Comparison and Evaluation of Statistical-Learning Methods for Gene Function Prediction in Arabidopsis thaliana

by

Hui Lan

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

Abstract

Comparison and Evaluation of Statistical-Learning Methods for Gene Function Prediction in Arabidopsis thaliana

Hui Lan

Master of Science

Graduate Department of Computer Science

University of Toronto

2005

Approximately 30,000 genes have been discovered by genome sequencing in *Arabidopsis thaliana* completed in 2000. However, about half of these genes have not been assigned any function yet. The goal of this study is to identify unknown genes that are potentially involved in plant responses to stresses. We evaluated and compared five basic statistical learning methods for gene function prediction on a genome-wide scale using gene expression data. None of these methods was uniformly better than the others. In addition, we investigated combining these methods for prediction. The combined method achieved better classification performance than the basic methods for the top "response to stress" function. With precision above 50%, we identified a considerable number of unknown genes that are potentially stress-associated, which are currently being validated by biologists.

Acknowledgements

I gratefully acknowledge the contribution of the following people to the completion of this thesis. Professor Anthony Bonner, my supervisor, worked closely with me in this study. His expertise has greatly influenced my thinking on bioinformatics. Professor Nicholas Provart collaborated with us by providing data, discussing with us, and allowing me to use the supercomputer in his lab. Kiana Toufighi and Rachel Carson helped me with problems on biology. Xiaodan Zhu and Miles Trochesset discussed statistical learning with me. Eric Hsu proofread the draft of this thesis. Finally, I would like to thank my parents for instilling in me a love of knowledge.

Contents

Chapter 1

Introduction

Assigning functions to genes with unknown function, identified by genome sequencing and other methods, is the goal of functional genomics. Many approaches have been proposed for large-scale prediction of gene function [2, 5, 9, 11, 26, 31]. These approaches are mostly based on physical association, genetic interaction, sequence relationships and patterns of gene expression.

The research in this thesis focuses on gene function prediction in *Arabidopsis thaliana* using statistical learning algorithms based on gene expression data. *A. thaliana* is a small flowering plant that is widely used by biologists who study cellular and molecular functions of flowering plants [22]. It is an ideal model organism for study since it has a relatively small genome (125Mb) with a set of representative genes for controlling developmental processes, responses to environmental changes and disease resistance. Also, its small size, short life cycle and prodigious seed production make it easy to cultivate in the laboratory. Figure 1.1 shows a wild-type *A. thaliana*. Gene expression is the process by which the information coded in a gene is transcribed into mRNA, which then is translated into protein, the active manifestations of the genetic information. Biologists use microarrays to measure gene expression levels of tens of thousands of genes simultaneously. The gene expression data reflect the gene activity levels during mRNA transcription: High

Figure 1.1: A picture of *A. thaliana* (`http://www.mpimp-golm.mpg.de/arabidopsis/thaliana-e.html`)

gene expression level indicates high activity of the particular gene and vice versa.

With the *A. thaliana* genome completely sequenced [36], gene functional annotation for all the genes in the genome remains a key challenge for biologists. Currently, approximately 50% of the 28,000 genes have not been assigned any function. Predicting gene functions based on gene expression data is an attractive strategy since many pathways display coordinated transcriptional regulation [5, 12]. This research is a successor to the finished genome sequencing of *A. thaliana* and is made possible by the gene expression data from the Department of Botany at University of Toronto [34] and from the AtGenExpression Consortium, archived at NASCArrays (`http://arabidopsis.info/`), as well as GO functional annotations from TAIR (`http://www.arabidopsis.org/`).

1.1 Gene Function Prediction Using Unsupervised and Supervised Learning Algorithms

Previously, most microarray classification work was on cancer classification [1, 6, 8, 14, 18, 20, 25, 28, 37]. The gene expression data of normal tissues and tumor tissues are collected and a classifier is learned using these data. When the testing tissue appears, the classifier determines if the tissue is normal or cancerous based on its gene expression.

In recent years, researchers have conducted experiments for predicting gene function in various microorganisms using statistical learning algorithms based on microarray data. These algorithms can be divided into two categories: unsupervised and supervised [10].

Unsupervised clustering methods group genes that have similar gene expressions. The most widely used similarity metric is the Pearson correlation coefficient, whose value ranges from -1 to +1. The genes in the same group are assumed to have similar functions. Based on this assumption, the functions of unknown genes can be inferred from known genes in the same group. Many studies have taken place in this way to identify functions of unknown genes [5, 7, 12, 32, 33, 35, 38].

However, unsupervised clustering methods cannot take advantage of the gene function information in the process of learning. A better approach of identifying gene functions of unknown genes based on gene expression data is supervised learning. A few attempts have been made. Hvidsten et al. modeled the relationships between gene expression of serum response in serum-starved human fibroblasts as a function of time and the involvement of a gene in a given biological process using Rough Set methods [13, 17]. The resulting model was used to predict the biological process roles of unknown genes. Midelfart et al. extended the above work by developing a method to learn an ontology by which biological processes are organized [23]. For the yeast *Saccharomyces cerevisiae*, Support Vector Machines (SVMs) have been extensively applied to functional classification and/or function prediction [2, 16, 19, 24, 27]. In addition, other supervised learning methods such as Multilayer Perceptrons [21] and Logistic Regression [35] have been investigated for similar purpose. Most recently, researchers have investigated gene function prediction in more complex organisms such as mice [39]. The results show that gene expression for mammals can also be used to predict gene function.

The above studies show gene expression data can be used to predict gene function in microorganisms such as *S. cerevisiae* and mammals such as mice using unsupervised and supervised learning methods. Nevertheless, it still remains unknown whether this is true in plants. This study addresses this question.

1.2 Research Goals

Since many genes in *A. thaliana* still have no known function, the research goal is to provide hypotheses of stress gene functions for these unknown genes for biologists to test. In the context of plants, a stress (biotic or abiotic) causes a decrease in plant growth or yield. Our biological collaborators are particularly interested in genes that are potentially involved in response to stresses, such as drought, cold, salinity, etc. Gene ontology (GO) is proposed by Gene Ontology Consortium [3] and it provides a dynamic controlled vocabulary for gene functions. It has three broad categories: molecular function, biological process and cellular component. For convenience, we describe gene functions in terms of Gene Ontology Biological Processes (GOBPs). For any particular GOBP, we call the genes that belong to it positives and the others that do not belong to it negatives. For economic reasons, we want a low false-positive rate (e.g., 50%) for the predictions: The majority of the predicted positives should be true positives. On the other hand, a high false-negative rate is acceptable since the cost of false negatives is much smaller than the cost of false positives.

1.3 Unbalanced Data with Few Positives but Many Negatives

Unbalanced data constitute the main difficulty in this study. Each of the stress GOBPs of interest typically contains lots of negatives and few positives. In fact, more than 92% of the training data are composed of negatives for the stress GOBPs we have learned. The unbalanced data made accurate predictions using most statistical learning algorithms difficult.

1.4 Using Supervised Learning Methods to Learn Gene Function

Supervised learning methods can take advantage of functional annotations of genes and perform feature selection. Since genes can either be positives (belonging to a GOBP) or negatives (not belonging to a GOBP), the gene function prediction problem can be simplified to a binary classification problem by treating each gene function independently. For each gene function, we learn a classifier $\hat{f}(\mathbf{x})$ using the training data. Cross-validation is then used to assess the classifier as well as to measure the prediction precision. The unknown genes in the prediction data are then classified as either positives or negatives by the resulting classifier. Unknown genes with discriminant values greater than a certain threshold are deemed positives and vice versa. However, a gene can belong to multiple GOBPs; reducing a multi-class classification problem to a binary classification problem cannot take advantage of the correlation and structure information among gene functions.

In this study, we used GOBP as the definition of gene function because we believe that biological processes are well correlated with gene expression. We use GOBP to refer to gene function in general and use a GOBP term to refer to a specific function. For instance, GO:0009409 represents the “response to cold” function. GOBPs are organized in a Directed Acyclic Graph (DAG) as we will describe in Chapter 2. In this graph, parent GOBP terms are subdivided into increasingly specific child GOBP terms. However, this graph differs from the hierarchy in that a child GOBP term may have multiple parents.

The significance of this study is threefold. First, to our knowledge, this study for the first time investigated gene function prediction in *A. thaliana* based solely on gene expressions using supervised learning methods. Second, a thorough evaluation and comparison for several well-known supervised statistical learning methods [10], Logistic Regression (LR), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes (NB) and K-Nearest Neighbors (KNN), was made; in addition, a combination of them using averaging and stacking was investigated. Third, this work enabled biologists to carry out directed biological experiments for determining gene functions and thus will contribute to the accomplishment of the goal to annotate the function of each gene in the whole *A. thaliana* genome by 2010.

1.5 Outline

The organization of this thesis is as follows:

- Chapter 2 gives an overview of the microarray data for *A. thaliana*, gene ontology and gene annotations. In addition, statistics about the microarray data and the current knowledge of gene functions for *A. thaliana* will be presented.

- In Chapter 3, various supervised learning methods (LR, LDA, QDA, NB, KNN) will be discussed. It follows the approaches of evaluation and comparison of these methods: Cross-validation, randomizations, and permutations are combined to picture the performance of individual methods. Finally, we discuss how to make predictions and how to measure the prediction precision.

- The evaluation and comparison of the above supervised learning methods as well as predictions are given in Chapter 4.

- In Chapter 5, we describe methods to combine the five learning methods discussed in Chapter 3 and show the evaluation results as well as prediction results.
- We conclude and discuss in Chapter 6.

Chapter 2

Gene Expression Data, Gene Ontology and Gene Annotations

As mentioned in the previous section, the goal in this study is to provide hypotheses on the stress functions for unknown genes in *A. thaliana*. To achieve this goal, we need to have access to two pieces of information: (1) features: the gene expression levels, and (2) labels: 0-1 arrays representing membership and nonmembership of each gene for a specific gene function. Features are from microarray data and labels from gene annotations. Let **P** be an $H \times K$ real-valued gene expression matrix, with H unknown genes and K experimental conditions. Let **T** be an $I \times K$ real-valued gene expression matrix, each row representing a known gene and each column representing an experimental condition. Let **M** be an $I \times J$ 0-1 membership matrix, each row representing a known gene and each column a GOBP. $m_{ij} = 1$ if and only if the ith gene is involved in GOBP$_j$; otherwise $m_{ij} = 0$. Thus, for the particular GOBP$_j$, we have a column vector $GOBP_j = (m_{1j}, m_{2j},\ldots, m_{Ij})^T$, where I equals the number of known genes. **P** formed the prediction data (with no label for any GOBP); **T** and **M** formed the training data. Figure 2.4 shows the number of positives of the individual GOBPs in the training data.

The prediction data:

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1K} \\ p_{21} & p_{22} & \cdots & p_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ p_{H1} & p_{H2} & \cdots & p_{HK} \end{pmatrix} \tag{2.1}$$

The training data:

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1K} \\ t_{21} & t_{22} & \cdots & t_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I1} & t_{I2} & \cdots & t_{IK} \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1J} \\ m_{21} & m_{22} & \cdots & m_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ m_{I1} & m_{I2} & \cdots & m_{IJ} \end{pmatrix} \tag{2.2}$$

2.1 Gene Expression Data for *A. thaliana*

In this section, we briefly describe the gene expression data used in this study. An oligonucleotide microarray uses the sequence resources of a particular organism to answer the question of what genes are expressed in specific tissues at particular developmental stages or under various conditions [29, 30]. It is used to record the expression levels, i.e., transcriptional profiles, of tens of thousands of genes in *A. thaliana* simultaneously on a genome-wide scale in a single assay.

2.1.1 Raw Gene Expression Data

In this study, we used two microarray datasets for *A. thaliana*: one from the Department of Botany at University of Toronto and the other from the AtGenExpression Consortium. The dataset from the Department of Botany at University of Toronto has 54 features, such as plant physiology, plant microbe, environmental stress, and biotechnology. The dataset from the AtGenExpression Consortium has 255 features, including various stresses (e.g., osmotic stress, heat stress, cold stress, salt stress, drought stress, UV-B stress, wounding stress, water deprivation stress and oxidative stress). These two datasets were combined into one. The resulting dataset contains the expression levels for 22,746 genes under 309 different conditions. Thus, the whole gene expression matrix

$$\mathbf{E} = \begin{pmatrix} \mathbf{P} \\ \mathbf{T} \end{pmatrix}$$

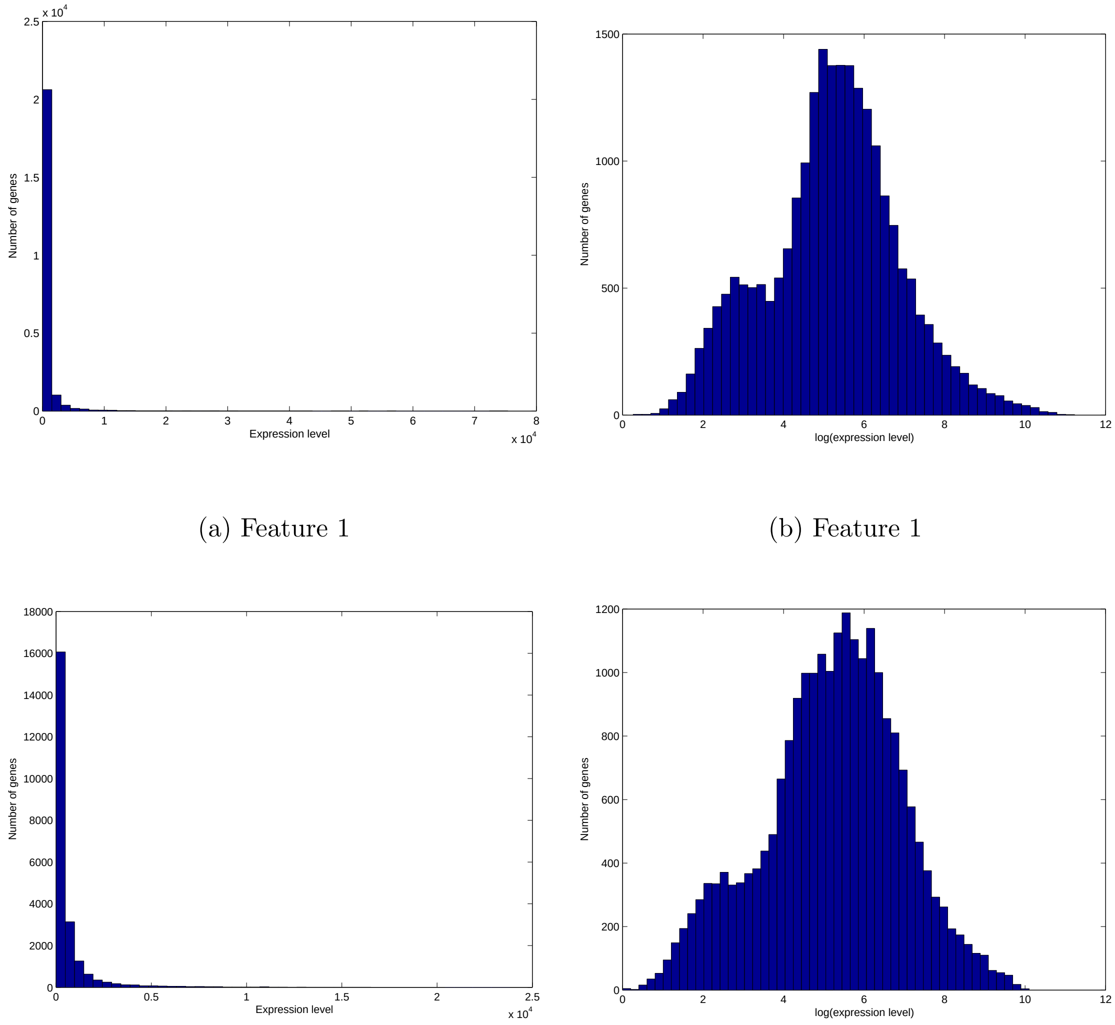
has 22,746 rows and 309 columns.

2.1.2 Data Preprocessing

The microarray data from the Department of Botany contain *detection calls*: P, M and A. P = Present, M = Marginal, and A = Absent. The detection call determines whether a transcript is reliably detected (present), partially detected (marginal), or not detected (absent). Following is an example for the gene AT3G24440 under three selected conditions:

AT3G24440	243.10 P	120.90 A	109.40 M

We simply removed these detection calls (P, A, and M) in this study. In addition, as a common practice, gene expression levels were logarithmized. The log-transformed gene expression data have approximately normal distributions while the raw data have approximately exponential distributions, as shown in Figure 2.1. Figure 2.2 shows the quantile test for normal distributions for two randomly selected features. A quantile-quantile (q-q) plot is used to determine whether two samples come from a population with a common distribution (normal distribution). The quantiles for one of the data samples were replaced with the quantiles of a normal distribution. There is a reference line (dashdot) with slope 1 in the plot. If two samples come from a population with a same distribution, the points should lie approximately on the reference line.



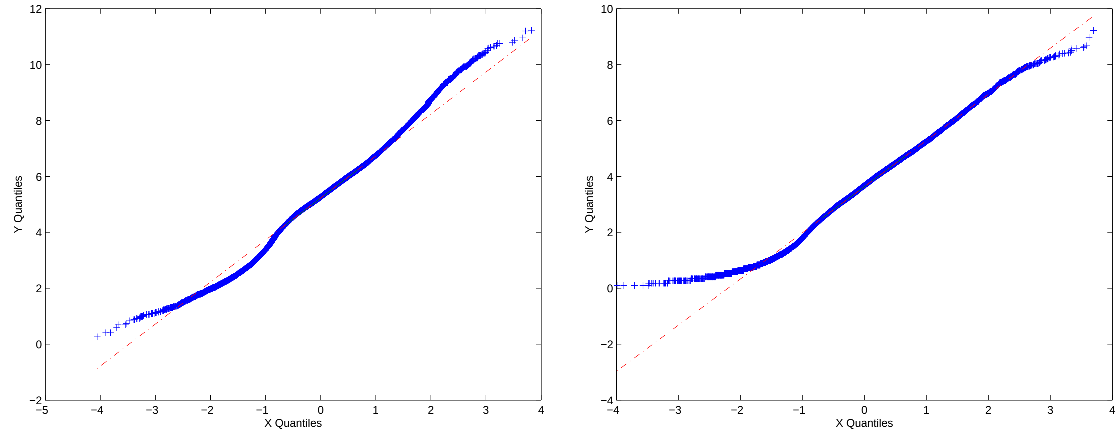
(a) Feature 1

(b) Feature 1

(c) Feature 110

(d) Feature 110

Figure 2.1: Histograms of raw gene expression levels (left panel) and log-transformed gene expression levels (right panel) for the two randomly selected features



(a) Feature 1

(b) Feature 110

Figure 2.2: The q-q plots for testing normality of the two randomly selected features

2.2 Gene Ontology and Gene Annotations in *A. thaliana*

Since *A. thaliana* was completely sequenced, biologists have attempted to provide functional annotations to these newly discovered genes [4]. Assigning functions to these genes is a key step towards understanding the genome of this species.

2.2.1 Gene Ontology

Gene ontology (GO) provides a dynamic controlled vocabulary for describing the role of all genes in all organisms [3]. GO terms are organized in a DAG to reflect the hierarchical relationships between them, as shown in Figure 2.3. A GO term can have several parents and children. GO terms fall into three broad categories: molecular function, biological process and cellular component, each of which groups several thousands of GO terms. This structure allows us to describe a gene's role at different levels of granularity based on the amount of information that is known for the gene. Lower levels in the hierarchy correspond to more specific functions and vice versa. The genes whose detailed functional information is known are assigned to low levels in the hierarchy, whereas the genes whose functional information is limited are assigned to high levels in the hierarchy.

In this study, we focused on the stress GOBPs (see Figure 2.3), which are a small part in the gene ontology. Specifically, we were concerned with the gene functions below and including the GO term *GO:0006950[response to stress]* in the hierarchy, under which there are 18 children such as *GO:0009409[response to cold]*, *GO:0009408[response to heat]*, and *GO:0009414[response to water deprivation]*. We have more detailed functional information on the genes annotated as *GO:0009409[response to cold]* than on those annotated as *GO:0006950[response to stress]*. The genes in *GO:0009409[response to cold]* can be propagated up to *GO:0006950[response to stress]*. On the other hand, we can move genes downward in the hierarchy when more knowledge of these genes is obtained.

2.2.2 Gene Annotations

Gene annotations using GO terms for all the *A. thaliana* genes is an ongoing project started in 2002. The weekly updated gene annotations can be downloaded from TAIR. The annotations we used are from the version for November 13, 2004.

Using these annotations, we categorized the genes into *labeled* genes and *unlabeled* genes. The labeled genes are those which have at least one GOBP annotation; the unlabeled genes are those which have no GOBP annotations. The unlabeled genes formed the prediction data (**P**) and the labeled genes formed the training data (**T** and **M**). There were 14,285 labeled genes and 8,461 unlabeled genes in our dataset.

