Machine Learning Approaches to Biological Sequence and Phenotype Data Analysis

by

Renqiang Min

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Machine Learning Approaches to Biological Sequence and Phenotype Data Analysis

Renqiang Min

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2010

To understand biology at a system level, I presented novel machine learning algorithms to reveal the underlying mechanisms of how genes and their products function in different biological levels in this thesis. Specifically, at sequence level, based on Kernel Support Vector Machines (SVMs), I proposed learned random-walk kernel and learned empirical-map kernel to identify protein remote homology solely based on sequence data, and I proposed a discriminative motif discovery algorithm to identify sequence motifs that characterize protein sequences' remote homology membership. The proposed approaches significantly outperform previous methods, especially on some challenging protein families. At expression and protein level, using hierarchical Bayesian graphical models, I developed the first high-throughput computational predictive model to filter sequence-based predictions of microRNA targets by incorporating the proteomic data of putative microRNA target genes, and I proposed another probabilistic model to explore the underlying mechanisms of microRNA regulation by combining the expression profile data of messenger RNAs and microRNAs. At cellular level, I further investigated how yeast genes manifest their functions in cell morphology by performing gene function prediction from the morphology data of yeast temperature-sensitive alleles. The developed prediction models enable biologists to choose some interesting yeast essential genes and study their predicted novel functions.

Acknowledgements

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

During the last decade, machine learning methods have received wide applicability in the computational biology and bioinformatics research field. On one hand, machine learning methods help to extract important information from high-throughput datasets to explain biological phenomena; on the other hand, some challenging biological problems motivate machine learning researchers to propose novel algorithms. In this thesis, I will derive new machine learning methods to analyze bio-sequence data such as protein sequence data, and phenotype data such as gene expression data, protein abundance data and quantitative morphology data, to reveal the underlying mechanisms of gene regulation and gene function in different biological levels.

At sequence level, using kernel Support Vector Machines (SVMs), I will describe how to identify proteins' remote homology solely based on their sequence data, and I will discuss how to identify discriminative motifs that characterize protein sequences' remote homology. SVMs have been applied to solve this problem, achieving reasonable success, however, they perform poorly on some difficult protein families. I have developed two approaches to solve this problem: the first uses a learned random-walk kernel based on sequence data, and the second constructs an empirical kernel map using a profile kernel. The resulting kernels all have much better prediction performance than the profile kernel directly derived from protein sequences. On a competitive SCOP benchmark dataset, the overall mean ROC_{50} scores on 54 protein families I obtained using my approaches are above 0.90, which significantly outperform previous published results. My methods here were also used to discover biologically meaningful sequence motifs that characterize the super-family membership of protein sequences. The methods above can be readily applied to discover Transcription Factor Binding Sites (TFBSs) and microRNA binding sites given some positive sequences containing binding sites for common regulators and negative control sequences without containing any binding sites of interest.

At expression and protein level, by further investigating predicted binding sites of microRNAs, I propose hierarchical Bayesian graphical models to explore how microRNAs regulate genes, which is insightful in understanding how proteins are effectively produced from messenger RNAs (mRNAs). In details, microRNAs are involved in many crucial biological processes such as development, differentiation, and tumorigenesis. It is estimated that mammalian genomes contains hundreds of microRNA and over one-third of the genes are under their regulation. A number of computational prediction tools are currently available, which predict the target binding sites of microRNAs based on the sequence complementarity between microRNA and the target sites, and the evolutionary conservation of such sites. In collaboration with biologists, I propose a novel Bayesian probabilistic approach, which is motivated by the intuition that a highly confident true microRNA target gene should have lower protein abundance, associated with a high expression level for the microRNA regulator. It is demonstrated that the proposed approach can improve the prediction accuracy by removing false-positives. In addition, the approach can also be used to infer the regulatory mechanisms of miRNAs. This approach offers the first attempt in incorporating proteomic data in prediction and characterization of microRNA regulations, which will become very valuable when more high-throughput protein abundance data become available.

At cellular level, based on recent available yeast morphology data, I have further investigated how yeast genes manifest their functions in observable cell phenotype. In this project, my collaborators created temperature-sensitive, viable alleles of about 50% of the essential genes of yeast and performed a large-scale microscopy screen of the resulting phenotype. Based on these cellular morphology data, they study how essential genes respond to environments at varying temperatures. For each marker of a given mutant, they measure the values of a set of features, giving each feature a distribution of values. I propose several novel feature representation methods to transform this complex dataset into a standard feature matrix and then I use kernel SVMs to analyze the functions of temperature-sensitive alleles of yeast mutants on this challenging dataset. The results obtained provide new insight into understanding how genes manifest their functions in morphology, and the prediction methods make it possible to accurately study the novel functions of well characterized essential genes.

1.2 Thesis Organization

I will organize this thesis as follows:

In chapter 2, I will discuss how to identify proteins' remote homology solely based on sequence data and how to discover discriminative protein sequence motifs characterizing their remote homology memberships using kernel machine learning methods. The approaches to protein remote homology identification was published in [51], and the approach to discriminative motif discovery was published in [52]. In this project, I compiled the data, conceived the ideas of learned random-walk kernel and learned empirical map kernel, and proposed the discriminative motif discovery method, and Prof. Rui Kuang helped construct the state-of-the-art profile kernel, and Jingjing Li helped interpret the biological results, and Prof. Anthony Bonner and Prof. Zhaolei Zhang supervised the research.

In chapter 3, I will present probabilistic graphical models to predict microRNAs' targets and their regulatory mechanisms by integrating sequence data, microRNA expression profile data, mRNA expression profile data, and proteomic data. A preliminary version of this chapter was published in [45]. Jingjing Li conceived the project of using proteomic data for microRNA regulation analysis, and I built the hierarchical Bayesian graphical model for this project and did the technical writing, and Prof. Anthony Bonner and Prof. Zhaolei Zhang supervised the research.

In chapter 4, I will describe how to generate effective feature representations from the complex morphology data for yeast temperature-sensitive mutants, thereby to predict gene functions using kernel SVMs. In this project, Dr. Franco Vizeacoumar provided the extracted feature data of microscopy images, and Jingjing Li and Dr. Ke Jin constructed a database and cleaned the data, and I did all the technical modeling and writing, and Prof. Anthony Bonner and Prof. Zhaolei Zhang supervised the research. This chapter will be submitted to a journal for publication.

In chapter 5, I will conclude this thesis with some discussions, and propose possible extensions of the work described in this thesis and future research directions.

Chapter 2

Protein Sequence Classification and Motif Discovery

The functions of genes are often mainly determined by the structures of their protein products, and the protein structures are mainly determined by protein sequences. To accurately predict gene functions from sequences, we need to predict protein structures from sequences, which becomes a classical sequence classification problem. In this chapter, I will present machine learning approaches to protein remote homology identification on a benchmark dataset for protein structure classification from sequences.

2.1 Background

Machine learning researchers are often faced with classification problems with limited labeled data and a large amount of unlabeled data. In biological problems, this is almost always the case. It takes a lot of manual work or expensive biological experiments to label data. Like the protein remote homology problem I will describe here, we often have several positive training cases, many negative training cases, and a lot of unlabeled data for many protein families. Therefore, we need good algorithms that can best take advantage of the unlabeled data. Moreover, classifying biological sequences is an important and challenging problem both in computational biology and machine learning. On the biological side, it helps to identify interesting sequence regions and protein domains that are related to a particular biological function; on the computational side, it motivates many novel and effective new classification approaches specifically for sequence data. Generative models (e.g., profile HMMs [39], [7]), discriminative models (e.g., kernel SVMs [32], [42], [46]), and graph-based models [69] have been applied to solve this problem.

In [32], [42], [74], [46] and [40], it has been shown that kernel SVMs have better prediction performance on biological sequence data than other methods. Moreover, it was shown in [74] that random-walk kernels [65] and empirical-map kernels [57] produced promising results on protein remote homology detection. However, the process of deciding the optimal number of random steps in a random-walk kernel and the process of deciding the scaling parameter in an empirical-map kernel remain as challenging problems [74]. In this chapter, I present two approaches to address these problems that improve prediction accuracy. In the first approach, I use label information of training data and a positive linear combination of random-walk kernels to approximate the random-walk kernel with the optimum steps of a random walk, thereby obtaining a convex combination of random-walk kernels with different random-walk steps which achieves the best classification confidence on the labeled training set. In the second approach, I construct an empirical kernel map using profile kernels. The scaling parameter of the empirical map is decided by minimizing the Leave-One-Out (LOO) nearest neighbor classification error.

As is described in [40], kernel SVMs can not only be applied to classify biological sequences, but also they can be used to extract discriminative sequence motifs that explain the classification results. In this chapter, I will use SVMs based on learned random-walk kernels to extract protein sequence motifs contributing to discriminating protein sequences' remote homology. Experimental results on the SCOP benchmark dataset show that learned random-walk kernel not only achieves significant improvement over the best published result and the result given by the random-walk kernel with a fixed number of random steps, but also effectively extracts meaningful protein sequence motifs that are responsible for discriminating the memberships of protein sequences' remote homology in SCOP.

2.2 SVM for biological sequence classification using mismatch string kernels

A SVM ([57] and [70]) is a discriminative model proposed especially for classification. Consider a two-class training set, $\{X, y\}$ and a test set U, where X is a matrix whose i-th column, X_i, is the feature vector of data point i in the training set, U is a matrix whose j-th column, U_j, is the feature vector of data point j in the test set, and y, a column vector whose i-th component y_i is the label of data point i in the labeled set, $y_i \in \{-1, 1\}$, $X_i, U_j \in R^d$, $i = 1, \cdots, N, j = 1, \cdots, M$. A linear SVM gives a separating hyper-plane that maximizes the margin between the sample data points of the two classes. The primal problem of a soft-margin SVM is as follows:

$$\begin{aligned} min_{w,b,\xi} \qquad & L(w) = \frac{1}{2}\|w\|^2 + C\left(\Sigma_i \xi_i\right), \qquad (2.1) \\ & w^T X_i + b \geq +1 - \xi_i \quad for \;\; y_i = +1, \\ & w^T X_i + b \leq -1 + \xi_i \quad for \;\; y_i = -1, \\ & \xi_i \geq 0 \quad \forall i \in \{1, \cdots, N\}, \end{aligned}$$

where C is the penalty coefficient penalizing margin violations, and the ξ_i are non-negative slack variables, which will be set to 0 when the dataset is separable. The dual problem of the soft-margin SVM can be formulated as follows:

$$\begin{aligned} max_{\alpha} \quad & 2\alpha^T\mathbf{1} - \alpha^T(K_{tr} \otimes yy^T)\alpha, \\ s.t. \quad & \alpha^T y = 0, \\ & \mathbf{0} \leq \alpha \leq C\mathbf{1}, \end{aligned} \tag{2.2}$$

where **1** and **0** are column vectors containing all ones and zeros respectively, $\otimes$ is the component-wise matrix multiplication operator, $K = [X|U]^T[X|U]$, is the dot product between feature vectors of pairwise data points, and K_{tr} is the training part of K where $K_{tr} = X^T X$. As the above dual problem is only dependent on dot-products between feature vectors, we can discard the original feature vectors of data points and calculate a kernel matrix K directly to represent the relationship between the original data points. As is discussed in [57], any symmetric positive semi-definite matrix can be used as a valid kernel matrix K. Therefore by constructing a kernel, K, we can use the induced kernel map to map every data point, X_i, to a high-dimensional feature space, in which a SVM can be used to generate a separating hyper-plane.

For biological sequences, a kernel function can be used to map these sequences consisting of characters representing amino acids to a higher dimensional feature space on which a max-margin classifier is trained. All the computations of a SVM are performed on the dot products of the pairwise feature vectors stored in the kernel matrix. For example, suppose A is an alphabet of ℓ symbols ($\ell = 20$ for protein sequences), then k-mer string kernel maps every sequence in A to a ℓ^k-dimensional feature space in which coordinates are indexed by all possible sub-sequences of length k (k-mers). Specifically, the feature map of a k-mer string kernel is given by

$$\Phi_k(x) = \left(\Phi_{\alpha_1}(x), \Phi_{\alpha_2}(x), \cdots, \Phi_{\alpha_{l^k}}(x)\right)^T, \tag{2.3}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_{\ell^k}$ is an ordering of all the ℓ^k possible k-mers, and $\Phi_\alpha(x)$ is the number of occurrences of k-mer α in sequence x. The corresponding kernel matrix is

$$K_k(x, y) = \Phi_k(x)^T \Phi_k(y). \tag{2.4}$$

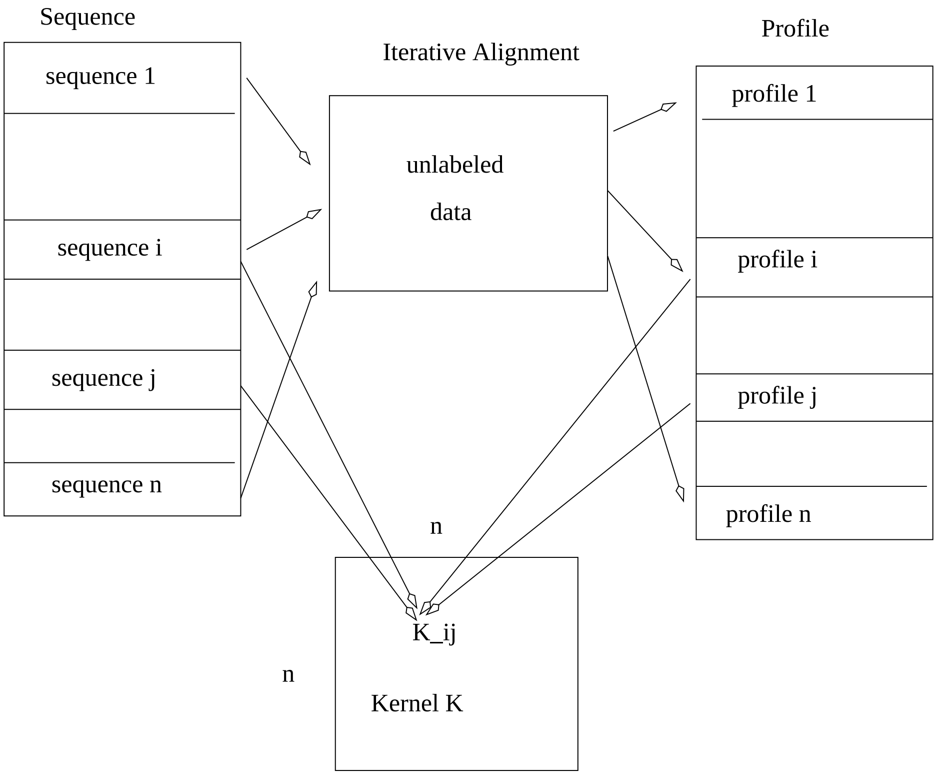


Figure 2.1: The work-flow of constructing a profile kernel in [40].

The mismatch string kernel extends this idea by accommodating mismatches when counting the number of occurrences of a k-mer in an input sequence. In particular, for any k-mer, α, let $N_{(\alpha,m)}$ be the set of all k-mers that differ from α by at most m mismatches. The kernel mapping and kernel matrix are then defined as follows:

$$\begin{aligned}\Phi_{(k,m)}\left(x\right) = (\Phi_{(k,m),\alpha_1}\left(x\right), \\ \cdots, \Phi_{(k,m),\alpha_{\ell k}}\left(x\right))^T,\end{aligned} \tag{2.5}$$

$$\Phi_{(k,m),\alpha}\left(x\right) = \sum_{\beta \in N_{(\alpha,m)}(x)} \Phi_{\beta}\left(x\right), \tag{2.6}$$

$$K_{(k,m)}\left(x,y\right) = \Phi_{(k,m)}\left(x\right)^T \Phi_{(k,m)}\left(y\right). \tag{2.7}$$

A profile of a protein sequence is a sequence of multinomial distributions. Each position of a protein sequence's profile is a multinomial distribution on 20 amino acids, representing the emission probabilities of the 20 amino acids at each position in that sequence. A Profile Kernel [40] extends the mismatch-string kernel by using additional profile information of each sequence. Instead of treating all k-mers with less than m mismatches similarly as the mismatch-string kernel described above, the profile-kernel examines these k-mers further by looking at the emission probabilities (profiles) at the mismatched positions and only accepts those mismatches that pass a certain threshold. The work-flow for constructing a profile kernel as described in [40] is shown in Fig. 2.1. Each sequence has a profile, which is obtained by iteratively aligning each sequence to the sequences in an unlabeled set using PSI-BLAST [2]. Suppose we have a sequence $x = x_1x_2...x_N$ of amino acids of length N, then $P(x) = \{p_i^x(a), a \in \Sigma\}_{i=1}^N$ is the profile of sequence x, where Σ is the set of 20 amino acids and $p_i^x(\cdot)$ is the multinomial distribution on the 20 amino acids at the i-th position of the profile of sequence x. For e.g., $p_i^x(a)$, is the emission probability of amino acid a at position i, such that $\sum_{a\in\Sigma} p_i(a) = 1$ at each position i. In the Profile Kernel, the neighborhood of a k-mer $x[j+1 : j+k]$=$x_{j+1}x_{j+2}...x_{j+k}$ in sequence x is defined as:

$$\begin{aligned} M_{(k,\sigma)}(P(x[j+1:j+k])) = \\ \{\beta = b_1...b_k : -\sum_{i=1}^{k} logp_{j+i}^x(b_i) < \sigma\}, \end{aligned} \tag{2.8}$$

where the free parameter σ controls the size of the neighborhood, and $p_{j+i}^x(b)$ for $i = 1, ..., k$ is obtained from the profile of sequence x, $0 \leq j \leq |x| - k$. Further, $p_{j+i}^x(b)$ can be smoothed using the background frequency of amino acid b. The feature vector of sequence x in the Profile Kernel is defined as the following:

$$\begin{aligned} \Phi_{(k,\sigma)}(x) = \sum_{j=0}^{|x|-k} (\phi_{\beta_1}(P(x[j+1:j+k])), \\ \ldots, \phi_{\beta_{\ell^k}}(P(x[j+1:j+k])))^T, \end{aligned} \tag{2.9}$$

where $\beta_1, \ldots, \beta_{\ell^k}$ is an ordering of all possible k-mers, and, the coordinate $\phi_\beta(P(x[j+1 : j+k]))$ is 1 if $\beta \in M_{(k,\sigma)}(P(x[j+1 : j+k]))$, and 0 otherwise. The profile kernel uses the profile to measure the mismatch information between different letters at each position of each sequence. Therefore, it's more accurate than the mismatch string kernel. In this chapter, I will use the profile kernel discussed above as the base kernel in the derivation of the random-walk kernel.

2.3 SCOP dataset for protein remote homology detection

To compare different methods for protein remote homology detection, I used the benchmark dataset, derived by Jaakkola from the SCOP database for this purpose (see [55] and [32]). In SCOP, protein sequences are classified into a three-level hierarchy: Fold, Superfamily, and Family, starting from the top. Remote homology is simulated by choosing all the members of a family as positive test data, some families in the same super-family of the test data as positive training data, all sequences outside the fold of the test data as either negative training data or negative test data, and sequences that are neither in the training set nor in the test set as unlabeled data. This data splitting scheme has been used in several previous papers (see [32], [46], and [74]). I used the same training and test data split as that used in [46] and [74]. I used version 1.59 of the SCOP dataset (http://astral.berkeley.edu), in which no pair of sequences share more than 95% identity.

In the data splits, of most experiments, there are only a few positive test cases but, hundreds, or even thousands of negative test cases. The maximum number of positive test cases is usually below 30, but the maximum number of negative test cases is above 2600. The minimum number of positive test cases is 1, but the minimum number of negative test cases is still above 250. In the experiments with a very limited number of positive test cases and a large number of negative test cases, we can almost ignore the ranking of positive cases below 50 negative cases. In such situations, I consider the ROC_{50} score much more informative of prediction performance of different methods than the ROC score. Here, a ROC curve plots the rate of true positives as a function of the rate of false positives at different decision thresholds. The ROC score is the area under the curve. The ROC_{50} score is the ROC score computed up to the first 50 false positives. Thus, in my experiments, I only compare the ROC_{50} scores corresponding to different kernels. We should note that Precision (the number of true positives divided the total number of predicted positives)-Recall (true positive rate) curve is another way to measure the performance of prediction programs. In biological experiments, we often have much more negatives than positives as discussed above, and biologists would rather have some reasonable number of false positives to get as many true positives as possible than get a reasonably high precision but miss a lot of other positives. In another word, biologists do not care precision as much as we evaluate the performance a search engine. For example, there are 6 positives and 6,000 negatives in our test set, and 5 positives are predicted to be ranked from 30th to 34th, and another positive is predicted to be ranked at the 99th position, and it will have a reasonably high ROC_{50} score because most of the positives are ranked among top 35 out of 6006. But if a search engine such as Google or Microsoft Bing ranks the positives like this, we probably would never use it again, because we would rather have 2 positives ranked among the top 2 positions and 3 positives ranked very very low than get all 5 positives ranked between 30 and 100.

2.4 Learned random-walk kernels and empirical-map kernels for protein remote homology identification

2.4.1 Learned random-walk kernels

In this section, I will describe learned random-walk kernels. As is discussed in section 1, we are often faced with classification problems with limited labeled data and a large amount of unlabeled data. These problems are often solved using similarity-propagation-based methods such as the method discussed in [77]. Random-walk based approaches are also examples of similarity-propagation based methods. My motivation for using a random-walk kernel is its ability to coerce data points in the same cluster to stay closer while making data points in different clusters to stay farther apart by propagating similarity on both labeled data and unlabeled data (see [65] and [74]). If we view a set of data points as a complete (or sparse) graph, in which the weights between data points are viewed as similarity scores, then we can make use of unlabeled data to help propagate similarity information through the whole graph. For e.g., we have a graph containing two labeled data points, i and j, and two unlabeled data points, s and t, i is highly similar to s, s is highly similar to t, and t is highly similar to j, but i and j are not very similar to each other in the given graph. After two steps of similarity propagation, i and j will become similar in a new similarity graph. When the similarity-propagation process is over, we hope that data points in the same class (having the same label) will stay relatively closer while data points in different classes (having different labels) will stay relatively farther apart (see [65], [12] and [74]). However, when the weight matrix connecting data points is not completely consistent with the labels of data points, excessive similarity propagation through the graph will harm the classification, therefore, we can use label information to guide the similarity-propagation process on the graph. This motivated me to use the label information of training data to optimize the parameter in a random-walk kernel.

A t-step random-walk kernel is generally derived from a transition matrix with a t-step random walk by normalization and symmetrization. Given a base kernel K with positive entries (in this chapter, I use profile kernels), the transition matrix P of a one-step random walk is defined as follows: let P_{ij} be the probability $P(x_i \rightarrow x_j)$, then after t steps of a random walk, the transition probability can be calculated as $P^t = (D^{-1}K)^t$, where D is a diagonal matrix with $D_{ii} = \sum_k K_{ik}$. Ideally, I want to use P^t as the kernel matrix for SVM classification. However, a kernel matrix must be a symmetric positive semi-definite matrix, therefore, I do the following manipulations to derive a kernel matrix from P^t. As is described in [74], let $L = D^{-\frac{1}{2}} K D^{-\frac{1}{2}}$, with its eigen-decomposition, $L = U \Lambda U^T$, and $\tilde{L} = U \Lambda^t U^T$, where, t denotes the exponent, and, T, denotes the transpose. Then, the new kernel corresponding to a t-step random walk is calculated as $\tilde{K} = \tilde{D}^{-\frac{1}{2}} \tilde{L} \tilde{D}^{-\frac{1}{2}}$, where $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \tilde{L}_{ii}$. We can see that the derived kernel $\tilde{K}$ relates to the transition matrix after t-steps of a random walk P^t as follows: $\tilde{K} = \tilde{D}^{-\frac{1}{2}} D^{\frac{1}{2}} P^t D^{-\frac{1}{2}} \tilde{D}^{-\frac{1}{2}}$.

A random-walk kernel based on PSI-BLAST E-values has been tried in [74] for protein remote homology detection. The challenge in random-walk kernels is how to decide the optimal number of random steps. Since random walks exploit both labeled data and unlabeled data to estimate the manifold structure of data, performing too many steps of a random walk can lead to the possibility of nearby clusters joining together, resulting in data points in different classes come closer. On the other hand, if the number of steps is too small, it can lead to a separation of data points in the same class. My goal is to find the optimum number of steps that is most consistent with the class memberships of the data points. Using the label information of training data to learn the parameters of kernel functions has been successfully adopted by researchers. Related research can be found in [77], [41] and [50]. Here, we need to learn the parameters of the random-walk kernel that achieves the goal of max-margin classification using the label information of training data. A brute-force solution to this problem results in a non-convex optimization problem, therefore, I propose using a positive linear combination of the base kernel and random-walk kernels from one step to m steps to calculate a new kernel to approximate the kernel with the optimum number of random steps by optimizing the dual objective function of the resulting SVM. I call the resulting kernel "learned random-walk kernel". Since every t-step random-walk kernel has trace n, if the base kernel also has trace n, by restricting the learned kernel to have trace n too, a positive linear combination of the base kernel and the random-walk kernels leads to a convex combination of these kernels, where n is the total number of training data and test data points. The result is the following optimization problem:

$$\begin{array}{rc} min_{\mu} max_{\alpha} & 2\alpha^T\mathbf{1} - \alpha^T(K_{tr} \otimes yy^T)\alpha, \\ s.t. & \alpha^T y = 0 \\ & \mathbf{0} \leq \alpha \leq C\mathbf{1}, \\ & K = \mu_0 \tilde{K}^0 + \sum_{k=1}^{m} \mu_k \tilde{K}^k, \\ & \sum_{k=0}^{m} \mu_k = 1, \\ & \mu_k \geq 0, \qquad k = 0, \ldots, m, \end{array} \tag{2.10}$$

where $\tilde{K}^0$ is the base kernel for deriving the learned random-walk kernel, $\tilde{K}^k$ is the random-walk kernel with a k-step random walk, and, m, is the maximal number of random steps performed. The above optimization problem is a special case of the optimization problem discussed in [41]. I follow the framework as is shown in [41], and show that the above problem is equivalent to the following quadratically constrained convex optimization problem:

$$\begin{array}{rc} min_{\alpha,t} & t, \\ s.t. & t \geq \alpha^T(\tilde{K}^k_{tr} \otimes yy^T)\alpha - 2\alpha^T\mathbf{1}, \quad k = 0, \ldots, m, \\ & \alpha^T y = 0 \\ & \mathbf{0} \leq \alpha \leq C\mathbf{1}, \end{array} \tag{2.11}$$

where tr denotes the training part of the corresponding kernel. The optimal values of parameters $\mu_k, k = 0, \ldots, m$ are exactly the dual solution to the above quadratic constrained convex optimization problem. They can be found using the standard optimization software SeDuMi [63] or MOSEK [4] which solve the primal and dual of an optimization problem simultaneously. For huge datasets, we can use SMO-like gradient-based algorithms [56] to solve the above problem. In this work, all the optimization problems were solved using MOSEK.

Theorem 1 *The optimization problem in equation 2.10 is equivalent to the optimization problem in equation 2.11.*

Proof 1 *It's easy to see that all the constraints in equation 2.10 are linear thus convex with respect to α and μ. Let $\ell = 2\alpha^T\mathbf{1} - \alpha^T(K_{tr} \otimes yy^T)\alpha$, since only K_{tr} appears in ℓ in equation 2.10, K_{tr} is the only part we need from K to solve equation 2.10. ℓ is linear thus convex with respect to μ. The Hessian of ℓ with respect to α is $-(K_{tr} \otimes yy^T)$, which is negative semi-definite, hence, ℓ is concave with respect to α. And ℓ is continuous with respect to α and μ. Therefore, we have the following equations:*

$$\begin{aligned}
& min_{\mu:\ \mu\geq\mathbf{0},\ \sum_{k=0}^m \mu_k=1}\ max_{\alpha:\ \alpha^T y=0,\ \mathbf{0}\leq\alpha\leq C\mathbf{1}} \\
& \qquad 2\alpha^T\mathbf{1} - \alpha^T[(\sum_{k=0}^m \mu_k \tilde{K}_{tr}^k) \otimes yy^T]\alpha \\
= {} & max_{\alpha:\ \alpha^T y=0,\ \mathbf{0}\leq\alpha\leq C\mathbf{1}}\ min_{\mu:\ \mu\geq\mathbf{0},\ \sum_{k=0}^m \mu_k=1} \\
& \qquad 2\alpha^T\mathbf{1} - \alpha^T[(\sum_{k=0}^m \mu_k \tilde{K}_{tr}^k) \otimes yy^T]\alpha \\
= {} & max_{\alpha:\ \alpha^T y=0,\ \mathbf{0}\leq\alpha\leq C\mathbf{1}}\ min_{\mu:\ \mu\geq\mathbf{0},\ \sum_{k=0}^m \mu_k=1} \\
& \qquad \sum_{k=0}^m \mu_k[2\alpha^T\mathbf{1} - \alpha^T(\tilde{K}_{tr}^k \otimes yy^T)\alpha] \\
= {} & max_{\alpha:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1}} min_k[2\alpha^T\mathbf{1} - \alpha^T(\tilde{K}_{tr}^k \otimes yy^T)\alpha] \\
= {} & max_{\alpha,t:\ \ \alpha^T y=0,\ \ \mathbf{0}\leq\alpha\leq C\mathbf{1},t\leq 2\alpha^T\mathbf{1}-\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha} \quad t \\
= {} & min_{\alpha,t:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1},\ \ t\geq\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha-2\alpha^T\mathbf{1}} \quad t
\end{aligned} \tag{2.12}$$

The first equality holds due to the special property of ℓ described above according to [11]. The second and third equalities hold due to the properties of the simplex defined by μ. The last two equalities hold due to the rewriting of the optimization problems in different formats. The last equality shows that the optimization problem in equation 2.10 is equivalent to the optimization problem in equation 2.11.

As is described in [37], the ideas of random walks and diffusion are closely related. Given a kernel matrix K, we can view it as a similarity matrix and compute the graph laplacian as $Q = D - K$, where D is a diagonal matrix described in this section. Instead of taking the form of the t-th power of the transition matrix P as in random-walk kernels, a diffusion kernel $K^{diffuse}$ takes a form of the matrix exponential of Q:

$$\begin{aligned} K^{diffuse} &= e^{\beta Q} = lim_{n->\infty; n \in \mathcal{N}} \;\; (I + \frac{\beta Q}{n})^n \\ &= I + \beta Q + \frac{\beta^2}{2} Q^2 + \ldots + \frac{\beta^t}{t!} Q^t + \ldots \\ &= \sum_i v_i e^{\beta \lambda_i} v_i^T, \end{aligned} \tag{2.13}$$

where β is a real parameter to control the diffusion, which is analogous to the minus inverse squared variance parameter in Gaussian kernels, I is an identity matrix, $\mathcal{N}$ is the integer set, and, v_i and λ_i are the i-th eigenvalue and eigenvector of K respectively. The first line in the above equation can be interpreted as a random walk with an infinite number of infinitesimally small steps. In this chapter, I compute diffusion kernels based on profile kernels, and compare their performance to that of learned random-walk kernels shown in the experimental results section.

