Telikicherla, Rajesh Raju, Beema Shafreen, Abhilash Venugopal, Lavanya Balakrishnan, Arivusudar Marimuthu, Sutopa Banerjee, Devi S. Somanathan, Aimy Sebastian, Sandhya Rani, Somak Ray, C.J. Harrys Kishore, Sashi Kanth, Mukhtar Ahmed, Manoj K. Kashyap, Riaz Mohmood, Y.L. Ramachandra, V. Krishna, B. Abdul Rahiman, S. Sujatha Mohan, Prathibha Ranganathan, Subhashri Ramabadran, Raghothama Chaerkady, and Akhilesh Pandey. Human protein reference database - 2009 update. *Nucleic Acids Research*, 37(Database-Issue):767–772, 2009.

[61] Nataša Pržulj. Protein-protein interactions: making sense of networks via graph-theoretic modeling. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 33(2):115–123, February 2011. ISSN 1521-1878. doi: 10.1002/bies.201000044.

[62] Nataša Pržulj, Oleksii Kuchaiev, Aleksandar Stevanovic, and Wayne Hayes. Geometric evolutionary dynamics of protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 178–189, 2010.

[63] Marc Rehmsmeier, Peter Steffen, Matthias Hochsmann, and Robert Giegerich. Fast and effective prediction of microRNA/target duplexes. *RNA (New York, N.Y.)*, 10(10):1507–1517, October 2004. ISSN 1355-8382. doi: 10.1261/rna.5248604.

[64] William Ritchie, Megha Rajasekhar, Stephane Flamant, and John E.J. Rasko. Conserved expression patterns predict microrna targets. *PLoS Comput Biol*, 5(9):e1000513, 09 2009. doi: 10.1371/journal.pcbi.1000513.

[65] Javier De Las Rivas and Celia Fontanillo. Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, 6(6), 2010.

[66] Lecturer Roded, Sharan Scribers, Ofer Lavi, and Lev Ferdinskoif. Protein-protein interaction: Network alignment, 2006.

[67] Antony Rodriguez, Sam Griffiths-Jones, Jennifer L. Ashurst, and Allan Bradley. Identification of mammalian microrna host genes and transcription units. *Genome Res*, 14 (10A):1902–10, 2004. ISSN 1088-9051.

[68] Jean-Francois Rual, Kavitha Venkatesan, and Tong Hao et al. Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437:1173–1178, 2005.

[69] Heinz Ruffner, Andreas Bauer, and Tewis Bouwmeester. Human protein-protein interaction networks and the value for drug discovery. *Drug Discovery Today*, 12(17-18):709–716, September 2007. ISSN 13596446. doi: 10.1016/j.drudis.2007.07.011.

[70] Lukasz Salwinski, Christopher S. Miller, Adam J. Smith, Frank K. Pettit, James U. Bowie, and David Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(Database-Issue):449–451, 2004.

[71] Carolina Sanchez-Hernandez, Doreen S. Boyd, and Giles M. Foody. One-class classification for mapping a specific land-cover class: Svdd classification of fenland. *IEEE T. Geoscience and Remote Sensing*, 45(4):1061–1073, 2007.

[72] B. Schoelkopf, K. Tsuda, and J.-P. Vert, editors. *Kernel methods in computational biology.* MIT Press, 2004.

[73] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection, 2000.

[74] Matthias Selbach, Bjorn Schwanhausser, Nadine Thierfelder, Zhuo Fang, Raya Khanin, and Nikolaus Rajewsky. Widespread changes in protein synthesis induced by micrornas. *Nature*, 455:58–63, 2008.

[75] Praveen Sethupathy, Benoit Corda, and Artemis G. Hatzigeorgiou. TarBase: A comprehensive database of experimentally supported animal microRNA targets. *RNA*, 12(2): 192–197, 2006. doi: 10.1261/rna.2239606.

[76] Praveen Sethupathy, Molly Megraw, and Artemis G Hatzigeorgiou. A guide through present computational approaches for the identification of mammalian microRNA targets. *Nature Methods*, 3:881–886, 2006.

[77] Shai Shalev-Shwartz and Nathan Srebro. SVM optimization: Inverse dependence on training set size. In *25th International Conference on Machine Learning (ICML)*, July 2008.

[78] Alexander Stark, Julius Brennecke, Robert B Russell, and Stephen M Cohen. Identification of drosophila microrna targets. *PLoS Biology*, 1(3), 2003. doi: 10.1371/journal.pbio.0000060.

[79] C. Stark, B.J. Breitkreutz, A. Chatr-Aryamontri, L. Boucher, R. Oughtred, M.S. Livstone, J. Nixon, K. Van Auken, X. Wang, X. Shi, T. Reguly, J.M. Rust, A. Winter, K. Dolinski, and M. Tyers. The BioGRID interaction database: 2011 update. *Nucleic Acids Res*, 39(Database issue):D698–704. doi: 10.1093/nar/gkq1116.

[80] W. Nick Street. Cancer diagnosis and prognosis via linear-programming-based machine learning. *Operations Research*, 43:570–577, 1994.