We further divided the 14,285 labeled genes into *positives* and *negatives* for each GOBP: Positives are the genes that belong to this GOBP; negatives are the genes that do not belong to this GOBP. Positives are manually curated by experts and we treat them as true. On the other hand, negatives may contain noise: Current annotations only provide knowledge of postives and are incomplete; some negatives may be actually positives but were regarded as negatives because of lack of annotations. However, the noise should be negligible since most GOBPs involve very few positives but many more negatives;

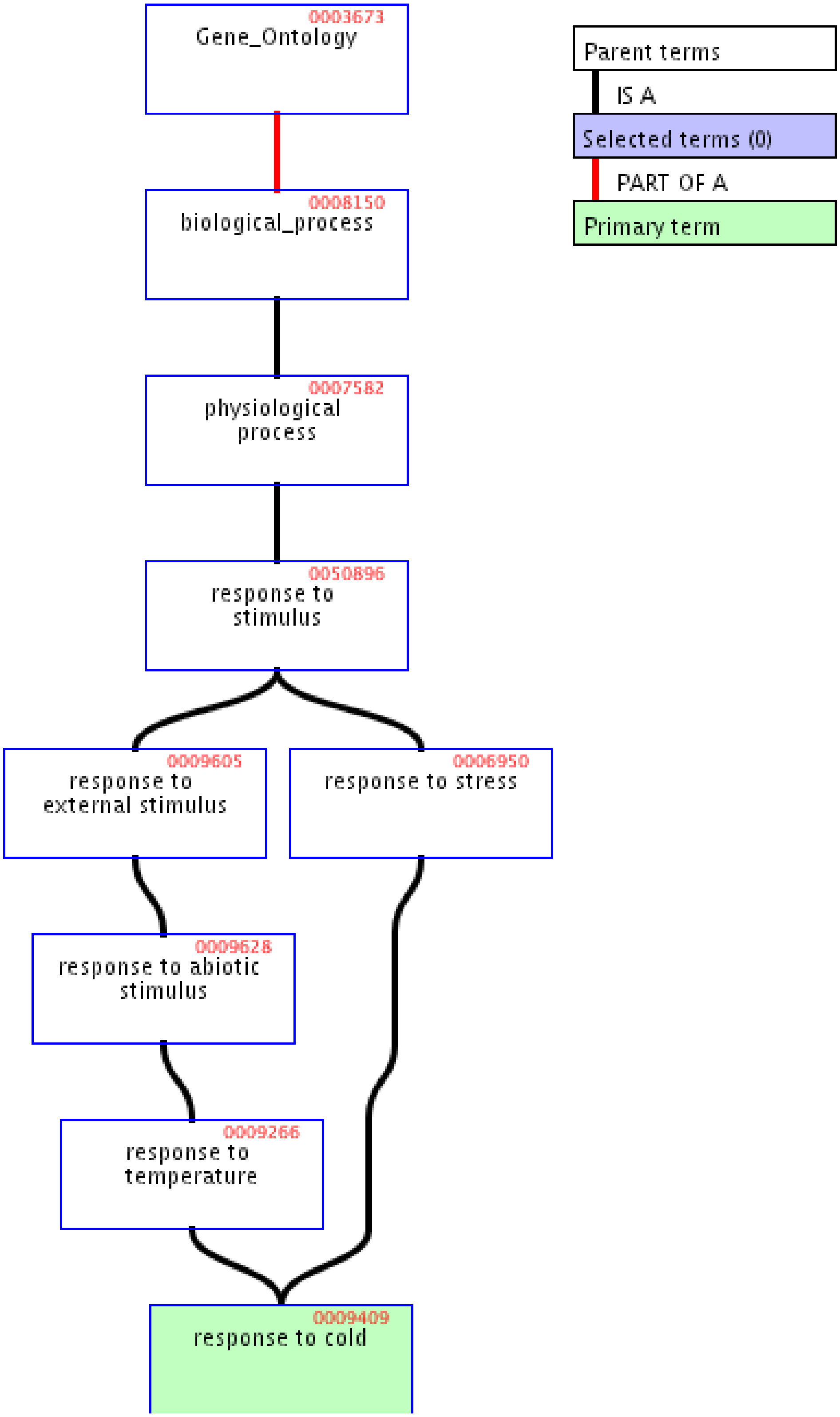
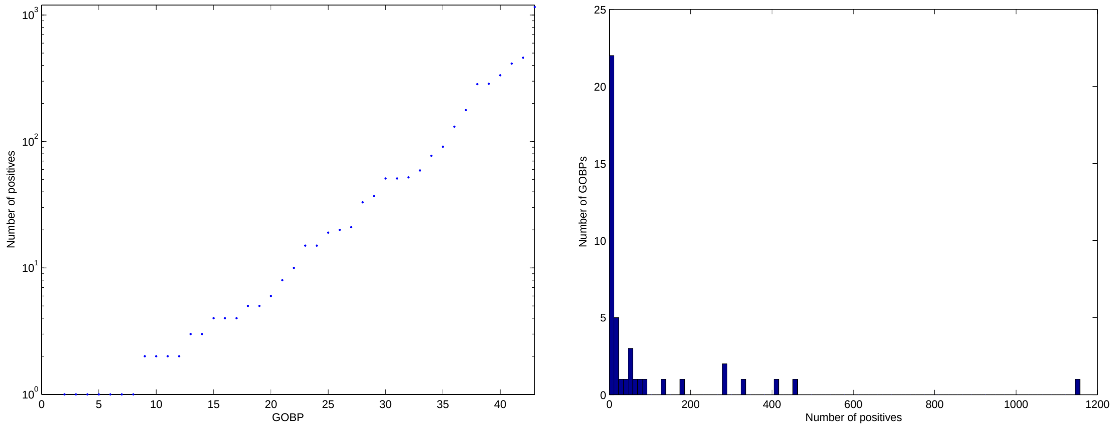


Figure 2.3: A part of the gene ontology

most negatives are real negatives. Figure 2.4 summarizes the number of positives of the up-propagated stress GOBPs of interest. The training data were unbalanced: The ratio of the number of positives to the number of negatives was less than 1:10 for all these GOBPs.



(a) Semilog plot

(b) Histogram

Figure 2.4: Statistics about the number of positives of the up-propagated stress GOBPs. After up-propagation, 42 GOBPs included at least one gene and 22 included at least 10 genes. Only those GOBPs with at least 10 genes were studied. The top stress GOBP (*GO:0006950[response to stress]*) included 1,157 genes, which consisted of only 8% of the genes in the training data. (a) From left to right, the number of positives for each stress GOBP is plotted in increasing order; (b) Most of the stress GOBPs had a number of positives less than 200.

Chapter 3

Comparison and Evaluation of Classifiers in Gene Function Prediction

We studied 22 stress GOBPs (up-propagated). For each of these GOBPs, all its offspring propagated their genes upward to it in the hierarchy. These GOBPs were not selected according to their suitability for learning; they were selected because they had at least 10 positives. As discussed in Chapter 2, the functional classes we learned were extremely unbalanced: many negatives and few positives. We did not expect any classification method to function well with less than 10 positives in a sea of more than ten thousand negatives.

We will describe the discrimination classification methods in Section 3.1. In Section 3.2, we will present our approaches to evaluate these methods. Prediction methods as well as precision estimates will be discussed in Section 3.3.

3.1 Overview of Discrimination Classification Methods

This section briefly describes the classification methods used in this study. All these methods are discriminative. In binary classification, they produce a discriminant value $dv0$, the probability that a test sample belongs to Class 0, and a discriminant value $dv1$, the probability that a test sample belongs to Class 1. A test sample is assigned to Class 1 if and only if $dv1/dv0 > t$, for a chosen decision threshold, t. In our application, Class 0 refers to not belonging to a specific GOBP; Class 1 refers to belonging to a specific GOBP; a test sample is a gene represented by a p-dimensional feature vector $\mathbf{x} = (x_1, x_2, ..., x_p)^T$ whose values are from gene expression levels. Genes in Class 1 are positives and genes in Class 0 are negatives.

3.1.1 Logistic Regression

Logistic Regression (LR) generates linear boundaries between classes: It models posterior probabilities for K classes as linear functions of $\mathbf{x}$, and thus defines linear decision boundaries between the K classes. In the two-class classification problem, the model has the simple form

$$log\,\frac{p(k=1|\mathbf{x})}{p(k=0|\mathbf{x})} = \beta_0 + \beta_1^T\mathbf{x} \tag{3.1}$$

Hence,

$$\begin{aligned} p(k=1|\mathbf{x}) &= \frac{e^{\beta_0+\beta_1^T\mathbf{x}}}{1+e^{\beta_0+\beta_1^T\mathbf{x}}} & (3.2)\\ p(k=0|\mathbf{x}) &= \frac{1}{1+e^{\beta_0+\beta_1\mathbf{x}}} & (3.3)\end{aligned}$$

and $p(k=1|\mathbf{x}) + p(k=0|\mathbf{x}) = 1$. Parameters $\beta = \{\beta_0, \beta_1\}$ are fitted using maximum likelihood [10].

3.1.2 Linear Discriminant Analysis

In Linear Discriminant Analysis (LDA), feature vectors $\mathbf{x}$ are modeled as a multivariate Gaussian:

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{(\mathbf{x}-\mu_\mathbf{k})^T \Sigma^{-1}(\mathbf{x}-\mu_\mathbf{k})}{2}} \tag{3.4}$$

where μ_k is a p-dimensional vector denoting the mean for class k, and Σ, the covariance matrix, is a $p \times p$ matrix. Each class is assumed to have a common covariance matrix Σ. The linear discriminant function for class k is

$$\delta_k = \mathbf{x}^T \Sigma^{-1} \mu_k - \frac{1}{2} {\mu_k}^T \Sigma^{-1} \mu_k + log\ \pi_k \tag{3.5}$$

The parameters π_k, μ_k and Σ can be estimated using maximum likelihood [10].

$$\begin{aligned} \pi_k &= \frac{n_k}{n} && (3.6) \\ \mu_k &= \sum_i \frac{\mathbf{x}_i}{n_k} && (3.7) \\ \Sigma &= \sum_k \sum_{g_i \in k} \frac{(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{(n-K)} && (3.8) \end{aligned}$$

where n is the number of total samples, n_k is the number of samples in class k, and K is the number of classes. In this study, $K = 2$.

3.1.3 Quadratic Discriminant Analysis

Without the common covariance matrix assumption in LDA, each class in Quadratic Discriminant Analysis (QDA) has a separate covariance matrix Σ_k. Therefore, QDA can be thought as a generalization of LDA. The discriminant function for class k is

$$\delta_k = -\frac{1}{2} log(|\Sigma_k|) - \frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k) + log\ \pi_k \tag{3.9}$$

3.1.4 Naive Bayes

Naive Bayes (NB) is based on the independent variable assumption. Variables in the feature vector $\mathbf{x}$ are assumed to be independent. This assumption allows class conditional density $p(x_i|k)$ to be estimated separately for each variable. In essence, Naive Bayes simplifies a multidimensional density estimation to a one-dimensional density estimation: Given a class k, each variable in the p-dimensional feature vector $\mathbf{x} = (x_1, x_2, ..., x_p)^T$ is independent; so

$$p(\mathbf{x}|k) = \prod_{i}^{p} p(x_i|k) \tag{3.10}$$

where $p(x_i|k)$ is the class conditional probability of x_i in class k. For each class k, estimate the distribution of the ith variable $p(x_i|k)$. Using Bayes Rule, we obtain

$$p(k|\mathbf{x}) \propto p(k) \prod_{i}^{p} p(x_i|k) \tag{3.11}$$

where $p(k)$ is the ratio of the number of the samples in class k to the number of total samples. To obtain $p(x_i|k)$, a typical way is to model the distribution of each variable as a Gaussian, $p(x_i|k) = N(\mu_i, \sigma_i)$. An alternative to estimating $p(x_i|k)$ is to discretize the continuous variables. After discretization, any original quantitative value $x_i \in (l_i, u_i]$ is replaced by x'_i. Hence estimating $p(x_i|k)$ converts into estimating $p(x'_i|k)$, which can be properly estimated from corresponding frequencies. In our data, the variables are continuous and are roughly Gaussian; hence, we assumed the Gaussian distribution for each variable.

3.1.5 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a nonparametric model since it needs the entire training set but requires few other parameters. The only parameter is K, the number of nearest neighbors.

Given a sample represented by a feature vector $\mathbf{x}$, KNN finds its K nearest neighbors using a distance metric. We used the Pearson correlation coefficient as the distance metric in this study. The distance d_{ij} of two sample vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as follows:

$$d_{ij} = 1 - \frac{(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T(\mathbf{x}_j - \bar{\mathbf{x}}_j)}{\sqrt{(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_j)^T(\mathbf{x}_j - \bar{\mathbf{x}}_j)}} \tag{3.12}$$

To classify a test sample $\mathbf{x}$, its nearest K neighbors are selected. Assume K_0 neighbors are in Class 0 and K_1 neighbors in Class 1. $K_0 + K_1 = K$. A test sample is assigned to Class 1 if and only if $K_1/K_0 > t$, where t is a decision threshold.

3.2 Comparison and Evaluation of the Classification Methods

In this section we will describe the evaluation approaches of assessing all the classification methods mentioned above when they were used for gene function prediction. The three key methods of evaluation were cross-validation (Section 3.2.1), randomizations (Section 3.2.5) and permutations (Section 3.2.6).

3.2.1 Cross-Validation

A 20-fold cross-validation was used to assess classification quality of the methods as well as to estimate precision of predictions. We randomly divided the training data into 20 nonoverlapping equal-sized parts. One part was used for testing; the other 19 parts were used to train a model. The discriminant value for each of the testing samples was predicted using this model. This procedure was repeated 20 times so that we obtained the discriminant values for all genes in the training data.

We then fitted a model using the whole training set and predicted on prediction data. Each gene in the prediction data was assigned a discriminant value. These genes were sorted in decreasing order according to their discriminant values. From top to bottom, we picked up one gene as a prediction with discriminant value dv and estimated the precision of the prediction using the cross-validation results in the training data. Let PP be the number of predicted positives in the training data whose discriminant values were greater than dv. Let TP be the number of true positives in the set of the predicted positives. The precision of the prediction was TP/PP.

3.2.2 Precision vs. Classification Rate

A confusion matrix represents the classifications predicted by a classification method versus the correct classifications. For a two-class classification problem, Table 3.1 shows the confusion matrix.

Table 3.1: A confusion matrix for the two-class classification problem. TP = the number of true positives; FN = the number of false negatives; FP = the number of false positives; TN = the number of true negatives.

	Predicted	
True	1	0
1	TP	FN
0	FP	TN

Precision is defined as the ratio of the number of true positives to the number of predicted positives (TP+FP), i.e.,

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.13}$$

Classification rate is defined as the ratio of the number of correctly classified samples (TP+TN) to the number of total samples, i.e.,

$$\text{classification rate} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{3.14}$$

We were concerned with precision rather than classification rate. The classification rate can be high while the precision is low, particularly when the number of positives is far fewer than the number of negatives. For example, suppose there are 92 negatives and 8 positives, and the confusion matrix is in Tabel 3.2, then precision = 40% and classification rate = 90%.

Since running biological experiments in the wet lab is time-consuming and expensive, a high precision for the predictions is much more desirable than the overall classification

Table 3.2: An example of confusion matrix

	Predicted	
True	1	0
1	4	4
0	6	86

rate. Few predictions, but accurate ones, are important. Biologists are interested in knowing the number of true and false positives; we tried to minimize false positives. In addition, biologists are much more interested in positive predictions than negative predictions, since negative predictions are hard to be confirmed in the laboratory.

3.2.3 The Reasons Not to Use the ROC Curve for Evaluating Classification Methods

The receiver operating characteristic (ROC) curve is a plot commonly used to assess a classifier by summarizing the tradeoffs between sensitivity (Sn) and specificity (Sp). The terms sensitivity and specificity are defined as follows:

- *Sensitivity:* the probability of predicting 1 given true class is 1, i.e., $\text{Sn} = \frac{\text{TP}}{\text{TP+FN}}$;
- *Specificity:* the probability of predicting 0 given true class is 0, i.e., $\text{Sp} = \frac{\text{TN}}{\text{FP+TN}}$.

Despite its popularity, we decided not to use it because of two reasons. First, the ROC curve does not show precision; precision cannot be readily interpreted from the ROC curve. Second, we faced highly unbalanced classes having many more negatives than positives (see Table 3.3). We only expected a small number of good predictions, thus knowing how many were true positives among the predictions was more important than knowing sensitivity and specificity. Therefore, we used the TP vs. PP plot to evaluate the classification method, as shown in Figure 3.1, from which we can read off

Table 3.3: The number of positives of the up-propagated stress GOBPs (#pos + #neg = 14,285)

GO-BP	#pos
GO:0006950[response to stress]	1157
GO:0009613[response to pest, pathogen or parasite]	460
GO:0042828[response to pathogen]	413
GO:0042829[defense response to pathogen]	334
GO:0006974[response to DNA damage stimulus]	286
GO:0006281[DNA repair]	284
GO:0009814[defense response to pathogen, incompatible interaction]	177
GO:0006979[response to oxidative stress]	131
GO:0009627[systemic acquired resistance]	77
GO:0009861[jasmonic acid and ethylene-dependent systemic resistance]	59
GO:0009618[response to pathogenic bacteria]	52
GO:0009409[response to cold]	51

the precision by computing TP/PP. The TP vs. PP plot will be described in detail in the next three sections.

3.2.4 The TP vs. PP Plot

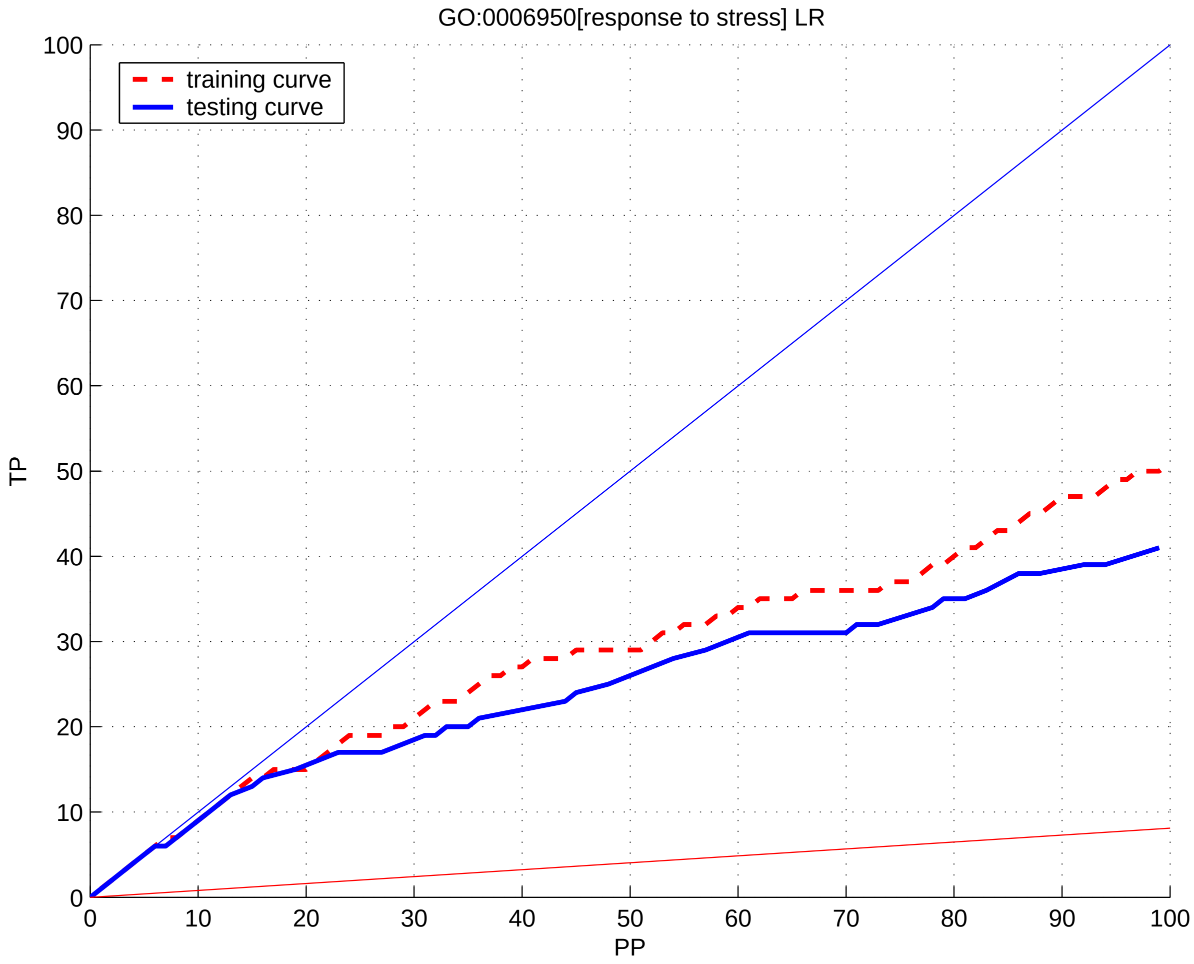


Figure 3.1: The simple TP vs. PP plot

Figure 3.1 shows an example of the TP vs. PP plot. The horizontal axis is the number of predicted positives (PP) that are predicted to belong to the GOBP *GO:0006950[response to stress]*; the vertical axis is the number true positives (TP). The upper diagonal line (blue) is the performance of a perfect classifier, in which all the predicted positives are true positives. The blue testing curve shows the testing performance of LR. It is comparable to a ROC curve since it measures the performance of the classifier at different thresholds. The dotted red curve is the training performance of LR. The lower diagonal line (red) is the expected performance of a random classifier that makes random guesses. The blue testing curve is much higher than the red diagonal line, which means LR is far better than a random classifier.

This plot was generated based on the 20-fold cross-validation results in the training data. As discussed in Section 3.1, for any classification method, we would have the discriminant value for each gene after 20-fold cross-validation. We sorted these values decreasingly, and then counted PP and TP as moving down the sorted list.

The TP vs. PP plot emphasizes the absolute number of positives and negatives but not proportion. It is useful when the number of predictions is small. For a certain number of predictions, the precision can be read readily from the plot by calculating TP/PP. From this plot, it is also easy to judge if a classifier is better than a random classifier; if it is not, then we have no confidence on the predictions and simply discard all the predictions.

3.2.5 Randomizations

In Figure 3.1, the testing curve is based on only one randomization (one random split of the training data into 20 folds) of the training data. One would expect that different randomizations may result in different curves. That is true. Different randomizations may produce different discriminant values for each sample in the training data because the classifier used to compute the discriminant value for a particular testing sample varies in different randomizations. One exception is leave-one-out cross-validation in which the number of parts is equal to the number of samples.