The computation of both a random-walk kernel and a diffusion kernel requires the eigen-decomposition of a base kernel, which has a worst-case time complexity $O(n^3)$. Computing the learned random-walk kernel described above requires solving in addition, the quadratically constrained convex optimization problem in equation 2.11, which has a worst-case time complexity $O(mn_{tr}^3)$ using an interior-point method, where n_{tr} is the number of training data points.

2.4.2 Learned empirical-map kernel

An empirical-map kernel based on PSI-BLAST E-values has been applied in the analysis of biological sequence data with reasonably good performance [74]. I compared the Leave-One-Out nearest neighbor classification errors on protein sequence classification produced using PSI-BLAST E-values to those produced using a normalized profile kernel, and found that the normalized profile kernel captures the neighborhood similarity much better than the PSI-BLAST E-values. This motivated me to use the normalized profile kernel to derive an empirical-map kernel for biological sequence classification. In [74], the authors report their best result after tuning a scaling parameter, however, they do not provide a method for calculating this parameter. In contrast, here I propose three approaches for calculating the scaling parameter in the empirical-map kernel.

Given a similarity matrix S where S_{ij} is the similarity score between data points X_i and X_j, the empirical map for data point x is defined as:

$$\Phi^{emp}(x) = (e^{-\lambda S(x,X_1)}, e^{-\lambda S(x,X_2)}, \ldots, e^{-\lambda S(x,X_P)})^T, \tag{2.14}$$

where P is the number of available data points including both labeled data points and unlabeled data points. The empirical-map kernel is defined as $K_{\lambda,ij} = \Phi^{emp}(X_i)^T \Phi^{emp}(X_j)$. The key to deriving the optimal empirical-map kernel is calculating the scaling parameter λ.

In this chapter, I use the normalized profile kernel matrix as the similarity matrix. Given a profile kernel matrix K^{prof}, I normalize it such that every sequence has a unit feature vector (the norm is 1) as follows:

$$K^{prof,norm} = \Delta^{-\frac{1}{2}} K^{prof} \Delta^{-\frac{1}{2}}, \tag{2.15}$$

where Δ is a diagonal matrix and $\Delta_{ii} = K^{prof}_{ii}$. Then the empirical-map kernel is given by:

$$K^{emp}_{\lambda,ij} = \sum_{k=1}^{P} e^{-\lambda(K^{prof,norm}_{ik} + K^{prof,norm}_{jk})}. \tag{2.16}$$

I normalize K^{emp} again so that every sequence has a unit feature vector, giving the following normalized empirical-map kernel:

$$K^{emp,norm}_{\lambda,ij} = \frac{\sum_{k=1}^{P} e^{-\lambda(K^{prof,norm}_{ik} + K^{prof,norm}_{jk})}}{\sqrt{\sum_{k=1}^{P} e^{-2\lambda K^{prof,norm}_{ik}} \sum_{k=1}^{P} e^{-2\lambda K^{prof,norm}_{jk}}}}. \tag{2.17}$$

One way to calculate λ is by cross validation, however, it is computationally expensive to search over a long list of candidate values and often this method fails to produce good values of λ. Alternatively we can substitute K with $K_{\lambda}^{emp,norm}$ in Equation 2.2, perform the maximization with respect to α and then perform the minimization with respect to λ. However, this problem is non-convex with respect to λ, and, each iteration for calculating the optimal value of α with λ fixed involves a quadratic programming problem. Instead, in this chapter, I propose three different approaches to calculate λ. The first approach calculates λ by maximizing the Kernel Alignment Score (KAS) [14]. Given the labels of training data, the optimal kernel is given by $K^{opt} = YY^T$. I calculates λ by maximizing the alignment score as follows:

$$KAS = Trace(K^{emp,normT} K^{opt}) / \sqrt{Trace(K^{emp,norm^2}) Trace(K^{opt^2})}. \tag{2.18}$$

The second approach calculates λ in a way that encourages the similarities between data points within a class to be as large as possible. Given a kernel matrix K, I calculate the probability of sequence i and sequence j being in the same class as, $P_{ij} = \frac{K_{ij}}{\sum_{k=1}^{\ell} K_{ik}}$, where ℓ is the size of the labeled training set. I calculate the probability matrix $P_{\lambda}^{emp,norm}$ using $K_{\lambda}^{emp,norm}$, and ,P^{opt} using K^{opt}. To enforce the class-dependent constraint, I minimize the following KL-divergence between P^{opt} and $P_{\lambda}^{emp,norm}$:

$$KL = \sum_{ij} P_{ij}^{opt} log[P_{ij}^{opt} \quad / \quad P_{\lambda,ij}^{emp,norm}]. \tag{2.19}$$

In the third approach, λ is chosen such that the normalized empirical-map kernel in equation 2.17 corresponds to a good metric for defining a neighborhood consistent with the labels of the labeled data, i.e., I choose λ to minimize the Leave-One-Out Nearest Neighbor classification error over the labeled dataset. To limit the search space, I use the optimal λs found by the first approach and the second approach as reference values, and I always take the smallest λ when there are several local minima of λ achieving equally good classification error.

All the three approaches of computing λ described above have a worst-case time complexity $O(n_{tr}^2)$. The third approach is often the most stable and often works best in practice, therefore, I suggest using this approach as the default approach for computing

λ in the empirical-map kernel for possible future applications. Once λ is decided, the worst-case time complexity for computing an empirical-map kernel is $O(n^3)$ using traditional matrix multiplications, but this time complexity is reduced to $O(n^{2.376})$ using advanced matrix multiplication algorithms in [13]. In contrast, computing an improved random-walk kernel has a worst-case time complexity $O(n^3)$ dominated by the eigen-decomposition of the base kernel matrix.

2.4.3 Experimental results based on learned random-walk and empirical-map kernels

Since the optimization procedure for calculating the convex combination coefficients for combining random-walk kernels is highly dependent on labels, I adopted the following approach: prior to training the SVM, I added to the positive training set labeled as positive, close homologs of the positive training data in the unlabeled set found by PSI-BLAST with E-value less than 0.05. When training the SVM based on random-walk kernels with a fixed number of random steps, diffusion kernels, and empirical-map kernels, I also used unlabeled data as discussed above. The improved random-walk kernel and the empirical-map kernel are based on the two profile kernels which produced the top 2 results on SCOP in [40]. Both profile kernels were obtained by setting the k-mer length to 5 and the parameter σ to 7.5. However, the best profile kernel was obtained using the PSI-BLAST profile trained up to 5 search iterations while the second best profile kernel was obtained using the PSI-BLAST profile trained up to 2 search iterations. The profile kernels were normalized to have trace n as in equation 2.15 before they were used for the SVM classification and the calculation of improved random-walk kernels. In [42], [40] and [74], it has been shown that normalized mismatch string kernels including the profile kernels are very effective for protein classification. And in the experiments, the maximum number of steps m of random walks for the improved random-walk kernel was set to 6 (when it was set to 7, 8, 9, or 10, I saw an increasing computational time but no significant improvement in the results over that of $m = 6$). To compare improved random-walk kernels to diffusion kernels, the free parameter β in $K^{diffuse}$ was decided by 5-fold cross validation. When using Leave-One-Out Nearest Neighbor classification error to decide λ, I used the values found by the first and the second approaches as reference and limit the search space to regions around the reference values. I used a hard-margin SVM to identify protein remote homology (the free parameter C in the SVM was set to infinity, which has been shown to be very effective for protein classification [74]).

Table 2.1 shows the ROC$_{50}$ scores produced by the random-walk kernels with the best fixed number of random steps, the scores produced by the improved random-walk kernels, and, the scores produced by the diffusion kernels based on the best and second best profile kernels. It clearly shows that the improved random-walk kernels have much better performance than the profile kernels and the diffusion kernels. Moreover, based on the best profile kernel, the random-walk kernel with the best fixed number of random steps (2 steps) has a worse performance than the base kernel; and based on both profile kernels, the diffusion kernels have a worse performance than the two base kernels. The poor performance of the diffusion kernels here is probably due to the very limited positive labeled data and the non-optimality of the parameter β decided by cross validation. From Table 2.1, I conclude that the convex combination of random-walk kernels is an effective way of using random walks. Table 2.2 lists the ROC$_{50}$ scores by the empirical-map kernels with λ calculated using three different approaches. From Table 2.2, I see that the first and the second approaches have similar performance, while the third approach outperforms these two. In the remainder of this chapter, the empirical-map kernel is taken in reference to the kernel with λ calculated using the third approach except where explicitly stated.

Table 2.3 gives the overall mean ROC$_{50}$ scores over 54 protein families obtained by several previous representative approaches and my improved random-walk kernels and empirical-map kernels. It can be clearly seen that previous approaches except for the

Random-Walk Kernel (RWK)	Overall Mean ROC_{50}
2-step RWK using the second best profile kernel	0.847
improved RWK using the second best profile kernel	0.867
diffusion kernel using the second best profile kernel	0.746
the second best profile kernel (base kernel)	0.824
2-step RWK using the best profile kernel	0.862
improved RWK using the best profile kernel	0.901
diffusion kernel using the best profile kernel	0.790
the best profile kernel (base kernel)	0.874

Table 2.1: Overall mean ROC_{50} scores over 54 protein families corresponding to different random-walk kernels and diffusion kernels. 2-step random-walk kernels work best on the SCOP dataset among all the random-walk kernels with a fixed number of random steps. We see that improved random-walk kernels outperform random-walk kernels with the best fixed number of random steps and diffusion kernels. Using Wilcoxon Matched-Pairs Signed-Rank test, we have the following p-values: Based on the second best profile kernel, the p-value for the ROC_{50} score difference between the learned random-walk kernel and the base kernel was 1.52×10^{-4}, and the p-value for the pair between the learned random-walk kernel and the second best profile kernel with 2 steps of random walks was 0.14, and the p-value for the pair between the learned random-walk kernel and the diffusion kernel was 1.39×10^{-7}. However, the p-value for the ROC_{50} score difference between the second best profile kernel with 2 steps of random walks and the base kernel was 0.06. Based on the best profile kernel, the p-value for the ROC_{50} score difference between the learned random-walk kernel and the base kernel was 3.30×10^{-3}, and the p-value for the pair between the learned random-walk kernel and the best profile kernel with 2 steps of random walks was 1.27×10^{-2}, and the p-value for the pair between the learned random-walk kernel and the diffusion kernel was 8.84×10^{-7}. However, the p-value for the ROC_{50} score difference between the best profile kernel with 2 steps of random walks and the best profile kernel was 0.62.

Empirical-Map Kernels	Overall Mean ROC_{50}
using the second best profile kernel[1]	0.862
using the second best profile kernel[2]	0.866
using the second best profile kernel[3]	0.878
the second best profile kernel (base kernel)	0.824
using the best profile kernel [1]	0.904
using the best profile kernel [2]	0.900
using the best profile kernel [3]	0.911
the best profile kernel (base kernel)	0.874

Table 2.2: Overall mean ROC_{50} scores over 54 protein families corresponding to different empirical-map kernels with λ calculated using three different approaches. '$[i]$' denotes the empirical-map kernel with λ calculated using the i-th approach, $i = 1$, 2, or 3. From this table, we find that the third approach for deciding λ is the best. In contrast, the cross validation procedure (see [50] for details) to choose λ gives overall mean ROC_{50} scores 0.848 based on the second best profile kernel and 0.891 based on the best profile kernel, and the searching procedure is very slow.

Methods	Overall Mean ROC_{50}
eMOTIF (see reference [9] and [40])	0.247
SVM-pairwise [PSI-BLAST] (see reference [46] and [40])	0.533
spectrum-kernel [PSI-BLAST] (see reference [42])	0.545
neighborhood (see reference [74])	0.699
the second best profile kernel (**the second best result)**	0.821
the best profile kernel (**the best result)**	0.874
improved RWK using the second best profile kernel	0.867
empirical-map kernel using the second best profile kernel	0.878
improved RWK using the best profile kernel	0.901
empirical-map kernel using the best profile kernel	0.911

Table 2.3: Overall mean ROC_{50} scores over 54 protein families corresponding to different kernels. Here the empirical-map kernels refer to the kernels with λ calculated using the third approach. In this table, the top rows show the results produced by several previous representative approaches and the best published results; the middle rows show my results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the second best profile kernel; and the bottom rows show my results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the best profile kernel. Using Wilcoxon Matched-Pairs Signed-Rank Test, we have the following p-values: Based on the second best profile kernel, the p-value for the ROC_{50} score difference between the improved random-walk kernel and the base kernel was 1.52×10^{-4}, and the p-value for the pair between the empirical-map kernel and the base kernel was 1.67×10^{-4}. Based on the best profile kernel, the p-value for the pair between the improved random-walk kernel and the base kernel was 3.30×10^{-3}, and the p-value for the pair between the empirical-map kernel and the base kernel was 3.05×10^{-2}.

Methods	ROC_{50} on the hardest protein family
eMOTIF (see reference [9] and [40])	0.000
SVM-pairwise [PSI-BLAST] (see reference [46] and [40])	0.000
spectrum-kernel [PSI-BLAST] (see reference [42])	0.000
neighborhood (see reference [74])	0.000
the second best profile kernel (**the second best result)**	0.045
the best profile kernel (**the best result)**	0.122
improved RWK using the second best profile kernel	0.454
empirical-map kernel using the second best profile kernel	0.455
improved RWK using the best profile kernel	0.509
empirical-map kernel using the best profile kernel	0.903

Table 2.4: The ROC_{50} scores on the most difficult protein family **Glutathione S-transferases, N-terminal domain** corresponding to different kernels. Here the empirical-map kernels refer to the kernels with λ calculated using the third approach. In this table, the top rows show the results produced by several previous representative approaches and the best published results; the middle rows show my results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the second best profile kernel; and the bottom rows show my results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the best profile kernel.

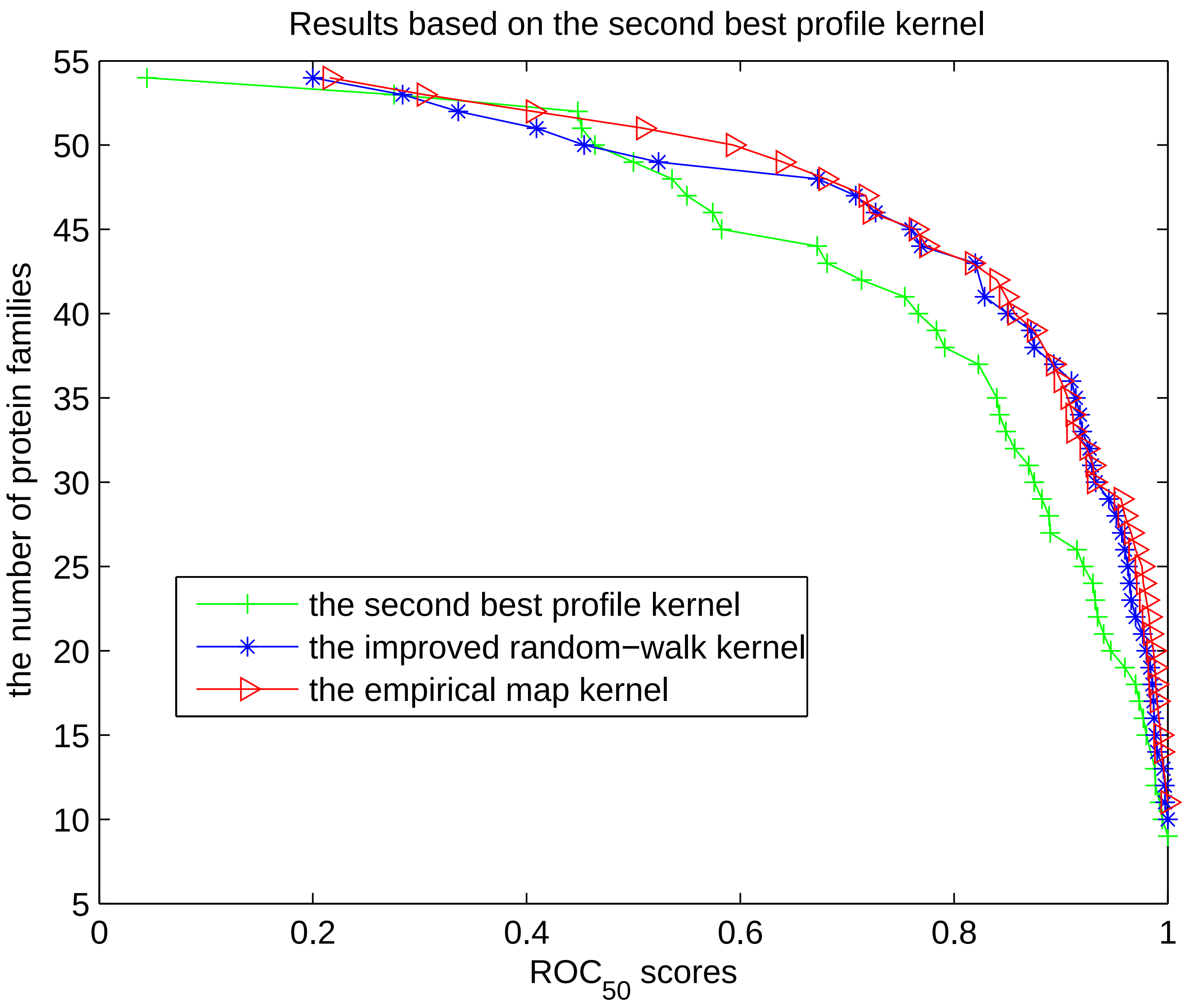


Figure 2.2: The number of protein families with ROC$_{50}$ scores above different thresholds for different kernels using the second best profile kernel.

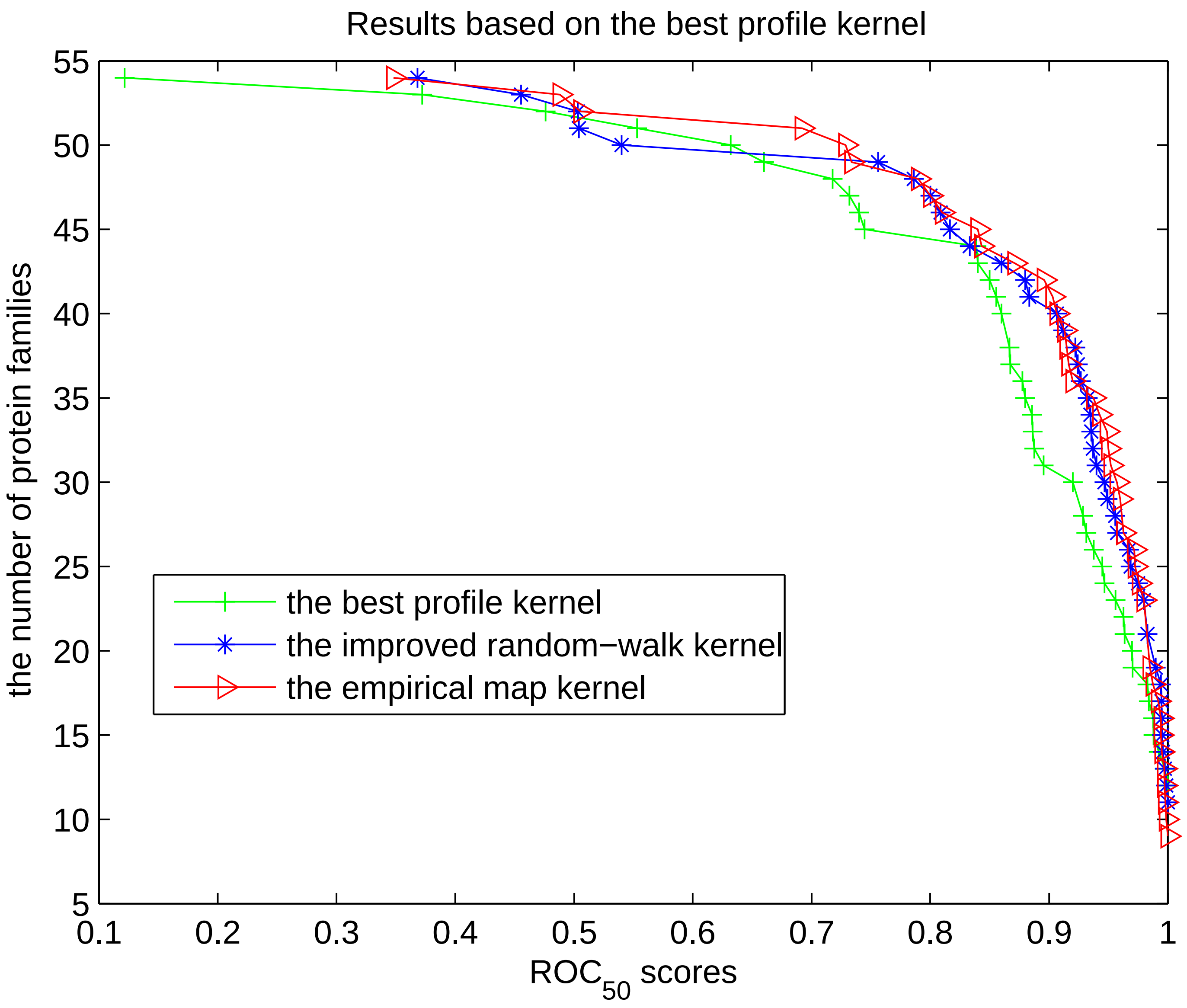


Figure 2.3: The number of protein families with ROC$_{50}$ scores above different thresholds for different kernels using the best profile kernel.

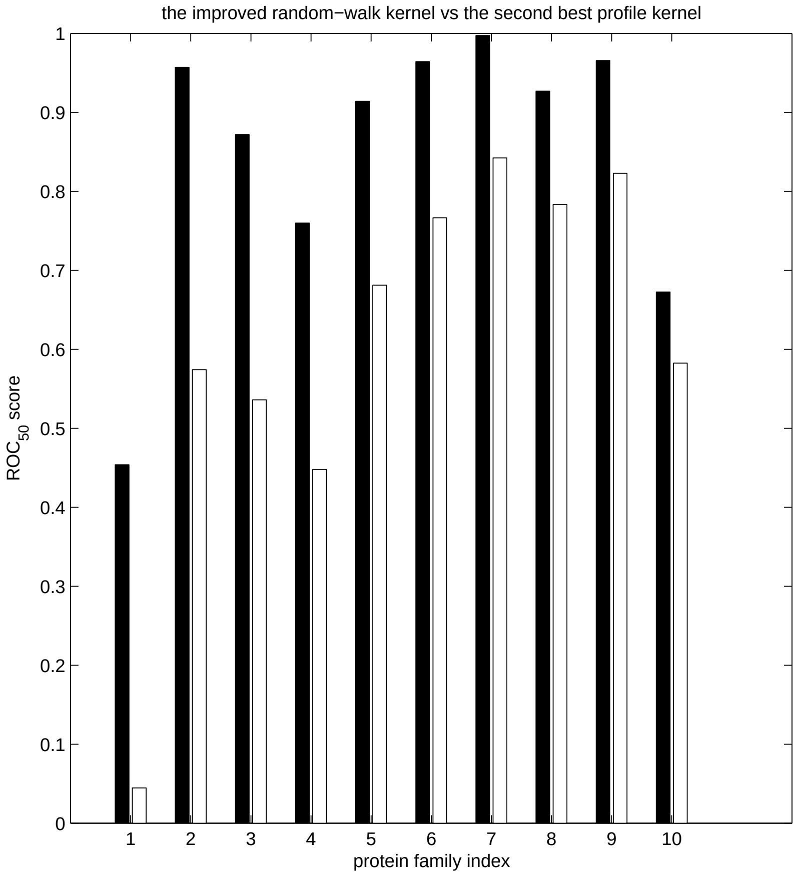


Figure 2.4: The results obtained using the second best profile kernel: the top 10 largest improvement in ROC$_{50}$ scores out of 54 protein families for the improved random-walk kernel based on the second best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the improved random-walk kernel based on the second best profile kernel, and the right white bar corresponds to the base kernel.

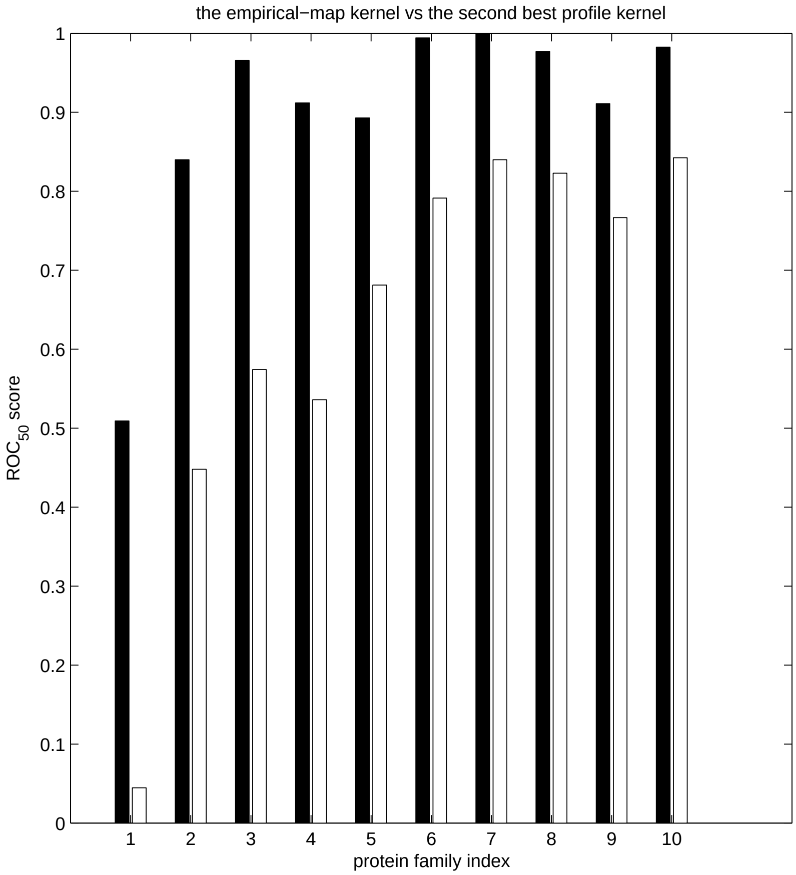


Figure 2.5: The results obtained using the second best profile kernel: the top 10 largest improvement in ROC_{50} scores out of 54 protein families for the empirical-map kernel based on the second best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the empirical-map kernel based on the second best profile kernel, and the right white bar corresponds to the base kernel.

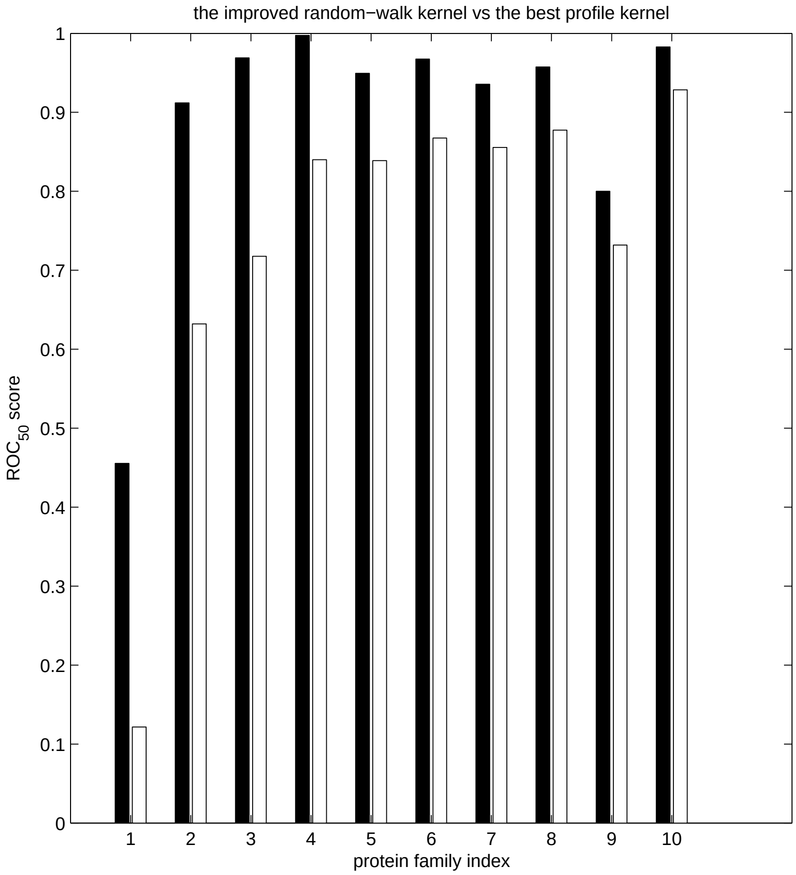


Figure 2.6: The results obtained using the best profile kernel: the top 10 largest improvement in ROC$_{50}$ scores out of 54 protein families for the improved random-walk kernel based on the best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the improved random-walk kernel based on the best profile kernel, and the right white bar corresponds to the base kernel.

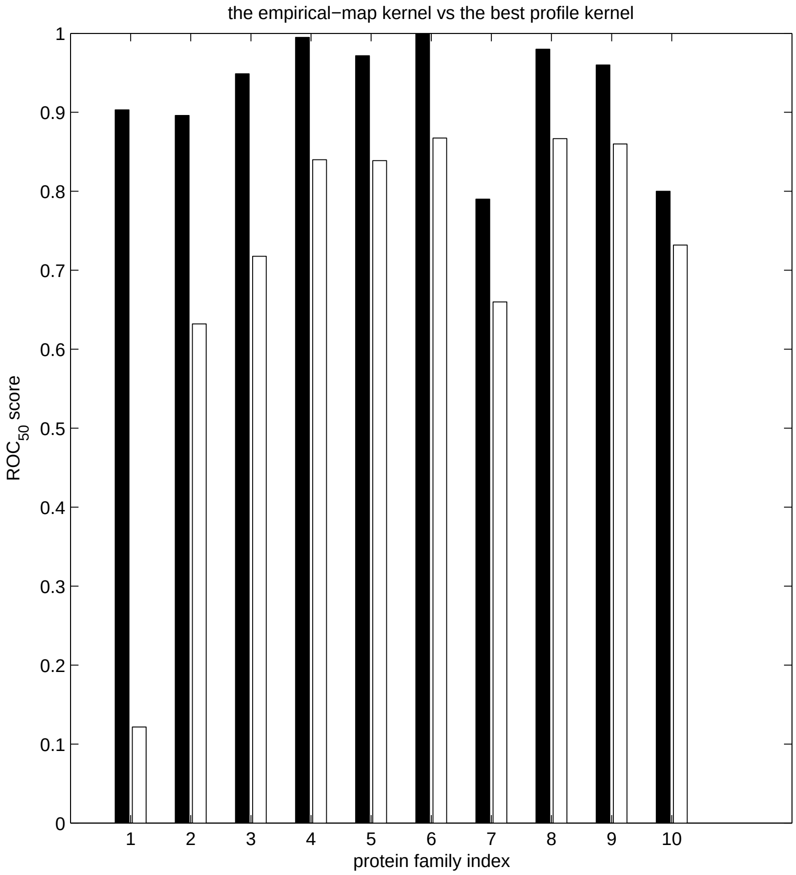


Figure 2.7: The results obtained using the best profile kernel: the top 10 largest improvement in ROC_{50} scores out of 54 protein families for the empirical-map kernel based on the best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the empirical-map kernel based on the best profile kernel, and the right white bar corresponds to the base kernel.