[81] Michael P.H. Stumpf, Carsten Wiuf, and Robert M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4221–4224, March 2005. ISSN 0027-8424. doi: 10.1073/pnas.0501179102.

[82] Bernhard Suter, Saranya Kittanakom, and Igor Stagljar. Two-hybrid technologies in proteomics research. *Curr Opin Biotechnol*, 19(4):316–23, 2008. ISSN 0958-1669.

[83] Damian Szklarczyk, Andrea Franceschini, Michael Kuhn, Milan Simonovic, Alexander Roth, Pablo Minguez, Tobias Doerks, Manuel Stark, Jean Muller, Peer Bork, Lars Juhl Jensen, and Christian von Mering. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39 (Database-Issue):561–568, 2011.

[84] David M.J. Tax and Robert P.W. Duin. Support vector data description. *Mach. Learn.*, 54(1):45–66, January 2004. ISSN 0885-6125. doi: 10.1023/B:MACH.0000008084.60811.49.

[85] D.M.J. Tax. *One-class classification.* PhD thesis, 2001.

[86] D.M.J. Tax and Robert P.W. Duin. Characterizing one-class datasets. In *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, pages 21–26. PRASA, PRASA, 2005. ISBN 0-7992-2264-X.

[87] Vladimir N. Vapnik. *Statistical learning theory.* Wiley, 1 edition, September 1998. ISBN 0471030031.

[88] V.N. Vapnik. *Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control.* Wiley, Wiley New York, 1998.

[89] QingHua Wang and Luis Seabra Lopes. One-class learning for human-robot interaction. In Luis M. Camarinha-Matos, editor, *BASYS*, volume 159 of *IFIP International Federation for Information Processing*, pages 489–498. Springer, 2005. ISBN 0-387-22828-4.

[90] QingHua Wang, Luis Seabra Lopes, and David M. J. Tax. Visual object recognition through one-class learning. In Aurelio C. Campilho and Mohamed S. Kamel, editors, *ICIAR (1)*, volume 3212 of *Lecture Notes in Computer Science*, pages 463–470. Springer, 2004. ISBN 3-540-23240-0.

[91] D. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393 (6684):440–442, June 1998.

[92] Nadir Yehya, Adi N. Yerrapureddy, John W. Tobias, and Susan S. Margulies. Differential Expression Profiling Of MicroRNAs In Stretched Alveolar Epithelial Cells. *Am. J. Respir. Crit. Care Med.*, 181(1):A2036–, 2010.

[93] Malik Yousef, Segun Jung, Andrew V.V. Kossenkov, Louise C.C. Showe, and Michael K.K. Showe. Naive bayes for microrna target predictions machine learning for microrna targets. *Bioinformatics*, October 2007. ISSN 1460-2059. doi: 10.1093/bioinformatics/btm484.

[94] H. Yu, P. Braun, M.A. Yildirim, I. Lemmens, K. Venkatesan, J. Sahalie, Hirozane T. Kishikawa, F. Gebreab, N. Li, N. Simonis, and Others. High-Quality Binary Protein Interaction Map of the Yeast Interactome Network. *Science*, 322(5898):104, 2008.

[95] Xiongying Yuan, Changning Liu, Pengcheng Yang, Shunmin He, Qi Liao, Shuli Kang, and Yi Zhao. Clustered microRNAs' coordination in regulating protein-protein interaction network. *BMC systems biology*, 3(1):65+, June 2009. ISSN 1752-0509. doi: 10.1186/1752-0509-3-65.

[96] Dong Yue, Hui Liu, and Yufei Huang. Survey of computational algorithms for microrna target prediction. *Curr Genomics*, 10(7):478–92, 2009. ISSN 1875-5488.

[97] Y. Zhao, J.F. Ransom, and A. Li et al. Dysregulation of cardiogenesis, cardiac conduction, and cell cycle in mice lacking miRNA-1-2. *Cell*, 129(2):303–317, 2007.

[98] Mingzhu Zhu, Lei Gao, Xia Li, Zhicheng Liu, Chun Xu, Yuqing Yan, Erin Walker, Wei Jiang, Bin Su, Xiujie Chen, and Hui Lin. The analysis of the drug-targets based on the topological properties in the human protein-protein interaction network. *J Drug Target*, 17(7):524–32, 2009. ISSN 1029-2330.

[99] Ann S.S. Zweig, Donna Karolchik, Robert M.M. Kuhn, David Haussler, and W. James J. Kent. UCSC genome browser tutorial. *Genomics*, May 2008. ISSN 1089-8646. doi: 10.1016/j.ygeno.2008.02.003.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Sualp, Merter
Nationality: Turkish (TC)
Date and Place of Birth: May 2, 1979, İzmir
Marital Status: Married
Phone: +90 533 353 8879
Fax: +90 312 507 6163

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Computer Engineering Dept., METU	2005
B.S.	Computer Engineering Dept., METU	2001
High School	İzmir Fen Lisesi	1997

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2001 - Current	Central Bank of Republic of Turkey	Computer Engineer

PUBLICATIONS

International Journal Publications

Merter Sualp and Tolga Can, Using network context as a filter for miRNA target prediction, Biosystems, 105(3):201-209, 2011. doi:10.1016/j.biosystems.2011.04.002.