To evaluate a classifier more thoroughly, we examined the distribution of these testing curves. Thus we repeated the cross-validation procedure 100 times, each corresponding to a different randomization of the training data (see Algorithm C.1.1 in Appendix C).

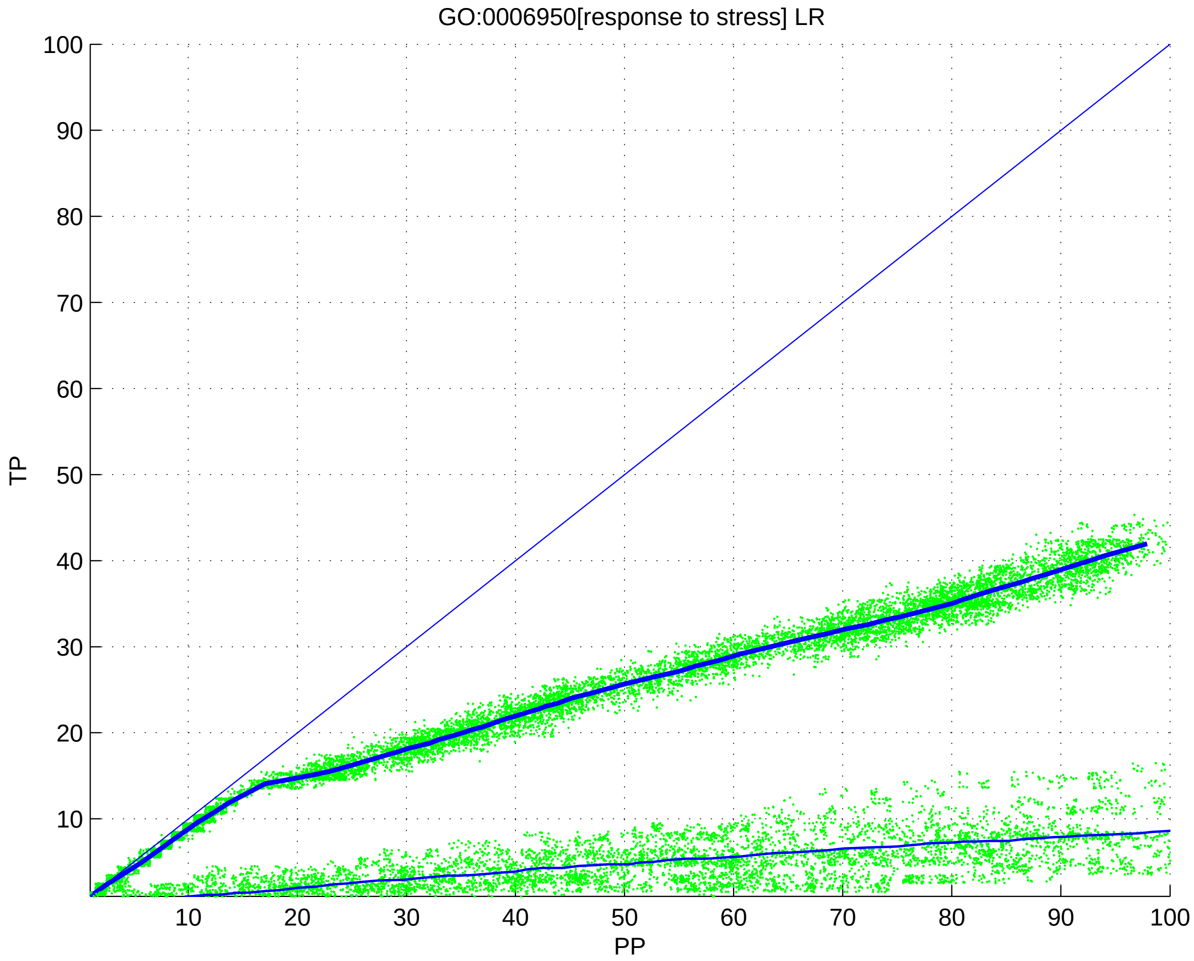


Figure 3.2: The TP vs. PP plot with randomizations and permutations

As shown in Figure 3.2, the upper green curve cloud represents the 100 randomizations and the blue curve represents the mean of these randomizations. It can be seen that LR is quite stable in classifying *GO:0006950[response to stress]* since the cloud band is quite narrow.

3.2.6 Permutations

Another important issue is whether a classifier learned from the training data is good just by chance. To answer this question, we permuted the GOBP label (the label was randomized, but the number of positives as well as the number of negatives were unchanged) 100 times, and learned random classifiers correspondingly using the permuted data (see Algorithm C.1.2 in Appendix C).

What would the testing curves of the random classifiers look like? One would expect that the random classifiers would perform poorly because their expected performance is poor, as shown in Figure 3.1. The lower green curve cloud in Figure 3.2 shows the testing curves for such 100 random classifiers. The lower blue curve is the mean of these curves. In Figure 3.2, the testing curves using the original data are much higher than those using the permuted data; hence, the classifier learned from the original data is not good just by chance. In cases the two clouds of curves overlap, we simply discard the classifier.

3.3 Gene Function Prediction for Unlabeled Genes and Precision Estimate

Thus far, we have discussed how to evaluate a classifier on the training data using 20-fold cross-validation, randomizations and permutations. Our final goal is to make some useful predictions of gene functions for biologists to validate. The general procedure was as follows. First, for a particular GOBP of interest, a classifier was trained using the training data. Second, the prediction data were predicted using this classifier. Each gene in the prediction data would have a discriminant value after this step. Third, these genes were sorted by their discriminant values in decreasing order. From bottom to top, the confidence level of the prediction increased. Top genes in the sorted list were more confident predictions. The final step was to associate each of these gene with a mean precision and a standard deviation of the precision. The precision associated with a gene is the precision of all the predictions in the sorted list above and including the gene, that is, the precision of all the predictions that are at least as confident as the gene in question. We used the testing data to estimate these precisions.

Given a gene with discriminant value dv, we associated a precision with the gene in three different ways: $p1$ was the unweighted mean precision over all randomizations; $p2$ was the weighted mean precision; $p3$ regarded the mean precision estimation as a regression problem.

$$p_1 = \frac{1}{N}\sum_i^N \frac{\mathrm{TP}_i}{\mathrm{PP}_i} \tag{3.15}$$

$$\begin{aligned} p_2 &= \frac{\sum_i^N \mathrm{TP}_i}{\sum_i^N \mathrm{PP}_i} \\ &= \sum_i^N w_i \times \frac{\mathrm{TP}_i}{\mathrm{PP}_i} \end{aligned} \tag{3.16}$$

$$p_3 = \frac{\sum_i^N \mathrm{TP}_i \mathrm{PP}_i}{\sum_i^N \mathrm{PP}_i \mathrm{PP}_i} \tag{3.17}$$

where $w_i = \mathrm{PP}_i / \sum_j^N \mathrm{PP}_j$, N is the number of randomizations of the testing data, TP_i and PP_i are the number of true positives and the number of predicted positives in the ith randomization, respectively. More specifically, PP_i is the number of genes in the testing data whose discriminant values are greater than dv, and TP_i is the number of these predictions that are true.

Figure 3.3 shows the three mean precision estimates (p_1, p_2, and p_3) of the top 100 predictions made by LR for *GO:0006950[response to stress]*. The mean precision these estimates produced was almost the same (the three mean precision curves overlap); we

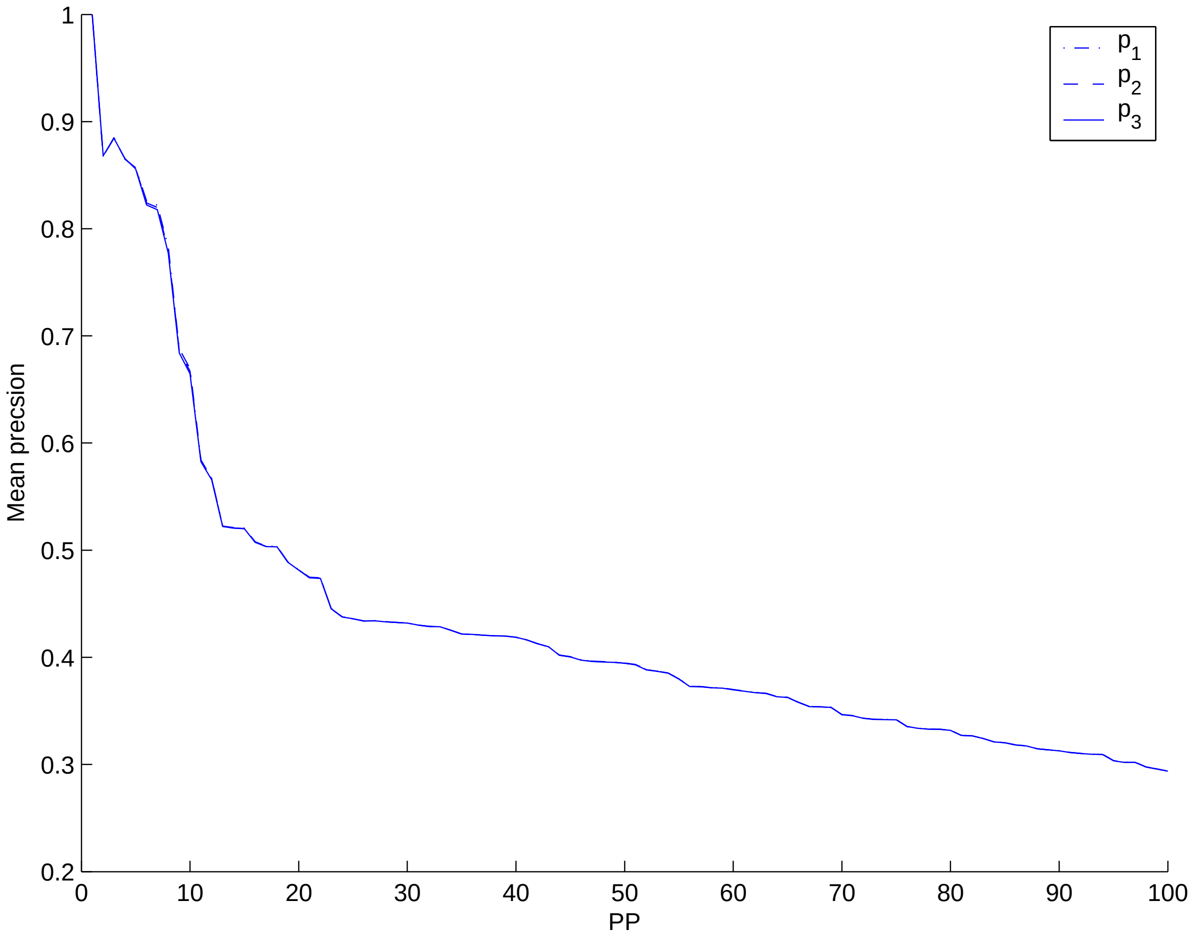


Figure 3.3: The mean precision vs. the number of predictions (PP) plot generated using the 100 randomizations

chose p_2 as our precision estimate.

The standard deviation of the precision for a prediction is calculated using the following formula:

$$sd = \frac{std(\mathrm{TP})}{mean(\mathrm{PP})} \tag{3.18}$$

where $\mathrm{TP} = \{\mathrm{TP}_1, \mathrm{TP}_2, \ldots, \mathrm{TP}_N\}$, $\mathrm{PP} = \{\mathrm{PP}_1, \mathrm{PP}_2, \ldots, \mathrm{PP}_N\}$, and N is the number of randomizations. $N = 100$ in this study. We chose not to use $std(\mathrm{TP}_i/\mathrm{PP}_i)$ because when both TP_i and PP_i are 0, $\mathrm{TP}_i/\mathrm{PP}_i$ is not a number (NaN).

Table 3.4: The examples of predictions for *GO:0006950[response to stress]* made by LR

gene	DV	M	SD
AT3G22240	0.8007	1.0000	0.3313
AT3G28290;AT3G28300	0.7357	0.8678	0.1566
AT4G39675	0.6691	0.8847	0.0936
AT3G14210	0.6229	0.8653	0.0698
AT1G16850	0.6052	0.8564	0.0478

Table 3.4 shows the top 5 predictions for *GO:0006950[response to stress]* made by LR (see Algorithm C.2.1 in Appendix C). The columns, from left to right, are gene name, discriminant value, mean precision, and standard deviation of the precision. Each gene, g, in the table is a prediction. If a gene in the table has a mean precision of, for example, 0.75, then we expect that 75% of the genes in the table above and including g will be correct predictions. More predictions can be added to this table; but as the number of predictions increases, the mean precision tends to decrease, as shown in Figure 3.4 (the precision vs. PP plot).

Figure 3.4 summarizes the precision of the top 100 predictions for *GO:0006950[response to stress]* made by LR (see Algorithm C.3.1 in Appendix C). The cloudy, multi-coloured curve represents the estimated precision in 100 randomizations. It shows the standard deviation of the precision vividly. The blue curve is the mean precision. The precision

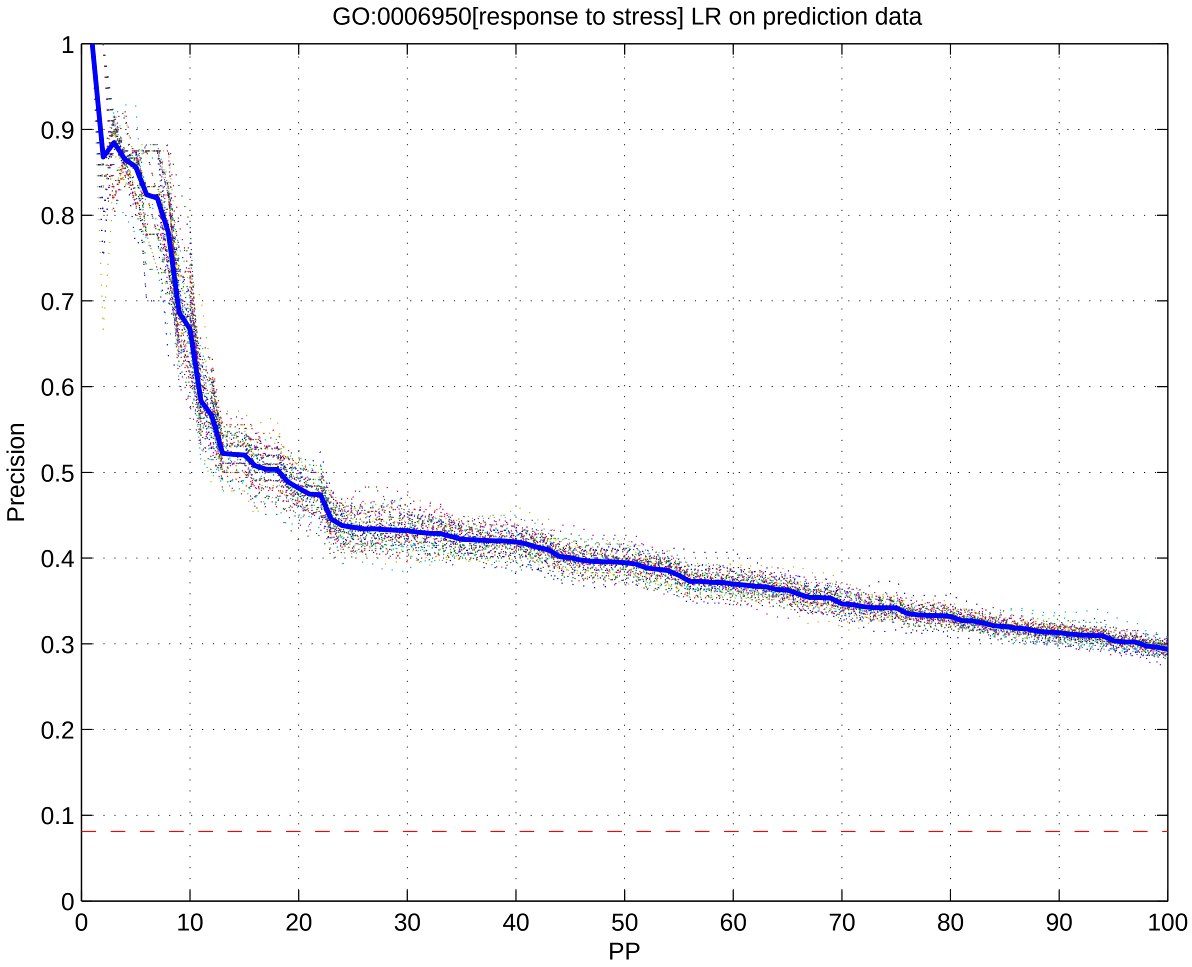


Figure 3.4: The precision vs. the number of predictions (PP) plot

that a random classifier could achieve is illustrated by the dashed horizontal line, which is much lower than the cloudy curve. The height of the dashed horizontal line is equal to the ratio of the number of positives to the number of total genes in the testing data.

Chapter 4

Results of Evaluation, Comparison and Prediction

In this chapter, we will present our results from evaluation and comparison of all the classification methods mentioned in Chapter 3 in predicting gene function in *A. thaliana*, as well as from the predictions we made using these classification methods for stress GOBPs. The evaluation approaches have been discussed in detail in Chapter 3. For clarity, we just present the results for the top stress GOBP *GO:0006950[response to stress]* but leave the results for other stress GOBPs to Appendix A and Appendix B.

4.1 The TP vs. PP Plots and Precision vs. PP Plots for the Classifiers

As discussed in Chapter 3, the two main approaches we used to evaluate the performance of classifiers in classification and prediction were the TP vs. PP plot and precision vs. PP plot. The TP vs. PP plot is useful for evaluation when the training data are highly unbalanced, and when we only expect a few accurate predictions. The precision vs. PP plot summarizes the quality of the predictions we made in the prediction data.

Figures 4.1, 4.2 and 4.3 show the evaluation results (the plots on the left panel) in the training data as well as the summaries of prediction quality (the plots on the right panel) for LR, LDA, QDA, NB and KNN.

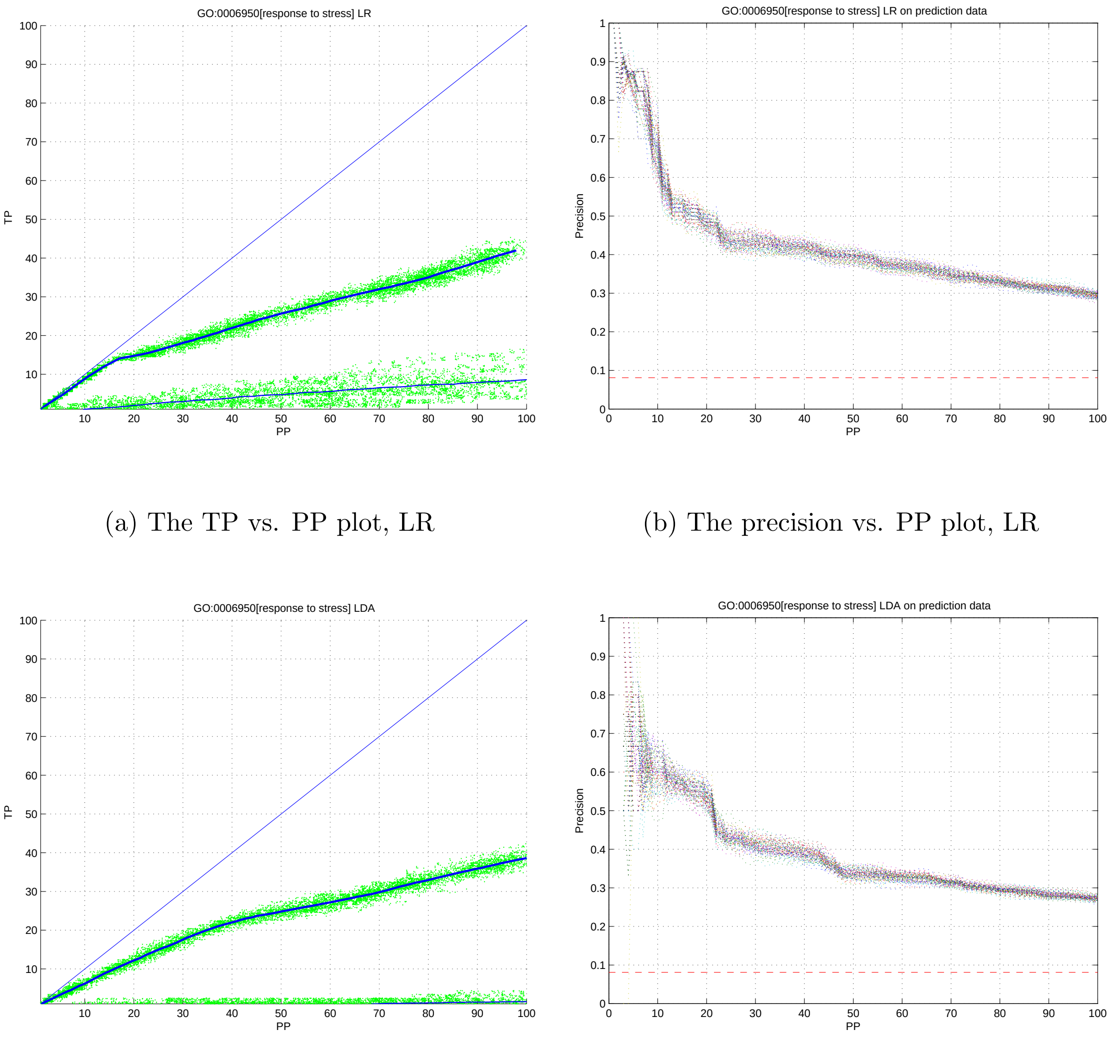
In the TP vs. PP plots, it can be seen that each classifier is far better than the random classifier, which demonstrates that each classifier can capture the correlation between gene expressions and gene functions. In other words, these plots confirm that gene expressions and gene functions correlate to some degree, but not completely, suggesting how complex life could be, even for this simple plant.