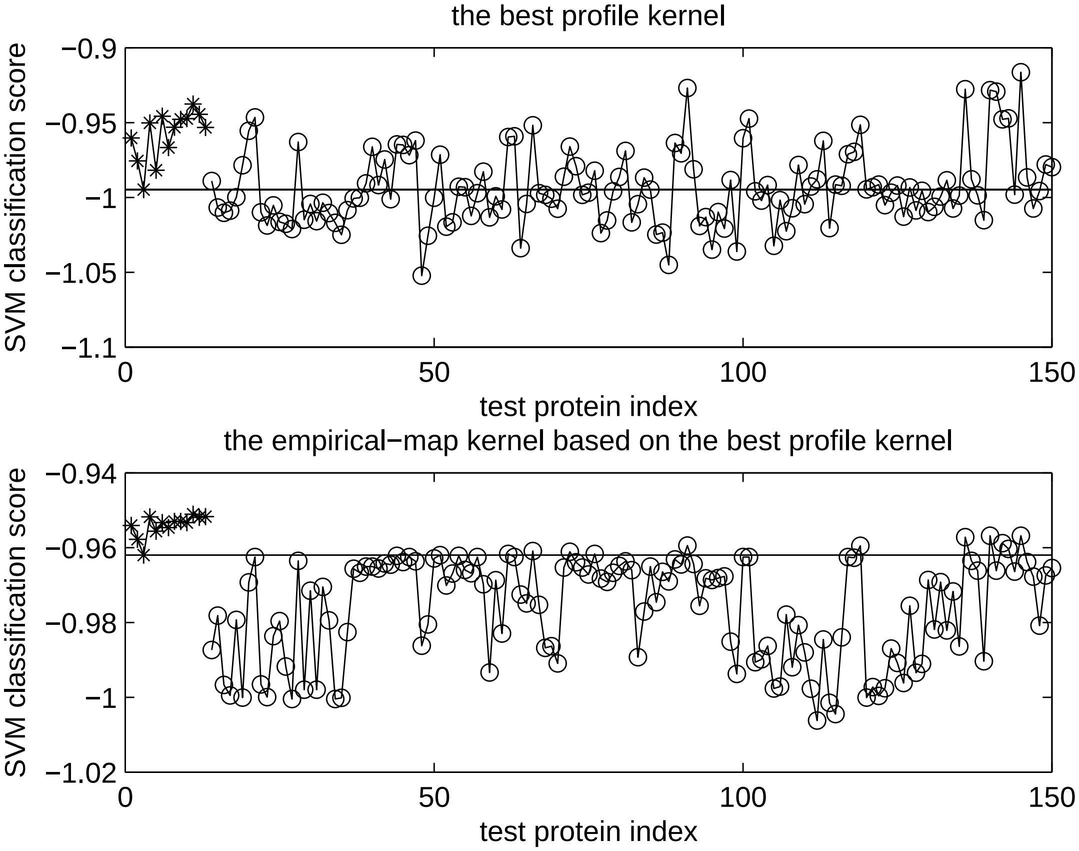


Figure 2.8: The SVM classification scores calculated using the best profile kernel and the empirical-map kernel based on the best profile kernel on the most difficult protein family **Glutathione S-transferases, N-terminal domain**. The top plot corresponds to the best profile kernel and the bottom plot corresponds to the empirical-map kernel. In both plots, the stars represent all positive test proteins and the circles represent some negative test proteins. In each plot, the horizontal line sits at the smallest value of the classification scores for all the positive test proteins. The top plot shows that if we want to classify most of the positive test proteins correctly by setting an appropriate threshold, there will be a lot of false positives; however, the bottom plot clearly shows that we can almost classify all the positive test proteins correctly by setting an appropriate threshold while only introducing a very small number of false positives.

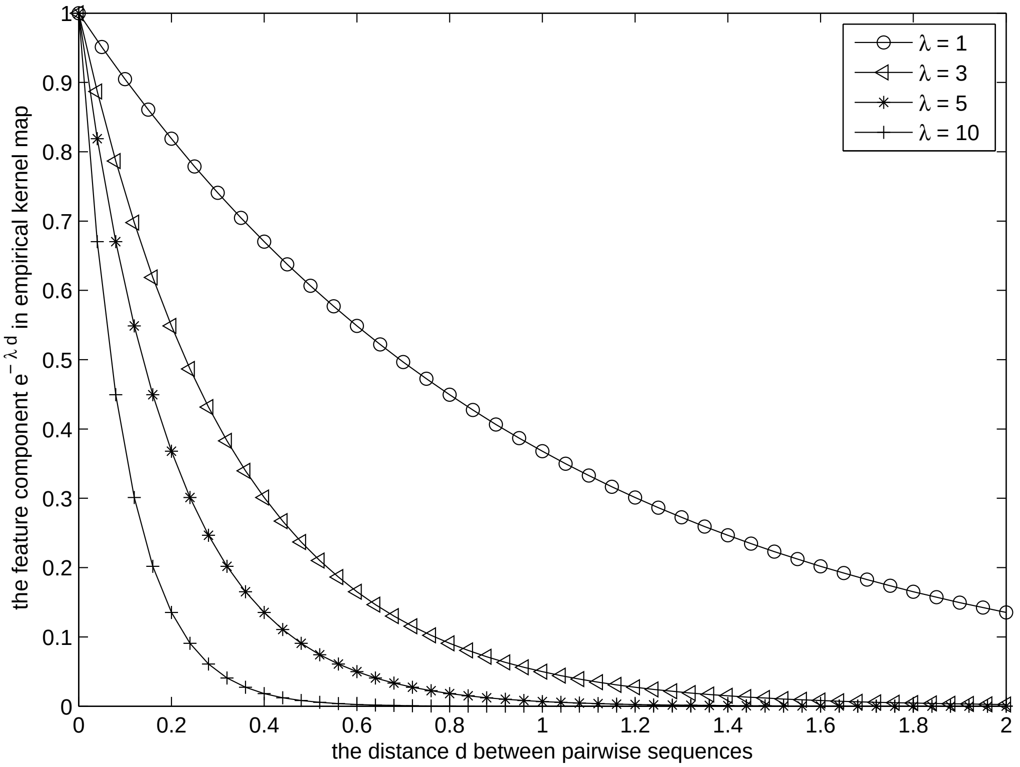


Figure 2.9: This figure shows how the feature component $exp(-\lambda d)$ in the empirical-map varies with the distance d between pairwise sequences for different λ values. When the normalized profile kernels are used to calculate the distances between pairwise sequences, the distances are always between 0 and 2.

profile kernels have low ROC$_{50}$ scores, below 0.70. The two profile kernels produce ROC$_{50}$ scores above 0.80. In contrast, my improved random-walk kernels and empirical-map kernels produce ROC$_{50}$ scores above 0.90. Because I have only a few positive test cases but, hundreds, or even thousands of negative test cases in most of the 54 experiments, the mean ROC$_{50}$ score produced by a random predictor is close to 0. We can see that all the approaches listed in the table have much better performance than a random predictor.

Table 2.4 shows the ROC$_{50}$ scores for the most difficult protein family **Glutathione S-transferases, N-terminal domain** on which all the previous approaches produced very poor performance while my approaches performed well. In the experiment for this protein family, I have 13 positive test proteins and 927 negative test proteins, therefore, the ROC$_{50}$ score produced by a random predictor should be close to 0, while the profile kernels and my proposed kernels performed much better than a random predictor.

Figures 2.2-2.8 show my results in detail. Figures 2.2-2.3 show the number of protein families that score above the different ROC$_{50}$ threshold values for my kernels and the top two profile kernels (note that they are not ROC_{50} curves but the summarization of all the ROC_{50} scores). Figures 2.4-2.7 show the ten largest improvements in ROC$_{50}$ scores for my kernels over the top two best profile kernels. Figure 2.8 compares the SVM classification scores calculated using the best profile kernel and the scores calculated using the empirical-map kernel based on the best profile kernel on the most difficult protein family **Glutathione S-transferases, N-terminal domain**. On this special family, I tried to find the obvious changes of cluster patterns by visualizing the kernel matrices, but, it turned out that the improvements of ROC$_{50}$ scores were due to subtle changes of some kernel entries, which are hard to capture by eyes. Besides, I found that, based on the best profile kernel, the empirical-map kernel resulted in much more support vectors than the improved random-walk kernel and the base kernel. Out of 1943 training protein sequences, the empirical-map kernel with the learned $\lambda = 3$ resulted in 1901 support vectors, while the random-walk kernel and the base kernel resulted in 1026 and 1083 support vectors respectively. In the empirical-map kernel, $\lambda = 3$ allows a lot of weak pairwise sequence similarities contributing to the construction of the kernel, moreover, almost every training sequence for this protein family were learned to be a support vector using this kernel, therefore, the drastic improvement given by the empirical-map kernel for this protein family is probably due to the combination of a lot of weak pairwise sequence similarities, which might correspond to the combination of a lot of short sequence motifs. In the next section, I will discuss how to extract biologically meaningful sequence motifs that are crucial for determining each positive test protein's superfamily membership using the random-walk kernel to rank protein sub-sequences.

To determine whether the improvements obtained by the improved random-walk kernels and the empirical-map kernels are statistically significant, I performed Wilcoxon Matched-Pairs Signed-Ranks Tests on the differences between paired kernels. All the resulting p-values were below 0.05. The resulting p-value for the ROC_{50} score difference between the improved random-walk kernel based on the second best profile kernel and the base kernel was 1.52×10^{-4}. The p-value for the pair between the improved random-walk kernel based on the best profile kernel and the base kernel was 3.30×10^{-3}. The p-value for the pair between the empirical-map kernel based on the second best profile kernel and the base kernel was 1.67×10^{-4}, and the p-value for the pair between the empirical-map kernel based on the best profile kernel and the base kernel was 3.05×10^{-2}.

However, based on both the profile kernels, the p-values for the ROC_{50} score differences between the empirical-map kernels and the improved random-walk kernels are both greater than 0.15, which are not statistically significant.

2.4.4 Discussions about learned random-walk and empirical-map kernels

In this chapter, I proposed two kernel learning approaches for protein remote homology detection based solely on protein sequence data. One approach approximates the random-walk kernel with the optimal number of random steps by calculating a convex combination of random-walk kernels with different numbers of random steps, and, the other approach uses profile kernels to derive empirical-map kernels with the scaling parameter λ calculated using a principled approach. The first approach reduces to a convex optimization problem avoiding concerns of local minima. The second approach initializes the value of λ in the empirical-map kernels by minimizing the KL divergence and maximizing the Kernel Alignment Score, and, then refining the value of λ by minimizing the Leave-One-Out nearest neighbor classification errors. It is a robust approach. I ran the procedure for calculating λ several times, each time obtaining the same refined value of λ on each protein family.

Both approaches make use of a large number of pairwise sequence similarities and unlabeled data to derive new kernels, which corresponds to new similarity metrics for pairwise sequences. In the first approach, pairwise sequence similarities contribute to defining the transition probability matrix for the random walks. The convex optimization procedure induces the new kernel to reflect the manifold structure of the sequences that is optimally consistent with the labeled training sequences.

In the second approach, the scaling parameter λ plays the role of selecting features in a soft way for the empirical-map kernel. When λ is small, small pairwise sequence similarities contribute weakly to the construction of the kernel based on the empirical map. When λ is large, only large pairwise sequence similarities contribute to the construction of the new kernel. Figure 2.9 illustrates how the feature component in the empirical map varies with distance between pairwise sequences for different λ values (note that the distances between pairwise sequences based on the normalized profile kernels are between 0 and 2). All the three procedures in the second approach make use of the label information of training sequences to calculate λ in order to achieve good separability between positive sequences and negative sequences.

The experimental results on protein remote homology detection show that the improved random-walk kernels and the empirical-map kernels proposed here produce strikingly better results than previous methods, including the best approaches for solving this problem proposed to date. Out of 54 protein families, the best profile kernel produced ROC$_{50}$ scores above 0.90 for 30 families while the empirical-map kernel based on the best profile kernel produced ROC$_{50}$ scores above 0.90 for 41 families. On one hand, this shows the effectiveness of the empirical map kernel, and on the other hand, it shows that the base kernel (the best profile kernel) has very good performance producing almost perfect results on more than half of the protein families. From Figures 2.4-2.7, I find that my approaches give more than 10% improvement over the base kernels on many difficult protein families. In particular, on the most difficult protein family **Glutathione S-transferases, N-terminal domain** on which all the previous approaches failed to produce useful results (ROC$_{50}$ scores of zero or close to zero), my approaches produced very good results.

My approaches are general and are readily applicable to other biological classification problems such as Transcription Factor Binding Site prediction and gene function prediction etc. The approaches described here can also be applied to non-biological problems such as document classification, handwritten digit classification and face recognition etc where kernels are constructed on texts and images instead of on biological data.

2.5 Protein sequence motif discovery using learned random-walk kernels

To identify sequence motifs making important contributions to discriminating the remote homology membership of a protein sequence x, I calculate the j-th positional contribution to the positive classification of sequence x using the following equation:

$$\sigma(x[j]) = max(\sum_{i=1}^{n_{tr}} \alpha_i K(i, x[j-k+1 : j+k-1]), 0), \tag{2.20}$$

where i indexes training sequences, $x[j-r+1 : j+r-1])$ represents a subsequence window with radius r centered at position j, $K(i, x[j-r+1 : j+r-1])$ represents the contribution to the kernel entry $K(i,x)$ made by $x[j-r+1 : j+r-1])$, K is the learned random-walk kernel, and α is the dual parameter of the SVM based on K. However, the mapping from the base kernel which is a profile kernel to the learned random-walk kernel is not linear, so there is no closed-form solution to calculate $K(i, x[j-r+1 : j+r-1])$. Instead, I resort to the following algorithm to calculate $\hat{K}(i, x[j-r+1 : j+r-1])$, which is an approximation to $K(i, x[j-r+1 : j+r-1])$.

Algorithm 1 *The algorithm for computing positional contribution to positive classification $\sigma(x[j])$*

Input: sequence profiles P, sequence x, position j, radius r, μ, α, profile kernel matrix K^{prof}, and learned random-walk kernel matrix K.

Output: $\hat{K}(\cdot, x[j-r+1 : j+r-1])$ and $\sigma(x[j])$

1. Use P to compute the contribution to the profile-kernel matrix made by $x[j-r+1 : j+r-1]$, denoted by M, which is symmetric and has non-zero entries only in the row and the column corresponding to sequence x.

2. Normalize K^{prof} and M using diagonal entries in K^{prof}. $\tilde{K}^0_{ij} = \frac{K^{prof}_{ij}}{\sqrt{K^{prof}_{ii} K^{prof}_{jj}}}$, and $\tilde{M}_{ij} = \frac{M_{ij}}{\sqrt{K^{prof}_{ii} K^{prof}_{jj}}}$.

3. Compute new learned random-walk kernel matrix K' based on new base kernel matrix $(\tilde{K}^0 - \tilde{M})$ and input combination coefficient μ.

4. $\hat{K}(\cdot, x[j-r+1 : j+r-1]) = K(\cdot, x) - K'(\cdot, x)$.

5. Replace K *with* $\hat{K}$ *in equation 2.20 to compute* $\sigma(x[j])$.

I can use the above algorithm to compute the positional contribution score to positive classification for both positive training and test sequences. Then I can rank the positions by the positional contribution score σ, and the top ranked positions, which occupy above 90% of the total positional contribution score mass, can be regarded as essential regions discriminating the remote homology membership of the considered sequences.

2.5.1 Experimental results on protein sequence motif discovery

In this subsection, I present the results of motif discovery using the SVMs based on the learned random-walk kernels. I set the radius parameter r in section 2.5 to 5. My experimental results show that the important discriminative motifs for a protein sequence often lie in the regions connecting or bordering at common structure motifs such as α-helixes and β-sheets. This completely makes sense in biology. Common structure motifs occur frequently in all kinds of protein sequences, while the regions connecting or bordering at these common motifs represent different ways of assembling these common structures, which should be more important identifiers of remote homology than other regions.

In the following, I will perform a case study for the identification of super-family **ConA-like lectins/glucanases**. On this super-family, the ROC_{50} scores produced by the state-of-the-art profile kernel and the random-walk kernel with 2 random steps are, respectively, 0.63 and 0.74, while the the ROC_{50} score produced by the learned random-walk kernel is 0.93.

Figure 2.10 shows the distributions of positional contribution scores to the positive classification of 4 positive training sequences with PDB id 1a8d-1, 3btaa1, 1epwa1, and

1kit-2. This figure shows that a small fraction of positions, which are peaky positions in the figure, have much higher scores, meaning that they are much more important than other positions for the identification of remote homology. Figure 2.11 shows the distribution of the positional contribution scores to positive classification of another positive training sequence with PDB id 2nlra. The blue positions are local peak positions, and the red positions correspond to the top 10 highest positions. Figure 2.12 shows the motif of sequence 2nlra annotated by PDB, and Figure 2.13 shows the motif predicted by the SVM based on the learned random-walk kernel, in which the blue and red positions in Figure 2.11 are also marked blue and red respectively here. From this figure, we can see that the blue and red regions lie either in the center of a standard structure motif, which may represent a standard motif, or lie in the regions connecting or bordering at standard motifs, which may act as bridge motifs.

Figures 2.14, 2.15, 2.16 and 2.17 show the results for a positive test sequence with PDB id 1c1l. In details, Figure 2.14 shows the distribution of the positional contribution scores to positive classification of sequence 1c1l, and the red positions correspond to the top 15 highest positions. Figure 2.15 shows the ROC$_{50}$ scores of predicting the superfamily of sequence 1c1l by training SVMs on learned random-walk kernels by respectively removing the subsequence window with radius 5 centered at each position. We can see that the results in Figure 2.15 are consistent with the positional contribution scores in Figure 2.14. Figure 2.16 shows the motif of this sequence annotated by PDB, and Figure 2.17 shows the motif predicted by the learned random-walk kernel. The red regions correspond to the red positions in Figure 2.14. Again, we can see that the red regions represent standard motifs or act as bridge motifs.

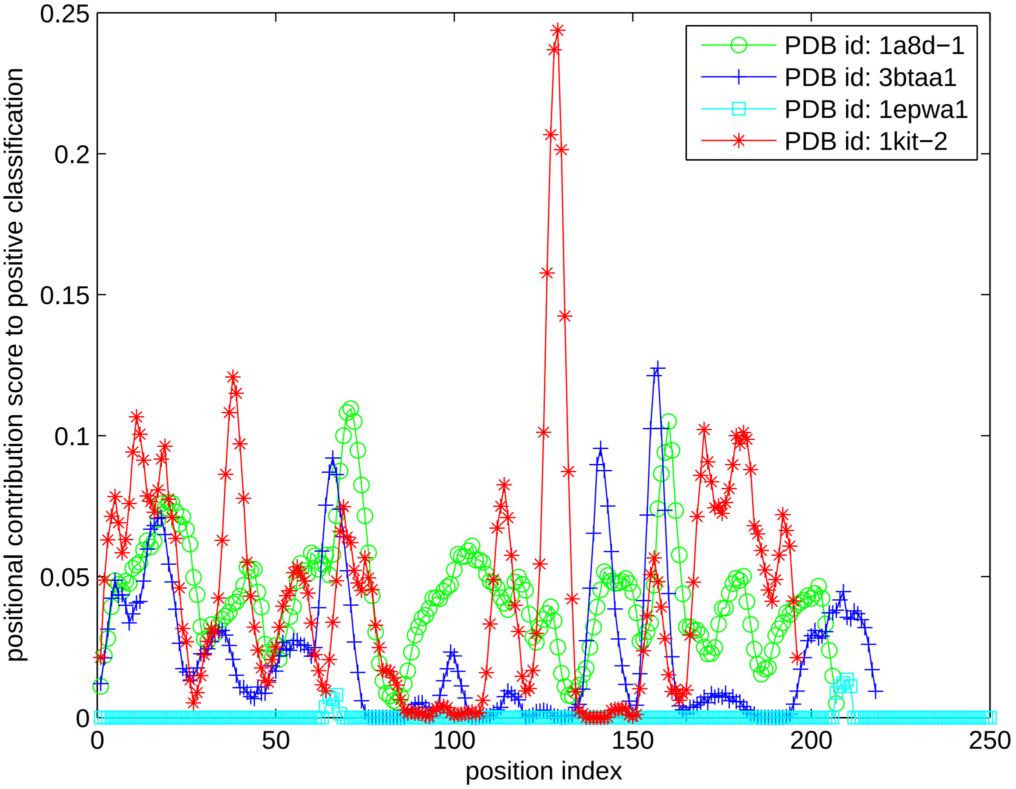


Figure 2.10: The distributions of positional contribution scores to positive classification for 4 positive training sequences.

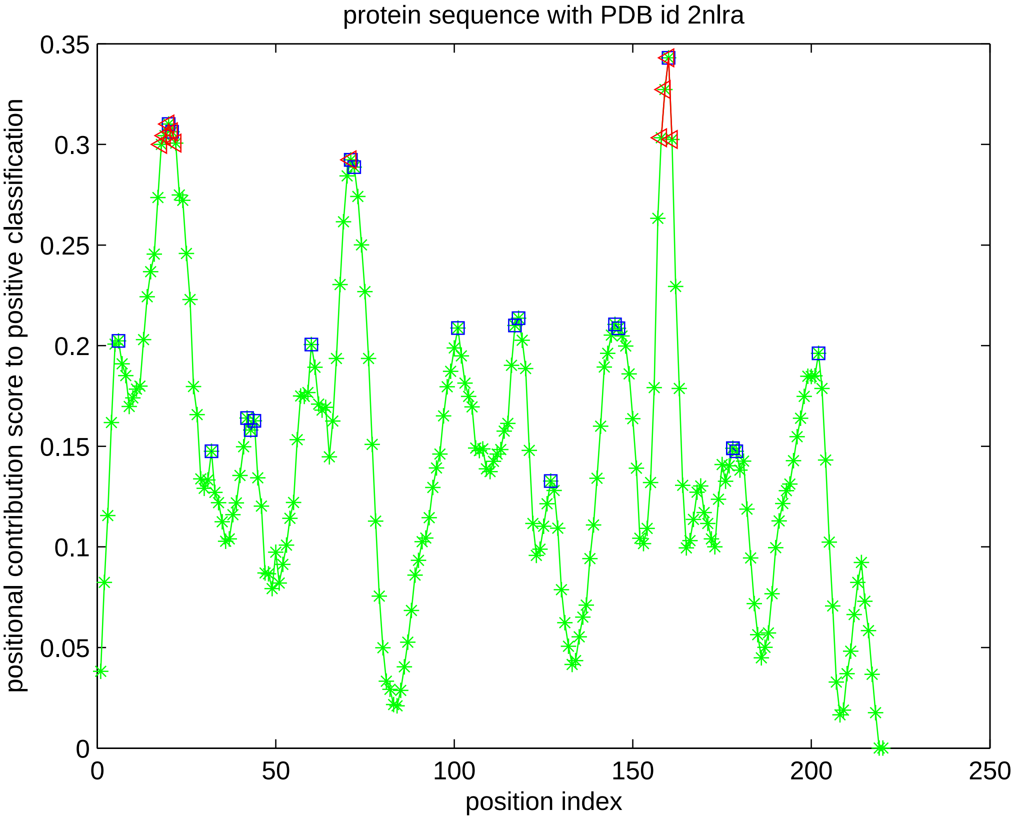


Figure 2.11: The positional contribution scores to positive classification of a positive training sequence with PDB id 2nlra. The blue positions are local peak positions, and the red positions correspond to the top 10 highest positions.

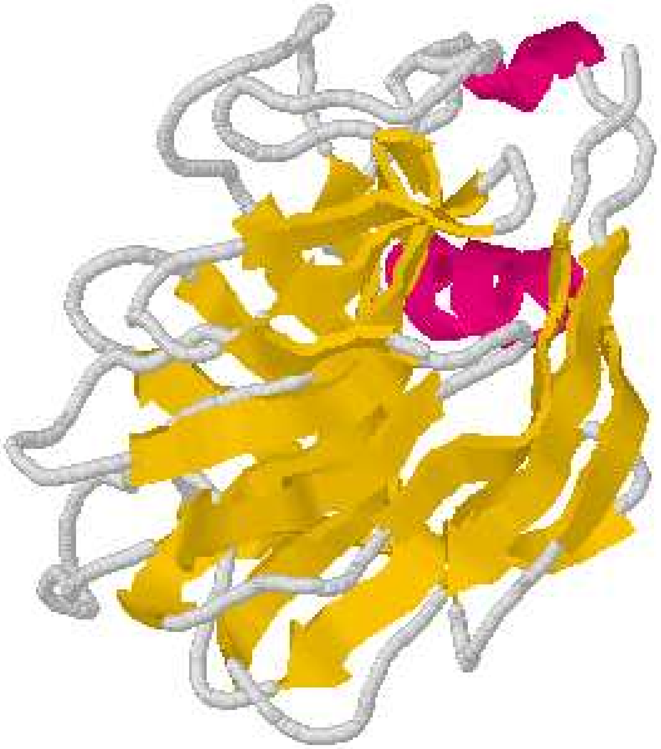


Figure 2.12: The structure motif annotated by PDB for protein sequence with PDB id 2nlra.

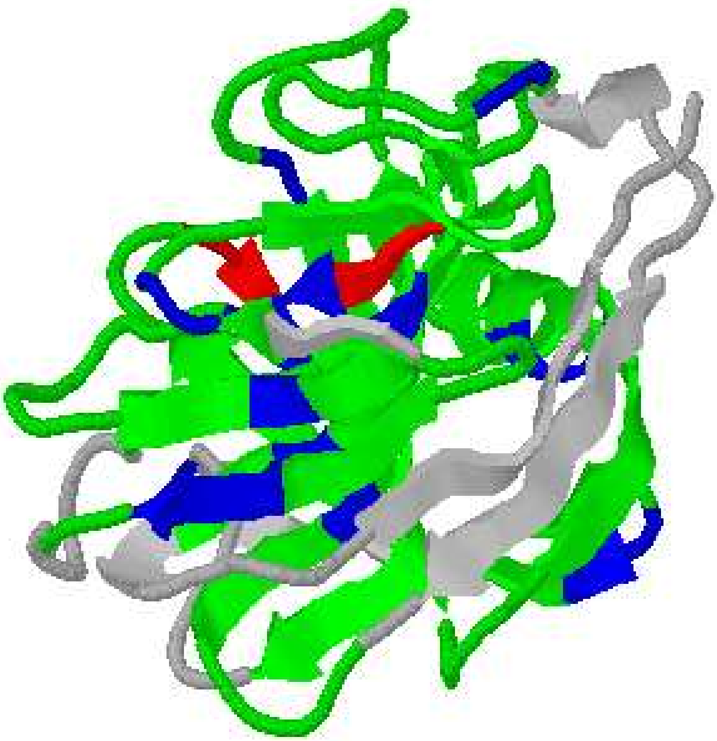


Figure 2.13: The sequence motif discovered by the SVM based on the learned random-walk kernel for protein sequence with PDB id 2nlra. The sum of the positional contribution scores of the green regions are above 80% of the sum of all the positional scores in 2nlra. The red regions correspond to the top 10 ranked positions, which correspond to the red positions in Figure 2.11. The blue regions correspond to the blue positions in Figure 2.11.

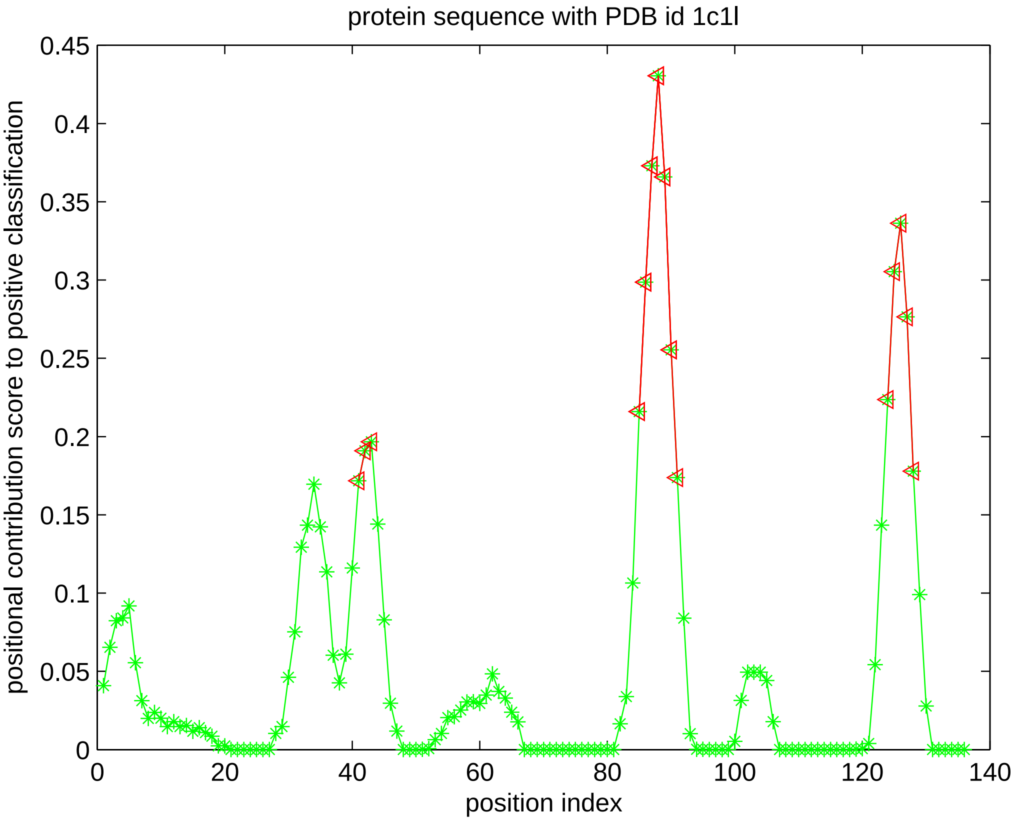


Figure 2.14: The positional contribution scores to positive classification of a positive test sequence with PDB id 1c1l. The red positions correspond to the top 15 highest positions.