The precision vs. PP plots summarize the precision of the top 100 predictions using the five classifiers. At the rightmost (low precision) end of each curve cloud, no significant difference can be observed among these classifiers, and the precision is roughly between 0.26 to 0.33. However, at the leftmost (high precision) end, LR almost always has lower variance and higher mean precision than the other methods; KNN and LDA have the highest variance.

4.2 Prediction Tables

Tables 4.1 to 4.5 show the top 10 predictions for the top stress GOBP *GO:0006950[response to stress]* made by each of the classifiers. Each row is a prediction. From left to right, the columns are gene name, discriminant value, mean precision, and standard deviation of the precision.

Examining these tables closely, we find that the gene AT4G39675 appears in all of the five tables; all the five classifiers predict it to be involved in response to stress. The gene AT3G28290;AT3G28300 wins 4 votes. AT3G22240, AT2G05510, and AT5G10040 obtain three votes each. These genes are believed to be high confidence predictions since not only they are among the top 10 predictions, but also they appear frequently in the prediction tables.



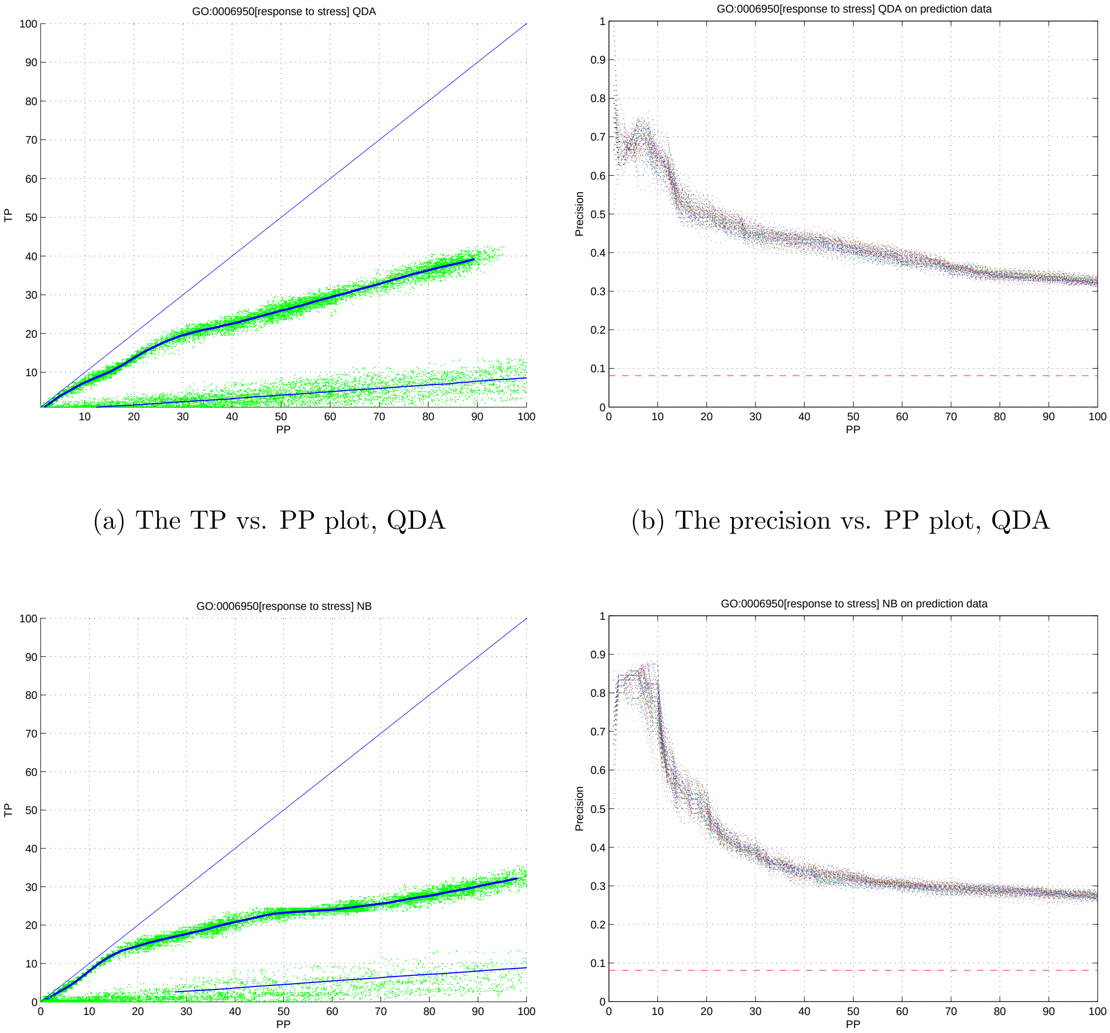
(a) The TP vs. PP plot, LR

(b) The precision vs. PP plot, LR

(c) The TP vs. PP plot, LDA

(d) The precision vs. PP plot, LDA

Figure 4.1: The evaluation of classification performance (left panel) and prediction performance (right panel) of the classifiers on the top stress GOBP *GO:0006950[response to stress]*



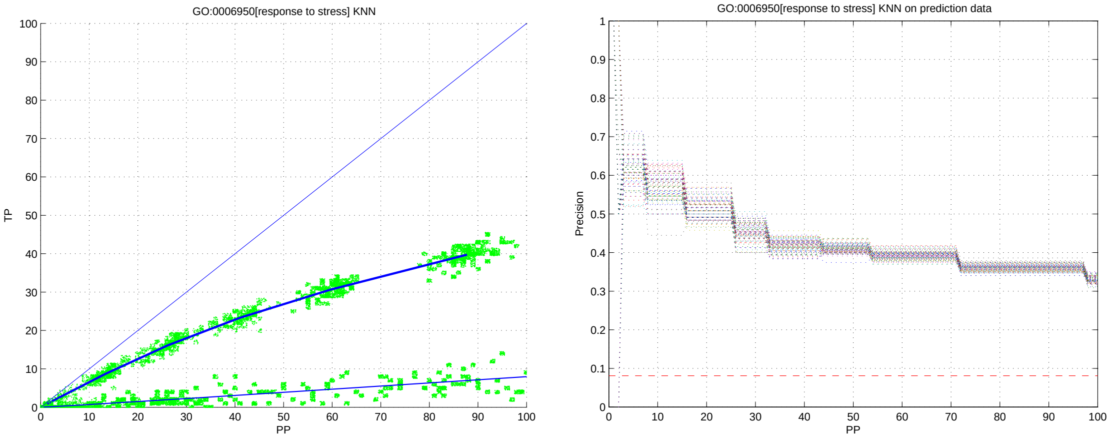
(a) The TP vs. PP plot, QDA

(b) The precision vs. PP plot, QDA

(c) The TP vs. PP plot, NB

(d) The precision vs. PP plot, NB

Figure 4.2: The evaluation of classification performance (left panel) and prediction performance (right panel) of the classifiers on the top stress GOBP *GO:0006950[response to stress]* (continued)



(a) The TP vs. PP plot, KNN

(b) The precision vs. PP plot, KNN

Figure 4.3: The evaluation of classification performance (left panel) and prediction performance (right panel) of the classifiers on the top stress GOBP *GO:0006950[response to stress]* (continued)

Table 4.1: The top 10 predictions made by LR for *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G22240	0.8007	1.0000	0.3313
AT3G28290;AT3G28300	0.7357	0.8678	0.1566
AT4G39675	0.6691	0.8847	0.0936
AT3G14210	0.6229	0.8653	0.0698
AT1G16850	0.6052	0.8564	0.0478
AT4G38080	0.5697	0.8241	0.0242
AT5G66985	0.5685	0.8199	0.0246
AT2G05510	0.5540	0.7799	0.0320
AT5G10040	0.5252	0.6860	0.0378
AT5G09530	0.5196	0.6671	0.0397

Table 4.2: The top 10 predictions made by LDA for *GO:0006950[response to stress]*

gene	DV	M	SD
AT4G39675	0.5230	NaN	NaN
AT3G22240	0.5138	NaN	NaN
AT3G28290;AT3G28300	0.4121	0.7164	0.3034
AT2G05510	0.4071	0.7029	0.2725
AT5G66985	0.3872	0.6673	0.1349
AT4G38080	0.3852	0.6619	0.1346
AT5G09530	0.3628	0.6263	0.1246
AT2G36220	0.3256	0.6156	0.0682
AT2G01520	0.3085	0.6021	0.0615
AT3G14210	0.3062	0.6009	0.0587

Table 4.3: The top 10 predictions made by QDA for *GO:0006950[response to stress]*

gene	DV	M	SD
AT4G39675	0.5798	0.7757	0.2149
AT3G28290;AT3G28300	0.5693	0.6694	0.0539
AT1G52070	0.5688	0.6693	0.0509
AT4G00680	0.5654	0.6716	0.0385
AT3G50480	0.5637	0.6831	0.0588
AT4G33560	0.5625	0.6911	0.0669
AT5G09530	0.5621	0.6891	0.0643
AT4G38080	0.5615	0.6856	0.0601
AT4G02270	0.5599	0.6611	0.0534
AT2G23540	0.5591	0.6404	0.0488

Table 4.4: The top 10 predictions made by NB for *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G16430;AT3G16420	0.5309	0.7220	0.1184
AT4G38080	0.5284	0.8288	0.0949
AT4G23680	0.5284	0.8294	0.0932
AT2G33850	0.5280	0.8321	0.0725
AT4G39675	0.5278	0.8308	0.0757
AT2G05510	0.5277	0.8310	0.0738
AT2G42610	0.5273	0.8215	0.0646
AT5G03350	0.5269	0.7998	0.0582
AT2G01520	0.5267	0.7734	0.0542
AT3G22240	0.5266	0.7627	0.0535

Table 4.5: The top 10 predictions made by KNN for *GO:0006950[response to stress]*

gene	DV	M	SD
AT5G38940;AT5G38930	0.4510	1.0000	1.4651
AT1G80960	0.4314	0.6446	0.5063
255181_at	0.3725	0.6164	0.0867
AT4G39675	0.3725	0.6164	0.0867
AT4G02270	0.3725	0.6164	0.0867
AT3G28290;AT3G28300	0.3725	0.6164	0.0867
AT2G39310	0.3725	0.6164	0.0867
AT3G44860	0.3529	0.5683	0.0466
AT5G05500	0.3529	0.5683	0.0466
AT3G50480	0.3529	0.5683	0.0466

4.3 The Mean TP vs. PP Plots and Mean Precision vs. PP Plots Comparing the Classifiers

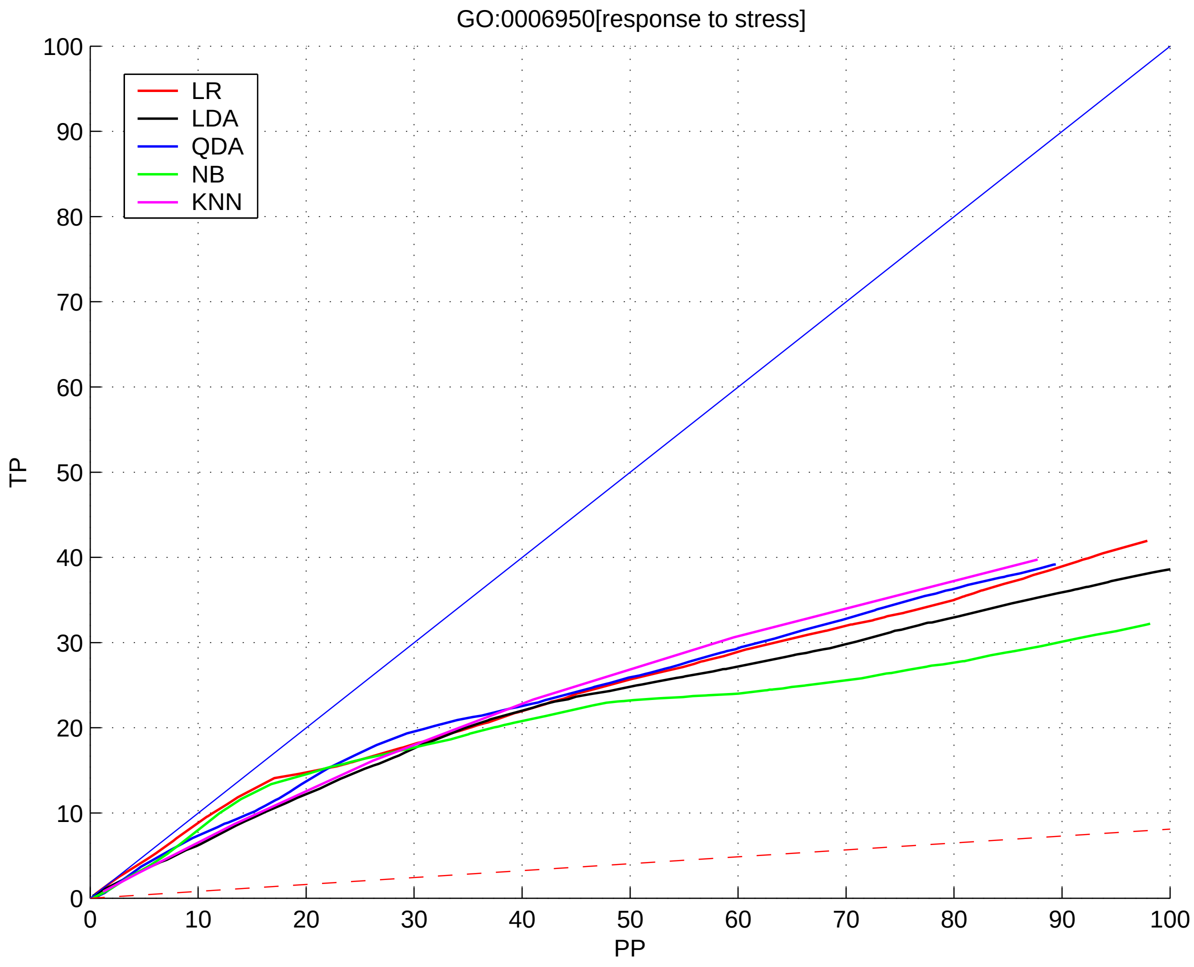


Figure 4.4: The mean TP vs. PP plot

Figure 4.4 and Figure 4.5 compare the mean performance of all the five classifiers in classification and prediction, respectively. None of the classifiers is consistently better than the others for all PP (the number of predicted positives).

In Figure 4.4, NB is comparable to the other methods when PP is less than 30, but its performance degrades greatly beyond PP = 30. LR has the best performance when PP is less than 23; however, it is outperformed gradually by QDA beyond PP = 23 and

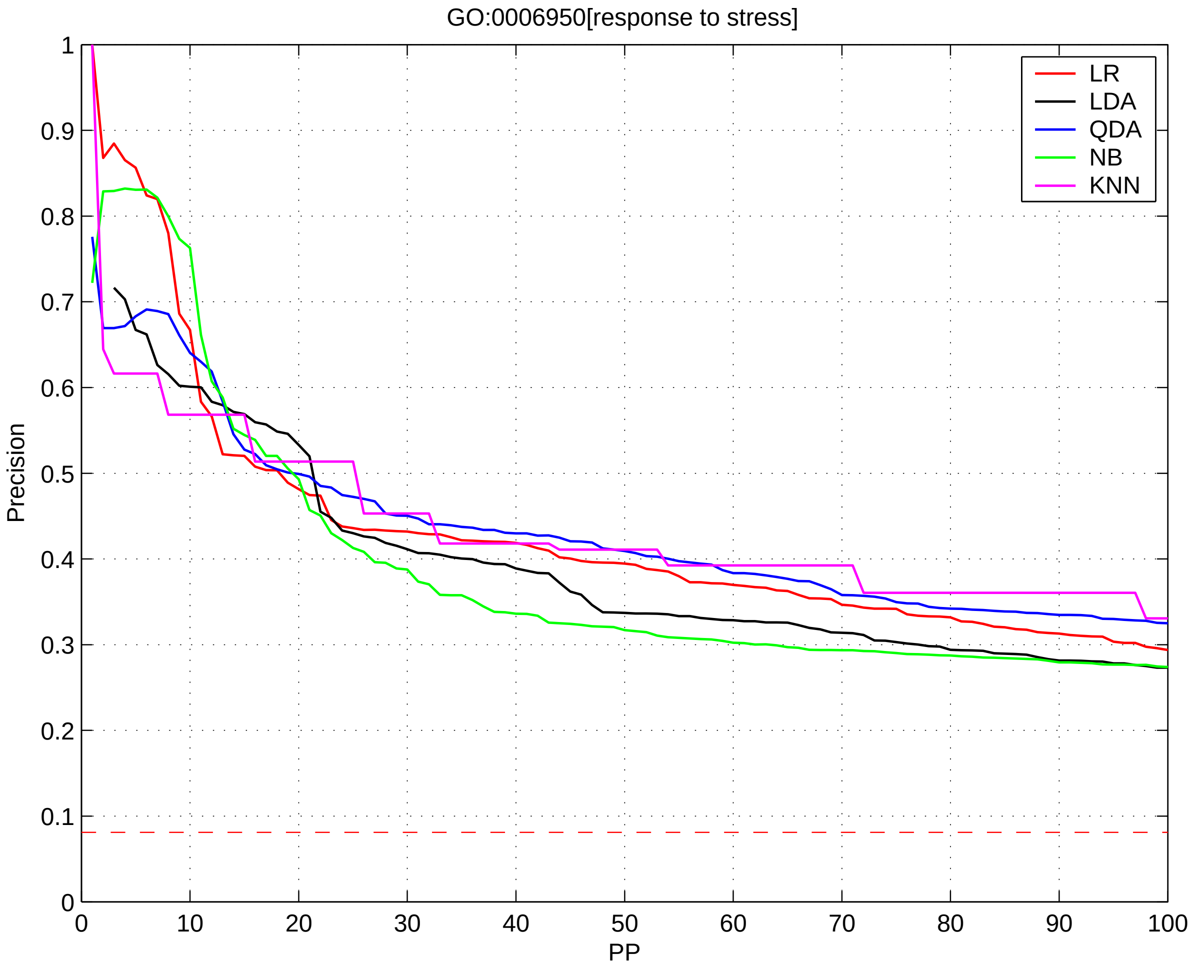


Figure 4.5: The mean precision vs. PP plot

by KNN beyond PP = 30. The performance of LDA remains the lowest in the range from PP = 7 to PP = 30, but starts to exceed NB after PP = 30, and gradually comes close to the performance of the other methods. Given this nonuniform situation, one cannot conclude that one classification method is better than the others in classification; however, when PP is given, we can tell exactly which classifier performs best.

A similar conclusion can be drawn from Figure 4.5: No classification method is uniformly better than the others in prediction. At the right end of the plot, KNN and QDA achieve higher precision than linear classifiers (LDA, LR) and NB do. At the left end of the plot, LDA, LR and NB have comparable prediction performance as KNN and QDA.

Chapter 5

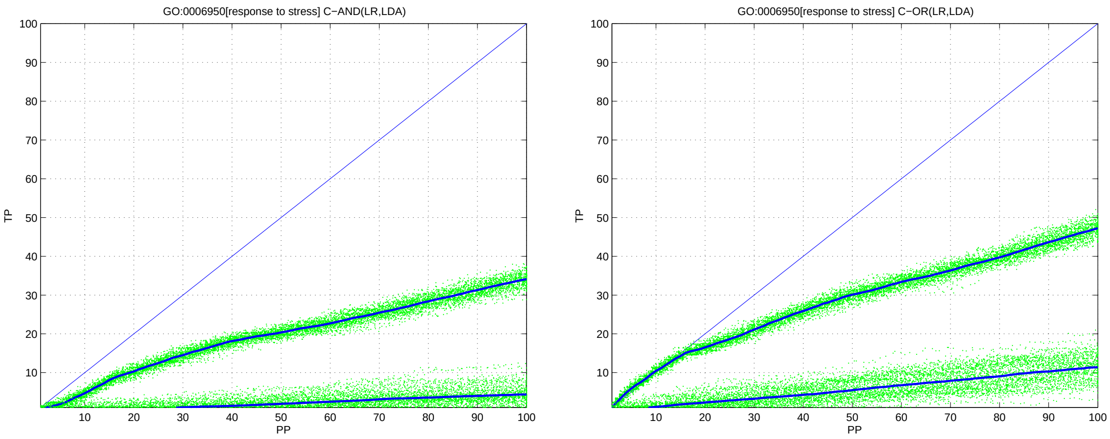
Combining Classifiers to Improve Classification Accuracy

Combining basic classifiers is an effective way to improve the accuracy of classification [10]. The basic idea is that an ensemble of experts tends to predict better than a single expert does: The other classifiers are expected to correct the error that a single classifier makes. In this chapter, we will investigate this strategy. In Section 5.1 and Section 5.2, we will discuss two simple approaches to combine basic classifiers. More sophisticated approaches, model averaging and stacking [10], will be treated in Section 5.3. We will present the results in Section 5.4. The combined classifiers were experimented on the top stress GOBP *GO:0006950[response to stress]*.

5.1 Intersection and Union Based on PP (Predicted Positives)

In statistical learning, a classifier is often referred to as a model. Define M1$\langle a \rangle$ to be Model 1 using threshold a. Define M2$\langle b \rangle$ to be Model 2 using threshold b. Define a classifier C-AND$\langle a, b \rangle$ as the Intersection model. A data point is accepted by C-AND$\langle a, b \rangle$ if and only if it is accepted by both M1$\langle a\rangle$ and M2$\langle b\rangle$. C-OR$\langle a, b\rangle$, the Union model, can be defined in the similar way except that a data point is accepted by C-OR$\langle a, b\rangle$ if and only if it is accepted by M1$\langle a\rangle$ or M2$\langle b\rangle$. The thresholds a and b are determined by the number of predicted positives (PP) of each classifier. That is, for a given number of predictions, n, a is the discriminant value of the nth best prediction of M1, and b is the discriminant value of the nth best prediction of M2. In this way, C-AND takes the intersection of the best n predictions of each classifier. (Likewise, C-OR takes the union.) This is done for $n = 1, 2, 3, ..., 100$. The pseudocode of the Intersection model can be found in Algorithm C.4.1 in Appendix C. The pseudocode for the Union model is similar to that of the Intersection model.

We chose LR as M1 and LDA as M2. Figure 5.1 presents the model evaluation for C-AND and C-OR (the TP vs. PP plots).



(a) C-AND

(b) C-OR

Figure 5.1: The TP vs. PP plots for the Intersection and Union models based on PP (predicted positives)

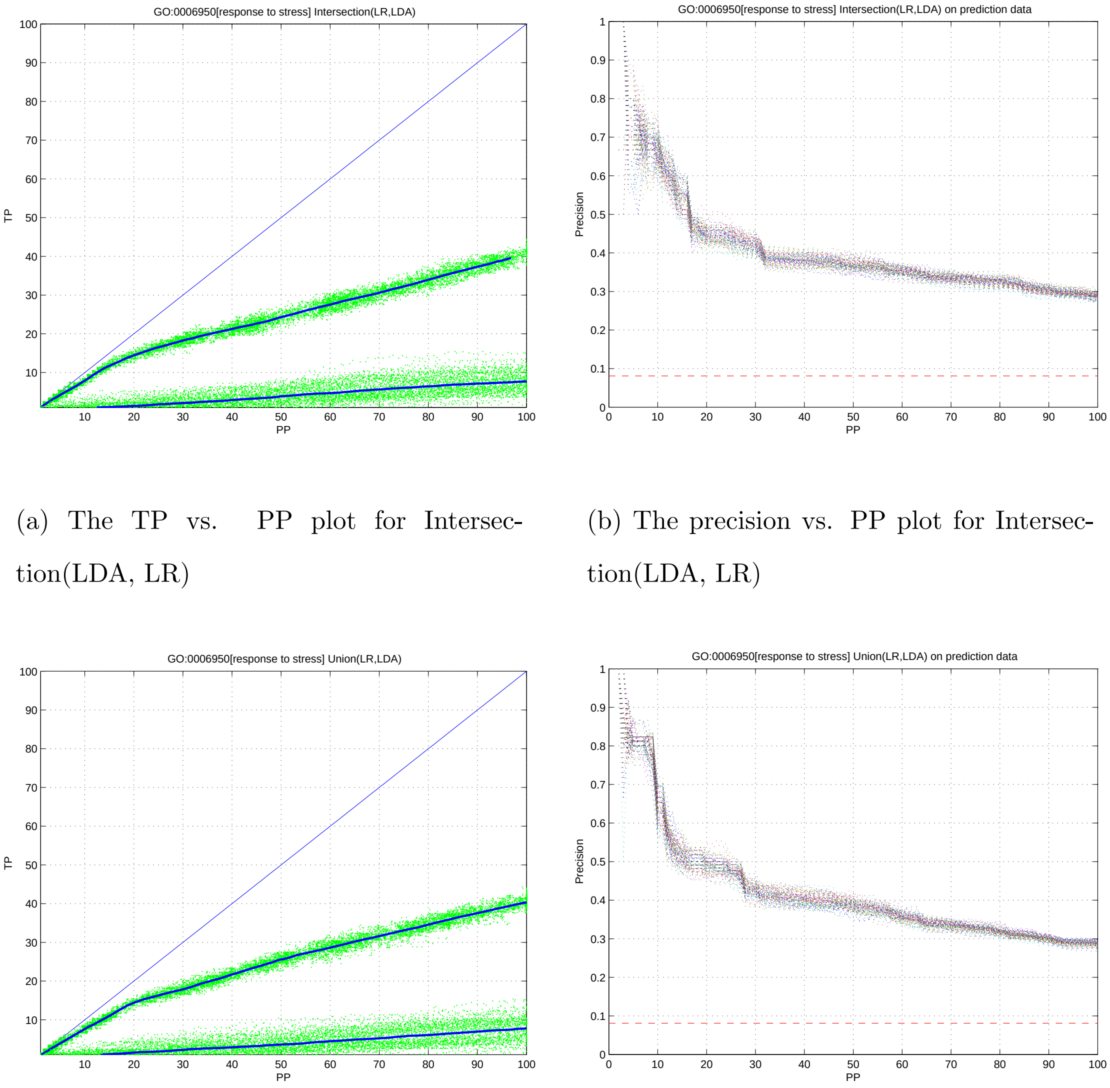
5.2 Intersection and Union Based on Discriminant Values

The intersection model described above is not discriminative in that it does not produce a discriminant value for each gene. Although each gene has two discriminant values, one from M1 and the other from M2, the model does not combine them into a single discriminant value for C-AND. In contrast, this section describes an intersection model that is discriminative.

Consider two models, M1 and M2. Each gene, i, is given two discriminant values, d_{i1} and d_{i2}, by M1 and M2, respectively. In the Intersection model, the discriminant value for gene i is defined as $d_i^{'} = min(d_{i1}, d_{i2})$; in the Union model, the discriminant value for gene i is defined as $d_i^{'} = max(d_{i1}, d_{i2})$. With this definition, for given values of d_{i1} and d_{i2}, the predictions made by the intersection model are the intersection of the predictions made by M1 and the predictions made by M2. Likewise, the union model produces the union of predictions.

As before, we chose LR as M1 and LDA as M2. Figure 5.2 shows the TP vs. PP plots as well as the precision vs. PP plots for the Intersection and Union models based on discriminant values.

Curiously, the performance of the Intersection model is not better than that of individual classifiers (see Figure 4.1(a) and Figure 4.1(c)), as shown in Figure 5.2(a). A possible explanation is given in Figure 5.3, which illustrates an intersection example. With this distribution of true positives and negatives, the Intersection model is worse than individual classifiers since the intersection set contains only false positives.



(a) The TP vs. PP plot for Intersection(LDA, LR)

(b) The precision vs. PP plot for Intersection(LDA, LR)

(c) The TP vs. PP plot for Union(LDA, LR)

(d) The precision vs. PP plot for Union(LDA, LR)

Figure 5.2: The TP vs. PP plots and precision vs. PP plots for the Intersection and Union model based on discriminant values

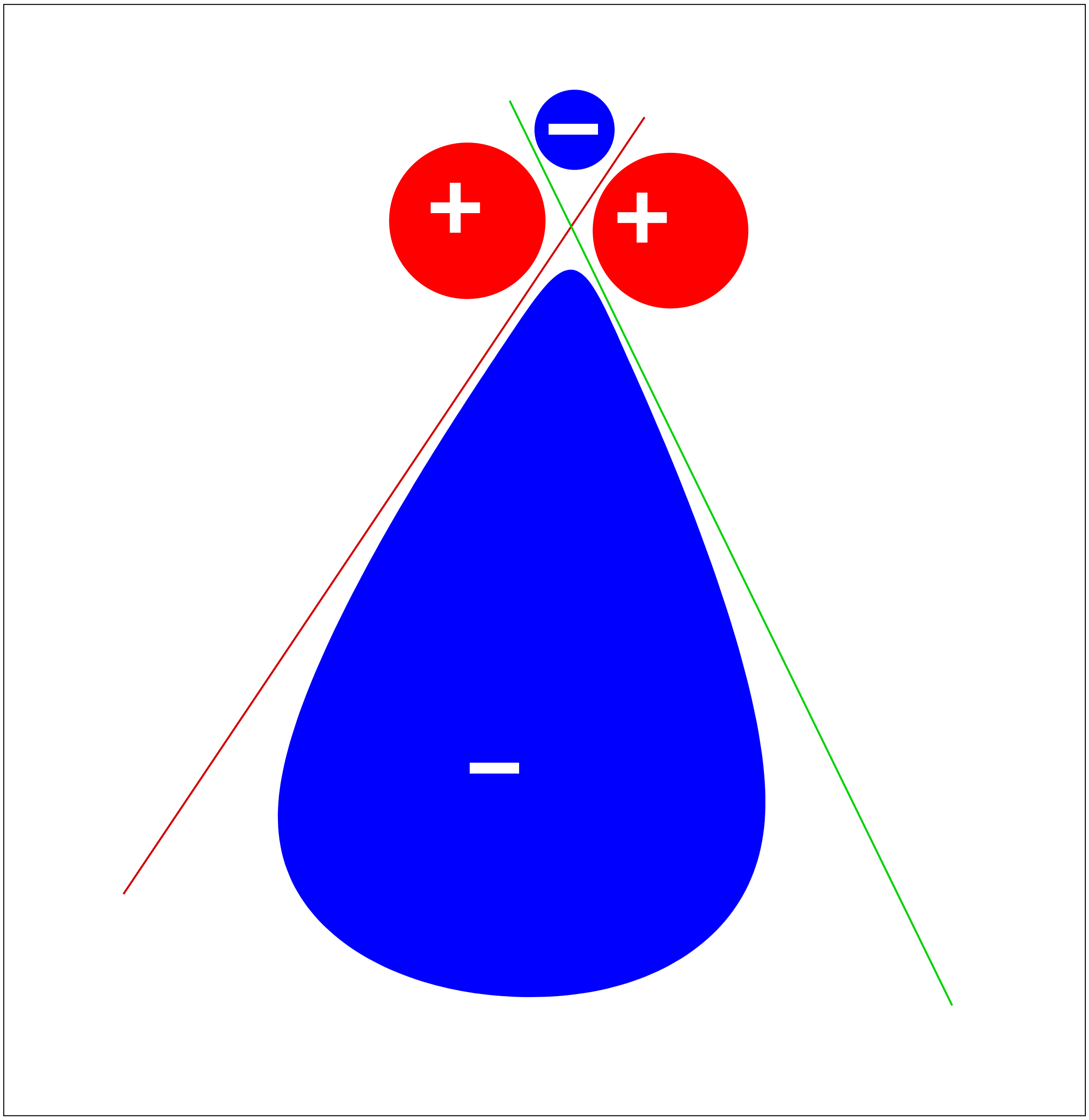


Figure 5.3: An Intersection model intersects LR and LDA. True positives are in the red areas with "+" signs. True negatives are in the blue areas with "-" signs. The green solid line is the decision boundary of LR: Positives are on the right side of the boundary; negatives are on the other side. Similarly, the red solid line is the decision boundary of LDA: Positives are on the left side of the boundary; negatives are on the right side. The top triangle area is classified as positives by the Intersection model. The Intersection model wrongly classifies the top circle as positives, which are actually negatives, as shown in the figure. The intersection model also wrongly classifies the two red areas as negatives.

5.3 Model Averaging and Stacking

Model averaging (see Algorithm C.5.1 in Appendix C) and stacking (see Algorithm C.5.2 in Appendix C) seek the best linear combination of the outputs of basic classifiers to generate a weighted output $\hat{f}(\mathbf{x})$, which is expected to be more accurate than any individual classifiers $\hat{f}_m(\mathbf{x})$ [10],

$$\hat{f}(\mathbf{x}) = \sum_m w_m \hat{f}_m(\mathbf{x}) \tag{5.1}$$

$\hat{f}_m(\mathbf{x})$ is the discriminant value of classifier $\hat{f}_m$ applied to sample $\mathbf{x}$.

Given M basic classifiers $\hat{f}_1$, $\hat{f}_2$, $\ldots$, $\hat{f}_M$, model averaging and model stacking search the best way to combine these basic classifiers. More specifically, the idea of model averaging and model stacking is to assign each classifier a weight $\hat{w}_m$, $m = 1, 2, \ldots, M$ to obtain a combined classifier, which minimizes squared error.

The weights for model averaging can be obtained from the training data by applying linear regression to the discriminant values of the basic classifiers,

$$\hat{w} = \arg\min_w \sum_i^N [y_i - \sum_m^M w_m \; \hat{f}_m(\mathbf{x}_i)]^2 \tag{5.2}$$

where N is the number of samples, and M is the number of classifiers to be combined. y_i is $+1$ or -1 depending on whether $\mathbf{x}_i$ is a true positive or a true negative. We obtain $\hat{f}_m(\mathbf{x}_i)$ by applying each classifier $\hat{f}_m$ induced from the whole training data back to the whole training data. $\hat{w}$ can be computed using linear regression on $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$. However, model averaging tends to put the most weight on the most complex classifier, since it fits the training data best.

Model stacking circumvents this problem using leave-one-out cross-validation in the training phase. The weights are set to minimize the average leave-one-out cross-validation error:

$$\hat{w} = \arg\min_w \sum_i^N [y_i - \sum_m^M w_m \; \hat{f}_m^{-i}(\mathbf{x}_i)]^2 \tag{5.3}$$

Instead of just training a classifier using all the samples in the training data and applying this classifier back to the whole training samples to obtain $\hat{f}_m(\mathbf{x}_i)$ $\forall i$, stacking removes one sample $\mathbf{x}_i$ at a time from the training data, and uses the remaining $N-1$ samples as training data to train a classifier $\hat{f}_m^{-i}$, and then computes $\hat{f}_m^{-i}(\mathbf{x}_i)$. $\hat{w}$ can be computed in the same way as model averaging.

Since we had more than ten thousand samples, leave-one-out cross-validation was computationally costly. Instead, we used 20-fold cross-validation.

The combined classifier is never worse than individual classifiers because

$$E_{\mathcal{P}}[Y - \sum_{m=1}^{M} \hat{w}_m \, \hat{f}_m(\mathbf{x})]^2 \leq E_{\mathcal{P}}[Y - \hat{f}_m(\mathbf{x})]^2 \quad \forall m \tag{5.4}$$

The combined classifier has smaller expected squared error than any single classifier.

The five basic classifiers were LR, LDA, QDA, NB and KNN, as described in Chapter 3. We also expanded these five basic classifiers by three pairwise operations: max, min, and products. For any two individual discriminant values of a sample $\mathbf{x}$, $\hat{f}_i(\mathbf{x})$ and $\hat{f}_j(\mathbf{x})$, the max operation adds 10 new discriminant values by using the maximum value of $\hat{f}_i(\mathbf{x})$ and $\hat{f}_j(\mathbf{x})$. Likewise, each of the min and products operations adds 10 new discriminant values by using the minimum value and products of $\hat{f}_i(\mathbf{x})$ and $\hat{f}_j(\mathbf{x})$, respectively. The products operation adds another 5 new discriminant values by self-production ($\hat{f}_i(\mathbf{x}) \cdot \hat{f}_i(\mathbf{x}), i = 1, 2, \ldots, 5$). Hence, the three operations add 35 new discriminant values to the original 5 discriminant values generated by the five basic classifiers.

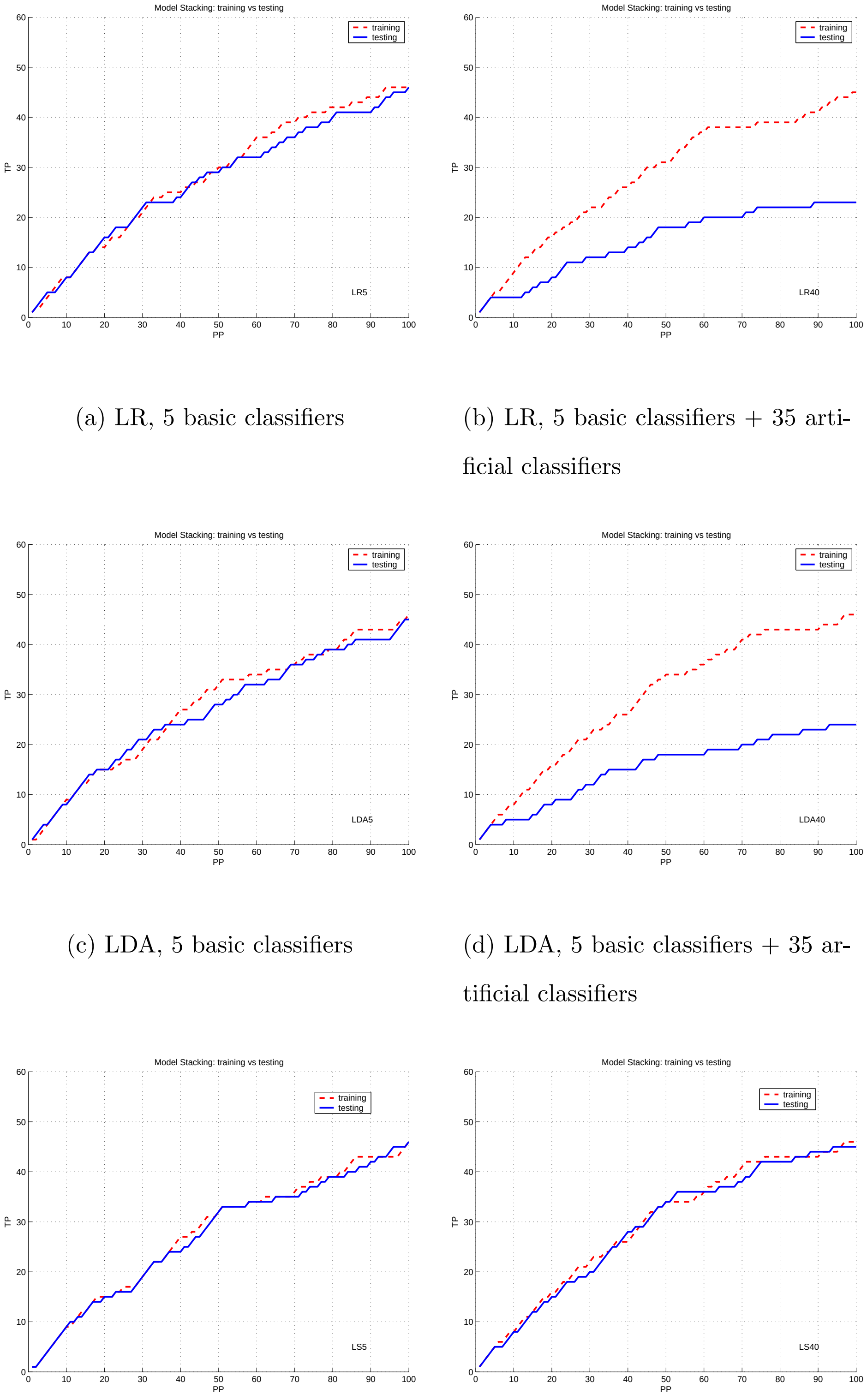
The above pairwise operations are non-linear. They combine the basic classifiers in a way that stacking cannot, and thus they effectively create new classifiers for the stacking process. In stacking, the discriminant values ($\hat{f}_i(\mathbf{x})$) output by the basic classifiers become feature values for the combining classifier. One can use any classification method, including LR, LDA and Least Squares (LS), as the combining classifier.

Figure 5.4 shows the comparison of combining the 5 basic classifiers and combining the 40 classifiers (including 35 expanded classifiers) using LR, LDA and LS. In Figures 5.4(a), 5.4(c) and 5.4(e), the training curves are close to the testing curves. However, in Figures 5.4(b) and 5.4(d), the testing curves are much lower than the training curves, which suggests overfitting . Interestingly, no overfitting occurred using LS (see

Figure 5.4(f)). LS is best at combining the 40 classifiers. The weights for the stacked classifier obtained by LS were much more constrained than the weights obtained by LDA and LR. In stacking, constrained weights achieve better classification accuracy than less constrained weights [10]. In addition, LR and LDA tended to more strictly fit to the training data than LS dose, and when raising thresholds, they made more wrong classification than LS did in the testing data.

5.4 Results

In Figure 5.4(e) and Figure 5.4(f), both testing curves are similar at the right end. However, at the left end (from PP = 0 to PP = 20), the testing curve in Figure 5.4(e) is higher than that in Figure 5.4(f), i.e., the classification precision using the combination of 5 basic classifiers is greater than that using the combination of 40 classifiers. Thus, in our subsequent tests, the combined classifiers were generated by linearly combining the 5 basic classifiers using least squares. Figure 5.5(a) shows the TP vs. PP plot for the combined classifier using model averaging. The pink curve (upper) and the dark curve (lower) are the training performance in the original training data and the training performance in the permuted data, respectively. The blue diagonal curve is the performance of a perfect classifier, and the red diagonal curve represents the expected performance of a theoretical random classifier. The upper bold curve is the mean performance of the combined classifier in the 100 randomizations in the original data; the lower bold curve is the mean performance of the combined classifer in the 100 permutations. The two green curve clouds show the performance distribution for all the randomizations and permutations, respectively. Figure 5.5(b) is the precision vs. PP plot for the same combined classifier (recall that the TP vs. PP plots always refer to training data, while the precision vs. PP plots refer to prediction data). Table 5.1 shows the predictions with precision higher than 0.5 made by the combined classifier using model averaging.