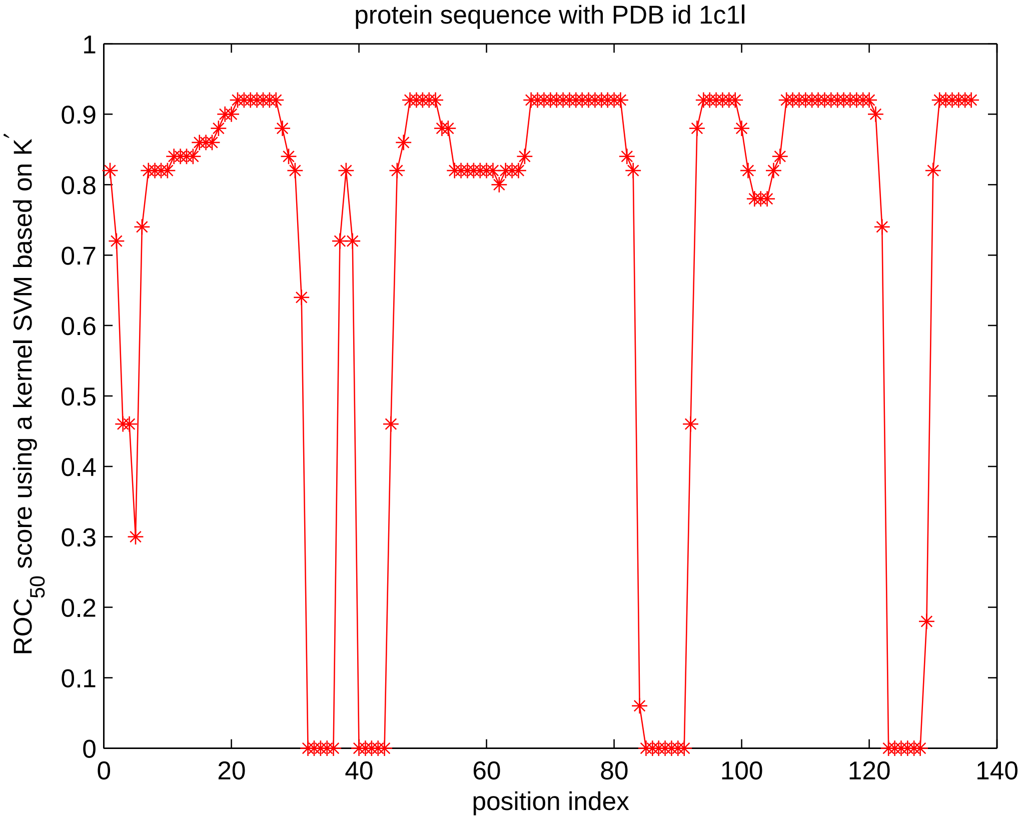


Figure 2.15: The ROC_{50} score obtained using a kernel SVM based on K' after removing the subsequence window with radius 5 centered at each position for the positive classification of the sequence with PDB id 1c1l.

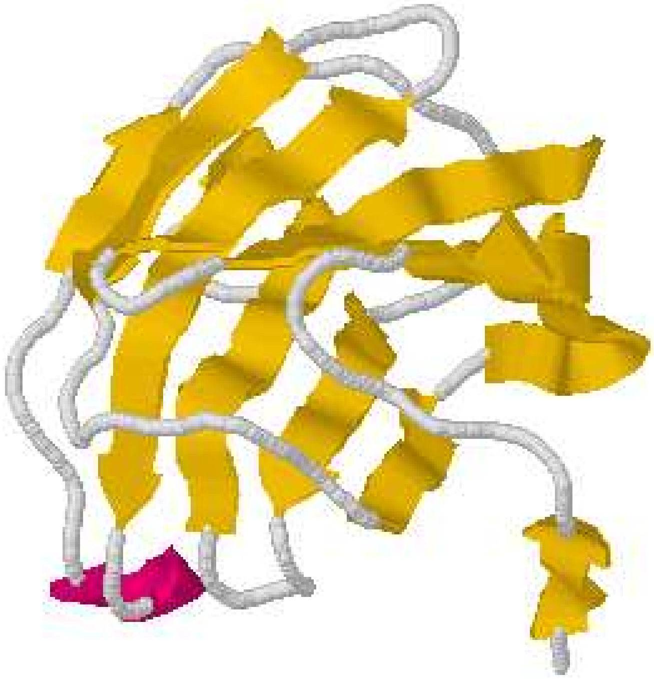


Figure 2.16: The structure motif annotated by PDB for protein sequence with PDB id 1c1l.

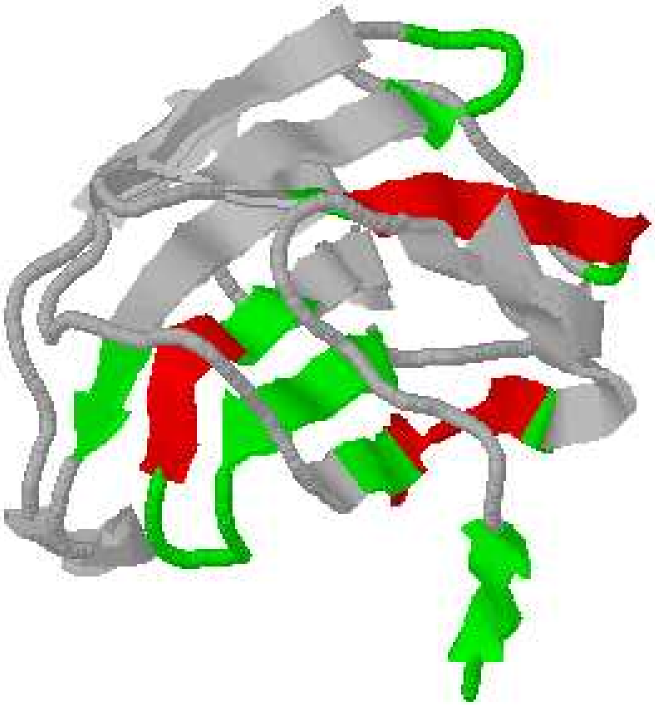


Figure 2.17: The sequence motif discovered by the SVM based on the learned random-walk kernel for protein sequence with PDB id 1c1l. The sum of the positional contribution scores of the green regions are above 90% of the sum of all the positional scores in 1c1l. And the red regions correspond to the top 15 ranked positions, which are also marked red in Figure 2.14.

Chapter 3

MicroRNA Regulation Prediction Using Proteomic Data

To understand the underlying mechanisms of gene function, it's important to know how gene expression is precisely regulated. Gene expression is mainly regulated by transcription factors at the transcriptional stage. Besides transcription factors, microRNAs (miRNAs) also play important roles in regulating metazoan gene expression, especially at the post-transcriptional stage. In this chapter, I will present computational approaches to miRNA target prediction and miRNA regulatory mechanism inference using proteomic data. Since this project is a collaborative project, I will use "we" subsequently in this chapter. We should note that this project was finished in 2007, and at that time, we did not have miRNA knockdown or miRNA transfection data, miRNA target prediction was very difficult, and we did not know which miRNA regulatory mechanism was dominant. Currently, biologists can perform miRNA transfection or knockdown experiments or use deep sequencing to get a lot of positive interactions.

Due to the difficulties in identifying microRNA (miRNA) targets experimentally in a high-throughput manner before 2008, several computational approaches were proposed, and most leading algorithms are based on sequence information alone. However, there has been limited overlap between these predictions, implying high false-positive rates, which underlines the limitation of sequence-based approaches. Considering the repressive nature of miRNAs at the mRNA translational level, here we describe a Bayesian model to make predictions by combining sequence complementarity, miRNA expression level, and protein abundance. Our underlying assumption is that, given sequence complementarity between a miRNA and its putative mRNA targets, high miRNA expression directly result in low protein abundance of the target gene. After having identified a set of confident predictions, we then built a second Bayesian model to trace back to the mRNA expression of the confident targets to investigate the mechanisms of the miRNA-mediated post-transcriptional regulation: translational repression, which has no effect on mRNA level, or mRNA degradation, which significantly reduces mRNA level.

3.1 Background

MicroRNAs (miRNAs) are a class of small non-coding RNAs, typically about 22 nucleotides in length, and are known to block protein synthesis of their target genes by binding to the 3' Untranslated Region (UTR) of the mRNA transcripts with perfect (in plants) or imperfect (in animals and C. elegans) base pairing [8]. It was estimated that thousands of genes in the mammalian genome are under regulation by miRNA at the post-transcriptional level [38], and they have been shown to have many important functional roles [3].

Despite microRNAs' importance and prevalence, it has proved to be difficult to experimentally identify and validate their target genes. To this date, only about 40 miRNA targets have been confirmed in mouse and about 200 in human [59]. As an alternative, a number of computational prediction programs had been developed and were widely used to predict miRNA targets in silico (see [17], [35], [38], and [44]). Most of these computational programs combine two types of data in making predictions: sequence complementarity between the miRNA and the putative target binding sites, and the evolutionary conservation of such sites (for a review, see [59]). Although great progress has been made in improving prediction accuracy, accurate prediction of miRNA targets remains challenging, with the major difficulty being the lack of agreement among these algorithms. A recent benchmark study has compared the predicted targets of several leading algorithms and reported significant discrepancy among them [59]. The disagreement among these algorithms can be largely attributed to the different scoring schemes and weights given to imperfect base pairing between miRNA and binding sites and evolutionary conservation of the binding sites. Moreover, some of these sequence-based algorithms are known to be less robust as slight changes in parameters often result in very different predictions [75].

Because of the repressive nature of miRNAs' regulatory roles and the availability of the genome-wide mRNA expression data, it was suggested that using gene expressions data could be helpful in predicting true miRNA targets [47]. The rationale of such approach is the following: if a miRNA is highly expressed and a putative target gene is lowly expressed in corresponding tissues, then it is considered as an additional evidence that the candidate gene is a true target. In [31] and [29], this idea was implemented as a probabilistic model which can simultaneously accounts for the interactions between multiple miRNAs and a target gene. The negative correlation between the expression levels of miRNA and target mRNA has been observed in various experiments (see [20], [47], and [60]). But some researchers claim that, in contrast to the miRNAs in plants, the miRNAs in animals typically have imperfect sequence complementarity to their target sites and function mostly by binding to the target sites to inhibit the translation process, instead of causing degradation of the mRNA transcripts ([19], [71]). And in some cases, a strong positive correlation has also been observed between the expression levels of miRNA and their target mRNAs [68], which could be attributed to common regulators shared by the miRNA and their target genes. However, several recent miRNA transfection and miRNA knockdown experiments ([6], [58]) show that, although a cohort of genes were modestly repressed by miRNAs at the protein level with little or no change at the mRNA level, almost all target genes that exhibite large protein level changes have large mRNA level changes. And a recent paper [25] shows that mRNA destablization caused by miRNAs is the dominant mechanism that accounts for the protein level changes of miRNA target genes. It's still controversial that whether mRNA degradation or translational repression is the dominant mechanism of miRNA regulation, but all the experimental results suggest that protein abundance data provides a direct and good alternative approach to predicting miRNA targets, given that we can get high-quality proteomic data.

All the evidence described above suggest that in animals, the repression effect of miRNAs on their target genes is more obviously manifested at the translational level (i.e. protein abundance), thus, identifying miRNA targets solely based on transcriptional data might miss some miRNA targets. In contrast, regardless of by degradation or by translational repression, the protein abundance of the target genes should be always negatively regulated. Motivated by this observation and by previous work, in this chapter we propose a new Bayesian approach to predicting miRNA targets in mouse using proteomic data in a high-throughput manner. In [66], researchers have used miRNA expression data and proteomic data to identify miRNA targets. However, their predictions are mainly based on biological experiments and no computational algorithm was proposed, and sequence-based methods such as TargetScan, PicTar and miRanda were only used to support their predictions. Besides, their experimental predictions were only performed in one tissue of rat, kidney. To our knowledge, our approach is the first computational one that incorporates miRNA expression data and proteomic data in multiple tissues to carry out high-throughput miRNA target predictions. In [31], proteomic data was only used to decide whether a probabilistic model for modeling translational repression or a model for modeling mRNA degradation should be used for miRNA target prediction, which is completely different from our model that directly uses proteomic data for target prediction.

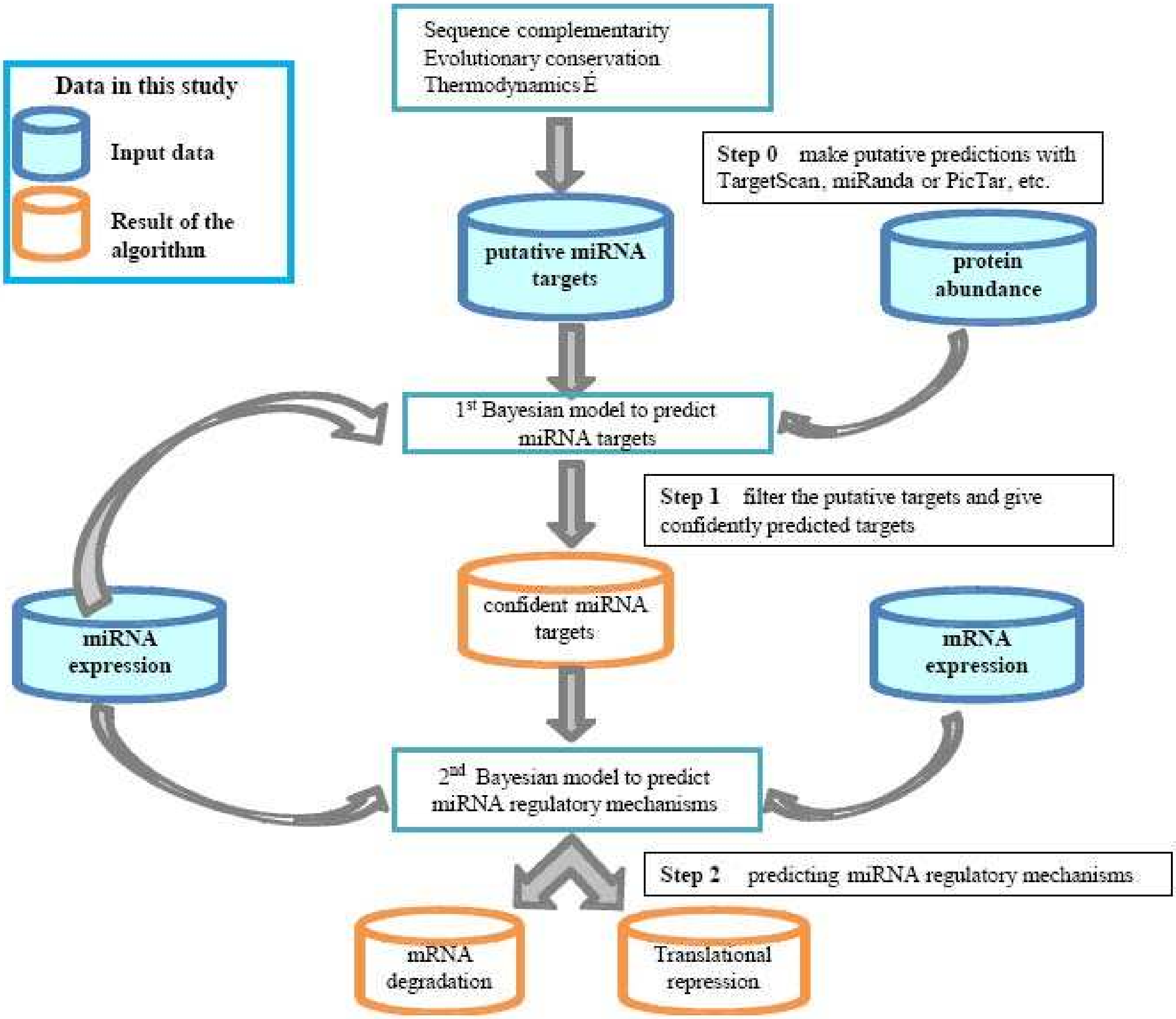

Data in this study
Input data
Result of the algorithm
Sequence complementarity
Evolutionary conservation
Thermodynamics E
Step 0 make putative predictions with TargetScan, miRanda or PicTar, etc.
putative miRNA targets
protein abundance
1st Bayesian model to predict miRNA targets
Step 1 filter the putative targets and give confidently predicted targets
miRNA expression
confident miRNA targets
mRNA expression
2nd Bayesian model to predict miRNA regulatory mechanisms
Step 2 predicting miRNA regulatory mechanisms
mRNA degradation
Translational repression

Figure 3.1: A flowchart of the algorithms described in this work. The described algorithm takes four types of experimental data: (1) a set of putative miRNA targets, (2) protein abundance, (3) miRNA expression profiles, and (4) mRNA expression profiles. Details can be found in the main text.

Our method consists of two steps. In the first step, it takes as input a set of putative miRNA target genes derived from sequence information alone; it then applies a Bayesian probabilistic model using protein abundance data to assign confidence scores to the predicted miRNA-target pairs. In the second step, another Bayesian probabilistic model is applied to the miRNA and mRNA expression data to predict whether the miRNA-mediated regulation is through translation repression or mRNA degradation. Figure 3.1 shows a flowchart of our approach.

3.2 Data Gathering

The mouse protein abundance data was derived from a recently published mass spectrometry study [36], in which the abundance of 4,768 proteins across 6 mouse tissues (brain, heart, liver, lung, placenta and kidney) was surveyed. After comparing with gene expression data from two microarray studies ([64], [76]), 1,758 proteins were confidently cross-mapped to their corresponding mRNAs. The incomplete coverage of the proteomic data was likely due to instrumentation bias, stringent filtering rules of database search or instability of low-level transcripts. The miRNA expression data was extracted from previous published microarray studies [5]. These authors also used TargetScan and miRanda separately to derive two lists of putative miRNA targets in mouse. The normalized mRNA expression profiling of 41,699 transcripts in 55 tissues was from [76]. Among these, 1,758 were confidently cross-mapped to the proteins (see above).

We chose to use full Bayesian model to make inference so as to take into account all possible uncertainties in our model. Inferences were made based on Gibbs sampling [22], which was performed in the WinBugs environment [61].

3.3 Method and Results

3.3.1 Deriving a list of putative miRNA targets by sequence data alone

We described the sources of the data in the above Methods section. As described in the Introduction, our current methods take as input a set of putative predictions from a sequence-based prediction algorithm. We decided to run our procedures twice using two different prediction algorithms: TargetScan [43] and miRanda [17]. The general results and conclusions are unchanged. Based on the intersection among the four types of datasets (predicted miRNA targets, miRNA expression, protein abundance and mRNA expression), we retained 21,721 putative interactions for TargetScan predictions (75 miRNAs, 1,404 cross-mapped mRNAs) and 17,339 putative interactions for miRanda predictions (70 miRNAs, 1408 cross-mapped mRNAs).

After compiling the datasets, we investigated mRNA or protein expression profiles in 6 tissues (brain, heart, liver, lung, placenta and kidney), among which expression in 4 tissues (brain, heart, liver and lung) were used for model construction and making predictions, while the remaining 2 tissue types (placenta and kidney) were used for blind tests.

3.3.2 Modeling protein abundance

Instead of looking for the degradation of mRNA transcripts by possible multiple miRNAs as previously described in [29], our method directly models the relationship between miRNA expression and protein abundance (see Figure 1). We start with a set of miRNA and target genes as predicted from a sequence-based approach, we then model the protein abundance of the putative targets and the miRNA in individual tissues. If we observe a negative correlation between the miRNA and the putative target across multiple tissues, then the algorithm will assign a higher confidence score to this miRNA-protein pair.

Conversely a positive correlation between miRNA expression and protein abundance, especially the cases where a high miRNA expression coincide with high protein abundance, will result in a low confidence score for the miRNA-protein pair.

We chose to use a probabilistic framework to model the relationship between miRNA expression profiles and protein abundance. The first challenge in this approach is to find an appropriate background distribution to model the protein abundance data. Unlike mRNA expression profiles, which can be effectively modeled using a Gaussian distribution, the peptide counts are discrete values. A possible choice is to use Poisson to model the count events; however, a simple Poisson model is not suitable for modeling the peptide counts in this study since there are excessive zeros in the dataset and the non-zero counts are also over-dispersed (variance are much greater than the mean). After comparing with other possible models such as zero-inflated Poisson and transformed Gaussian, we chose to use Negative Binomial model (NB) to characterize the peptide counts, with which the Poisson mean and over-dispersion can be considered simultaneously with lower model complexity. Recent research also suggested NB is an optimal choice to model the abundance data with excessive zeros [72].

The protein abundance data has discrete integer values corresponding to protein counts and a lot of zeros corresponding to no protein abundance, and has much larger sample variance than sample mean, which cannot be effectively modeled by a Poisson distribution but can be fitted very well by a Negative Binomial distribution empirically. A Negative Binomial distribution with a positive real parameter r and a real parameter γ ($0 < \gamma < 1$) is described in the following equation:

$$
\begin{aligned}
p(k|r,\gamma) &= \binom{k+r-1}{r-1}\gamma^r(1-\gamma)^k \\
&= \frac{\Gamma(k+r)}{k!\Gamma(r)}\gamma^r(1-\gamma)^k. \qquad (3.1)
\end{aligned}
$$

In the above equation, k is an integer, and the mean of the distribution is $r\frac{1-\gamma}{\gamma}$. With integer r, $p(k|r,\gamma)$ can be viewed as the probability of having observed k successes if we are observing a sequence of Bernoulli trials with success probability p until we have observed a pre-defined number of r failures, which is conversely analogous to a Binomial distribution, so it is called Negative Binomial distribution. If we re-parametrize the Negative Binomial distribution using the mean parameter $\lambda = r\frac{1-\gamma}{\gamma}$ and the positive real parameter r, we have the following equation:

$$\begin{aligned} p(k|r,\gamma) &= NB(k|\lambda, r) \\ &= \frac{\lambda^k}{k!}\frac{\Gamma(r+k)}{\Gamma(r)(r+\lambda)^k}\frac{1}{1+\frac{\lambda}{r}}. \end{aligned} \tag{3.2}$$

In the above equation, $\Gamma(\cdot)$ is the Gamma function, r controls the over-dispersion of the distribution, and, when the over-dispersion parameter r approaches infinity, $NB(k|\lambda, r)$ approaches a Poisson distribution with mean parameter λ. The above NB model uses r to adjust the variance independently of the mean parameter λ of the distribution, differing from a Poisson distribution which has equal mean and variance. We model protein abundance data by NB using the parametrization in Equation 3.2. We assume the abundance of each protein i in tissue type t, W_{it}, follows NB distribution, with two parameters θ_{it} and r_t, $1 \leq i \leq N$ and $1 \leq t \leq T$, where N and T are the total number of genes (proteins) and total number of tissues types, respectively. Thus, the probability of protein i with peptide count k in tissue t can be modeled as the following in Equation 3.3.

$$p(W_{it} = k|\theta_{it}, r_t) = NB(k|\theta_{it}, r_t). \tag{3.3}$$

In Equation 3.3, θ_{it} represents the Poisson mean for protein i in tissue t and r_t represents the over-dispersion of the data, which was shared by all the proteins in the same tissue t. We then used hierarchical Bayesian Negative Binomial regression to regress the Poisson mean θ_{it} with miRNA expression in corresponding tissues, M_{jt}, $1 \leq j \leq J$, and $1 \leq t \leq T$, where J is the total number of miRNAs in the dataset. Equation 3.4 gives the regression of the Poisson mean in the model. Thus,

$$ln(\theta_{it}) = ln(\tau_t) - \rho_t \sum_{j=1}^{J} \omega_j \delta_{ij} M_{jt} \tag{3.4}$$

In Equation 3.4, τ_t stands for the background protein abundance shared by all the proteins in the same tissue t. As suggested in [29], we introduced δ_{ij} as a binary latent variable indicating whether or not the miRNA j regulates the gene i; ω_j is a regression coefficient associated with j-th miRNA expression shared by all the tissue types, and ρ_t is a scaling parameter for tissue t accounting for the measurement difference in different tissues. Since sequence complementarity is a necessary condition for true targets, we use a binary variable S_{ij} as the putative predictions between miRNA j and protein i, which was derived from sequence-based predictions (TargetScanS, miRanda, or PicTar, etc); $S_{ij} = 1$ means that there is a putative predictions between i and j. The probability of a putative prediction being a true positive, p, is formally given in Equation 3.5.

$$\begin{aligned} p(\delta_{ij} = 1 | S_{ij} = 1) &= p, \\ p(\delta_{ij} = 1 | S_{ij} = 0) &= 0 \end{aligned} \tag{3.5}$$

To avoid over-fitting the data and to account for all possible uncertainties, we chose to use full Bayesian approach to infer δ_{ij} so that all the uncertainties and nuisance variables can be integrated out. Thus we assigned the priors to other parameters as follows (most were assigned flat priors):

$$\begin{aligned} p &\sim beta(1,1), \\ \tau_t &\sim uniform(0,50), \\ \rho_t &\sim gamma(\alpha,\alpha), \\ \alpha &\sim uniform(0,+\infty), \\ \omega_j &\sim exponential(\beta), \\ \beta &\sim uniform(0,1000), \\ r_t &\sim exponential(a), \\ a &\sim uniform(0,1000), \end{aligned} \tag{3.6}$$

We chose the above Beta prior because Beta distribution is the conjugate prior of Bernoulli distribution and Beta(1, 1) defines a uniform distribution between 0 and 1. We chose a uniform distribution between 0 and 50 as the prior of τ_t because the peptide count in our dataset never exceeds 50. We chose a Gamma prior for the scaling parameter ρ_t because we just wanted to define an arbitrary prior over a non-negative parameter. And we chose exponential priors for the over-dispersion parameter r_t and the regression coefficient ω_j because we wanted to have a prior on non-negative parameters and we wanted to penalize very large values. The upper bound in the last two uniform distributions were set large enough to make sure a wide range of exponential distributions can be sampled for r_t and ω_j. With the likelihood and priors defined above, we then implemented Gibbs sampling [22] to compute marginal distribution of δ_{ij} conditioned on all evidence. All the inferences were made on drawing 5,000 samples after 10,000 iterations.

3.3.3 Apply Bayesian model to predict miRNA targets

We applied the model described above to the 21,712 putative miRNA-protein interactions derived from TargetScan, and assigned a confidence score to each putative interaction. Then we ranked the 21,712 putative interactions from the highest to the lowest confidence, and grouped them into 44 bins with each bin containing 500 ranked interactions. The results are shown in Figure 3.2A-D for 4 different tissue types.

As shown in Figure 3.2, our model can well capture the miRNAs' repression effects in these four tissues. The miRNA-protein pairs that are predicted to have the highest confidence scores have lower protein abundance and higher miRNA expression; conversely the miRNA-protein pairs with the lowest confidence scores also have higher protein abundance and higher miRNA expression. For the interactions ranked with intermediate confidence scores, the miRNA expression is low, and the protein abundance can be either low or high. The fact that a large number of TargetScan predictions are located in the right side of the curve, i.e. low confidence score with high miRNA expression and high

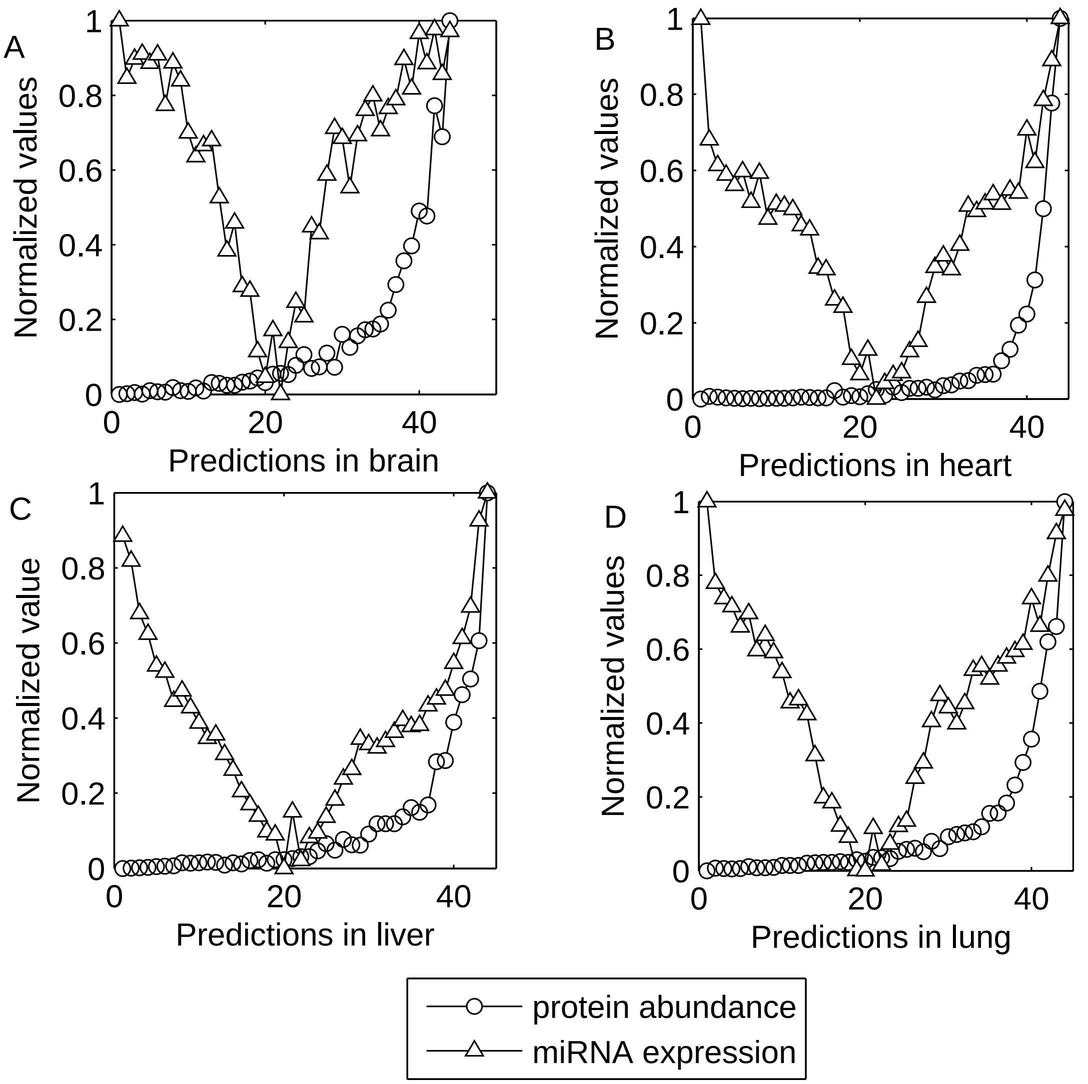


Figure 3.2: miRNA targets prediction using miRNA expression and protein abundance. With our model, in the 4 tissues (panel A-D), the most confident predictions (on the left) have the lowest protein abundance and the highest miRNA expression; while the least confident predictions (on the right) are high in both protein abundance and miRNA expression. All the data were scaled between 0 and 1. The putative predictions were from (Babak, et al, 2004) using TargetScan.

protein abundance, indicates the extent of possible false-positives in the predictions made from sequence data alone. Because the Bayesian approach is intrinsically evidence-based, a prediction can only be made with high confidence if the miRNA is highly expressed in a certain tissue.

Note that the high-confidence miRNA-protein interaction pairs as shown in Figure 3.2 are predictions pooled from all 4 tissues. We do not explicitly model the tissue specificity of miRNAs in our model (see Equation 3.3); instead, the strengths of the miRNA regulation in specific tissues are inferred from the expression level of miRNAs. For instance, a miRNA can be interpreted as a functional regulator in a given tissue only if it is highly expressed and it has high confidence score with a potential target protein that is lowly expressed in that tissue.

3.3.4 Blind tests for the Bayesian predictions method

As described above, we only used the protein abundance and miRNA expression in brain, lung, heart and liver to train our model and make predictions; the data in the remaining two tissues (placenta and kidney) was left out during the model construction stage. To further validate our method, we subsequently conducted a blind test on the placenta and kidney data sets.

Figure 3.3A shows the results of the blind test. On the X-axis, we sorted the miRNA-protein pairs according to the confidence scores predicted by using the four training tissues; on the Y-axis, we plotted the protein abundance and miRNA expression level that are observed in placenta. The results indicate that, as a general trend, the predicted interactions can also reflect the desired tendency in placenta. The predicted interactions with high confidence usually have low protein abundance and high miRNA expression. The least confident predictions also have highly expressed proteins and highly expressed miRNAs, indicating those proteins are unlikely to be repressed by the miRNAs in placenta.