(a) LR, 5 basic classifiers

(b) LR, 5 basic classifiers + 35 artificial classifiers

(c) LDA, 5 basic classifiers

(d) LDA, 5 basic classifiers + 35 artificial classifiers

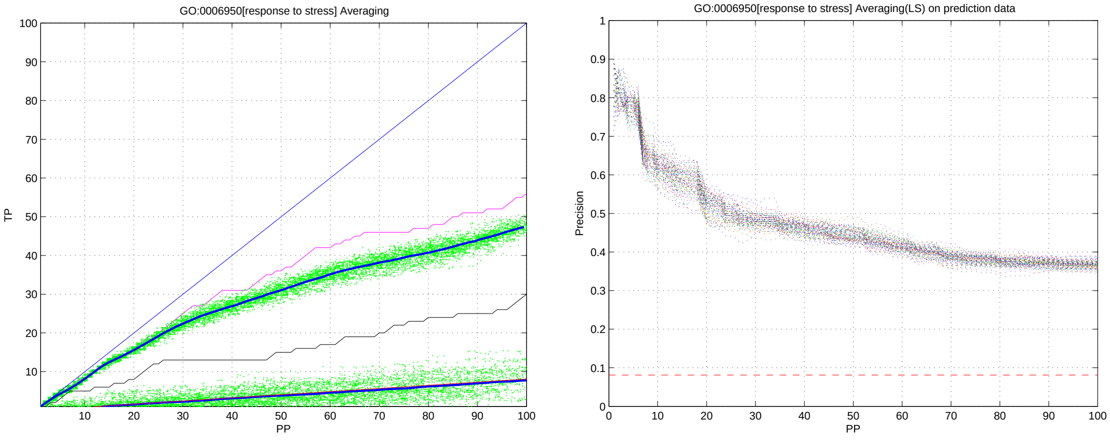
(e) LS, 5 basic classifiers

(f) LS, 5 basic classifiers + 35 artificial classifiers

Figure 5.4: Stacking used to combine classifiers using different regression methods

Table 5.1: The predictions with precision above 0.5 for *GO:0006950[response to stress]* made by the combined classifier using model averaging

gene	DV	M	SD
AT3G28290;AT3G28300	0.8764	0.8122	0.1232
AT3G22240	0.7618	0.8045	0.0426
AT3G50480	0.7412	0.7929	0.0420
AT4G39675	0.7044	0.7775	0.0530
AT1G16850	0.7021	0.7758	0.0521
AT3G44860	0.6784	0.7681	0.0578
AT3G14210	0.6192	0.6673	0.0381
AT5G44820	0.6060	0.6416	0.0331
AT1G80960	0.6052	0.6397	0.0337
AT5G38940;AT5G38930	0.5947	0.6202	0.0346
AT4G38080	0.5872	0.6083	0.0321
AT3G16440	0.5835	0.6008	0.0297
AT4G25790	0.5776	0.5884	0.0284
AT5G50670;AT5G50570	0.5749	0.5853	0.0289
AT4G02270	0.5742	0.5839	0.0283
AT1G70830;AT1G70850	0.5731	0.5818	0.0279
AT2G01530	0.5729	0.5820	0.0280
AT4G16960;AT4G16880;AT4G16940	0.5622	0.5515	0.0261
AT1G52070	0.5511	0.5248	0.0260
AT5G05500	0.5489	0.5201	0.0249
AT3G59930;AT5G33355	0.5474	0.5163	0.0244
AT3G11550	0.5463	0.5128	0.0249



(a) The TP vs. PP plot

(b) The precision vs. PP plot

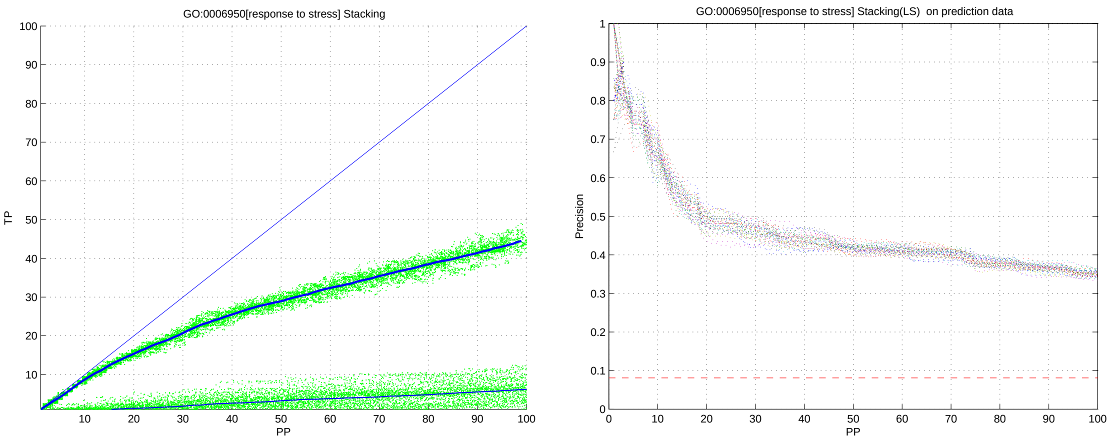
Figure 5.5: The performance evaluation of the combined classifier using model averaging based on least squares

Figure 5.6 shows the evaluation results of the combined classifier using model stacking. We can observe low variance of TP (left panel) and precision (right panel): The combined classifier was quite stable both in classification and prediction. In the main comparison based on the TP vs. PP plot, the combined classifier almost always has better generalization performance than individual classifiers, as shown in Figure 5.7. Table 5.2 shows the predictions with precision greater than 0.5 made by the combined classifier using stacking.

Figure 5.7 compares the mean performance of individual classifiers to the combined classifier using model stacking. It can be seen that the combined classifier combined the best aspects of the individual classifiers and therefore has the best performance in classification for almost all PP. Figure 5.8 compares the mean precision of the predictions made by all the classifiers in Figure 5.7.

Table 5.2: The predictions with precision above 0.5 for *GO:0006950[response to stress]* made by the combined classifier using model stacking

gene	DV	M	SD
AT3G28290;AT3G28300	0.9150	0.8755	0.2290
AT3G22240	0.8875	0.8484	0.1590
AT4G39675	0.8267	0.8421	0.1459
AT1G16850	0.7103	0.7926	0.0641
AT3G14210	0.6687	0.7551	0.0485
AT4G38080	0.6661	0.7542	0.0471
AT3G50480	0.6590	0.7455	0.0476
AT2G05510	0.6268	0.7038	0.0698
AT3G44860	0.6002	0.6753	0.0540
AT5G09530	0.5940	0.6639	0.0506
AT2G01520	0.5750	0.6333	0.0378
AT5G66985	0.5587	0.6037	0.0317
AT5G10040	0.5451	0.5761	0.0315
AT5G44820	0.5374	0.5612	0.0318
AT1G70830;AT1G70850	0.5291	0.5471	0.0339
AT5G50670;AT5G50570	0.5234	0.5348	0.0322
AT1G27030	0.5155	0.5194	0.0311
AT4G33720	0.5150	0.5199	0.0305



(a) TP vs. PP

(b) precision vs. PP

Figure 5.6: The performance evaluation of the combined classifier using model stacking

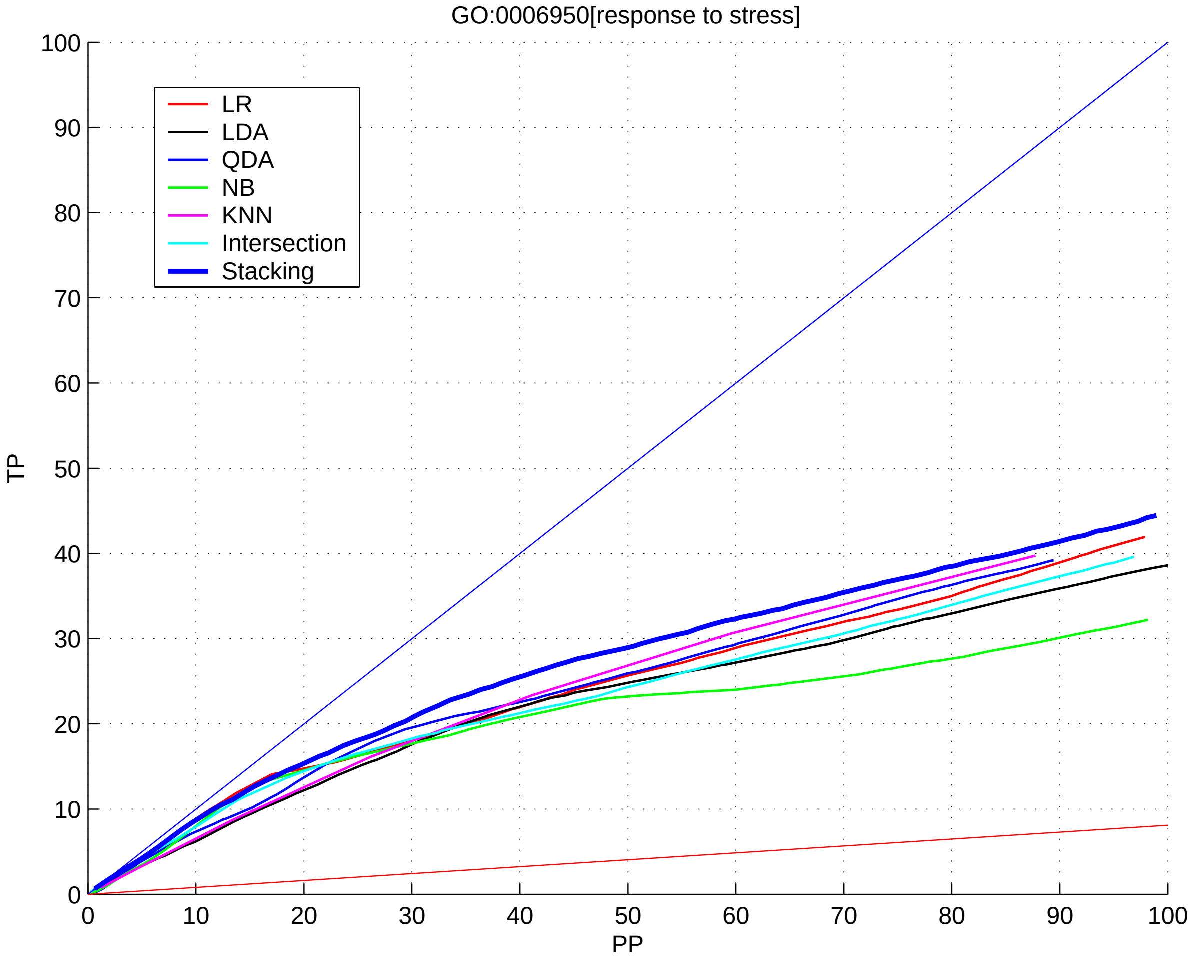


Figure 5.7: The mean TP vs. PP plot comparing the performance of the classifiers in classification

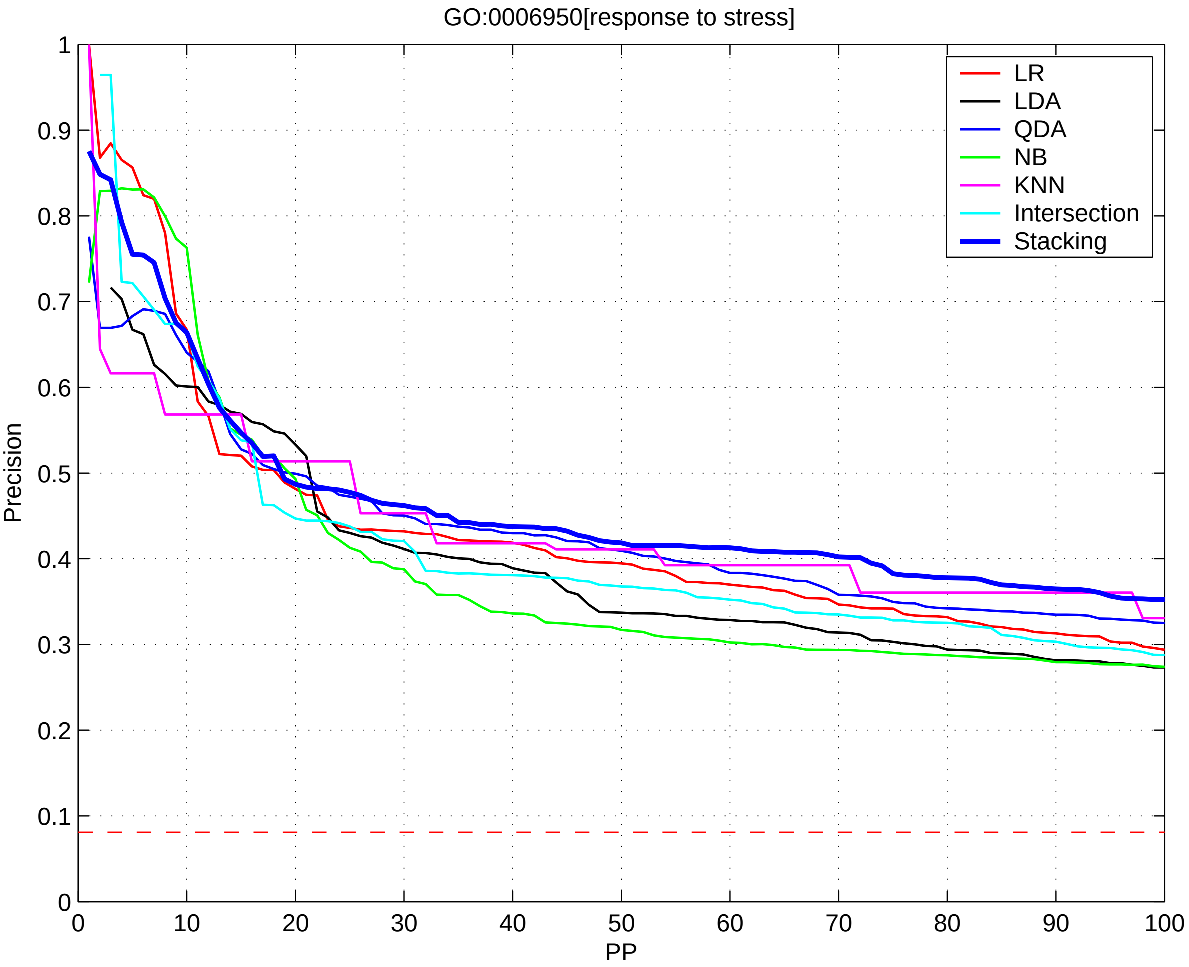


Figure 5.8: The mean precision vs. PP plot comparing the performance of the classifiers in prediction

Chapter 6

Conclusions and Discussion

In this study, we evaluated and compared five basic supervised learning methods (LR, LDA, QDA, NB and KNN) for gene function prediction in *A. thaliana* based solely on gene expression data. The major advantage of supervised methods over unsupervised methods is that by including the prior knowledge of class information, supervised methods can ignore uninformative features and select informative features that are useful to separate classes. 20-fold cross-validation, 100 randomizations, and 100 permutations were the key elements in evaluating as well as comparing these learning methods. 22 up-propagated stress GOBPs were studied. The results show some of these GOBPs are learnable based on gene expression data alone. For each of the GOBPs, we found that no method is uniformly better than the other methods in either classification or prediction.

We also investigated combining these basic classifiers using model averaging and stacking [10], as well as two simple combination strategies. The results show that the combined classifier using stacking outperforms all the basic classifiers for the top stress GOBP. However, establishing the optimal supervised classification method is not the goal of this work; it is possible other supervised methods such as SVMs, or Neural Networks can achieve comparable success.

Our results also show that the precision of predictions varied widely depending on the GOBPs we learned. For some GOBPs, the precision we could achieve was high; for others, the precision was low. In the latter case, we believe that the data were simply not informative: The genes involved in these GOBPs are not regulated at the level of mRNA transcription which can be detected by microarray chips; hence, these GOBPs are not learnable using microarray data alone. This observation suggests that developing algorithms which could effectively incorporate additional types of data in the learning process, for example, phenotype, sequence and homology data, might be useful. Also, we found that the precision of predictions made by a classifier could be affected by class homogeneity, class size, and variability of gene expression. Larger class size of positives generally results in higher precision (see Figure 4.1, Figure 4.2, and Figure 4.3). However, the opposite is not necessarily true (see Figure A.2(e) and Figure A.4(e)).

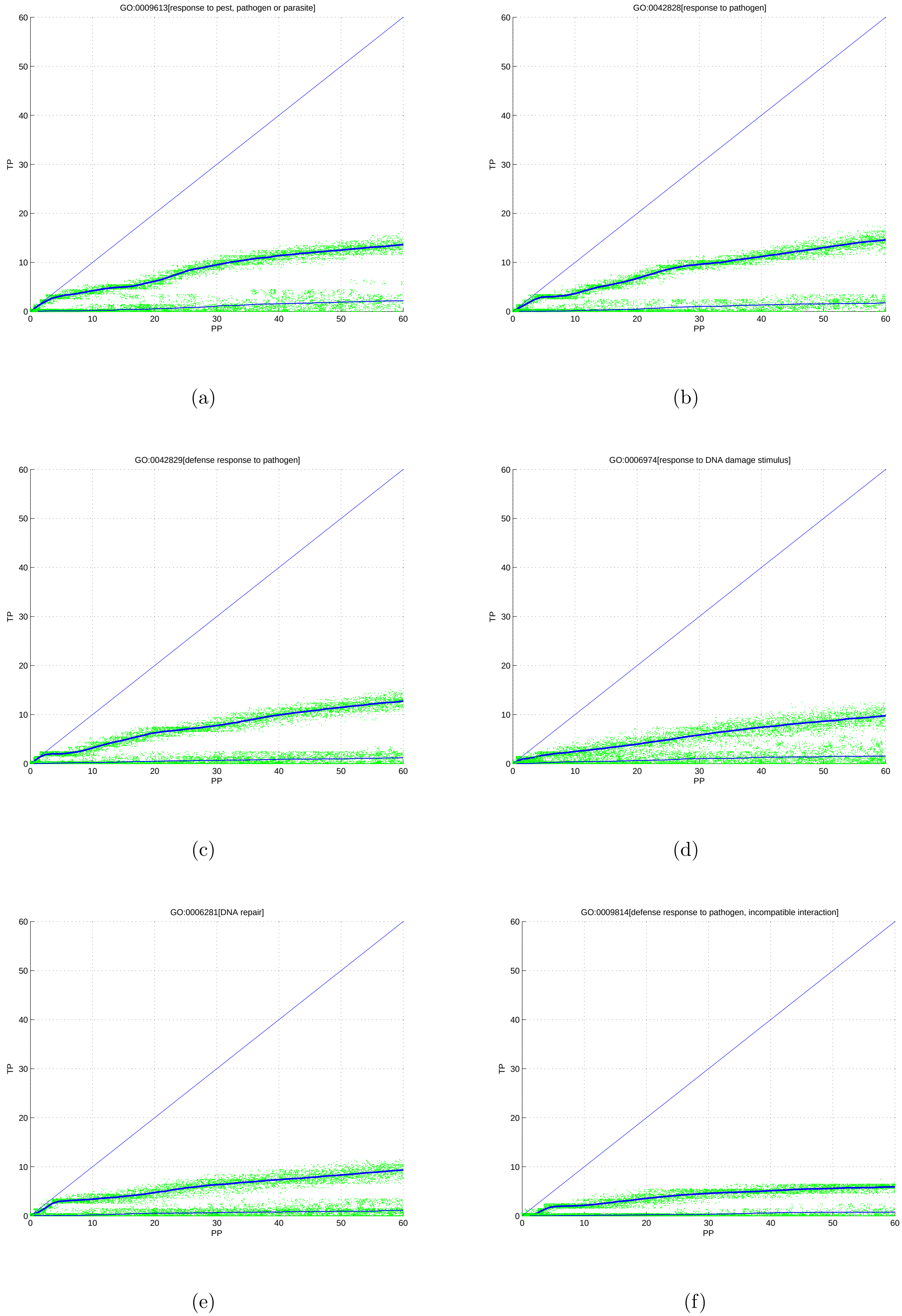
Using Electronic Northern analysis, our biological collaborators observed strong gene expression of many of our predictions. The predictions that had consistent up-regulation during stress were characterized for known protein motifs. These predictions are being validated by the biologists in the Department of Botany at University of Toronto. With more genes added to particular GOBPs of interest and more accurate gene functional annotations available, the advantage of supervised learning methods would become more evident.

This study suggests several avenues for future research. First, adding other types of data to gene expression data could be tried. Second, since many genes have multiple functions and these functions are organized in a hierarchy, we are considering methods that could take advantage of the correlation and structure information existing among GOBPs. The initial attempts in this direction were made by King et al. [15] and Midelfart et al. [23]

Appendix A

Plots

In this Chapter, we present the evaluation results for LR and LDA in classifying various stress GOBPs. The evaluation results for the other classifiers can be found in the supplementary data (`http://www.cs.toronto.edu/pub/lanhui/Supplementary_data/`). Figure A.1 and Figure A.2 show the TP vs. PP plots for LR; Figure A.3 and Figure A.4 show the TP vs. PP plots for LDA. It can be seen that *GO:0009409[response to cold]* (Figure A.2(e) and Figure A.4(e)) is a GOBP that is easily learned by both LR and LDA using the gene expression data alone. *GO:0009618[response to pathogenic bacteria]* (Figure A.2(d) and Figure A.4(d)) is just unlearnable: LDA and LR have almost the same performance as a random classifier.