In kidney as the second blind test, shown in Figure 3.3B, although the miRNA expression data were not available for this tissue type, clearly our predictions were also effective and the most confident predictions have the lowest protein abundance and vice versa. All the above analysis were based on the sequence-based predictions from TargetScan. The same results also hold true after we repeated the analysis using predictions from another program miRanda [17]. Next, to test the robustness of our method, we shuffled the gene labels to randomize the proteomic data, and we also applied our Bayesian model on the random data. We ranked the interaction pairs from the most confident to the least confident, and Figure 3.2C shows the grouped protein abundance of the ranked miRNA targets for the random data with the same grouping used for the results in Figure 3.2 and 3.3AB. To compare the ranked protein abundance in the brain tissue for the real data in Figure 3.2 to that in Figure 3.3C, we grouped 5 consecutive groups (dots) into one region, the standard deviations of the first 7 disjoint neighboring regions of the real data are, respectively, 0.0039, 0.0049, 0.0091, 0.0069, 0.023, 0.039, and 0.024, and the standard deviations of the first 7 regions of the random data are, respectively, 0.17, 0.17, 0.16, 0.063, 0.075, 0.082, and 0.064. We can see that, on the real data, the standard deviations of protein abundance for the highly-ranked miRNA-protein interaction pairs are much smaller, and the p-value for the two sequences of standard deviations based on Wilcoxon Signed-Rank Test is 0.016. Moreover, the results in Figure 3.2C from the shuffled data appear clearly random, which strongly suggests that our predictions did not occur by chance.

3.3.5 Comparison with TarBase and other methods

We further searched for published experimental evidence for our predicted interactions in TarBase [59], which is a comprehensive database containing experimentally verified miRNA targets in a number of organisms. However, to this date, there are only 41 experimentally verified miRNA targets for mouse in the database. Since in the database

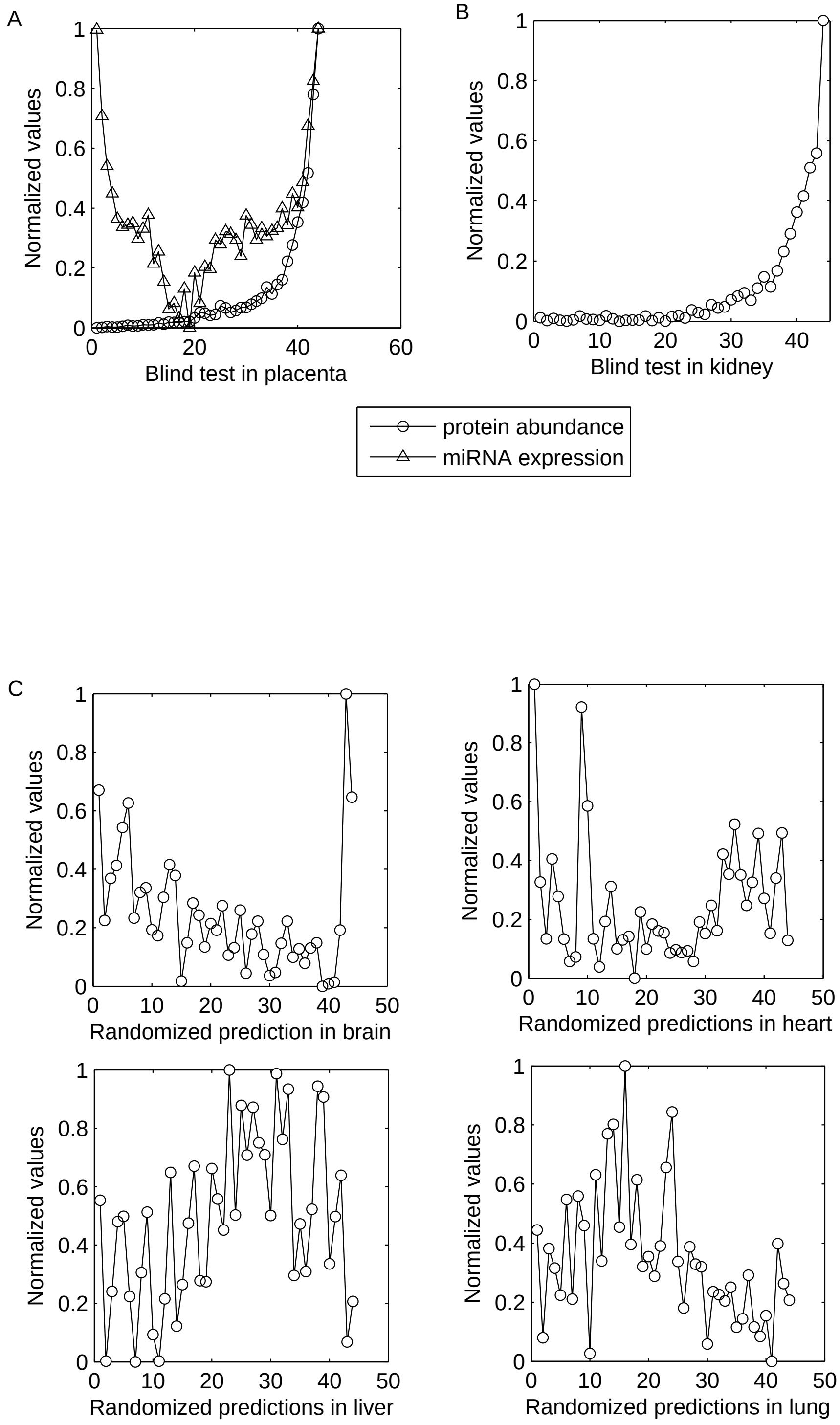


Figure 3.3: Blind test of our predictions on placenta (A) and kidney (B). (C) Result from randomization with protein labels shuffled.

all targets used gene symbol, we then converted the Swiss-Prot protein names in our study to corresponding gene symbols via http://idconverter.bioinfo.cnio.es/ [1]. However, except for the gene Arid3a (ARI3A_MOUSE), all other genes were not included in our dataset as they do not have protein abundance data compiled in this study. For Arida3a, in TarBase, it was annotated to be regulated by miR-125b. From our predictions based on miRanda predictions, the interactions between miR-125b and ARI3A_MOUSE was ranked among top 5% in all the 17,339 putative interactions, suggesting ARI3A_MOUSE is likely a true target. However, this interaction was not detected by TargetScan as compiled in [5].

We also compared our prediction results to the results obtained using the method in [29]. Most of the predictions by both models are consistent, but our approach directly used the proteomic data for target predictions [1], and we believe that modeling proteomic data is the most reliable way of filtering miRNA target predictions when large-scale proteomic data become available. In details, the miRNA/target interactions such as mmu-mir-214/Q8R399_MOUSE, mmu-mir-211/Q8BYX4_MOUSE, mmu-miR-292-5p/KCNN3_MOUSE, and mmu-miR-298/RRAS2_MOUSE all ranked among top 10% in all the putative interactions in both models. However, the miRNA/target interactions such as mmu-miR-298/PLF3_MOUSE, mmu-miR-210/Q8BSZ8_MOUSE, and mmu-miR-92/8BZZ4_MOUSE all ranked among top 1% in all the putative interactions in our model, but they all ranked among bottom 15% in all the putative interactions in the model by [29]. We found that these miRNA/target pairs all have very good relatively high miRNA expression vs. relatively low protein abundance patterns, but they don't have very clear relatively high miRNA expression vs. relatively low mRNA expression patterns. Since miRNAs can either degrade mRNAs or repress mRNA translation, which will be discussed later in this chapter, we believe that these interactions are likely to be

[1] Our Bayesian model filters sequence-based predictions and removes a lot of false positives predicted by sequence models.

false negatives predicted by the model in [29].

3.3.6 Two possible mechanisms: mRNA degradation vs. translational repression

In this section, we tried investigating whether the predicted target genes are regulated by translational repression or by mRNA degradation using a Bayesian model, which was only an attempt to understand miRNA regulatory mechanism based on all the data published before 2007. In [31], Bayesian Networks were constructed to test whether a mRNA degradation model or a translational repression model is more appropriate for fitting the mRNA expression and protein abundance data to predict miRNA targets by comparing the Bayesian scores of the Bayesian Networks, and this model provided a very reasonable model selection method for restricting all miRNA and target gene interactions to one type of miRNA regulatory mechanism and completely ignoring the other one, but the model is not capable of inferring which mechanism is more likely for a specific miRNA and target gene interaction. In another word, the model in [29] can only predict one type of miRNA targets with the assumed regulatory mechanism. An improved version of this model was published in [30], which infers the regulatory mechanism of a specific miRNA-protein interaction by regressing the target protein abundance based on either mRNA expression level alone or both mRNA expression level and miRNA expression level using linear models. By comparing the sum of least squared errors of the two different linear models using cross validation on different tissues, the improved model can predict the miRNA regulatory mechanism for the given interaction, but the model suffers from the limited available tissues for fitting each linear model for each miRNA-protein interaction. In this section, we presented an alternative Bayesian model for inferring miRNA regulatory mechanism. For the top miRNA-protein pairs that are predicted to be true regulator and targets, we can distinguish between these two possible regulatory mechanisms by analyzing the correlation between the miRNA expression and the mRNA expression.

For example, if a top-ranked miRNA-protein pair has high miRNA expression and high mRNA expression, then it is a strong indicator that the protein target is regulated by translational expression. In contrast, if a predicted miRNA target has low mRNA expression, then it is likely regulated by mRNA degradation.

There are two common concerns in modeling the mRNA expression data: (1) the intrinsic low signal-to-noise ratio of microarray data, (2) the potential problem of missing values since a large number of the genes have expression levels measured as 0 [76]. To overcome these difficulties, we elected to discretize the mRNA expression data by using a cutoff of 0.1 to binarize the expression level to either low or high, which was decided by checking the histogram of the expression level and fitting a Mixture of Gaussians with two mixture components. For a given mRNA i in tissue t, $1 \leq i \leq L$ and $1 \leq t \leq T$, where L is the total number of mRNAs in the confident predictions derived from the first model, its mRNA expression R_{it} can be either low ($R_{it} = 0$) or high ($R_{it} = 1$). Let the probability of degradation for mRNA i in tissue t be q_{it}, we assume,

$$P(R_{it} = k) = q_{it}^{(1-k)}(1 - q_{it})^k, k = 0 \quad or \quad 1. \tag{3.7}$$

We next used logistic regression to regress q_{it} with the expression of miRNAs that regulates gene i, in tissue t. Then we have the following equation,

$$logit(q_{it}) = log(\frac{q_{it}}{1 - q_{it}}) = \sum_{j=1}^{H} \Phi_j b_{ij} M_{jt} \tag{3.8}$$

in which H is the total number of miRNAs in the miRNA-protein interactions, M_{jt} is the expression of the j-th miRNA in tissue t, b_{ij} is a binary latent variable indicating whether or not the gene i is degraded by miRNA j , and Φ_j is a scaling parameter associated with the j-th miRNA, shared by all tissue types. The rationale behind Equation 3.8 is that for a given gene i, if its expression is low in tissue t, i.e. $R_{it} = 0$, then from the perspective of maximum likelihood, we need to maximize q_{it} so that the interactions between gene i and its regulating miRNAs that are highly expressed in tissue t should be assigned a higher degradation score. In this sense, the observed low expression of mRNA and high expression of miRNA together lead to the assignment of a high degradation probability. Similarly, if in tissue t, $R_{it} = 1$, then q_{it} needs to be minimized, implying those highly expressed miRNAs should be associated with a low degradation score, so the highly expressed miRNAs and mRNAs indicate such regulation is more likely to be through translational repression than through degradation.

Regarding the latent variable b_{ij}, we further required that:

$$\begin{aligned} p(b_{ij} = 1 | \delta_{ij} = 1) &= h, \\ p(b_{ij} = 1 | \delta_{ij} = 0) &= 0, \end{aligned} \tag{3.9}$$

in which, δ_{ij} indicates whether or not mRNA i is targeted by miRNA j. If $\delta_{ij} = 1$, then the miRNA j has a probability h to cause degradation to its target mRNA i. We then used a full Bayesian approach to estimate the parameters in the model to avoid overfitting the data and to account for all potential uncertainties. In the Bayesian framework, we then assigned priors to other parameters in the model as follows:

$$\begin{aligned} \Phi_j &\sim exponential(\Psi), \\ \Psi &\sim uniform(0, +\infty), \\ h &\sim beta(1, 1). \end{aligned} \tag{3.10}$$

Having defined the likelihood and the priors, we then inferred the posterior marginal distribution of $p(b_{ij} = 1 | \mathbf{S}, W, M)$, conditioned on all the evidence. By implementing Gibbs sampling in the environment of WinBugs [61], all the inferences were based on drawing 5,000 samples after 10,000 iterations.

3.3.7 Apply the Bayesian model to mRNA data

By implementing the model described above, we calculated the confidence scores for mRNA degradation for each miRNA-protein interaction pair, which indicated the likelihood that miRNA causes degradation to their mRNA targets. The lower degradation score implies higher probability of being translationally repressed. Then, we ranked the scores from the highest to the lowest, and grouped them into 50 bins, each containing 100 ranked interactions. Figure 3.4 shows the mRNA expression level of the ranked miRNA targets across 4 tissue types. The miRNA targets near the top of the Figure 3.4(A) have the highest probability of being regulated by mRNA degradation, as demonstrated by their low mRNA expression level (details shown in Figure 3.4(B)). Conversely the targets near the bottom have the highest probability of being regulated by translational repression (details shown in Figure 3.4(C)). The top ranked interactions are associated with the low mRNA expression and the bottom ranked interactions are associated with high mRNA expression, which is consistent with our model assumptions. We can use the degradation probabilities to infer the exact regulatory mechanism for each miRNA-protein interaction pair, and we can verify the highly-ranked predictions by biological experiments, which will be done in the future.

3.4 Discussion

3.4.1 miRNA regulation by translational repression

In this chaper we described two novel formalisms in the computational analysis of miRNA regulation. We first introduced a Bayesian approach to identify miRNA targets based on protein abundance data. After having selected high confidence predictions, we then introduced a second Bayesian model to further distinguish the two possible regulatory mechanisms, i.e. mRNA degradation versus translational repression. We showed that our model is very effective in describing the three intertwining genomics data sets, i.e. miRNA expression, mRNA expression, and protein abundance. Our results demonstrated that protein abundance is a very useful resource in predicting miRNA targets. We would like to point out that although in this chapter our model takes as input the predictions from TargetScan and miRanda, essentially results from any other sequence-based predictions

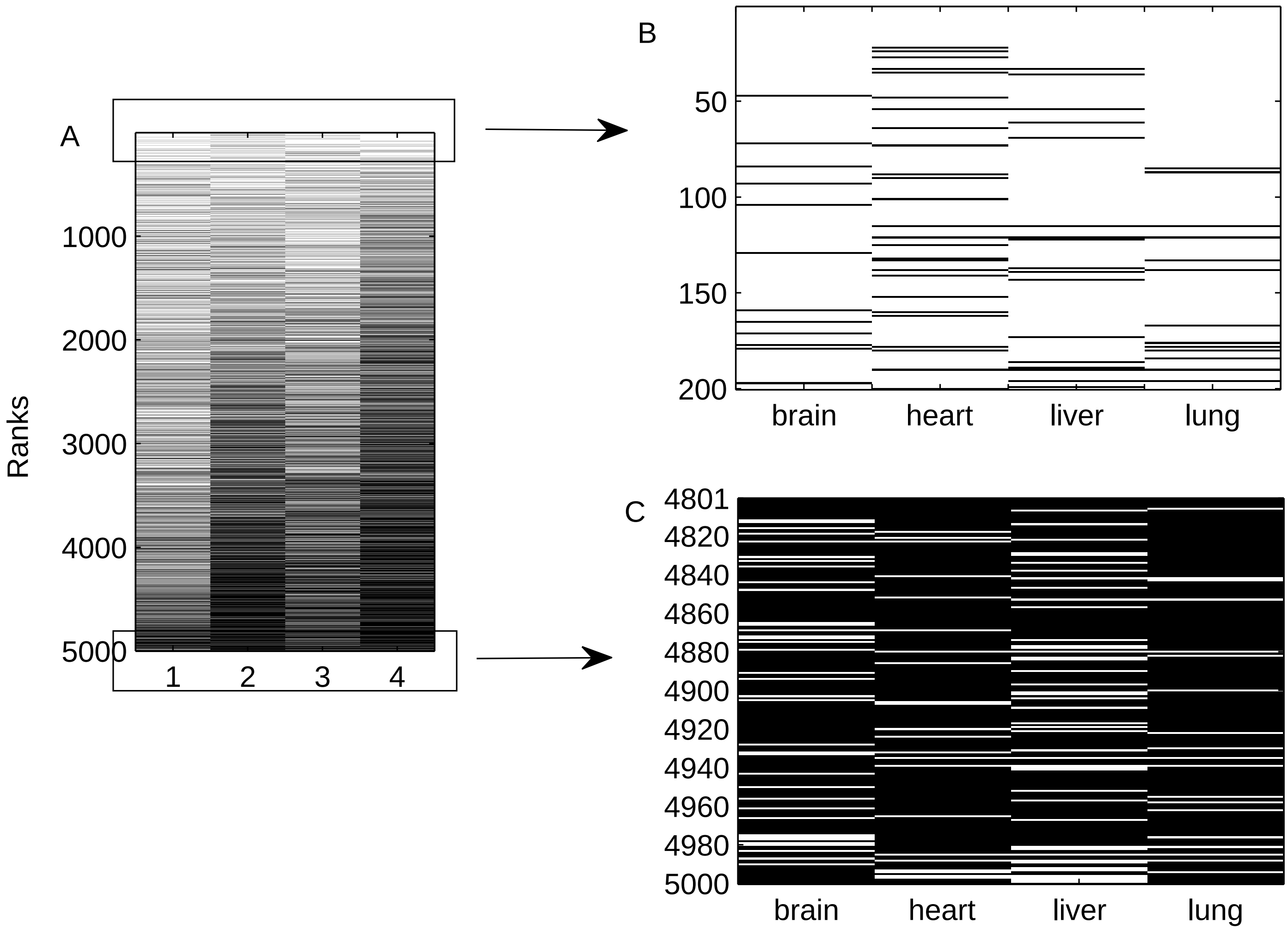


Figure 3.4: (A) Ranking of miRNA targets according to the probability of being regulated by the mechanism of mRNA degradation; the targets were ranked from the highest degradation probability (top) to the lowest degradation probability (bottom). It also showed mRNA expression of the targets across 4 tissue types. Black color denotes high expression and white color denotes low expression. (B) The top ranked interactions have the highest degradation probability, and are associated with the low mRNA expression. (C) The bottom ranked interactions have the lowest degradation probability, and thus are associated with high mRNA expression.

can be used in our model.

3.4.2 Potential limitations and future directions

Even though our framework has obtained encouraging results, it certainly has limitations. We envision that it can be improved in the following area. (1) As we noted, miRNA is not the only mechanism of gene regulation. Some of the observed variations in protein abundance across tissues are likely the result of regulation at the transcription level by transcription factors, or at the post-transcriptional level by mRNA degradation pathways. (2) Although it has been reported that the mechanism of translational repression by miRNAs has little impact on mRNA level, the mRNA expression is still helpful in predicting miRNA targets, and recent research suggested that targeted mRNA showing strong correlation (positive and negative) with miRNAs ([19] and [68]). In the future, we could incorporate the mRNA expression data with the proteomic data to build an integrated predictive model. (3) At this stage, our model takes as input the sequence-based predictions from another prediction programs such as TargetScan or miRANDA, therefore our algorithm does not explicitly consider the sequence complementarity and evolutionary conservation. As a future work, it would be interesting to extend our model to incorporate these properties into a unified probabilistic framework. (4) In our model, similar with [29], we assumed a single baseline distribution of protein abundance for all the genes in each tissue type. However, this is a significant simplification since different genes could have distinct baseline expression levels. The next step in this work is to take this into account and develop a more realistic expression baseline model. For example it would be possible to take into account the codon usage of the genes to infer the possible baseline expression of a given gene [48].

Chapter 4

Gene Function Prediction Using Yeast Morphology Data

Protein sequence determines gene function at sequence level by determining different structures, miRNAs influences or controls gene function at expression and protein abundance level by affecting the protein production quantities, and finally, different genes manifest their ultimate functions at the morphology level. In this chapter, I will discuss how to predict gene functions from yeast cellular morphology data.

4.1 Background

The budding yeast Saccharomyces cerevisiae has been thoroughly studied as model organism to reveal gene functions to answer many important biological questions. To study gene functions, gene knockout experiment is the most extensively used approach by biologists. However, biologists found that approximately 19% of yeast genes are essential for haploid viability under normal laboratory conditions (growth at 30 degree in rich medium with glucose) [23]. Because essential genes cannot be deleted in a yeast haploid strain, their functions cannot be studied directly by knockout experiments. Biological experiments show that a lot of genes involved in important biological processes such as transcription, splicing, ribosome biosynthesis, translation, cell wall and membrane biogenesis, DNA replication, nuclear transport, and basic cytoskeletal functions are all required for cell proliferation and are often essential. Although they are more important, essential genes are less known to biologist for their precise functions at the molecular level. Fortunately, conditional alleles provide opportunities to study the functions of essential genes. Conditional yeast gene mutants include temperature-sensitive (ts) alleles, cold-sensitive (cs) alleles, temperature-inducible degron (td) mutants [16], and tetracycline-regulatable promoter-replacement (tet) alleles [18].

Temperature-sensitive alleles (mutants) are the most commonly used conditional alleles in yeast. At the permissive temperature, the activity or phenotype of a ts mutant is very similar to that of the wild type. However, at the restrictive temperature, the activity of a ts mutant is reduced or abolished, resulting in a slow-growth or lethal phenotype. Given that many ts mutants are readily available and relatively easy to work with, temperature-sensitive mutants (ts allels) have been used by biologists for decades to study gene functions in vivo and to unravel the gene interaction networks. Biologists have developed several different methods for creating temperature sensitive mutants such as DNA shuffling and error-prone Polymerase Chain Reaction (PCR).

In this project, my collaborators constructed a collection of temperature-sensitive (ts) mutants of yeast essential genes in the S288c background. They gathered over 1000 yeast strains and DNA constructs carrying ts alleles from about 300 different laboratories. They constructed 795 ts strains representing more than 500 essential genes, accounting for about 50% of yeast essential genes, and they verified close to 99% of the ts alleles using Polymerase Chain Reaction (PCR) and by complementation of the ts phenotype with the cognate plasmid. Then the ts collection was characterized by quantitative phenotypic analysis called high content screening (HCS) using eight fluorescent markers on eight fundamental sub-cellular compartments or structures, including nucleus, DNA damage foci, mitochondria, mitotic spindle, actin patches, plasma membrane, Endoplasmic

Reticulum, and Peroxisome.

By quantifying these cell image data using an ImageXpress 5000A fluorescence microscopy system, my collaborators generated a morphological profile for every ts mutant at both permissive temperature (26 degree) and restrictive temperature (32 degree). On average, each marker has about 12 features. For each marker of a given mutant, they measured the values of the corresponding group of features from a lot of microscopy images in which each image contains a lot of cells, giving each feature a distribution of values. According to the diameter ratio between mother cell and daughter cell, we classified the cells obtained from microscopy images to 4 cell groups: large bud cells, middle bud cells, small bud cells, and unbud cells. Because small bud cells have a lot of missing values, we did not include them for analysis. Adding the features of 8 markers together, each cell group has about 100 features including general cellular parameters, such as cell shape, budding index, organelle density, as well as multiple marker-specific parameters such as Spindle Length, Spindle Fiber Length, Actin Area, Mitochondria Perimeter, Nucleus Shape Factor, Nucleus Elliptical Form Factor, Peroxisome Inner Radius, Peroxisome Outer Radius, etc. Adding the features from the 3 cell groups together, we have about 300 features in total for each ts allele in our final dataset.

By analyzing these cellular morphology data, we want to characterize how yeast essential genes affect morphology when mutated and what the functions of these essential genes are. Thereby, we aim to identify novel functions of well-characterized genes and new roles of uncharacterized genes in well-established pathways. The significance of this project is that my collaborators are the first to produce the whole-genome yeast ts alleles on the same yeast strain under the same conditions. Since yeast essential genes cannot be knocked out, by studying these ts alleles, they can generate a comprehensive map of yeast essential gene functions in vivo based on cell morphology changes obtained from microscopy image data. Moreover, ts alleles provide them with opportunities to study synthetic lethal genetic interactions (SGA) [67] within essential genes and between essential genes and non-essential genes, which significantly extends the traditional SGA technique that can only be applied to non-essential genes.

In this project, I proposed using Rank Sum (RS) test, Kolmogorov-Smirnov (KS) test [24, 28], and Kernel Density Estimation (KDE) [73] to transform this dataset over distributions of distributions to a feature matrix with one ts allele mutant having one feature vector. To perform function prediction, I used Kernel Support Vector Machine classifiers to evaluate the performance of different feature representations, and I evaluated the performance of the SVM using a combined kernel which is a convex combination of kernels based on different feature representations. I also used feature selection algorithms such as forward-backward feature selection [27], random forest [62], and recursive feature elimination method [26], to perform function prediction and informative feature extraction for these essential genes. These genes were assigned to 24 functional categories based on Gene Ontology assignment and manual curation of biological experts. Function prediction is very natural in this context because the original goal of ts allele studies is to explore the gene functions in vivo. To perform function prediction, I used RBF kernel SVMs based on different feature representations. The final results show that the features derived from the changes of mutant phenotype relative to the wild-type phenotype at 26 degree using a triple-valued RS test are the most informative ones for function prediction.

My contribution in the project is that I proposed novel data representation schemes to transform the complex microscopy image features of each ts allele to a sensible feature vector and that I can directly measure the similarities between pairwise ts alleles for function prediction. I proposed the first pipeline and method for analyzing such kind of morphology data of mutants for gene function prediction. And based on my effective data representation, I can further study which ts alleles result in the same type of abnormal phenotype.

4.2 Density Estimation Methods and Statistical Tests

Because we need to estimate the distribution of wild-type feature values and to compare the population of mutant feature values to the population of wild-type feature values, I will review some density estimation methods and statistical tests for comparing samples in this section.

4.2.1 Gaussian Mixture Models

Gaussian Mixture Model (GMM) [49], which is also called the Mixture of Gaussians, is the most commonly used parametric method for density estimation and clustering in machine learning. Given a dataset with N independent and identically distributed data points, $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}, \ldots, \mathbf{x}^{(N)}\}$, which is sampled from a Mixture of Gaussians with K components, GMM assumes the variable $\mathbf{x}$ takes the probability density function of the following form,

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma}_k), \tag{4.1}$$

where π_k is the mixing coefficient for component k, $\mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k)$ is the k-th Gaussian distribution with mean $\mu_{\mathbf{k}}$ and covariance $\mathbf{\Sigma}_k$, and $\sum_k \pi_k = 1, \pi_k \geq 0, k = 1, \ldots, K$.

The parameters of GMM $\{\pi_k, \mu_k, \mathbf{\Sigma}_k\}$ can be learned using standard Expectation Maximization (EM) algorithm [15]. EM algorithm is often used to learn the parameters of a model with latent variables in a maximum likelihood setting. It bounds the log likelihood of data with a function called free energy by assuming a probability distribution q over latent variables, and then it performs coordinate descent to minimize the free energy over the q distribution and the parameters of the model.

Use a discrete latent variable $z^{(n)}$ to indicate which component the data point n comes from, the EM algorithm for GMM iterates the following two steps:

E-step: Compute the posterior probability for each data point n,

$$p(z^{(n)} = k|\mathbf{x}^{(n)}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \mathbf{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \mathbf{\Sigma}_j)} \tag{4.2}$$

M-step: re-estimate the parameters of GMM using the newly computed posterior probabilities,

$$\mu_k^{new} = \frac{\sum_{n=1}^{N} p(z^{(n)} = k|\mathbf{x}^{(n)})\mathbf{x}^{(n)}}{\sum_{n=1}^{N} p(z^{(n)} = k|\mathbf{x}^{(n)})} \tag{4.3}$$

$$\mathbf{\Sigma}_k^{new} = \frac{\sum_{n=1}^{N} p(z^{(n)} = k|\mathbf{x}^{(n)})(\mathbf{x}^{(n)} - \mu_k^{new})(\mathbf{x}^{(n)} - \mu_k^{new})^T}{\sum_{n=1}^{n} p(z^{(n)} = k|\mathbf{x}^{(n)})} \tag{4.4}$$

$$\pi_k^{new} = \frac{\sum_{n=1}^{N} p(z^{(n)} = k|\mathbf{x}^{(n)})}{N}. \tag{4.5}$$

The EM algorithm iterates the E-step and the M-step until convergence. Although GMM has wide applicability, it's often hard to choose the optimal number of mixture components for it. For some special datasets which contains lots of data points taking several unique values, GMM often has trouble in estimating the covariance matrix.

4.2.2 Kernel Density Estimation

Kernel Density Estimation (KDE) [73, 10] is a non-parametric method that is extensively used for density estimation. It can be viewed as a smooth version of histogram method for density estimation. Suppose we have a one-dimensional dataset with N data points, in histogram methods, we partition the input space x into several bins with the same width w, and we count the number of data points N_i falling in bin i, then the probability density p_i over bin i can be approximated by,

$$p_i = \frac{N_i}{Nw}. \tag{4.6}$$

Histogram method is simple and easy to use, but it is not smooth due to coarsely divided bins. Instead of using bins, we can use a smooth kernel function to define smooth volumes around data points. Given a kernel function $\mathbf{K}$ and a positive number h called bandwidth, KDE defines the probability distribution of x as follows,

$$p(x) = \frac{1}{Nh}\sum_{n=1}^{N}\mathbf{K}(\frac{x - x^{(n)}}{h}), \tag{4.7}$$

where $\mathbf{K}$ denotes the kernel function, and a smooth Gaussian kernel with $\mathbf{K}(x) = \frac{1}{\sqrt{2\pi}}exp(-x^2)$ is often used. The bandwidth h plays an important role in smoothing: if h is too small, the estimated density function will be sensitive to noise; if h is too large, the estimated function will fail to capture the multi-modal properties of the underlying distribution. A lot of heuristic methods have been proposed for choosing h [33]. When applying KDE to estimate the wild-type feature value distributions on the morphology dataset, to estimate h, I used a heuristic method that estimates the variance of the median-subtracted residuals of log-transformed data and minimizes the L^2 risk function of the underlying probability density function of data, which is implemented in Matlab R2008b. And there are often a large number of values for each wild-type feature, choosing h becomes not crucial, which alleviates the sensitivity problem caused by the choice of bandwidth parameter [73].

4.2.3 Rank-Sum Test for Comparing Two Samples

Rank-Sum (RS) test [24, 28] is a non-parametric test for comparing two non-paired samples to check if they have the same median. Suppose we have one sample $\mathbf{X} = \{x^{(1)}, \ldots, x^{(n)}\}$ and another sample $\mathbf{Y} = \{y^{(1)}, \ldots, y^{(m)}\}$, the basic idea of the RS test is to combine X and Y together and sort the combined data with an increasing order to see if one sample is stochastically larger or smaller than the other. The RS test performs the following procedure for small n and m:

1. Let $N = n + m$, and let $R_1, \ldots, R_N$ be the rank of the combined data, with the smallest rank being 1 and the largest rank being N, and calculate the observed rank sum W of sample $\mathbf{X}$ (or $\mathbf{Y}$);

2. find all possible permutations of the ranks of size N, and assign n ranks to $\mathbf{X}$ and m

ranks to $\mathbf{Y}$;

3. For each permutation, calculate the sum of the ranks assigned to $\mathbf{X}$ (or $\mathbf{Y}$), denoted by S;

4. P-value$_{onesided}$ $= \frac{the number of times S > W}{\binom{n+m}{n}}$, and P-value$_{twosided}$ $= 2\times$ P-value$_{onesided}$.