(a)

(b)

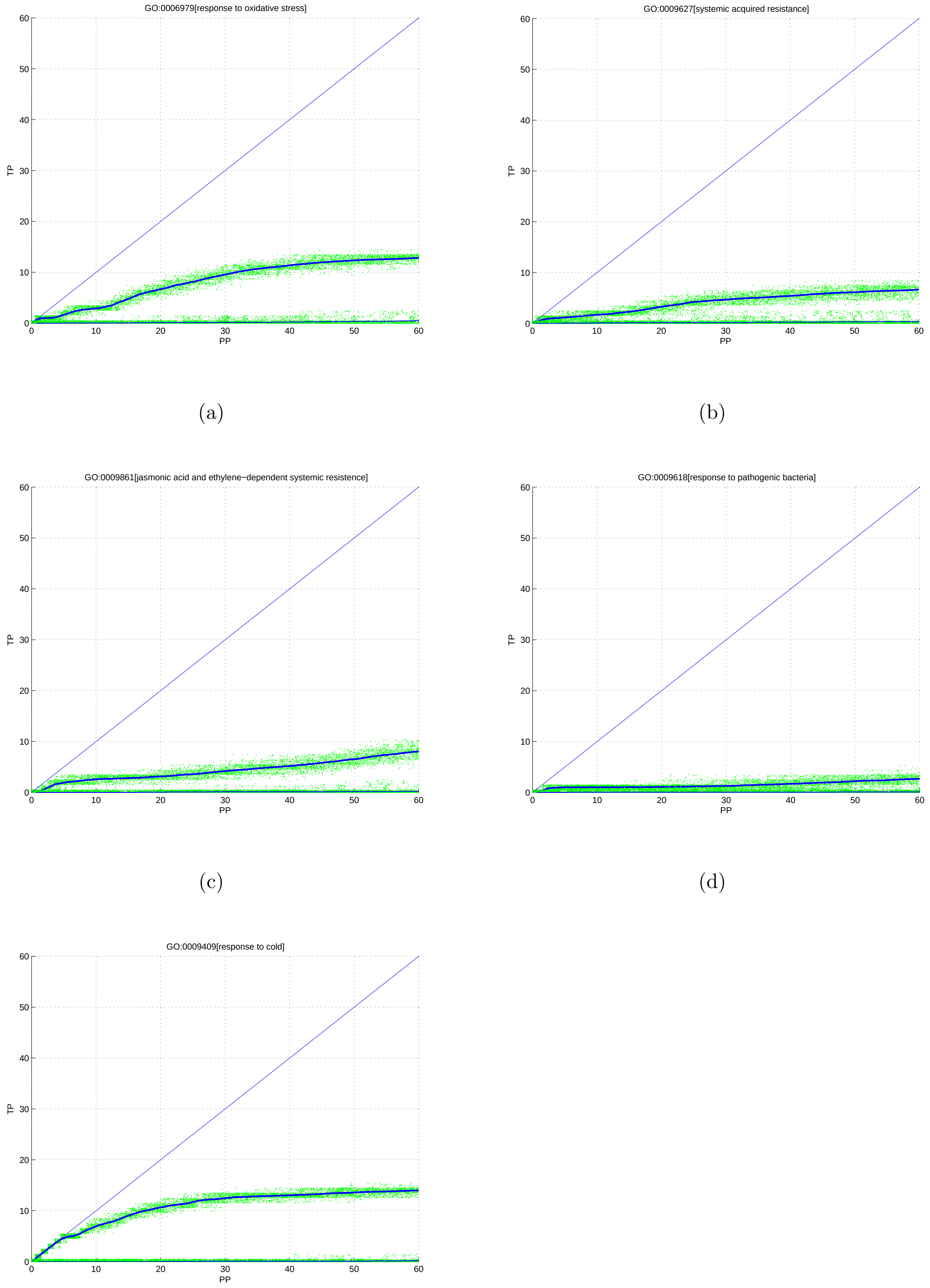
(c)

(d)

(e)

(f)

Figure A.1: The TP vs. PP plots of LR for various stress GOBPs



(a)

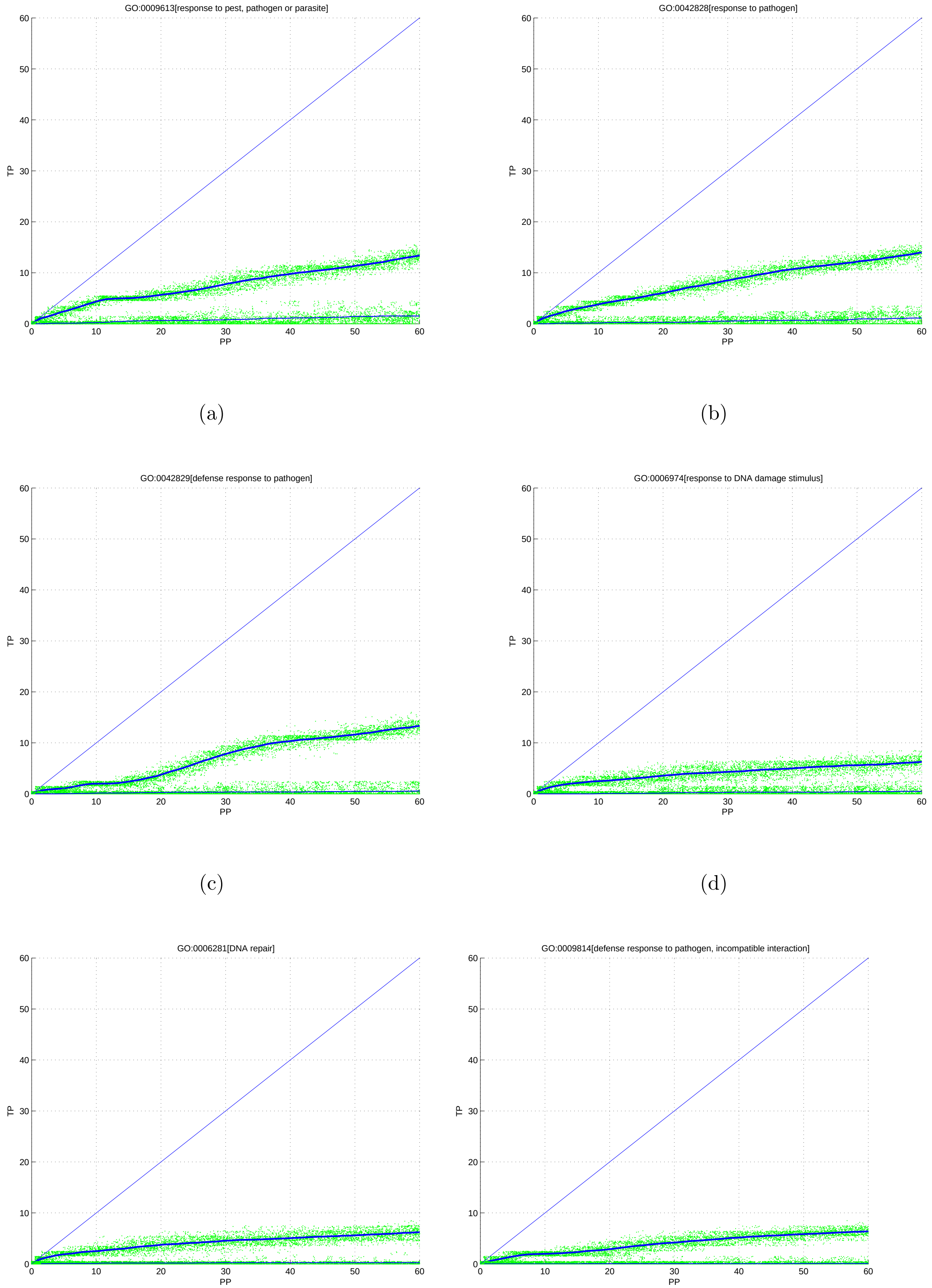
(b)

(c)

(d)

(e)

Figure A.2: TP vs. PP plots of LR for various stress GOBPs (continued)



(a)

(b)

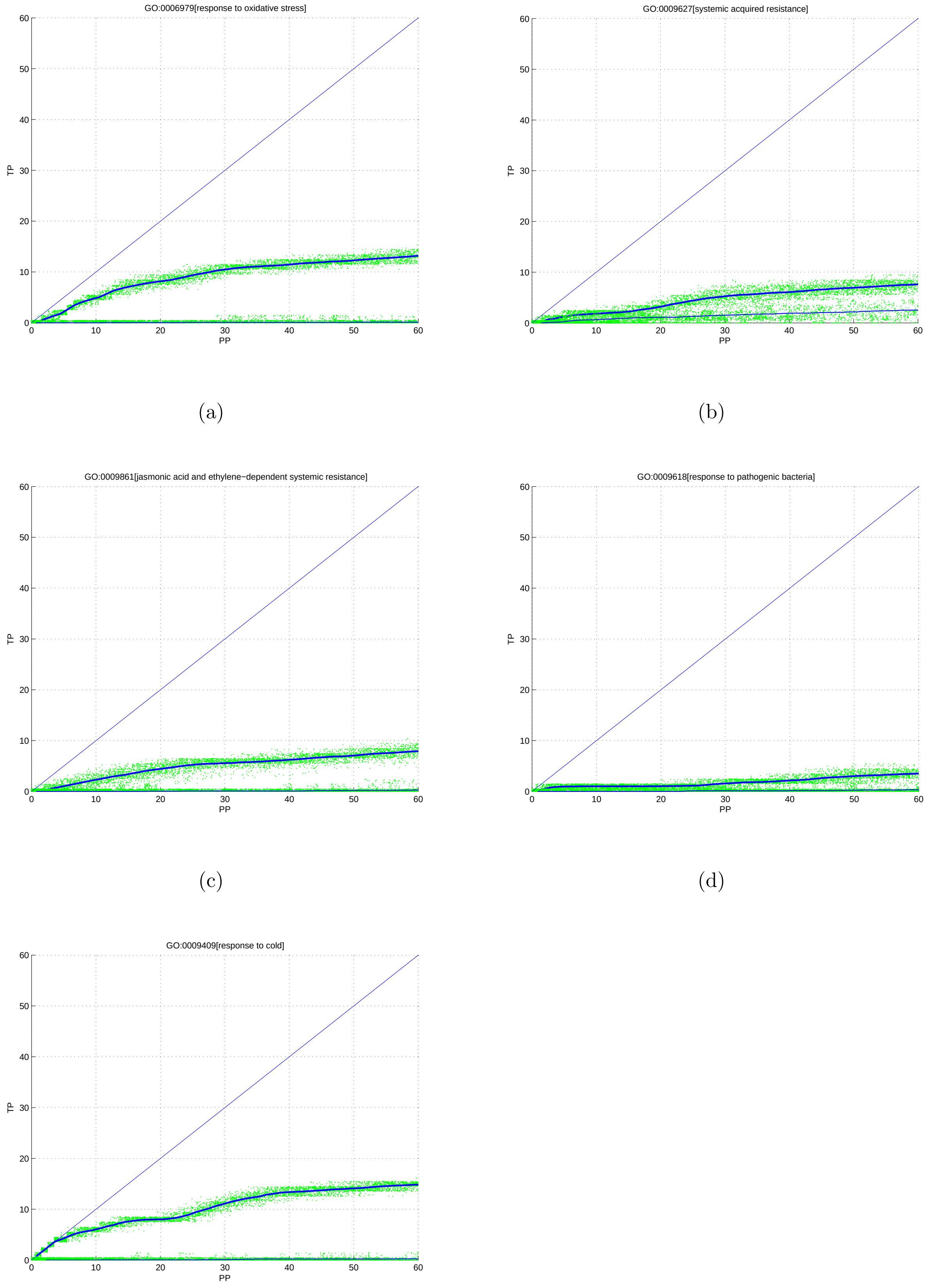
(c)

(d)

(e)

(f)

Figure A.3: The TP vs. PP plots of LDA for various stress GOBPs



(a)

(b)

(c)

(d)

(e)

Figure A.4: The TP vs. PP plots of LDA for various stress GOBPs (continued)

Appendix B

Tables

This chapter shows the prediction results for various stress GOBPs using LR, LDA, QDA, NB and KNN. The top 100 predictions made by each method for the top stress GOBP (*GO:0006950[response to stress]*) are shown. In addition, for LR and LDA, the predictions having rounded precision greater than or equal to 0.5 for various stress GOBPs are shown. The predictions for the other stress GOBPs using the other classifiers (QDA, NB and KNN) can be found in the supplementary data. The columns in the tables, from left to right, are gene name, discriminant value, mean precision and standard deviation of the precision.

B.1 Logistic Regression

Table B.1: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G22240	0.8007	1.0000	0.3313
AT3G28290;AT3G28300	0.7357	0.8678	0.1566
AT4G39675	0.6691	0.8847	0.0936
AT3G14210	0.6229	0.8653	0.0698

Table B.1: *GO:0006950[response to stress]*

gene	DV	M	SD
AT1G16850	0.6052	0.8564	0.0478
AT4G38080	0.5697	0.8241	0.0242
AT5G66985	0.5685	0.8199	0.0246
AT2G05510	0.5540	0.7799	0.0320
AT5G10040	0.5252	0.6860	0.0378
AT5G09530	0.5196	0.6671	0.0397
AT5G07010	0.4812	0.5833	0.0352
AT2G01520	0.4723	0.5663	0.0364
AT5G62520	0.4406	0.5222	0.0301
AT4G33720	0.4400	0.5208	0.0301
AT2G41730	0.4395	0.5203	0.0309
AT1G27030	0.4294	0.5076	0.0299
AT4G21840;AT4G21830	0.4266	0.5036	0.0292
AT5G18470	0.4266	0.5033	0.0296
AT3G50480	0.4156	0.4890	0.0272
AT2G23540	0.4088	0.4815	0.0250
AT2G33850	0.4039	0.4745	0.0223
AT2G36220	0.4028	0.4739	0.0228
AT5G10695	0.3786	0.4455	0.0198
AT2G32160	0.3678	0.4379	0.0198
AT2G26400	0.3620	0.4359	0.0215
AT3G44860	0.3595	0.4339	0.0219
AT5G22530	0.3592	0.4341	0.0215
AT5G44820	0.3551	0.4332	0.0227

Table B.1: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G28220	0.3533	0.4326	0.0224
AT5G25260;AT5G25250	0.3505	0.4319	0.0211
AT3G28210	0.3463	0.4301	0.0201
AT1G76960	0.3444	0.4289	0.0211
AT5G09480	0.3430	0.4286	0.0195
AT1G19020	0.3371	0.4253	0.0176
AT5G50670;AT5G50570	0.3284	0.4217	0.0180
AT1G31680	0.3281	0.4213	0.0184
AT5G03350	0.3269	0.4207	0.0195
AT3G29970	0.3256	0.4201	0.0189
AT2G32210	0.3255	0.4200	0.0188
AT5G42860	0.3242	0.4187	0.0186
AT4G37710	0.3218	0.4163	0.0176
AT5G45500	0.3181	0.4127	0.0182
AT3G26470	0.3139	0.4098	0.0163
AT3G59930;AT5G33355	0.3054	0.4020	0.0142
AT4G37070;AT4G37060	0.3042	0.4005	0.0135
AT1G61340	0.3019	0.3975	0.0131
AT1G64370	0.3008	0.3963	0.0126
AT1G67870	0.3006	0.3958	0.0127
AT4G23670	0.3002	0.3955	0.0123
AT4G33560	0.2997	0.3946	0.0121
AT1G80240	0.2989	0.3932	0.0121
AT2G19970	0.2964	0.3885	0.0121

Table B.1: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G26450	0.2955	0.3871	0.0120
AT1G29670	0.2943	0.3854	0.0118
AT1G70830;AT1G70850	0.2906	0.3799	0.0109
AT5G64510	0.2869	0.3728	0.0114
AT1G52070	0.2869	0.3727	0.0113
AT3G06390	0.2862	0.3716	0.0118
AT4G16960;AT4G16880;AT4G16940	0.2859	0.3713	0.0118
AT5G19250	0.2848	0.3698	0.0117
AT2G42610	0.2839	0.3685	0.0118
AT4G24110	0.2830	0.3672	0.0118
AT2G19990	0.2823	0.3663	0.0115
AT1G19960	0.2804	0.3634	0.0113
AT1G10990	0.2799	0.3626	0.0116
AT3G09350	0.2762	0.3581	0.0114
AT1G14870	0.2737	0.3541	0.0111
AT1G07500	0.2734	0.3538	0.0110
AT5G23840	0.2728	0.3533	0.0111
257874_at	0.2678	0.3465	0.0115
AT5G63560	0.2673	0.3456	0.0114
AT1G09950	0.2650	0.3431	0.0107
AT5G64870	0.2638	0.3421	0.0102
AT1G66500;AT5G43620	0.2636	0.3420	0.0103
AT4G00080	0.2631	0.3418	0.0096
AT5G60950	0.2579	0.3355	0.0097

Table B.1: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G47250	0.2565	0.3337	0.0095
AT3G16440	0.2560	0.3331	0.0095
AT4G21850	0.2559	0.3329	0.0095
AT2G01530	0.2548	0.3318	0.0090
AT1G23960	0.2504	0.3272	0.0081
AT1G19180	0.2498	0.3267	0.0082
AT3G16800	0.2480	0.3243	0.0076
AT4G01870	0.2453	0.3210	0.0080
248621_at	0.2447	0.3203	0.0081
AT1G11850	0.2430	0.3182	0.0078
AT5G61660	0.2424	0.3174	0.0077
AT1G49650	0.2405	0.3147	0.0076
AT3G52870	0.2398	0.3137	0.0078
AT4G18280	0.2393	0.3129	0.0076
AT1G21680	0.2384	0.3112	0.0077
AT1G11210	0.2378	0.3103	0.0074
AT3G03520	0.2374	0.3096	0.0078
AT4G39190	0.2373	0.3094	0.0078
AT5G13200	0.2336	0.3036	0.0069
AT4G25790	0.2328	0.3020	0.0069
AT5G37990	0.2327	0.3020	0.0069
AT5G35940	0.2301	0.2976	0.0066
AT3G04320	0.2288	0.2958	0.0065
AT2G27080;AT2G27090	0.2277	0.2938	0.0064

Table B.1: *GO:0006950[response to stress]*

gene	DV	M	SD

Table B.2: *GO:0009613[response to pest, pathogen or parasite]*

gene	DV	M	SD
AT3G14210	0.607	NaN	NaN
AT3G28290;AT3G28300	0.5800	NaN	NaN
AT3G50480	0.5469	1.0000	1.9400
AT2G42610	0.4525	0.7817	0.1475
AT1G76960	0.4205	0.6240	0.0887
AT5G03350	0.4145	0.5898	0.0880
AT3G44860	0.4012	0.4861	0.0754
AT1G19960	0.3980	0.4657	0.0739

Table B.3: *GO:0042828[response to pathogen].* $\sum_{i=1}^{100} \mathrm{TP}_i = 0$ caused the estimated mean precision for AT3G28290;AT3G28300 and AT3G50480 to be 0. See Section 3.3 in Chapter 3 for more details.

gene	DV	M	SD
AT3G14210	0.6122	NaN	NaN
AT3G28290;AT3G28300	0.5716	0	0
AT3G50480	0.5677	0	0
AT2G42610	0.4203	0.5928	0.0342
AT1G76960	0.3991	0.5000	0.0167
AT5G03350	0.3948	0.4709	0.0157
AT1G19960	0.3939	0.4644	0.0220

Table B.4: *GO:0042829[defense response to pathogen]*

gene	DV	M	SD
AT1G76960	0.8554	0.9545	2.0729
AT3G50480	0.8119	0.8790	0.4026
AT3G14210	0.7376	0.6387	0.0454

Table B.5: *GO:0006974[response to DNA damage stimulus]*

gene	DV	M	SD
AT4G19240	0.7508	0.9286	2.4143
AT2G06005	0.6477	0.7544	0.3303
AT3G09040	0.5862	0.4725	0.1376

Table B.6: *GO:0006281[DNA repair]*

gene	DV	M	SD
AT2G06005	0.7162	NaN	NaN
AT2G23470	0.5715	0.6183	0.4415
AT4G19240	0.4540	0.6819	0.1031
AT3G19670	0.4198	0.6095	0.0741
AT3G20810	0.4164	0.5992	0.0638
AT1G58025	0.4072	0.5725	0.0542

Table B.7: *GO:0006979[response to oxidative stress]*

gene	DV	M	SD
AT2G01520	0.8565	0.8000	0

Table B.8: *GO:0006972[hyperosmotic response]*

gene	DV	M	SD
AT5G11420	0.6809	0.5156	0.7384

Table B.9: *GO:0009409[response to cold]*

gene	DV	M	SD
AT1G16850	0.9474	1.0000	0.1913
AT3G28290;AT3G28300	0.8881	0.9849	0.2452
AT3G22240	0.3311	0.6181	0.0551
AT5G45500	0.3287	0.6154	0.0555

B.2 Linear Discriminant Analysis

Table B.10: *GO:0006950[response to stress]*

gene	DV	M	SD
AT4G39675	0.5230	NaN	NaN
AT3G22240	0.5138	NaN	NaN
AT3G28290;AT3G28300	0.4121	0.7164	0.3034
AT2G05510	0.4071	0.7029	0.2725
AT5G66985	0.3872	0.6673	0.1349
AT4G38080	0.3852	0.6619	0.1346
AT5G09530	0.3628	0.6263	0.1246
AT2G36220	0.3256	0.6156	0.0682
AT2G01520	0.3085	0.6021	0.0615
AT3G14210	0.3062	0.6009	0.0587
AT5G10040	0.3055	0.6002	0.0575
AT2G23540	0.2898	0.5835	0.0403
AT1G76960	0.2860	0.5794	0.0366
AT5G03350	0.2796	0.5717	0.0307
AT1G16850	0.2777	0.5690	0.0290
AT5G62520	0.2723	0.5596	0.0260
AT5G09480	0.2707	0.5569	0.0280
AT1G27030	0.2668	0.5487	0.0236
AT2G33850	0.2659	0.5461	0.0228
AT1G66500;AT5G43620	0.2605	0.5330	0.0234
AT2G26400	0.2559	0.5198	0.0234
AT1G19020	0.2404	0.4551	0.0219

Table B.10: *GO:0006950[response to stress]*

gene	DV	M	SD
AT5G45500	0.2389	0.4481	0.0200
AT1G29670	0.2340	0.4332	0.0219
AT3G29970	0.2330	0.4300	0.0217
AT1G70830;AT1G70850	0.2318	0.4264	0.0213
AT5G10695	0.2309	0.4246	0.0220
AT2G19970	0.2265	0.4188	0.0199
AT4G33560	0.2252	0.4154	0.0192
AT2G41730	0.2235	0.4114	0.0179
AT2G32210	0.2210	0.4070	0.0180
AT2G42610	0.2205	0.4066	0.0182
AT1G14870	0.2199	0.4049	0.0179
AT1G61340	0.2187	0.4021	0.0171
AT4G21840;AT4G21830	0.2178	0.4004	0.0168
AT5G18470	0.2176	0.3998	0.0172
AT5G25260;AT5G25250	0.2161	0.3959	0.0175
AT3G26450	0.2151	0.3940	0.0159
AT2G05380	0.2150	0.3940	0.0159
AT1G07500	0.2129	0.3889	0.0152
AT4G24110	0.2120	0.3864	0.0152
AT4G33720	0.2110	0.3838	0.0143
AT5G07010	0.2109	0.3833	0.0144
AT4G23670	0.2079	0.3723	0.0140
AT1G19960	0.2051	0.3618	0.0135
AT5G60950	0.2042	0.3585	0.0136