If N is large, performing all possible permutations is impossible. According to the sampling theory from a finite population, the mean of S is $E(S) = m\mu = \frac{n(N+1)}{2}$, where $\mu = \frac{\sum_{i=1}^{N} R_i}{N} = \frac{N+1}{2}$, and the variance of S is $var(S) = \frac{mn\sum_{i=1}^{N}(R_i-\mu)^2}{N(N-1)} = \frac{mn(N+1)}{12}$. If there are k groups of ties, let t_i denote the number of observations in the i-th tied group, the adjusted variance of S is $var(S_{withties}) = \frac{mn(N+1)}{12} - \frac{mn\sum_{i=1}^{k}(t_i^3-t_i)}{12N(N-1)}$, and the mean of S remains the same. Therefore, the p-value for rejecting the hypothesis can be computed using z-score $= \frac{W-E(S)}{\sqrt{var(S)}}$ to refer a unit Gaussian distribution.

4.2.4 Kolmogrov-Smirnov Test for Comparing Two Samples

Kolmogrov-Smirnov (KS) Test [24, 28] is another non-parametric statistical test for comparing two samples. Instead of testing if one sample tends to be stochastically larger or smaller than the other as the RS test, the KS test is a test that just tells if the two samples are different for any reason. Suppose we also have the two samples $\mathbf{X}$ and $\mathbf{Y}$ as in section 4.2.3, the KS test computes the empirical distribution function (EDF) F_1 and F_2 for the two samples, where $F_1(x) = \frac{1}{n}\mathcal{I}(x_i <= x)$, and $\mathcal{I}(\cdot)$ is an indicator function, and $F_2 y$ can be computed in a similar way. The KS statistics $D_{n,m}$ is the largest distance between the two EDFs, which can be computed as follows,

$$D_{n,m} = max_w|F_1(w) - F_2(w)|. \tag{4.8}$$

The score of $\sqrt{\frac{nm}{n+m}}D_{n,m}$ follows a Kolmogrov distribution, and the p-value can be readily computed.

4.3 Data and Methods

In this section, I will describe the morphology dataset and the feature representation methods. Suppose we have a n by m non-traditional matrix $\mathcal{X}$, where n is the number of ts alleles, m is the number of morphological features, each $(i, j)-th$ component of $\mathcal{X}$ contains a distribution of values with i indexing ts alleles and j indexing features, and each row of $\mathcal{X}$ corresponds to one ts allele, then we can represent $\mathcal{X}$ by $\mathcal{X} = [\mathbf{x}^{(1)}; \ldots; \mathbf{x}^{(i)}; \ldots; \mathbf{x}^{(n)}]$, and $\mathbf{x}^{(i)} = [\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_j^{(i)}, \ldots, \mathbf{x}_m^{(i)}]$, or equivalently, $\mathcal{X}_{ij} = \mathbf{x}_j^{(i)}$, $i = 1, \ldots, n$, and $j = 1, \ldots, m$. Here, the (i, j)-th component of $\mathcal{X}$ is a vector instead of a scalar. We should note that $\mathbf{x}^{(i)}$ has different lengths for different is, and $\mathbf{x}_j^{(i)}$ might have different lengths for different js. The reason is that the cell populations carrying different ts alleles obtained from microscopy images are different, and even for the same ts allele, the cell populations for different biological markers are different. We also have an additional 1 by m similar non-traditional matrix $\mathcal{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_j, \ldots, \mathbf{w}_m]$ representing the morphological profile of wild-type cells. Figure 4.1 illustrates the described dataset above. Because the given dataset $\mathcal{X}$ is a non-traditional matrix instead of a n by m mathematical matrix, each feature of each ts allele has a distribution of values with varying size, we cannot easily measure the similarity between pairwise ts alleles, which poses challenges for function prediction analysis using machine learning methods.

In this section, I will describe several methods which convert the given complex dataset to a mathematical n by m feature matrix, so that we can easily apply machine learning methods on it for function prediction analysis. Because ts mutants often show different phenotype rather than wild-type phenotype, in mathematical terms, the feature values of ts mutants often have different distributions compared to the corresponding feature value distribution of wild-type cells, I propose using statistical tests and density estimation methods described in section 4.2 to quantify differences between ts mutant feature distributions and wild-type feature distributions. In specific, I will use Rank-Sum (RS) test and Kolmogorov-Smirnov (KS) test to directly compare mutant

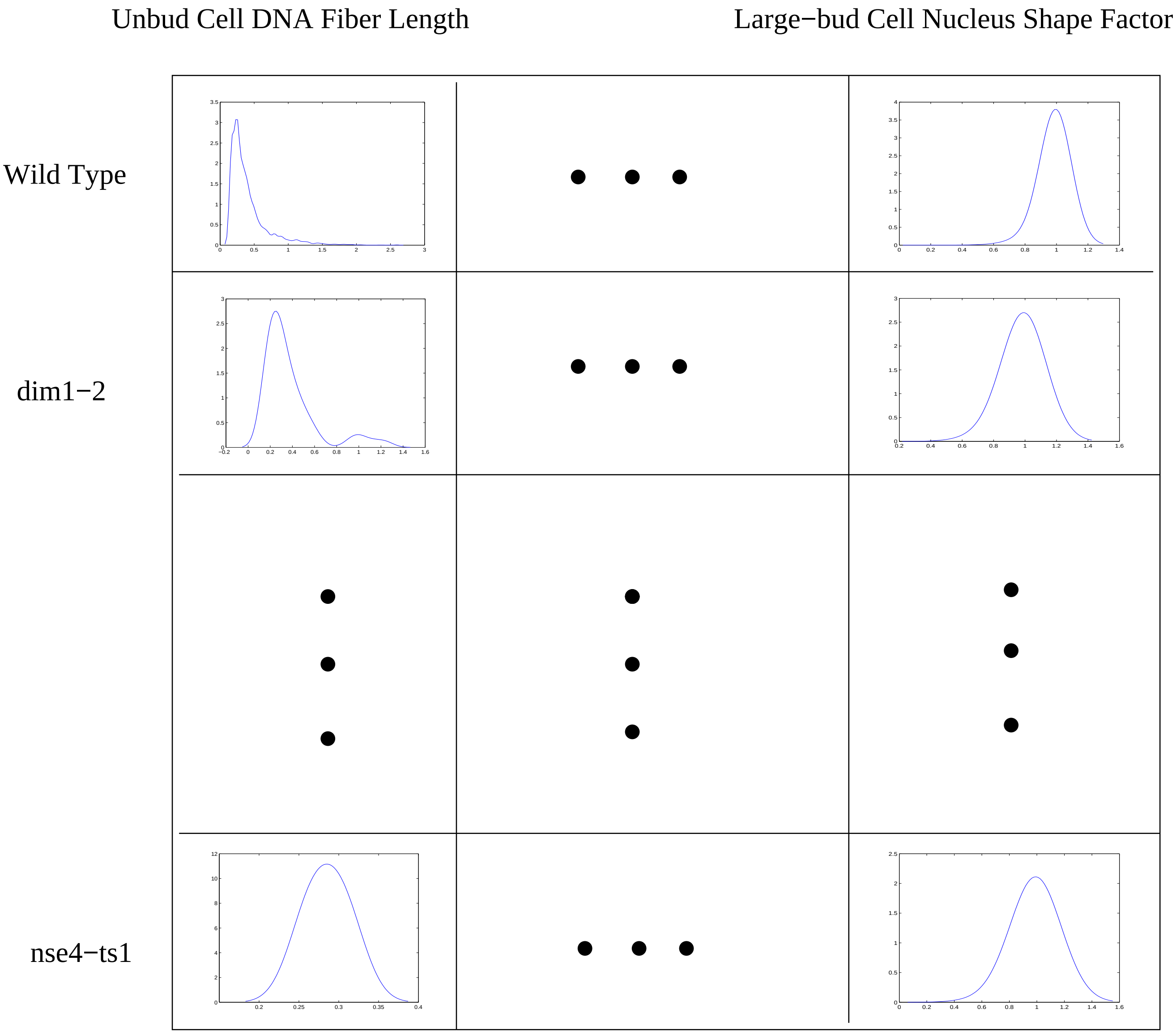


Figure 4.1: This figure shows the non-traditional matrix containing the values of different features for wild-type cells and different temperature-sensitive mutants. The curve in each grid illustrates the probability density function estimated from the distribution of values of the corresponding feature (column) for the corresponding temperature-sensitive mutant or wild type (row).

feature distributions to wild-type feature distributions, and I will evaluate the average log-probabilities of mutant feature values under the probability density function of corresponding wild-type feature values estimated using Kernel Density Estimation (KDE).

4.3.1 Calculate Features Using Rank-Sum Test

Algorithm 2 *The algorithm for calculating feature matrices using Rank-Sum test and Kolmogorov-Smirnov test.*

Input*: the profiles of ts mutants* $\mathcal{X}$ *and the wild-type profile* $\mathcal{W}$ *described in section 4.3, the number of ts mutants* n*, and the number of features* m*.*

Output*: the feature matrix* $\mathbf{F}_{rs}$ *based on the RS test and the feature matrix* $\mathbf{F}_{ks}$ *based on the KS test.*

```
For ts mutant $i$ = 1, ..., $n$:
    For morphology feature $j$ = 1, ..., $m$:
        rs_pvalue = Rank-Sum test($\mathbf{x}_j^{(i)}$, $\mathcal{W}_j$);
        ts_median = median ($\mathbf{x}_j^{(i)}$);
        wt_median = median ($\mathcal{W}_j$);
        $\mathbf{F}_{rs}(i,j) = \mathcal{I}(rs\_pvalue \leq 0.05) * sign(ts\_median - wt\_median)$;
        ks_pvalue = Kolmogorov-Smirnov test($\mathbf{x}_j^{(i)}$, $\mathcal{W}_j$);
        $\mathbf{F}_{ks}(i,j) = \mathcal{I}(ks\_pvalue \leq 0.05)$;
    End For loop over $j$;
End For loop over $i$;
Return $\mathbf{F}_{rs}$ and $\mathbf{F}_{ks}$.
```

Algorithm 2 describes how to compute a n by m feature matrix using the RS test from the given the non-traditional matrix $\mathcal{X}$ and wild-type feature data $\mathcal{W}$ in details, where $\mathcal{I}$ is an indicator function, and $sign(a) = +1$ if $a > 0$, and -1 otherwise. In fact, we choose the RS test because it is a non-parametric distribution-assumption-free test that can effectively test whether two population sets of data points are from the same underlying distribution. Moreover, instead of giving a binary test result, I also compare the median of mutant feature values to that of wild-type feature values to produce a triple-valued test result, which reflects whether each ts mutant manifests elevated feature values (+1), or reduced feature values (-1), or no change (0), relative to the corresponding wild-type feature values. Based on this triple-valued representation, only pairwise ts mutants that manifest the same direction of phenotype changes will have high similarities, and pairwise ts mutants that both have abnormal phenotype but manifest different directions of phenotype changes will have low similarities, which is different from what we get using a binary representation. Experimental results later in this chapter will show that the triple-valued representation is much better for function analysis than the binary representation using RS test.

4.3.2 Calculate Features Using Kolmogorov-Smirnov Test

Besides RS test, Kolmogorov-Smirnov (KS) test reviewed in 4.2 is another non-parametric distribution-assumption-free test for comparing two samples. Algorithm 2 shows how to compute a binary feature matrix from the given dataset using KS test. Although KS test can also be used here to generate feature representation, I believe that it is not appropriate for this particular dataset. This is because the sample size of wild-type feature values and that of mutant feature values differ a lot for some ts mutants. In details, we often collect feature values for a lot of features from more than 10,000 wild-type cells, but we often collect feature values for many features from below 50 cells carrying some particular ts mutants, for which Figure 4.2 gives an example. Due to the dramatic sample size difference, the KS test will simply output significant difference although the mutant feature values are just like sampled values from the wild-type feature distribution. And for some special ts mutants, the sample size of mutant feature values is very small, which will cause problems for estimating empirical cumulative distribution

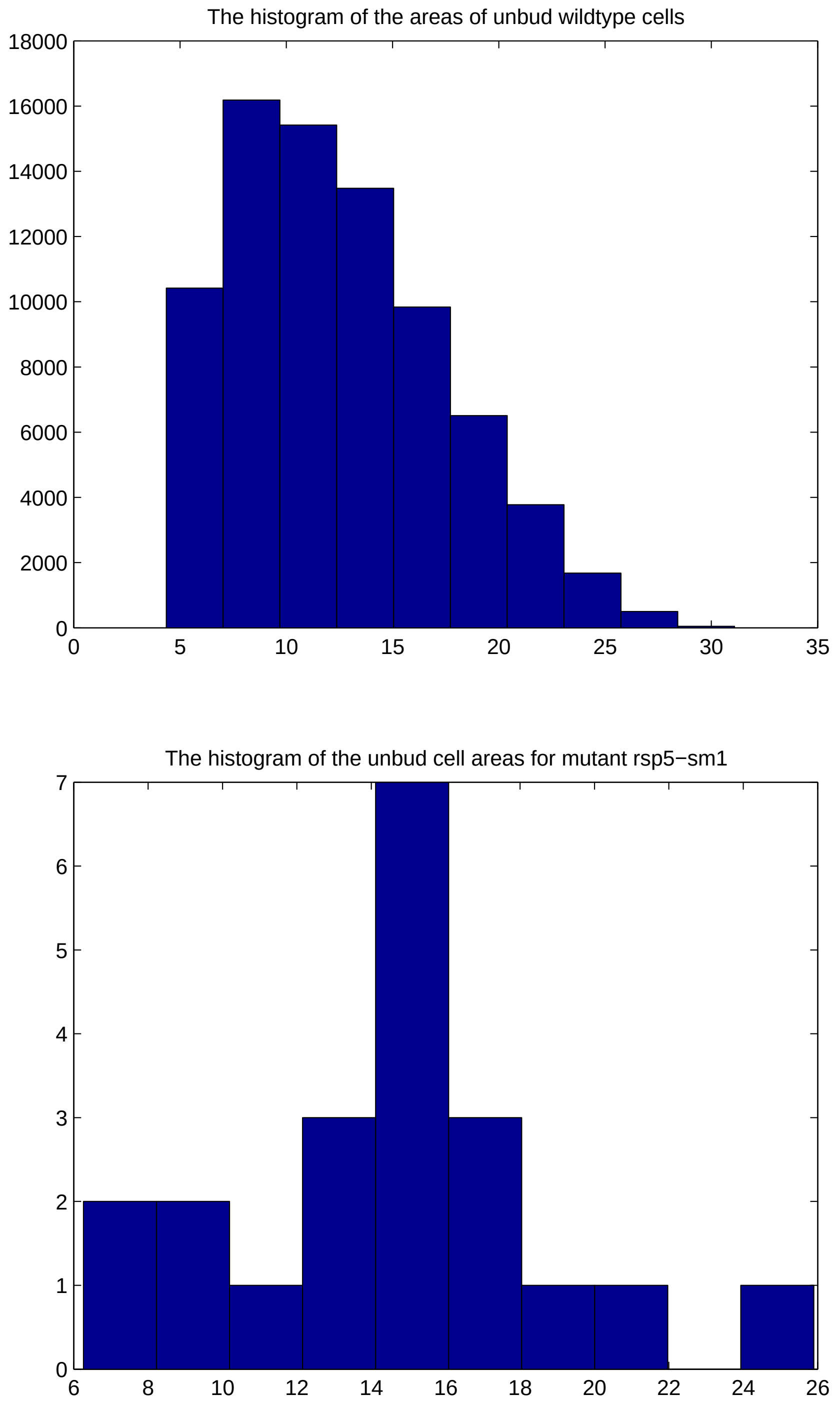
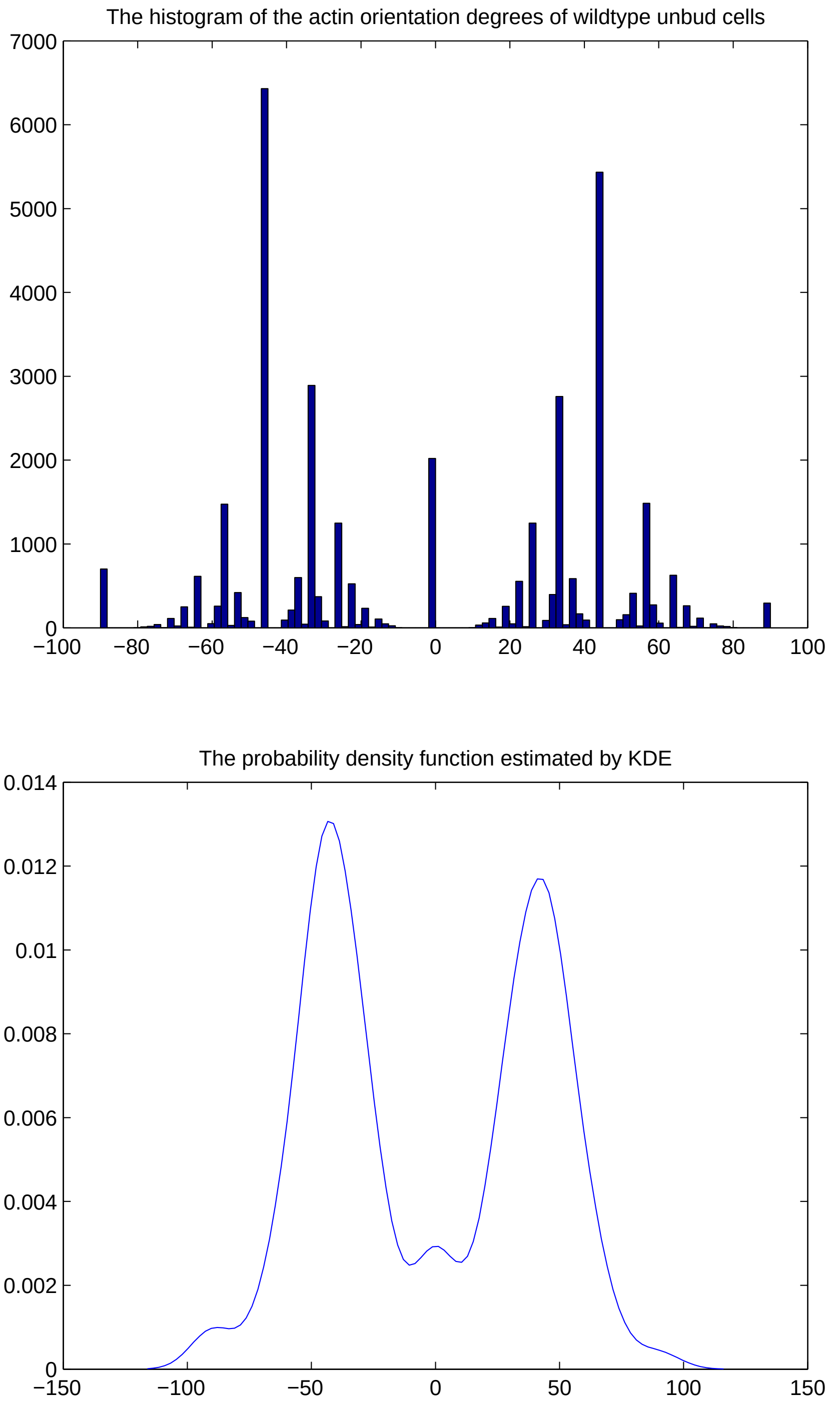


Figure 4.2: This figure shows the histogram of the areas of unbud wild-type cells (the top one) and the histogram of the areas of unbud cells carrying ts allele rsp5-sm1 (the bottom one). From this figure, we can see that, for the feature "Unbud Cell Area", there are above 60,000 wild-type sample cells available for measuring the wild-type feature values, but there are only about 20 sample cells carrying the ts allele available for measuring the mutant feature values.


The histogram of the actin orientation degrees of wildtype unbud cells

The probability density function estimated by KDE

Figure 4.3: This figure presents the histogram of the actin orientation degrees of unbud wild-type cells (the top one) and the corresponding probability density function of the wild-type feature values estimated by KDE (the bottom one). This figure clearly shows that the wild-type feature values here have a multi-modal distribution, which violates the assumption of applying a t-test.

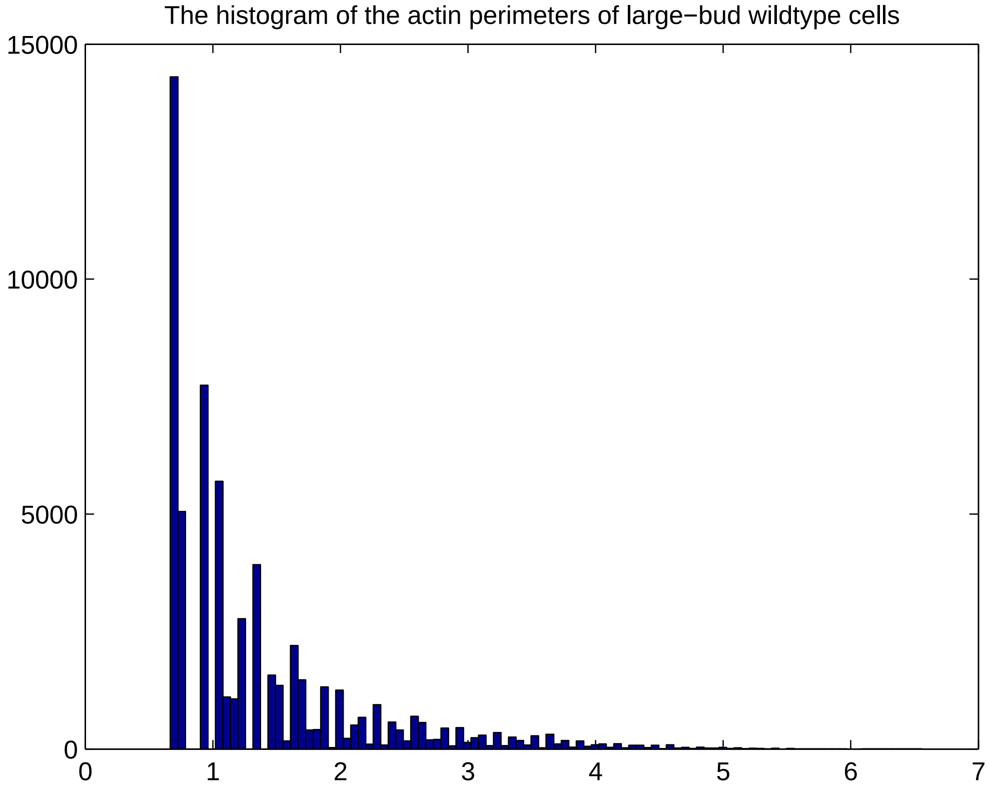


Figure 4.4: This figure plots the histogram of the actin perimeters of large-bud wild-type cells. Although there are a lot of wild-type large-bud cells measured here, the total number of unique numerical values is very small, which will cause problems for density estimation using GMM in a brute-force way.

thereby cause problems for KS test.

Some other researchers might want to use student t-test for comparing samples to generate features for ts mutants. However, t-test requires the assumption that both the mutant feature values and the wild-type feature values have Gaussian distribution, which is often violated in our problem. Figure 4.3 gives such an example. It shows the multi-modal distribution of the perimeters of large-bud wild-type cells.

4.3.3 Calculate Features Through Kernel Density Estimation

The computed features of ts mutants should reflect how different the mutant feature value distribution is from the corresponding wild-type feature value distribution. Besides using statistical tests to directly quantify the difference between the two distributions, we can use an alternative approach to quantify how the mutant feature values are distributed under the wild-type feature distribution. We can use the kernel density estimation methods described in section 4.2 to estimate the wild-type feature distributions, then either we can compute the proportion of deviant values in the sample of mutant feature values, where the deviant values refer to the values with cumulative probability ≥ 0.95 or ≤ 0.05 under the wild-type distribution, or we can compute the average log-probabilities of the mutant feature values under the wild-type distribution.

Gaussian Mixture Model (GMM) described in section 4.2 is a commonly used approach to density estimation, but it cannot be directly applied to our dataset to estimate wild-type feature distributions. The reason is that, in our dataset, a lot of wild-type cell feature values take a single numerical value, that is to say, although the population size of wild-type feature values for many features is above 10,000, the number of unique numerical values is very small, say, below 100. Figure 4.4 shows such an example. For these cases, GMM often has components trapped into regions near some highly-popular unique values, which will cause numerical problems for estimating variances.

Therefore, I turn to Kernel Density Estimation (KDE) for estimating wild-type feature distributions. I used the default heuristics procedure implemented in Matlab for estimating the bandwidth parameter in KDE. Because the population size of wild-type feature values is often very large, above 10,000, the final obtained distribution by KDE will not be sensitive to the chosen bandwidth parameter. Algorithm 3 describes how to compute the feature representation from the given dataset through KDE in details. In this algorithm, I also normalize the proportion of deviants and the average of log-probabilities to get the final feature matrix $\mathbf{F}_{dev}$ and $\mathbf{F}_{prob}$ so that each feature column has mean 0 and standard deviation 1. I also tried using KDE to estimate both wild-type feature distributions and mutant feature distributions, and then I used the KL-divergence between the corresponding two distributions as feature value. However, due to the small size of the feature value population for many features of some ts mutants, I cannot estimate the mutant feature distribution in a robust way, which will make the computed KL-divergence meaningless.

Algorithm 3 *The algorithm for calculating feature matrices through Kernel Density Estimation.*

Input*: the profiles of ts mutants $\mathcal{X}$ and the wild-type profile $\mathcal{W}$ described in section 4.3, the number of ts mutants n, and the number of features m.*

Output*: the feature matrix $\mathbf{F}_{dev}$ based on the proportion of deviants and the feature matrix $\mathbf{F}_{prob}$ based on the average log-probabilities.*

```
1. For morphology feature $j$ = 1, ..., $m$:
2. Estimate the probability density function $p$ and the cumulative distribution function $Pr$ for the population of wild-type feature values $\mathcal{W}_j$ using KDE;
3. For ts mutant $i$ = 1, ..., $n$:
4. $\mathbf{F}_{prob}(i,j)$ = average(log($p(\mathbf{x}_j^{(i)})$));
5. Estimate the cumulative probabilities of mutant feature values under $Pr$ using $\mathbf{v} = Pr(\mathbf{x}_j^{(i)})$;
6. $\mathbf{F}_{dev}(i,j)$ = the proportion of the elements in $\mathbf{v}$ with value $\leq 0.05$ or $\geq 0.95$;
7. End For loop over $i$
8. End For loop over $j$
9. Normalize $\mathbf{F}_{prob}$ and $\mathbf{F}_{dev}$ so that each column has mean 0 and standard deviation 1;
10. Return $\mathbf{F}_{prob}$ and $\mathbf{F}_{dev}$.
```

4.4 Results on Gene Function Prediction

In this section, at first, I will present experimental results for gene function prediction using the morphology profile of ts mutants at 26 degree (permissive temperature) and 32 degree (restrictive temperature) based on the feature representations calculated by the RS test, the KS test, and KDE, respectively, and I will perform comparisons between the results obtained using different feature representations and different degrees; then I will present the experimental results based on the combined features including all the feature representations described above; finally, I will use feature selection method to investigate the respective informative features for each functional category.

For all the gene function prediction experiments described in this chapter, there are 775 ts mutants and 293 features altogether. But a lot of ts mutants have missing values over many features, so I removed these ts mutants for the consideration of prediction analysis. After the preprocessing, there are 644 ts mutants remaining at 26 degree, and 636 ts mutants remaining at 32 degree. These two sets of ts mutants almost completely overlap. Based on Gene Ontology and my collaborators' manual curation, I assigned these ts mutants into 24 functional categories with each category containing more than 8 ts mutants. Table 4.1 shows all the 24 functional categories in details, and I will use the ranked number in the table to denote each functional category in the subsequent description of this chapter, and Table 4.2 shows the ts mutants without missing values contained in some functional categories in the final morphology dataset at 26 degree.

To enable consistent comparisons between different feature representations, I randomly split the ts mutants in each functional category without missing values at 26 degree into a positive training set containing two-thirds of the ts mutants and a positive test set containing the remaining one-third. All the other ts mutants outside each functional category were split into either a negative training set or a negative test set, also with two-thirds of them for training and one-third of them for test. I call the resulting dataset obtained by this splitting "Fixed-26" in the subsequent presentation of

Function No.	Function Name
1.	Protein Degradation
2.	Chromatin Structure and Maintenance
3.	Cell Cycle Regulation
4.	Ribosomal Biogenesis and Organization
5.	Actin Cytoskeleton
6.	Protein Modification
7.	Protein Targeting
8.	Mitosis
9.	Vesicle Mediated Transport
10.	Filamentous Growth
11.	Secretory pathway
12.	Translation
13.	RNA Splicing
14.	Microtubule Cytoskeleton
15.	Pol II transcription
16.	RNA Processing
17.	Cell Wall Biogenesis and Organization
18.	DNA Repair and Replication
19.	Meiosis
20.	Cytokinesis
21.	Transcription
22.	mitotic spindle organization and biogenesis in nucleus
23.	Lipid Metabolic process
24.	Mitochondrial Biogenesis and Organization

Table 4.1: The list of 24 functional categories that contain more than 8 ts alleles in each category.

Function No.	ts mutants
3.	cdc53-1, cdc34-2, cdc34-1, ubc9-2, sfi1-7, sfi1-3, ulp1-333, zpr1-1, cdc123-4
7.	sec63-1, sec62-ts, srp101-47, sec53-6, sec11-2, srp102-510, spc3-4, sec61-2, sec65-1
10.	STE4, cdc24-H, cdc24-5, cdc24-4, cdc24-3, cdc24-2, cdc24-11, cdc24-1, cdc42-1 srv2-ts, srv2-2
12.	cdc39-1, cdc36-16, mot1-1033, nab3-11, gcd1-502, ils1-1, efb1-4, yef3-F650S, rpg1-1, hyp2-ts, hyp2-3, hyp2-2, hyp2-1, cdc33-E72G, ded1-F144C, ded1-95, ded1-199, prt1-1
17.	gpi8-ts, gaa1-ts, gab1-3, gab1-2, gab1-1, gfa1-97, pkc1-ts, pkc1-4, pkc1-3, pkc1-2, pkc1-1, cdc10-5, cdc10-4, cdc10-2, cdc10-1, tsc11-7, tsc11-5, tsc11-1, act1-4, act1-3, act1-2, act1-159, act1-155, act1-136, act1-133, act1-132, act1-129, act1-125, act1-124, act1-122, act1-121, act1-120, act1-119, act1-112, act1-111, act1-108, act1-105, act1-101, cdc12-td, cdc12-1, cdc11-5, cdc11-4, cdc11-3, cdc11-2, cdc11-1, cdc3-3, cdc3-1, rho1-td, gpi19-2-XH, gpi19-2, gpi1-1, gwt1-20, mcd4-174, gpi13-5, gpi13-4, gpi13-3, smp3-2, smp3-1, gpi2-774, gpi2-1-7B, spt14-1-10C
24.	mas2-10, mas1-1, ole1-m2, arc15-10, tim22-19, tim44-8, pam16-3, pam18-1, mge1-100, sam35-2

Table 4.2: The final ts alleles without missing values contained in some functional categories in the morphological profile dataset at 26 degree.

Feature Representations	The Mean ROC Score	The Mean ROC_{50} Score
The triple-valued RS test	**0.74**	**0.37**
The binary RS test	0.59	0.22
The KS Test	0.62	0.23
The average log-probabilities (KDE)	0.67	0.32
The proportion of deviant values (KDE)	0.66	0.26

Table 4.3: The overall mean ROC scores and the overall mean ROC_{50} scores over 24 functional categories obtained by kernel SVMs based on different feature representations of the morphological profiles of ts mutants at 26 degree.

this chapter. I used linear SVM and RBF Kernel SVM to predict the functions of the ts mutants in the constructed test sets, and I used ROC score and ROC_{50} score described in chapter 2 to evaluate the performance of SVM classifiers based on different feature representations. The free parameter C in SVM and the variance parameter σ in RBF kernel were chosen by cross validation.

Because different ts mutants of the same essential gene are constructed by mutating the gene at different locations, they may cause very different phenotype. Therefore, it's reasonable to put multiple ts mutants in the same functional category for random training/test splitting for function prediction. To eliminate possible bias, I am doing additional experiments for function prediction by not allowing different ts mutants of the same essential gene to be split into both training set and test set.

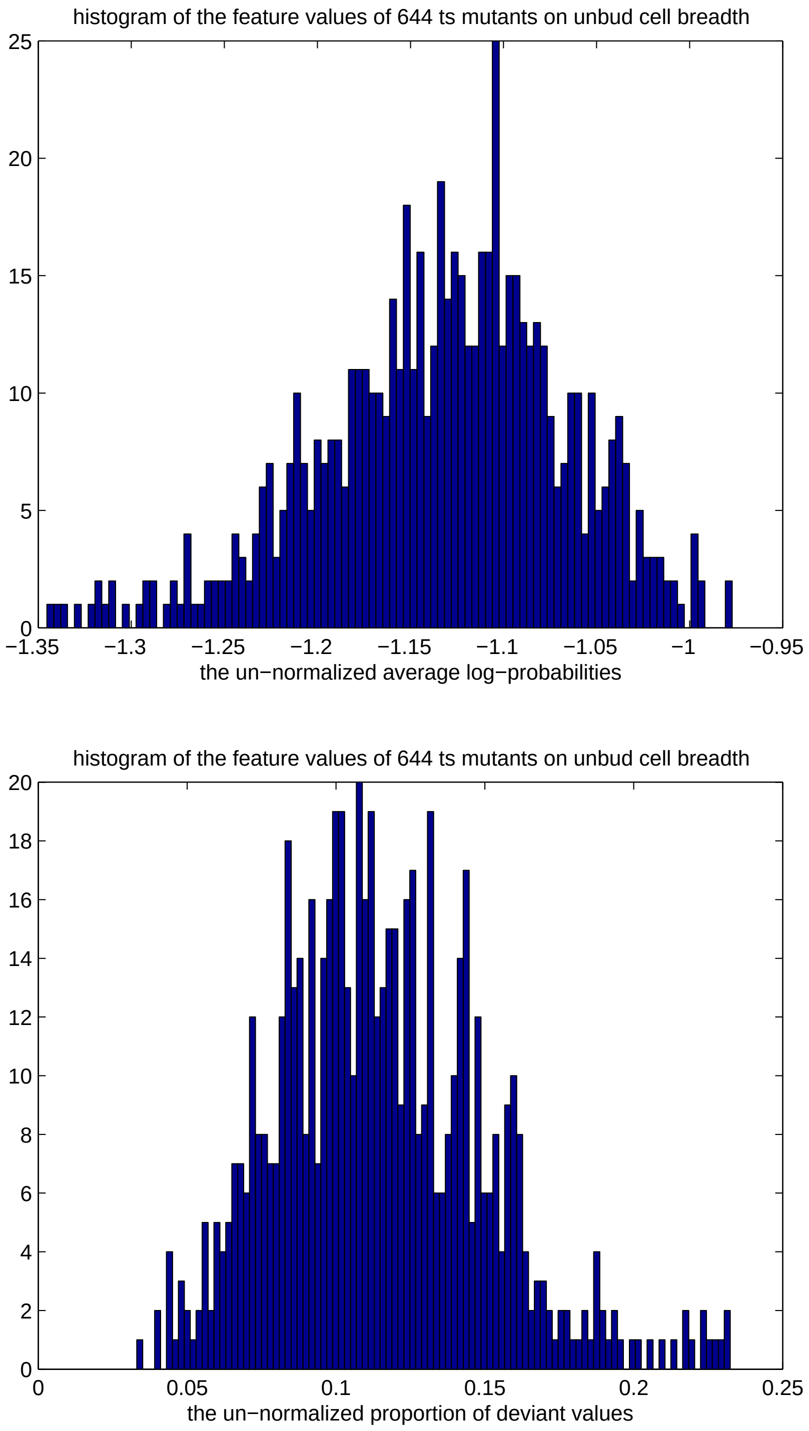


Figure 4.5: This figure shows the histograms of the feature values of all the ts mutants on the feature "UnBud Cell Breadth" at 26 degree, calculated using the average log-probabilities of mutant unbud cell breadths under the probability density function of the wild-type unbud cell breadths (the top sub-figure) and the proportion of deviant mutant unbud cell breadths under the wild-type feature value distribution (the bottom sub-figure).

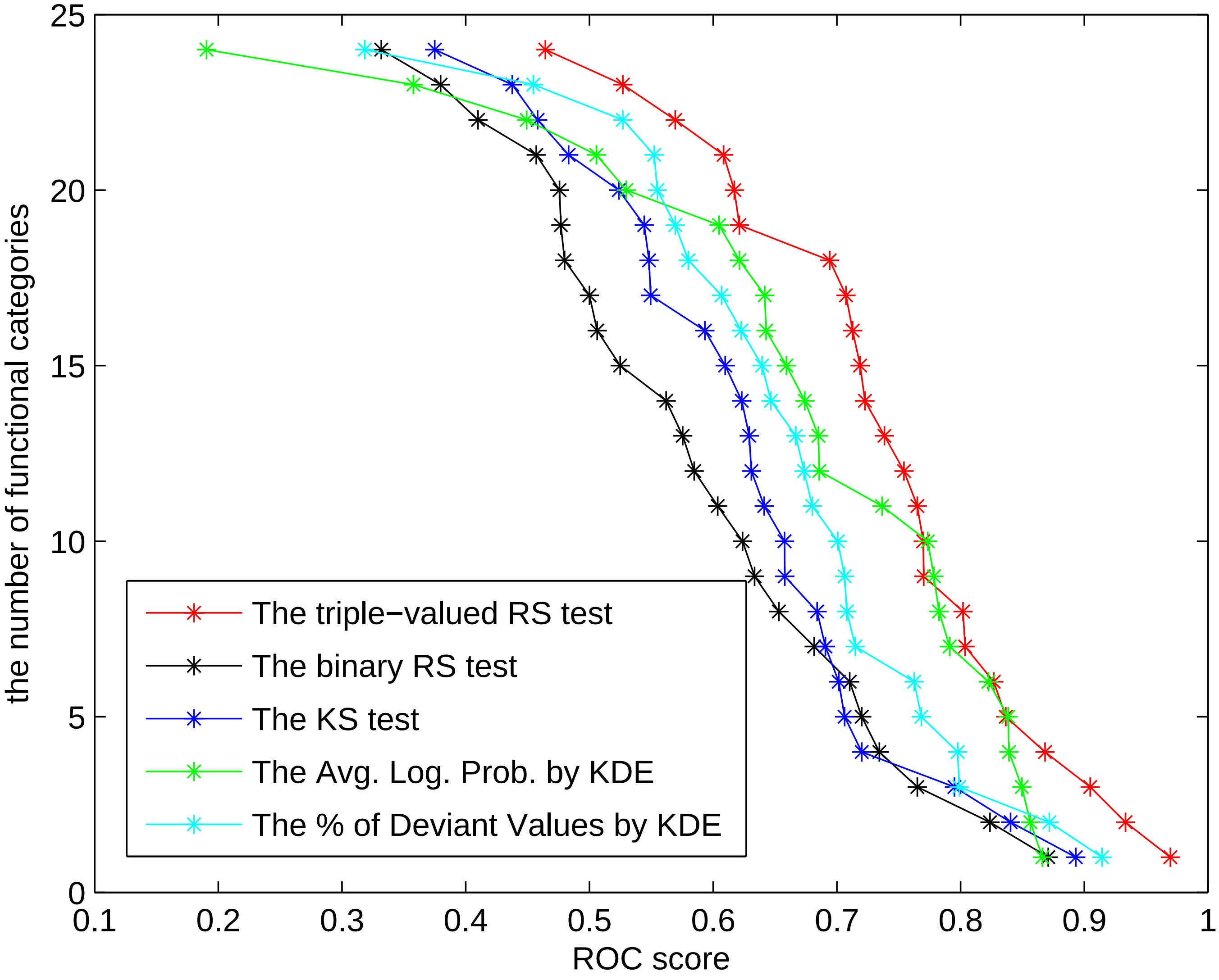


Figure 4.6: The number of functional categories with ROC scores above different thresholds using RBF kernels based on different feature representations.

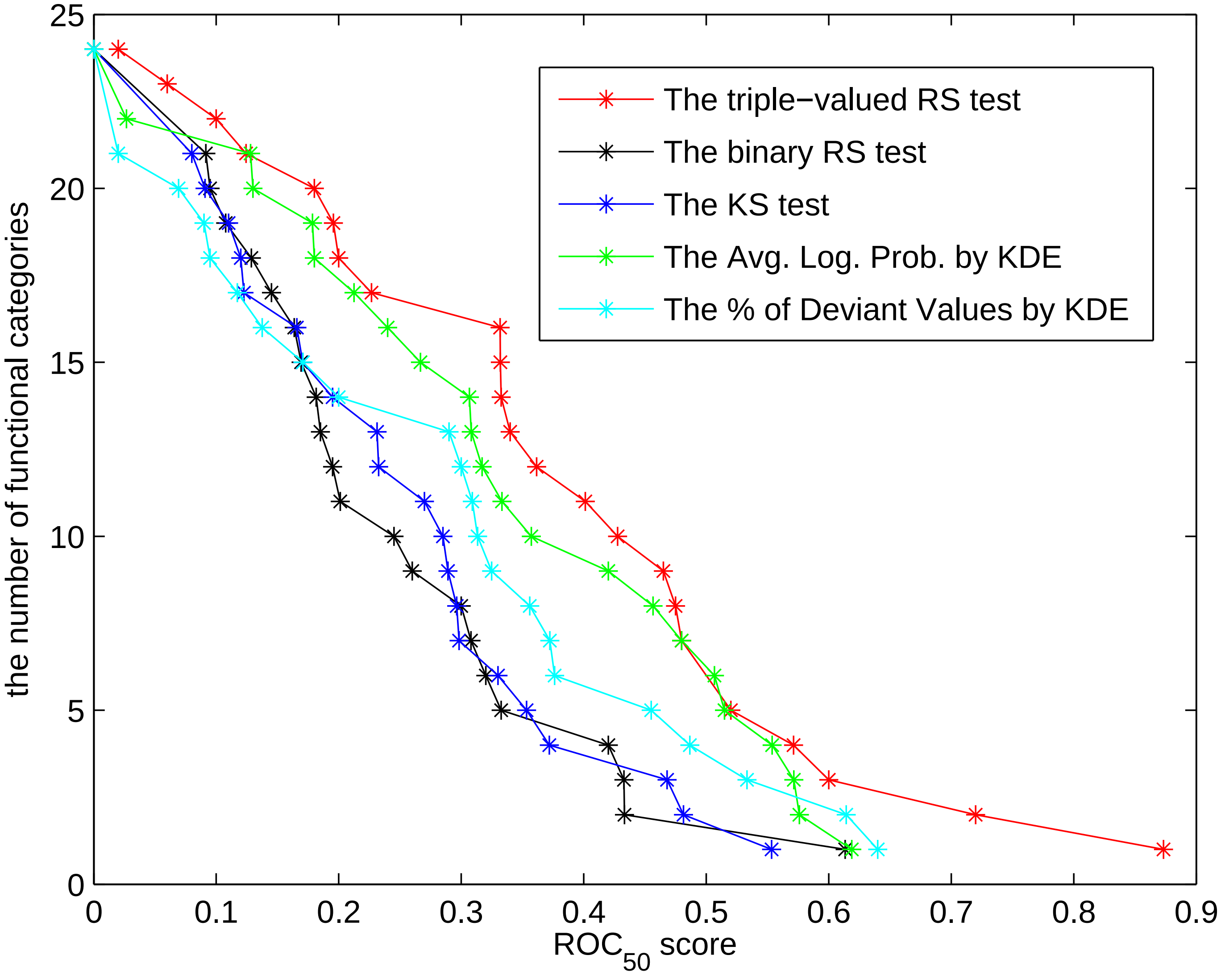


Figure 4.7: The number of functional categories with ROC_{50} scores above different thresholds using RBF kernels based on different feature representations.

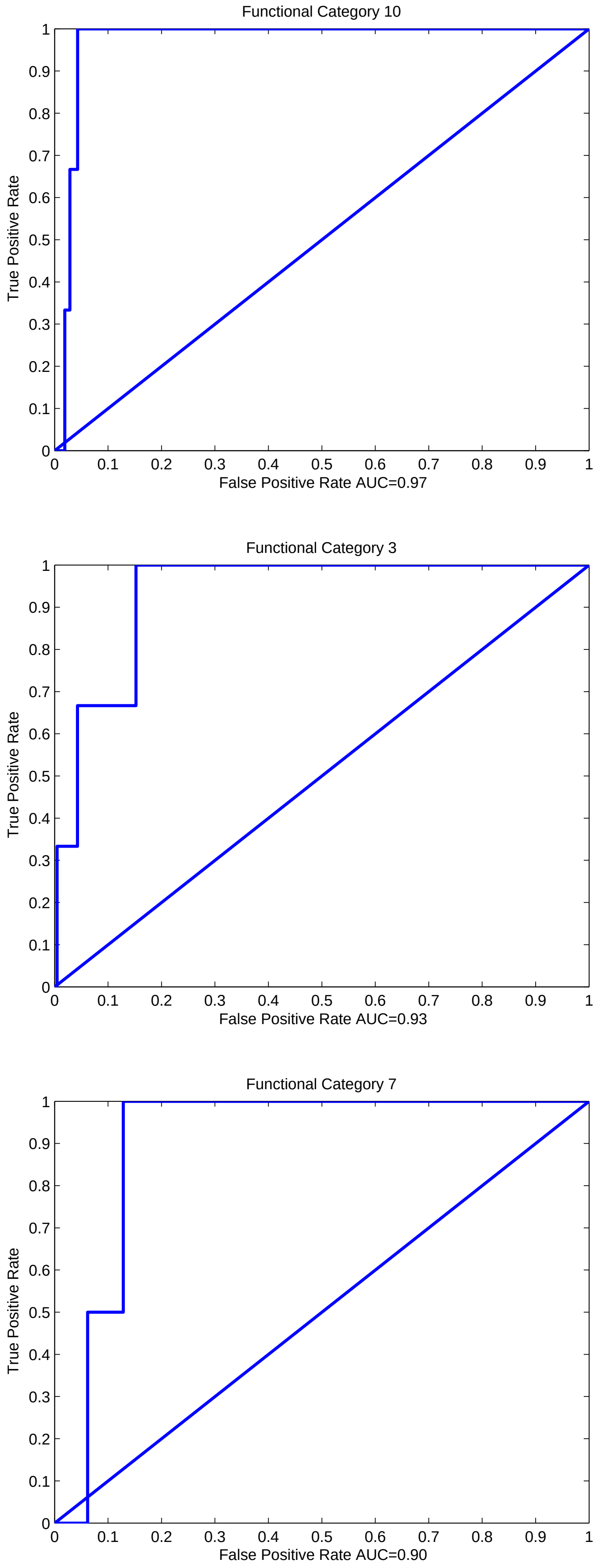


Figure 4.8: The ROC curves with the top 3 ranked ROC scores.

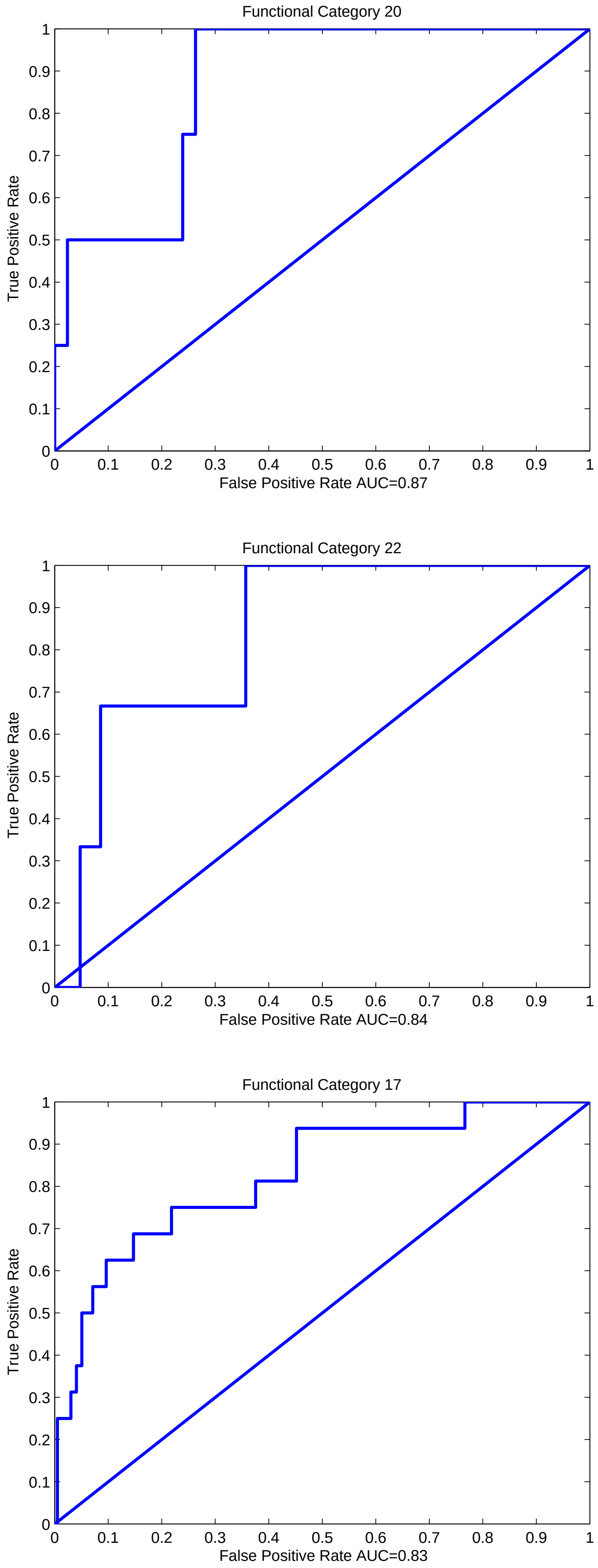


Figure 4.9: The ROC curves with the ROC scores ranked from 4 to 6.

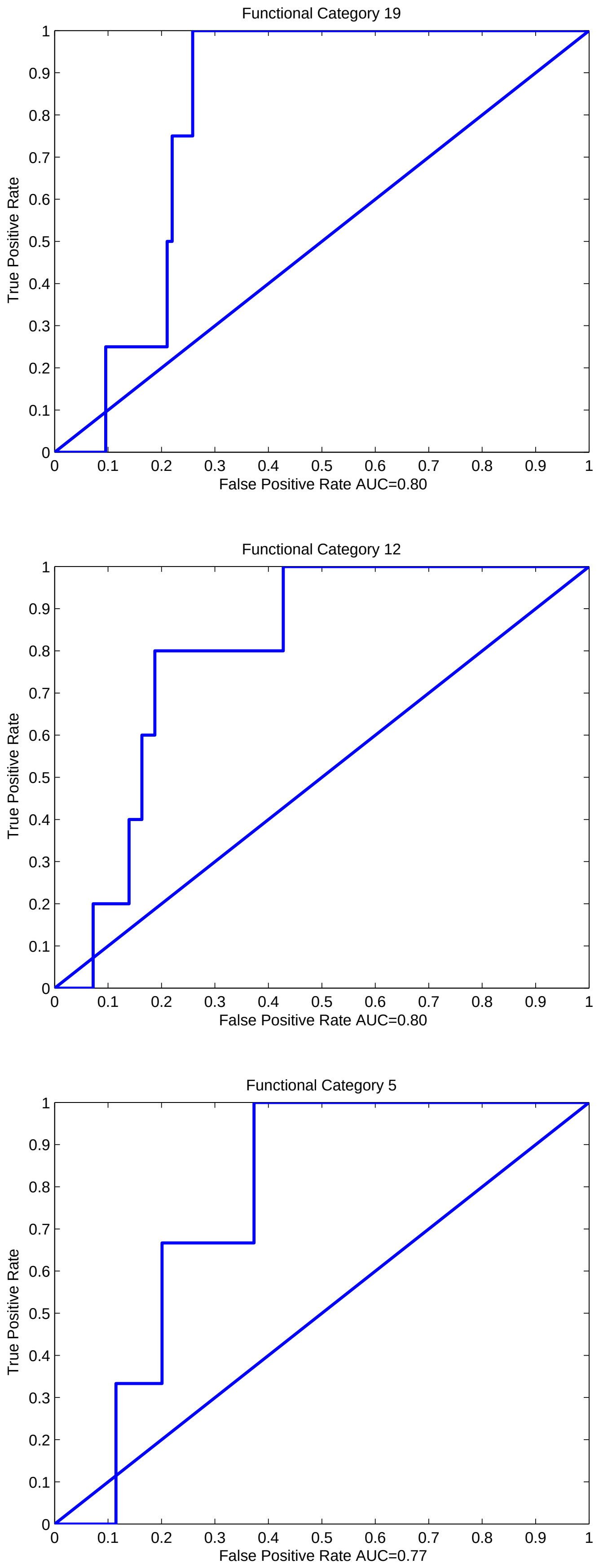


Figure 4.10: The ROC curves with the ROC scores ranked from 7 to 9.

4.4.1 Experimental Results Using RS Test, KS Test, and KDE

In this section, I will present the experimental results on gene function prediction using different feature representations. The top sub-figure and the bottom sub-figure in Figure 4.5, respectively, show the histogram of the feature values of 644 ts mutants on the feature "UnBud Cell Breadth" at 26 degree, calculated, respectively, using the average log-probabilities and the proportion of deviant values. Applying RBF kernel SVM based on different feature representations to the 26 degree morphology dataset, the mean ROC scores and the mean ROC_{50} scores over 24 functional categories obtained are summarized in Table 4.3. Figure 4.6 and Figure 4.7, respectively, show the number of functional categories that score above different threshold values of, respectively, ROC score and ROC_{50} score using the RBF kernels based on different feature representations (note that they are not ROC curves but the summarization of all the ROC and ROC_{50} scores).

We can see that the triple-valued RS test has the best performance among all the feature representations. Based on the test, out of 24 functional categories, there are 18 functional categories with ROC score above 0.65, and there are 12 functional categories with ROC score above 0.75. The 12 functional categories are Cell Cycle Regulation, Ribosomal Biogenesis and Organization, Actin Cytoskeleton, Protein Targeting, Vesicle Mediated Transport, Filamentous Growth, Translation, Nucleocytoplasmic transport, Cell Wall Biogenesis and Organization, Meiosis, Cytokinesis, mitotic spindle organization and biogenesis in nucleus, and Mitochondrial Biogenesis and Organization, which are closely related to the cell structures that carry fluorescence markers. Figure 4.8-4.10 show the ROC curves for the functional categories with ROC scores among the top 9 ranked ones using the triple-valued RS test. In contrast, on the same dataset, the mean ROC score and the mean ROC_{50} score obtained using the binary RS test are much worse than those obtained by the triple-valued RS test, and the results obtained using the KS test are close to those obtained by the binary RS test. The superior performance of triple-valued RS test clearly demonstrates the importance of distinguishing elevated feature deviations from reduced feature deviations.

To further demonstrate that the superior performance of the triple-valued RS test to that of the KS test is not random, I performed the following test: for each functional category, I randomly split $\frac{2}{3}$ ts mutants for training and $\frac{1}{3}$ ts mutants for test using the same splitting strategy for constructing ”fixed-26”, and then I calculated the ROC scores obtained by the RBF kernel SVM based on, respectively, different feature representations, and I repeated this procedure 100 times. The mean ROC scores obtained by the triple-valued RS test, the binary RS test, the KS test, the average log-probabilities of mutant feature values, and the proportion of deviant mutant feature values are, respectively, 0.70, 0.59, 0.60, 0.61, and 0.62, which again demonstrates the advantage of using the triple-valued RS test.

To determine whether one feature representation is statistically significantly better than another, I performed Wilcoxon Matched-Pairs Signed-Ranks Tests on the ROC score and ROC$_{50}$ score differences between pairwise feature representations. The resulting p-value for the ROC score difference between the triple-valued RS test and the binary RS test, the KS test, the proportion of deviant values, the average of log-probabilities, are, respectively, $2.67e-05$, $1.15e-04$, 0.016, and 0.17. The resulting p-value for the ROC$_{50}$ score difference between the triple-valued RS test and the binary RS test, the KS test, the proportion of deviant values, the average of log-probabilities, are, respectively, $6.32e-05$, $1.70e-03$, 0.032, and 0.46. Except the pairs mentioned above and the ROC and ROC$_{50}$ score differences between the average log-probabilities and the binary RS test, the score differences between any other pairs are not statistically significant. These statistical tests again show that the triple-valued RS test is the best feature representation.

Because the ts mutants respectively appearing in the 26 degree dataset and in the 32 degree dataset almost completely overlap, I mapped the training and test ts mutants for each functional category used in ”fixed-26” to the 32 degree dataset to make comparisons possible. Table 4.4 presents the results obtained using RBF kernel SVMs based on

Feature Representations	The Mean ROC Score	The Mean ROC_{50} Score
The triple-valued RS test	**0.60**	**0.29**
The binary RS test	0.58	0.22
The KS Test	0.57	0.22
The average log-probabilities (KDE)	0.59	0.23
The proportion of deviant values (KDE)	0.56	0.21

Table 4.4: The overall mean ROC scores and the overall mean ROC_{50} scores over 24 functional categories obtained by kernel SVMs based on different feature representations of the morphological profiles of ts mutants at 32 degree.

different feature representations of the morphological profiles of ts mutants at 32 degree. Again, we see that triple-valued RS test performs best. Comparing the results here to the results in Table 4.3, we can see that the morphology profiles of ts mutants at the restrictive temperature (32 degree) do not have as good predictive power as those at the permissive temperature (26 degree). I tried to combine the features extracted from the 26 degree profiles and the features extracted from the 32 degree profiles, but the results obtained did not improve upon those obtained using the 26 degree features alone.

4.4.2 Experimental Results Using Combined Features based on Multiple Kernel Learning

From the results described in the previous sections of this chapter, we can see that the triple-valued RS test has the best performance on function prediction, and the feature representations calculated by KDE also have reasonably good performance. Are different feature representations complement to each other? To answer this question, I propose using a convex combination of RBF kernels based on the four different feature representations for function prediction, which is analogous to the convex combination of random-walk kernels described in chapter 2. Therefore, the optimization problem here is almost the same as what I described in chapter 2, and I also used MOSEK to solve the resulting QCQP problem. Because Cross Validation (CV) set the variance parameter σ in the previously used RBF kernels either to 5 or 15, I used a convex combination of 8 RBF kernels, in specific, the RBF kernels based on the feature representations by the triple-valued RS test, the KS test, the average log-probabilities of mutant feature values, and the proportion of deviant mutant feature values, with $\sigma = 5$ and $\sigma = 15$.

The overall mean ROC score and the overall mean ROC$_{50}$ score obtained by the combined kernels are, respectively, 0.73 and 0.38, which are almost the same as the results produced by the best feature representation using the triple-valued RS test. Figure 4.11 shows the number of functional categories that score above different threshold values of ROC score and ROC$_{50}$ score using the combined kernel and the RBF kernel based on the triple-valued RS test. This figure also shows that the combined kernel performs comparably to the RBF kernel based on the best feature representation. I performed Wilcoxon Matched-Pairs Signed-Ranks Tests on the ROC score and ROC$_{50}$ score differences between the combined kernel and the RBF kernel based on the triple-valued RS test, and there is no significant difference between them.

Figure 4.12 presents the combination coefficients of the 8 kernels in 24 functional categories. Surprisingly, the kernel learning algorithm often places dominant weights on the kernels based on the feature representations calculated through KDE instead of the triple-valued RS test to construct the optimal kernel. However, the resulting kernel has comparable performance to that of the kernel based on the triple-valued RS test, which clearly shows the usefulness of the feature representations calculated by KDE.

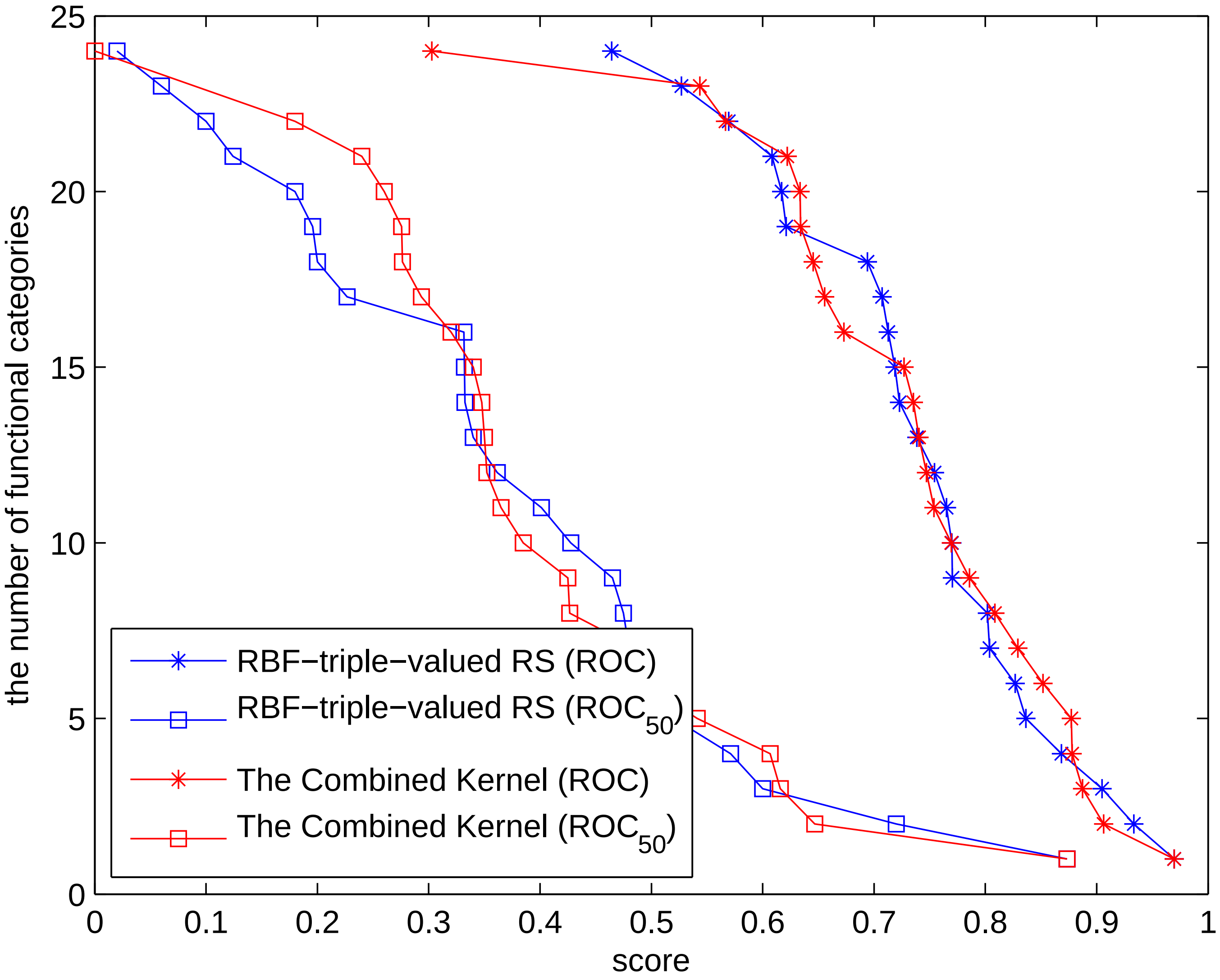


Figure 4.11: The number of functional categories with ROC scores and ROC$_{50}$ scores above different thresholds using the combined kernel and the RBF kernel based on the triple-valued RS test.

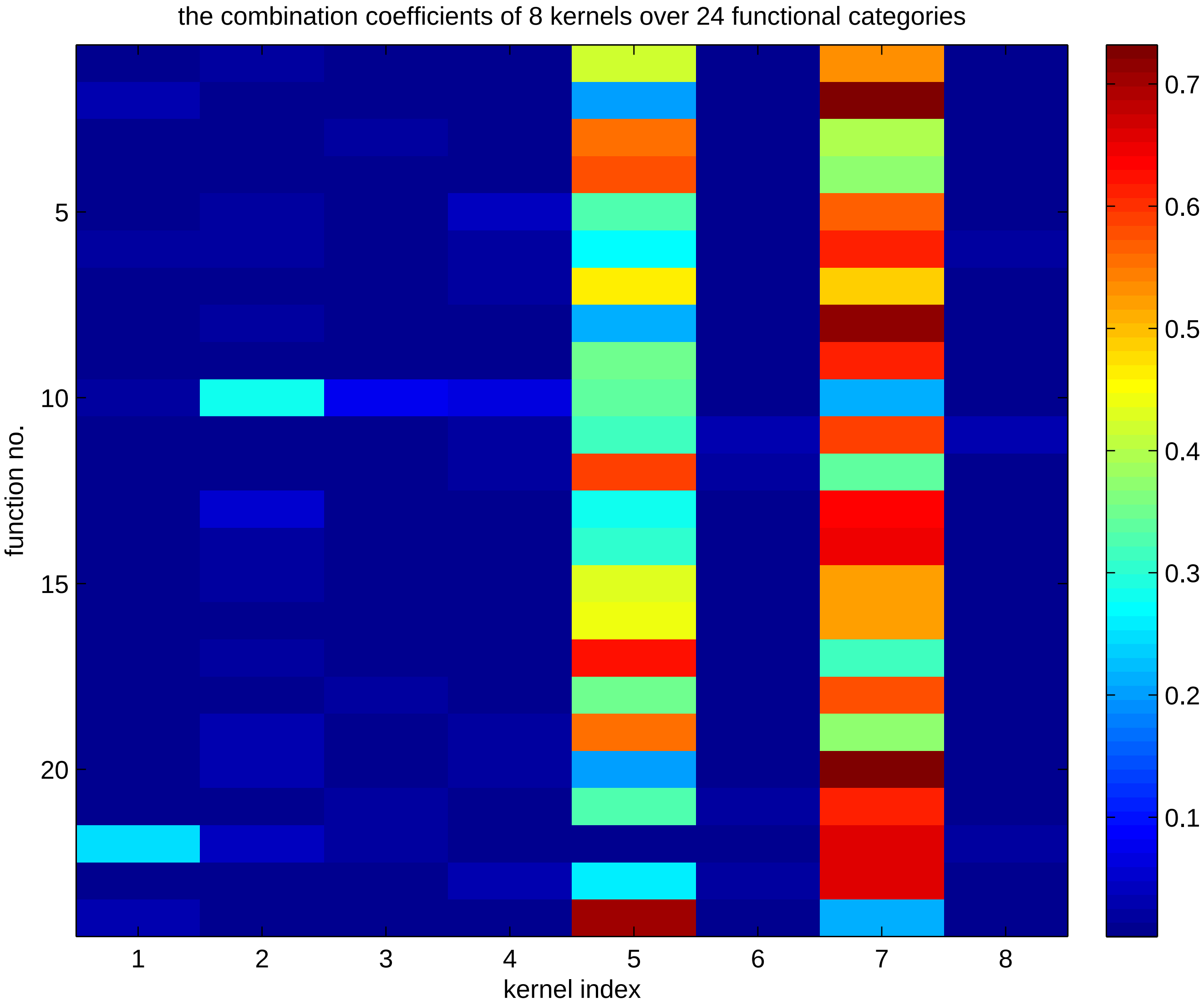


Figure 4.12: This figure shows the combination coefficients of different kernels. The odd columns correspond to the RBF kernels with $\sigma = 5$ and the even columns correspond to the RBF kernels with $\sigma = 15$. Column 1 and 2 correspond to the KS-test representation, column 3 and 4 correspond to the triple-valued RS-test representation, column 5 and 6 correspond to the average-log-probabilities representation, and column 7 and 8 correspond to the proportion-of-deviant-values representation.

4.4.3 Gene Function Prediction with Feature Selection

To identify informative features for each functional category, I used Backward Feature Selection (BFS) [27], Random Forest (RF) with 1000 trees [62], linear SVM with a fixed number of features, and Recursive Feature Elimination (RFE) [26] to perform feature selection. RFE is a recursive greedy feature selection method based on linear SVM. I didn't use Forward Feature Selection because SVM often failed to converge on this dataset with a small number of features. In BFS, the optimal number of features will be automatically determined, while the optimal number of features used in RF and RFE are chosen by CV. For a lot of functional categories on which SVM or Random Forest (RF) produces good results, the feature selection methods based on SVM or RF often output very good informative features. For a detailed case study, I used Functional Category 22, mitotic spindle organization and biogenesis in nucleus, as a running example.

In linear SVMs, I selected features by choosing the features with the largest magnitude of weights, since all the features take values on the same scale, -1, 0, or +1. For the linear SVMs with a fixed number of features, the total number of selected features was set to 100. For BFS, RBF kernel SVM was used as base classifier, in which the optimal value of σ found by CV was 5. Table 4.5 presents the mean ROC scores over 24 functional categories by different feature selection methods. We can see that the linear SVM with 100 features selected give the best mean ROC score. Table 4.6 presents the selected features by different methods for the spindle-related functional category in details. All the methods succeed in selecting spindle-related features as informative features. In specific, RFE selected 180 features, BFS selected 7 features, and RF selected 80 features. If I set the fixed number of features selected by linear SVM above 100, the performance doe not significantly improve, but if I set the number below 50, the performance dramatically degrades. When all the 198 features were used as input to train a linear SVM, the mean ROC score is 0.70, which is slightly better than what I got use only 100 selected features. Some other researchers might doubt the validity of using a linear SVM without feature normalization for feature selection on our triple-valued dataset, so I also normalized the dataset with each feature having mean 0 and standard deviation 1, and then I performed linear SVM with a fixed number of 100 features for feature selection. As is shown in Table 4.5, the overall performance is worse than that of the linear SVM without feature normalization. And on the spindle-related function, the features selected by the linear SVM with feature normalization are almost the same as those selected by the linear SVM without feature normalization, except that two more unbud cell spindle features are added and the two large-bud cell spindle features in Table 4.5 are removed. I performed Wilcoxon Signed-Ranks tests on the ROC score differences between pairwise feature selection methods, but I did not find significant difference between any pair of them.

Although feature selection methods can tell us which features are informative for each functional category, they did not improve the performance of the RBF kernel SVMs based on the triple-valued-RS-test feature representation.

4.5 Discussions

Because yeast essential genes cannot be deleted in biological knockout experiments for direct functional studies, my collaborators collected an array of yeast ts alleles, and measured the morphological profiles of these ts alleles using fluorescence markers at 8 key cell structures. It generates a dataset in which each feature of each ts allele has a distribution of values.

In this chapter, I proposed several novel feature representation methods to transform the morphological dataset into a mathematical feature matrix, which are based on the triple-valued RS test, the binary RS test, the KS test, the average log-probabilities of mutant feature values under the probability density function of wild-type feature values, and the proportion of deviant mutant feature values under the cumulative distribution of wild-type feature values. Both the probability density function and the cumulative

Feature Selection Methods	the Mean ROC Score	the ROC Score on the No. 22 Function	The number of features selected
RFE	0.65	0.83	180
linear SVM (100 features)	0.69	0.81	100
linear SVM (100 features, feature normalized)	0.67	0.83	100
BFS	0.68	0.82	7
RF	0.64	0.85	80

Table 4.5: This table gives the overall mean ROC scores over 24 functional categories and the ROC scores on the spindle-related functional category produced by different feature selection methods on the 26 degree morphology dataset using the triple-valued RS test. The linear SVM with 100 features selected works best, and the BFS method based on RBF kernels performs similarly to RFE, and RF performs worst overall.

Methods	Features Selected
linear SVMs	unbud spindle Orientation, unbud spindle Breadth, unbud spindle Fiber breadth, unbud spindle Equiv. oblate vol., midbud spindle Area, midbud spindle Perimeter, midbud spindle Length, midbud spindle Fiberlength, midbud spindle Shape factor, midbud spindle Equiv. oblate vol., largebud spindle area, largebud spindle Ell. form factor, and many other non-spindle features (100 features selected)
RFE	all the spindle-related features are included in the 180 selected features
BFS	unbud mito Fiberlength, unbud spindle Equivalent oblate volume, midbud spindle Perimeter, largebud nls Outer radius, largebud nls Equivalent radius, largebud nls Equivalent oblate volume, largebud pm Shape factor (7 features total)
RF	unbud spindle Perimeter, unbud spindle Orientation, unbud spindle Length, unbud spindle Breadth, unbud spindle Fiberlength, unbud spindle Shape factor, uunbud spindle Ell. form factor, unbud spindle Equiv. prolate vol., unbud spindle Equiv. oblate vol., midbud spindle Perimeter, midbud spindle Fiberlength, midbud spindle Fiber breadth, midbud spindle Shape factor, midbud spindle Ell. form factor, largebud spindle Area, largebud spindle Ell. form factor, largebud spindle Equiv. oblate vol., and other non-spindle features (80 features total)

Table 4.6: This table shows the detailed selected features by different feature selection methods for the spindle-related functional category in Table 4.5. In this table, ”midbud” denotes middle bud, ”mito” denotes mitochondria, ”nls” denotes nucleus, ”pm” denotes plasma membrane, and ”er” denotes Endoplasmic Reticulum.

distribution function of wild-type feature values were estimated using KDE. Experimental results show that the RBF kernel SVM based on the triple-valued RS test has the best predictive power, and the convex combination of RBF kernels based on the above different feature representations has comparable performance to the RBF kernel based on the best feature representation. The effectiveness of the triple-valued RS test demonstrates the importance of modeling the direction of mutant phenotype changes, that is, the reduced mutant feature values or elevated mutant feature values relative to the wild-type feature values.

For the learned kernel based on multiple kernel learning, the learned combination coefficients and the insignificant p-values for the ROC and ROC_{50} score differences between the triple-valued RS test and the average log-probabilities demonstrate the usefulness of the feature representations calculated through KDE.

Using feature selection methods based on BFS, linear SVMs, or RF, we can identify informative features for a lot of functional categories on which the RBF kernel SVMs have high predictive power. Although there is no statistically significant ROC/ROC_{50} score difference between pairs of used feature selection methods, on the 26 degree dataset with the triple-valued-RS-test representation, the mean ROC and ROC_{50} scores produced by the RBF kernel SVMs are, respectively, 0.74 and 0.37, in contrast, the mean ROC and ROC_{50} scores produced by linear SVMs are, respectively, 0.70 and 0.28. The p-values for the ROC and ROC_{50} score differences between the pair using Wilcoxon Matched-Pairs Signed-Ranks Tests are, respectively, $8.6e-3$ and $9.3e-3$, which clearly shows the advantage of using non-linear kernels. It proves that the dependencies between the features of ts mutants are highly non-linear. Therefore, it's difficult for linear feature selection methods and linear predictive models to produce highly accurate and informative features for each functional category. In the future, I will build more advanced feature selection methods to handle this problem.

The methods developed in this chapter make it possible to further study the relations between pairwise ts alleles and to study the novel functions of essential genes by further analyzing the highly-ranked false positives predicted by our prediction methods.

Chapter 5

Conclusions

5.1 Discussions

To understand gene regulation and gene function at a system level, I proposed machine learning methods to analyze protein sequence data, miRNA and mRNA expression data, proteomic data, and yeast cell morphology data in this thesis.

In details, I proposed learned random-walk kernels and learned empirical-map kernels based on protein sequence profile kernels to classify protein sequences into different remote homology categories. The overall mean ROC_{50} scores obtained by my approaches are above 0.90, moreover, my proposed approaches even work very well on the difficult protein families for all the other methods, which means that my approaches almost solved this challenging problem on the SCOP benchmark dataset. Furthermore, the learned random-walk kernels can be used to discover biologically meaningful sequence motifs that are responsible for determining protein sequences' remote homology membership.

In collaboration with biologists, I developed probabilistic graphical models to predict miRNAs' targets and to infer the possible mechanisms of miRNA regulation. The approach developed in this thesis is the first computational one that incorporates miRNA expression data and proteomic data in multiple tissues to perform high-throughput miRNA target predictions. Moreover, by combining miRNA expression profile data and mRNA expression profile data, our model can be used to infer the regulatory mechanism of miRNA regulation: translational repression which does not affect mRNA expression level, or mRNA degradation which significantly reduces mRNA expression level.

To study the functions of yeast essential genes that cannot be directly deleted in biological knock-out experiments, my collaborators created the first whole-genome ts alleles for about 50% of all the yeast essential genes, and profiled these ts mutants using high-content screening, which produces a complex dataset with each feature of each ts mutant having a distribution of values of varying size. I proposed several novel feature representation schemes based on the RS test, the KS test, and KDE, which enables kernel SVMs and feature selection methods to be applied on this dataset to perform functional analysis. The overall mean ROC score over 24 functional categories obtained by RBF kernel SVMs based on the best feature representation, the triple-valued RS test, is 0.74. Using the best feature representation, there are 18 out of 24 functional categories with ROC score above 0.65, and there are 12 out of 24 functional categories with ROC score above 0.75, which demonstrates the reasonably good predictive power of cell morphology data for gene functions. And my prediction methods enable biologists to pick and study novel functions of yeast essential genes by analyzing the highly-ranked false positives produced by the prediction methods.

5.2 Future Research

For protein sequence analysis, I will apply my methods to predict protein folds solely based on protein sequence data, which is a harder problem than protein remote homology identification. I also plan to extend the learned random-walk kernels to a semi-supervised setting, where I will use additional network information to leverage the unlabeled data to learn a better kernel instead of just focusing on labeled data. On the machine learning side, I will try to prove the generalization bound of the learned kernels in semi-supervised learning settings.

For miRNA regulation prediction, I plan to apply my approach to other organisms to further study the regulation mechanisms of miRNAs, and I plan to develop fast inference methods based on varational inference [34] to scale the approach to large datasets. Moreover, I will try to use the motif discovery method developed in this thesis to identify possible sequence patterns in the 3' UTRs that distinguish the mRNAs with different miRNA regulation mechanisms.

For gene function prediction based on the yeast morphology data, I mentioned that kernel SVMs perform much better than linear SVMs, which shows that the dependencies between morphological features are very non-linear. Therefore, instead of using template-like kernel methods, I will develop advanced methods that can capture higher-order statistics hidden in the morphology data based on the triple-valued RS test, and I plan to embed the data using the non-linear embedding methods developed in [53] and [54] to visualize the relationships between pairwise ts mutants explicitly, which will facilitate biologists to pick the ts mutants for further study. The current feature representations are based on comparing mutant feature value distribution to wild-type feature value distribution, which ignores the dependencies between features when deriving feature representation. In the future, I will build generative models that can directly generate all the feature values of ts mutants with dependencies considered, and I plan to use discriminative label information to help learn a hybrid model. Infinite Mixture Models based on Dirichlet Process [21] are possible good candidates. In the future, I will explore this line of research.

Bibliography

[1] A. Alibes, P. Yankilevich, A. Canada, and R. Diaz-Uriarte. IDconverter and IDClight: conversion and annotation of gene and protein IDs. *BMC Bioinformatics*, 8:9, 2007.

[2] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schffer, Ro A. Schffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25:3389–3402, 1997.

[3] V. Ambros. The functions of animal microRNAs. *Nature*, 431(7006):350–5, 2004.

[4] E. D. Andersen and A. D. Andersen. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. *High Performance Optimization*, 91(3):197–232, 2000.

[5] T. Babak, W. Zhang, Q. Morris, B. J. Blencowe, and T. R. Hughes. Probing microRNAs with microarrays: tissue specificity and functional inference. *Rna*, 10(11):1813–9, 2004.

[6] 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.

[7] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure. Hidden markov models of biological primary sequence information. *PNAS*, 91(3):1059–1063, 1994.

[8] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–97, 2004.

[9] Asa Ben-hur and Douglas Brutlag. Remote homology detection: a motif based approach. *ISMB*, 19:26–33, 2003.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, 2007.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[12] O. Chapelle, J. Weston, and B. Scholkopf. Cluster kernels for semi-supervised learning., 2003.

[13] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans. Group-theoretic algorithms for matrix multiplication, 2005.

[14] Nello Cristianini, John Shawe-Taylor, Andre Elissee, and Jaz Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT Press, 2001.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[16] R. J. Dohmen, P. Wu, and A. Varshavsky. Heat-inducible degron: a method for constructing temperature-sensitive mutants. *Science*, 263:1273–1276, 1994.

[17] A. J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks. MicroRNA targets in Drosophila. *Genome Biol*, 5(1):R1, 2003.

[18] S. Mnaimneh et al. Exploration of essential gene functions via titratable promoter alleles. *Cell*, 118:31–44, 2004.

[19] Ana Eulalio, Eric Huntzinger, and Elisa Izaurralde. Getting to the root of miRNA-mediated gene silencing. *Cell*, 132(1):9, 2008.

[20] K. K. Farh, A. Grimson, C. Jan, B. P. Lewis, W. K. Johnston, L. P. Lim, C. B. Burge, and D. P. Bartel. The widespread impact of mammalian microRNAs on mRNA repression and evolution. *Science*, 310(5755):1817–21, 2005.

[21] Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1, 1973.

[22] Andrew Gelman. *Bayesian data analysis*. Chapman and Hall, London, 1st edition, 1995.

[23] Guri Giaever, Angela M. Chu, Li Ni, Carla Connelly, Linda Riles, Steeve Veronneau, Sally Dow, Ankuta Lucau-Danila, Keith Anderson, Bruno Andre, Adam P. Arkin, Anna Astromoff, Mohamed El Bakkoury, Rhonda Bangham, Rocio Benito, Sophie Brachat, Stefano Campanaro, Matt Curtiss, Karen Davis, Adam Deutschbauer, Karl-Dieter Entian, Patrick Flaherty, Francoise Foury, David J. Garfinkel, Mark Gerstein, Deanna Gotte, Ulrich Guldener, Johannes H. Hegemann, Svenja Hempel, Zelek Herman, Daniel F. Jaramillo, Diane E. Kelly, Steven L. Kelly, Peter Kotter, Darlene LaBonte, David C. Lamb, Ning Lan, Hong Liang, Hong Liao, Lucy Liu, Chuanyun Luo, Marc Lussier, Rong Mao, Patrice Menard, Siew L. Ooi, Jose L. Revuelta, Christopher J. Roberts, Matthias Rose, Petra Ross-Macdonald, Bart Scherens, Greg Schimmack, Brenda Shafer, Daniel D. Shoemaker, Sharon Sookhai-Mahadeo, Reginald K. Storms, Jeffrey N. Strathern, Giorgio Valle, Marleen Voet, Guido Volckaert, Ching-yun Wang, Teresa R. Ward, Julie Wilhelmy, Elizabeth A. Winzeler, Yonghong Yang, Grace Yen, Elaine Youngman, Kexin Yu, Howard Bussey,

Jef D. Boeke, Michael Snyder, Peter Philippsen, Ronald W. Davis, and Mark Johnston. Functional profiling of the Saccharomyces cerevisiae genome. *Nature*, 418:387–391, 2002.

[24] Jean Dickinson Gibbons. *Nonparametric Statistical Inference.* Marcel Dekker Inc., 1985.

[25] Huili Guo, Nicholas T. Ingolia, Jonathan S. Weissman, and David P. Bartel. Mammalian microRNAs predominantly act to decrease target mRNA levels. *Nature*, 466:835–840, 2010.

[26] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using Support Vector Machines. *Machine Learning*, 46:389–422, 2002.

[27] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning.* Springer, 2003.

[28] James J. Higgins. *Introduction To Modern Nonparametric Statistics.* Duxbury Press, 2003.

[29] J. C. Huang, Q. D. Morris, and B. J. Frey. Bayesian inference of microRNA targets from sequence and expression data. *J Comput Biol*, 14(5):550–63, 2007.

[30] Jim C. Huang, Tomas Babak, Timothy W. Corson, Gordon Chua, Sofia Khan, Brenda L. Gallie, Timothy R. Hughes, Benjamin J. Blencowe, Brendan J. Frey, and Quaid D. Morris. Using expression profiling data to identify human microRNA targets. *Nature Methods*, 4:1045–1049, 2007.

[31] Jim C. Huang, Quaid D. Morris, and Brendan J. Frey. Detecting microRNA targets by linking sequence, microRNA and gene expression data. *In Proceedings of RECOMB*, 3909:114–129, 2006.

[32] Tommi Jaakkola, Mark Diekhans, and David Haussler. A discriminative framework for detecting remote protein homologies, 2000.

[33] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91:401–407, 1996.

[34] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[35] M. Kiriakidou, P. T. Nelson, A. Kouranov, P. Fitziev, C. Bouyioukos, Z. Mourelatos, and A. Hatzigeorgiou. A combined computational-experimental approach predicts human microRNA targets. *Genes Dev*, 18(10):1165–78, 2004.

[36] T. Kislinger, B. Cox, A. Kannan, C. Chung, P. Hu, A. Ignatchenko, M. S. Scott, A. O. Gramolini, Q. Morris, M. T. Hallett, J. Rossant, T. R. Hughes, B. Frey, and A. Emili. Global survey of organ and organelle protein expression in mouse: combined proteomic and transcriptomic profiling. *Cell*, 125(1):173–86, 2006.

[37] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *In Proceedings of the ICML*, pages 315–322, 2002.

[38] A. Krek, D. Grun, M. N. Poy, R. Wolf, L. Rosenberg, E. J. Epstein, P. MacMenamin, I. da Piedade, K. C. Gunsalus, M. Stoffel, and N. Rajewsky. Combinatorial microRNA target predictions. *Nat Genet*, 37(5):495–500, 2005.

[39] Anders Krogh, Michael Brown, I. Saira Mian, Kimmen Sjolander, and David Haussler. Hidden markov models in computational biology: Applications to protein modeling, 1994.

[40] Rui Kuang, Eugene Ie, Ke Wang, Kai Wang, Mahira Siddiqi, Yoav Freund, and Christina Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Journal of Bioinformatics and Computational Biology*, 3:152–160, 2005.

[41] Gert Lanckriet, Nello Cristianini, Peter Bartlett, and Laurent El Ghaoui. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[42] C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. In S. Becker, S. Thrun, and K. Obermayer, editors, *NIPS 2002*, volume 15, pages 1441 – 1448, Cambridge, MA, USA, 2003. MIT Press.

[43] B. P. Lewis, C. B. Burge, and D. P. Bartel. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, 120(1):15–20, 2005.

[44] B. P. Lewis, I. H. Shih, M. W. Jones-Rhoades, D. P. Bartel, and C. B. Burge. Prediction of mammalian microRNA targets. *Cell*, 115(7):787–98, 2003.

[45] Jingjing Li, Renqiang Min, Anthony J. Bonner, and Zhaolei Zhang. A probabilistic framework to improve microRNA target prediction by incorporating proteomics data. *J Bioinform Comput Biol*, 7(6):955–72, 2009.

[46] Li Liao and William Stafford Noble. Combining pairwise sequence similarity and Support Vector Machines for remote protein homology detection. In *Journal of Computational Biology*, pages 225–232, 2002.

[47] L. P. Lim, N. C. Lau, P. Garrett-Engele, A. Grimson, J. M. Schelter, J. Castle, D. P. Bartel, P. S. Linsley, and J. M. Johnson. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. *Nature*, 433(7027):769–73, 2005.

[48] A. C. McHardy, A. Puhler, J. Kalinowski, and F. Meyer. Comparing expression level-dependent features in codon usage with protein abundance: an analysis of 'predictive proteomics'. *Proteomics*, 4(1):46–58, 2004.

[49] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[50] Renqiang Min, Anthony Bonner, and Zhaolei Zhang. Modifying kernels using label information improves SVM classification performance. In *ICMLA '07: Proceedings of the Sixth International Conference on Machine Learning and Applications*, pages 13–18, Washington, DC, USA, 2007. IEEE Computer Society.

[51] Renqiang Min, Anthony J. Bonner, Jingjing Li, and Zhaolei Zhang. Learned random-walk kernels and empirical-map kernels for protein sequence classification. *J Comput Biol*, 16(3):457–74, 2009.

[52] Renqiang Min, Rui Kuang, Anthony J. Bonner, and Zhaolei Zhang. Learning random-walk kernels for protein remote homology identification and motif discovery. In *SIAM International Conference on Data Mining*, pages 133–144, 2009.

[53] Renqiang Min, David A. Stanley, Zineng Yuan, Anthony J. Bonner, and Zhaolei Zhang. A deep non-linear feature mapping for large-margin kNN classification. In *IEEE International Conference on Data Mining*, pages 357–366. IEEE Computer Society, 2009.

[54] Renqiang Min, Laures van der Maaten, Zineng Yuan, Anthony J. Bonner, and Zhaolei Zhang. Deep supervised t-distributed embedding. In *International Confernece on Machine Learning*, 2010.

[55] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

[56] John C. Platt. Fast training of Support Vector Machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, pages 185–208, Cambridge, MA, USA, 1999. MIT Press.

[57] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[58] 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.

[59] P. Sethupathy, B. Corda, and A. G. Hatzigeorgiou. TarBase: A comprehensive database of experimentally supported animal microRNA targets. *Rna*, 12(2):192–7, 2006.

[60] P. Sood, A. Krek, M. Zavolan, G. Macino, and N. Rajewsky. Cell-type-specific signatures of microRNAs on target mRNA expression. *Proc Natl Acad Sci U S A*, 103(8):2746–51, 2006.

[61] D. Spiegelhalter, A. Tomas, N. Best, W. Gilks, and D. Lunn. BUGS:Bayesian inference using Gibbs sampling. *MRC Biostatistics Unit, Cambridge, England. www.mrc-bsu.cam.ac.uk/bugs/*, 2003.

[62] Leo Breiman Statistics and Leo Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[63] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, 1999.

[64] A. I. Su, T. Wiltshire, S. Batalov, H. Lapp, K. A. Ching, D. Block, J. Zhang, R. Soden, M. Hayakawa, G. Kreiman, M. P. Cooke, J. R. Walker, and J. B. Hogenesch.

A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc Natl Acad Sci U S A*, 101(16):6062–7, 2004.

[65] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, pages 945–952. MIT Press, 2002.

[66] Z. Tian, A. Greene, J. Pietrusz, I. Matus, and M. Liang. MicroRNA-target pairs in the rat kidney identified by microRNA microarray, proteomic, and bioinformatic analysis. *Genome Research*, 18(3):404–411, March 2008.

[67] Amy H. Tong, Guillaume Lesage, Gary D. Bader, Huiming Ding, Hong Xu, Xiaofeng Xin, James Young, Gabriel F. Berriz, Renee L. Brost, Michael Chang, YiQun Chen, Xin Cheng, Gordon Chua, Helena Friesen, Debra S. Goldberg, Jennifer Haynes, Christine Humphries, Grace He, Shamiza Hussein, Lizhu Ke, Nevan Krogan, Zhijian Li, Joshua N. Levinson, Hong Lu, Patrice Menard, Christella Munyana, Ainslie B. Parsons, Owen Ryan, Raffi Tonikian, Tania Roberts, Anne-Marie Sdicu, Jesse Shapiro, Bilal Sheikh, Bernhard Suter, Sharyl L. Wong, Lan V. Zhang, Hongwei Zhu, Christopher G. Burd, Sean Munro, Chris Sander, Jasper Rine, Jack Greenblatt, Matthias Peter, Anthony Bretscher, Graham Bell, Frederick P. Roth, Grant W. Brown, Brenda Andrews, Howard Bussey, and Charles Boone. Global mapping of the yeast genetic interaction network. *Science*, 303:808–813, 2004.

[68] J. Tsang, J. Zhu, and A. van Oudenaarden. MicroRNA-mediated feedback and feedforward loops are recurrent network motifs in mammals. *Mol Cell*, 26(5):753–67, 2007.

[69] Koji Tsuda, Hyunjung Shin, and Bernhard Scholkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21(2):59–65, 2005.

[70] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[71] M. Wakiyama, K. Takimoto, O. Ohara, and S. Yokoyama. Let-7 microRNA-mediated mRNA deadenylation and translational repression in a mammalian cell-free system. *Genes Dev*, 21(15):1857–62, 2007.

[72] David Warton. Many zeros does not mean zero inflation: comparing the goodness-of-fit of parametric models to multivariate abundance data. *Environmetrics*, 16:275–289, 2005.

[73] Larry Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer, May 2007.

[74] Jason Weston, Christina Leslie, Eugene Ie, Dengyong Zhou, Andre Elisseeff, and William Stafford Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

[75] X. Xu. Same computational analysis, different miRNA target predictions. *Nat Methods*, 4(3):191; author reply 191, 2007.

[76] Wen Zhang, Quaid Morris, Richard Chang, Ofer Shai, Malina Bakowski, Nicholas Mitsakakis, Naveed Mohammad, Mark Robinson, Ralph Zirngibl, Eszter Somogyi, Nancy Laurin, Eftekhar Eftekharpour, Eric Sat, Jorg Grigull, Qun Pan, Wen T. Peng, Nevan Krogan, Jack Greenblatt, Michael Fehlings, Derek van der Kooy, Jane Aubin, Benoit Bruneau, Janet Rossant, Benjamin Blencowe, Brendan Frey, and Timothy Hughes. The functional landscape of mouse gene expression. *Journal of Biology*, 3:21+, 2004.

[77] Xiaojin Zhu, Jaz Kandola, Zoubin Ghahramani, and John Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In Lawrence K. Saul,

Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1641–1648. MIT Press, Cambridge, MA, 2005.