Table B.10: *GO:0006950[response to stress]*

gene	DV	M	SD
AT1G67870	0.1988	0.3462	0.0145
AT5G50670;AT5G50570	0.1932	0.3378	0.0143
AT1G80240	0.1927	0.3376	0.0142
AT3G28210	0.1924	0.3370	0.0146
AT4G37070;AT4G37060	0.1918	0.3363	0.0151
AT3G44860	0.1918	0.3363	0.0151
AT3G50480	0.1911	0.3361	0.0150
AT2G32160	0.1908	0.3355	0.0148
AT1G09310	0.1876	0.3333	0.0149
AT5G42860	0.1873	0.3333	0.0146
AT5G63180	0.1828	0.3311	0.0125
AT4G01870	0.1817	0.3300	0.0114
AT3G09350	0.1807	0.3288	0.0114
AT1G52070	0.1805	0.3286	0.0111
AT1G09950	0.1795	0.3274	0.0108
AT2G41380	0.1795	0.3274	0.0108
AT3G02840;AT3G02850	0.1788	0.3260	0.0101
AT1G11210	0.1788	0.3259	0.0101
AT5G22530	0.1787	0.3257	0.0101
AT5G37990	0.1772	0.3229	0.0094
AT5G45110	0.1759	0.3195	0.0095
AT1G64370	0.1751	0.3179	0.0090
AT5G53830	0.1739	0.3145	0.0081
AT1G13340	0.1738	0.3139	0.0083

Table B.10: *GO:0006950[response to stress]*

gene	DV	M	SD
AT2G47950	0.1735	0.3134	0.0085
AT1G65690	0.1726	0.3113	0.0084
AT3G03520	0.1701	0.3048	0.0072
AT5G19250	0.1701	0.3047	0.0073
AT3G06390	0.1695	0.3029	0.0072
AT5G64510	0.1690	0.3015	0.0072
AT1G70840	0.1686	0.3002	0.0073
AT5G61660	0.1680	0.2983	0.0075
AT3G59930;AT5G33355	0.1678	0.2977	0.0071
AT2G01530	0.1666	0.2940	0.0070
AT3G57380	0.1665	0.2936	0.0070
AT5G64170	0.1664	0.2934	0.0069
AT3G52870	0.1662	0.2929	0.0070
AT4G22530	0.1652	0.2901	0.0074
AT2G45760	0.1650	0.2896	0.0074
AT1G21680	0.1648	0.2890	0.0073
AT5G23840	0.1645	0.2884	0.0074
AT4G33050	0.1634	0.2855	0.0066
AT1G13470	0.1627	0.2832	0.0069
AT1G17830	0.1620	0.2815	0.0074
AT5G64870	0.1619	0.2814	0.0074
AT1G23960	0.1619	0.2812	0.0074
AT5G44820	0.1616	0.2805	0.0075
AT2G44010	0.1615	0.2803	0.0075

Table B.10: *GO:0006950[response to stress]*

gene	DV	M	SD
AT2G19990	0.1608	0.2781	0.0070
AT4G16960;AT4G16880;AT4G16940	0.1607	0.2781	0.0068
AT2G27080;AT2G27090	0.1600	0.2762	0.0065
AT2G46790;AT2G46670	0.1596	0.2751	0.0063
AT5G16030	0.1589	0.2731	0.0061
AT1G18980	0.1589	0.2731	0.0063

Table B.11: *GO:0009613[response to pest, pathogen or parasite]*

gene	DV	M	SD
AT3G50480	0.3509	1.0000	1.4321
AT3G15240	0.3509	1.0000	1.4321
AT3G28290;AT3G28300	0.3034	0.8115	0.2470
AT1G76960	0.2885	0.5955	0.2640
AT3G44860	0.2718	0.4937	0.1476

Table B.12: *GO:0042828[response to pathogen]*

gene	DV	M	SD
AT3G50480	0.3666	1.0000	0.0000
AT3G14210	0.3555	1.0000	3.1958
AT3G28290;AT3G28300	0.2991	0.6937	0.3190
AT1G76960	0.2861	0.5956	0.2653
AT3G44860	0.2602	0.4505	0.1000

Table B.13: *GO:0042829[defense response to pathogen]*

gene	DV	M	SD
AT1G76960	0.9110	NaN	NaN
AT3G50480	0.8323	NaN	NaN
AT5G03350	0.7370	0.4865	0.4513

Table B.14: *GO:0006974[response to DNA damage stimulus]*

gene	DV	M	SD
AT3G09040	0.2288	0.6138	0.2923
AT5G20460	0.2147	0.5158	0.1932
255186_at	0.2144	0.5112	0.1914

Table B.15: *GO:0006281[DNA repair]*

gene	DV	M	SD
AT3G09040	0.2387	0.6829	0.3956
AT5G20460	0.2189	0.5872	0.2266
255186_at	0.2097	0.4823	0.1616
AT2G44420	0.2078	0.4704	0.1565

Table B.16: *GO:0006979[response to oxidative stress]*

gene	DV	M	SD
AT4G02270	0.9739	0.5262	0.0753
AT2G19970	0.9754	0.5233	0.0791
AT2G23540	0.9644	0.5090	0.0781
AT2G01520	0.9625	0.5082	0.0768
AT4G39675	0.9351	0.4778	0.0584
AT4G38080	0.9313	0.4829	0.0581
AT4G33720	0.8967	0.4736	0.0549
AT4G00680	0.8916	0.4619	0.0511

Table B.17: *GO:0009409[response to cold]*

gene	DV	M	SD
AT1G16850	0.9978	1.0000	0.2479
AT3G28290;AT3G28300	0.9942	1.0000	0.1819
AT1G11210	0.8590	0.6624	0.0632
AT5G42900	0.7491	0.5571	0.0469
AT3G22240	0.7428	0.5509	0.0475
AT4G16146	0.7356	0.5457	0.0465
AT3G14210	0.6719	0.5191	0.0401
AT5G50360	0.5625	0.4513	0.0167

B.3 Quadratic Discriminant Analysis

Table B.18: *GO:0006950[response to stress]*

gene	DV	M	SD
AT4G39675	0.5798	0.7757	0.2149
AT3G28290;AT3G28300	0.5693	0.6694	0.0539
AT1G52070	0.5688	0.6693	0.0509
AT4G00680	0.5654	0.6716	0.0385
AT3G50480	0.5637	0.6831	0.0588
AT4G33560	0.5625	0.6911	0.0669
AT5G09530	0.5621	0.6891	0.0643
AT4G38080	0.5615	0.6856	0.0601
AT4G02270	0.5599	0.6611	0.0534
AT2G23540	0.5591	0.6404	0.0488
AT5G05500	0.5586	0.6301	0.0415
AT4G37070;AT4G37060	0.5581	0.6186	0.0381
AT5G03350	0.5571	0.5837	0.0326
AT3G47250	0.5557	0.5455	0.0336
AT2G19970	0.5548	0.5277	0.0333
255181_at	0.5545	0.5224	0.0318
AT3G29970	0.5539	0.5093	0.0275
AT1G18980	0.5535	0.5046	0.0270
AT1G16850	0.5532	0.5008	0.0248
AT1G70830;AT1G70850	0.5530	0.4992	0.0245
AT2G05510	0.5527	0.4960	0.0248
AT1G58025	0.5518	0.4852	0.0245

Table B.18: *GO:0006950[response to stress]*

gene	DV	M	SD
AT5G60950	0.5517	0.4833	0.0242
AT1G33055	0.5509	0.4745	0.0250
AT3G14210	0.5507	0.4723	0.0230
AT2G42610	0.5505	0.4700	0.0244
AT3G48640	0.5502	0.4673	0.0241
AT5G23830	0.5484	0.4531	0.0220
AT1G80240	0.5481	0.4508	0.0203
AT5G26280;AT5G26260	0.5481	0.4505	0.0203
AT5G38940;AT5G38930	0.5477	0.4472	0.0183
AT1G23960	0.5471	0.4405	0.0194
AT1G76960	0.5471	0.4405	0.0191
AT2G01520	0.5470	0.4394	0.0190
AT1G13470	0.5467	0.4374	0.0185
AT3G16450	0.5466	0.4364	0.0179
AT2G39310	0.5463	0.4339	0.0181
AT5G15360	0.5463	0.4339	0.0181
AT5G10040	0.5459	0.4306	0.0174
AT2G14560	0.5458	0.4298	0.0180
AT5G37990	0.5457	0.4298	0.0178
AT2G40330	0.5455	0.4273	0.0181
AT2G32160	0.5455	0.4274	0.0179
AT1G66690	0.5450	0.4248	0.0174
AT2G33850	0.5445	0.4207	0.0176
AT2G36220	0.5445	0.4204	0.0172

Table B.18: *GO:0006950[response to stress]*

gene	DV	M	SD
AT5G26300;AT5G26280;AT5G26260	0.5444	0.4193	0.0170
252346_at	0.5437	0.4123	0.0141
AT4G25790	0.5436	0.4110	0.0135
AT2G36100	0.5435	0.4093	0.0140
AT2G01530	0.5431	0.4068	0.0134
AT3G10930	0.5428	0.4033	0.0141
AT3G18170	0.5427	0.4025	0.0140
AT5G09480	0.5425	0.4002	0.0150
AT1G01750	0.5423	0.3974	0.0157
AT1G27030	0.5420	0.3959	0.0150
258246_s_at	0.5419	0.3945	0.0141
AT1G61340	0.5418	0.3934	0.0143
AT1G70890	0.5413	0.3870	0.0141
AT1G67865	0.5410	0.3835	0.0141
AT4G24110	0.5410	0.3834	0.0136
AT3G44860	0.5409	0.3825	0.0131
AT2G44010	0.5408	0.3808	0.0127
AT1G11210	0.5406	0.3789	0.0119
AT3G12540	0.5404	0.3768	0.0112
AT5G50670;AT5G50570	0.5402	0.3743	0.0107
AT1G23130	0.5401	0.3739	0.0105
AT3G14440	0.5398	0.3695	0.0102
AT1G66500;AT5G43620	0.5394	0.3648	0.0100
AT1G80960	0.5388	0.3579	0.0095

Table B.18: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G11550	0.5388	0.3577	0.0096
AT3G16440	0.5386	0.3570	0.0099
AT1G58270	0.5385	0.3560	0.0097
AT5G25260;AT5G25250	0.5383	0.3538	0.0101
AT4G16960;AT4G16880;AT4G16940	0.5379	0.3497	0.0098
AT1G25097;AT1G24822;AT1G24996;AT1G25170	0.5377	0.3483	0.0094
AT3G20370	0.5377	0.3479	0.0091
AT3G22240	0.5374	0.3442	0.0090
AT1G19960	0.5372	0.3427	0.0092
AT2G15560	0.5371	0.3421	0.0089
AT3G45160	0.5371	0.3418	0.0087
AT4G22510	0.5370	0.3409	0.0087
AT5G61660	0.5370	0.3403	0.0085
AT3G47480	0.5369	0.3395	0.0087
AT1G50060	0.5368	0.3387	0.0083
AT4G33730	0.5368	0.3386	0.0083
AT2G37750	0.5366	0.3370	0.0084
AT1G67870	0.5366	0.3368	0.0082
AT3G04320	0.5364	0.3356	0.0081
AT1G17830	0.5363	0.3348	0.0081
AT5G44580	0.5363	0.3348	0.0081
AT1G17380	0.5363	0.3346	0.0081
AT1G35140	0.5362	0.3336	0.0080
AT5G44820	0.5357	0.3301	0.0086

Table B.18: *GO:0006950[response to stress]*

gene	DV	M	SD
AT4G30140	0.5357	0.3300	0.0090
AT2G19990	0.5355	0.3291	0.0086
AT4G15390	0.5355	0.3283	0.0085
AT5G61160	0.5354	0.3279	0.0085
AT3G16430;AT3G16420	0.5351	0.3256	0.0086
AT4G16146	0.5350	0.3251	0.0087

B.4 Naive Bayes

Table B.19: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G16430;AT3G16420	0.5309	0.7220	0.1184
AT4G38080	0.5284	0.8288	0.0949
AT4G23680	0.5284	0.8294	0.0932
AT2G33850	0.5280	0.8321	0.0725
AT4G39675	0.5278	0.8308	0.0757
AT2G05510	0.5277	0.8310	0.0738
AT2G42610	0.5273	0.8215	0.0646
AT5G03350	0.5269	0.7998	0.0582
AT2G01520	0.5267	0.7734	0.0542
AT3G22240	0.5266	0.7627	0.0535
AT5G09530	0.5259	0.6613	0.0391

Table B.19: *GO:0006950[response to stress]*

gene	DV	M	SD
AT1G76960	0.5254	0.6073	0.0314
AT2G19970	0.5252	0.5884	0.0290
AT1G13470	0.5248	0.5519	0.0311
AT5G05060	0.5246	0.5447	0.0322
AT5G42900	0.5246	0.5389	0.0335
AT1G23960	0.5243	0.5202	0.0292
AT3G30720	0.5243	0.5203	0.0286
AT5G62280	0.5241	0.5056	0.0254
AT1G66690	0.5240	0.4931	0.0249
AT4G33720	0.5235	0.4570	0.0149
AT5G26300;AT5G26280;AT5G26260	0.5235	0.4506	0.0134
AT1G24020	0.5232	0.4301	0.0124
AT5G61160	0.5231	0.4221	0.0122
AT1G73260	0.5230	0.4127	0.0103
AT2G23540	0.5230	0.4082	0.0095
AT4G15390	0.5228	0.3961	0.0091
AT5G66985	0.5228	0.3954	0.0091
AT3G28290;AT3G28300	0.5226	0.3889	0.0112
AT3G14210	0.5226	0.3878	0.0114
AT2G25625	0.5224	0.3736	0.0135
AT1G14120	0.5223	0.3705	0.0137
AT1G23130	0.5220	0.3582	0.0141
AT4G15620	0.5220	0.3577	0.0144
AT4G27860	0.5220	0.3577	0.0144

Table B.19: *GO:0006950[response to stress]*

gene	DV	M	SD
AT2G05380	0.5219	0.3521	0.0156
AT2G39310	0.5217	0.3446	0.0148
AT2G41230	0.5215	0.3384	0.0142
AT3G02480	0.5215	0.3378	0.0139
AT1G29670	0.5215	0.3360	0.0140
AT3G28220	0.5215	0.3360	0.0139
AT3G16390	0.5214	0.3337	0.0148
AT4G01390	0.5210	0.3256	0.0161
AT3G05730	0.5209	0.3249	0.0154
AT5G46960	0.5208	0.3243	0.0137
AT4G33560	0.5207	0.3232	0.0141
AT5G22460	0.5206	0.3216	0.0145
AT1G03940	0.5206	0.3209	0.0142
AT2G15560	0.5205	0.3206	0.0140
AT5G26280;AT5G26260	0.5203	0.3169	0.0127
AT5G10040	0.5202	0.3159	0.0122
AT3G10320	0.5201	0.3146	0.0106
AT5G64510	0.5200	0.3106	0.0101
AT1G16850	0.5199	0.3087	0.0100
AT2G41380	0.5199	0.3081	0.0100
AT2G41730	0.5198	0.3073	0.0100
AT3G55970	0.5198	0.3067	0.0095
AT3G16450	0.5197	0.3060	0.0092
AT2G19850	0.5197	0.3044	0.0094

Table B.19: *GO:0006950[response to stress]*

gene	DV	M	SD
AT1G53885	0.5196	0.3023	0.0092
AT1G80240	0.5196	0.3019	0.0089
AT4G23670	0.5195	0.3002	0.0091
AT3G59930;AT5G33355	0.5195	0.3003	0.0091
AT2G24762	0.5194	0.2993	0.0089
AT3G18250	0.5193	0.2971	0.0087
AT5G07010	0.5193	0.2963	0.0086
AT1G18980	0.5192	0.2940	0.0088
AT1G67870	0.5192	0.2938	0.0089
AT1G52070	0.5192	0.2937	0.0089
AT5G38940;AT5G38930	0.5192	0.2936	0.0089
AT5G37990	0.5192	0.2936	0.0088
AT1G11210	0.5192	0.2927	0.0085
AT3G16530	0.5191	0.2923	0.0083
AT4G30140	0.5191	0.2911	0.0080
AT2G32160	0.5190	0.2903	0.0078
AT5G09480	0.5190	0.2890	0.0082
AT5G45500	0.5190	0.2888	0.0082
AT3G10930	0.5189	0.2883	0.0089
AT5G25460	0.5189	0.2877	0.0090
AT3G45730	0.5189	0.2874	0.0087
AT1G19960	0.5188	0.2863	0.0087
AT5G20790	0.5188	0.2861	0.0087
AT1G80130	0.5187	0.2849	0.0090

Table B.19: *GO:0006950[response to stress]*

gene	DV	M	SD
AT3G48640	0.5187	0.2847	0.0089
AT1G33055	0.5187	0.2844	0.0086
AT2G14560	0.5186	0.2838	0.0085
AT1G25097;AT1G24822;AT1G24996;AT1G25170	0.5186	0.2835	0.0084
AT1G78450	0.5185	0.2829	0.0085
AT3G50480	0.5184	0.2813	0.0087
AT4G14060	0.5183	0.2794	0.0085
AT1G09310	0.5183	0.2794	0.0084
AT4G37070;AT4G37060	0.5182	0.2788	0.0090
AT5G37300	0.5181	0.2785	0.0090
AT5G23820	0.5180	0.2770	0.0090
AT2G46790;AT2G46670	0.5180	0.2768	0.0089
AT1G58270	0.5180	0.2767	0.0090
AT2G44240	0.5179	0.2762	0.0091
AT2G26400	0.5179	0.2765	0.0089
AT1G70830;AT1G70850	0.5178	0.2747	0.0079
AT5G02580	0.5177	0.2740	0.0075

B.5 K-Nearest Neighbors

Table B.20: *GO:0006950[response to stress]*

gene	DV	M	SD
AT5G38940;AT5G38930	0.4510	1.0000	1.4651
AT1G80960	0.4314	0.6446	0.5063
255181_at	0.3725	0.6164	0.0867
AT4G39675	0.3725	0.6164	0.0867
AT4G02270	0.3725	0.6164	0.0867
AT3G28290;AT3G28300	0.3725	0.6164	0.0867
AT2G39310	0.3725	0.6164	0.0867
AT3G44860	0.3529	0.5683	0.0466
AT5G05500	0.3529	0.5683	0.0466
AT3G50480	0.3529	0.5683	0.0466
AT4G30140	0.3529	0.5683	0.0466
AT4G00680	0.3529	0.5683	0.0466
AT1G21360	0.3529	0.5683	0.0466
AT1G01750	0.3529	0.5683	0.0466
AT1G70830;AT1G70850	0.3529	0.5683	0.0466
245079_at	0.3333	0.5137	0.0323
AT2G27370	0.3333	0.5137	0.0323
AT4G25790	0.3333	0.5137	0.0323
AT3G23190	0.3333	0.5137	0.0323
AT3G11550	0.3333	0.5137	0.0323
AT3G16460	0.3333	0.5137	0.0323
AT3G16440	0.3333	0.5137	0.0323
AT3G16450	0.3333	0.5137	0.0323
AT1G30750	0.3333	0.5137	0.0323

Table B.20: *GO:0006950[response to stress]*

gene	DV	M	SD
AT2G01530	0.3333	0.5137	0.0323
AT5G40730	0.3137	0.4529	0.0267
AT3G54040	0.3137	0.4529	0.0267
AT3G22240	0.3137	0.4529	0.0267
AT1G33700	0.3137	0.4529	0.0267
AT1G53870;AT1G53890	0.3137	0.4529	0.0267
AT2G36100	0.3137	0.4529	0.0267
AT1G12080	0.3137	0.4529	0.0267
258246_s_at	0.2941	0.4179	0.0187
AT4G16960;AT4G16880;AT4G16940	0.2941	0.4179	0.0187
AT5G26300;AT5G26280;AT5G26260	0.2941	0.4179	0.0187
AT5G50670;AT5G50570	0.2941	0.4179	0.0187
AT5G23830	0.2941	0.4179	0.0187
AT4G30670	0.2941	0.4179	0.0187
AT4G30320	0.2941	0.4179	0.0187
AT2G07777;ATMG01090	0.2941	0.4179	0.0187
AT3G09350	0.2941	0.4179	0.0187
AT1G14120	0.2941	0.4179	0.0187
AT1G52070	0.2941	0.4179	0.0187
AT5G26280;AT5G26260	0.2745	0.4109	0.0138
AT3G59930;AT5G33355	0.2745	0.4109	0.0138
AT3G47250	0.2745	0.4109	0.0138
AT4G29270	0.2745	0.4109	0.0138
AT4G23680	0.2745	0.4109	0.0138

Table B.20: *GO:0006950[response to stress]*

gene	DV	M	SD
AT5G44820	0.2745	0.4109	0.0138
AT3G16430;AT3G16420	0.2745	0.4109	0.0138
AT1G18980	0.2745	0.4109	0.0138
AT1G80240	0.2745	0.4109	0.0138
AT2G44010	0.2745	0.4109	0.0138
AT1G30250	0.2549	0.3926	0.0123
AT4G36500	0.2549	0.3926	0.0123
AT5G60950	0.2549	0.3926	0.0123
AT5G51620	0.2549	0.3926	0.0123
AT5G50565;AT5G50665	0.2549	0.3926	0.0123
AT5G44610	0.2549	0.3926	0.0123
AT5G44820	0.2549	0.3926	0.0123
AT4G38080	0.2549	0.3926	0.0123
AT4G33730	0.2549	0.3926	0.0123
AT4G00080	0.2549	0.3926	0.0123
AT1G13750	0.2549	0.3926	0.0123
AT3G16800	0.2549	0.3926	0.0123
AT3G20370	0.2549	0.3926	0.0123
AT3G06500	0.2549	0.3926	0.0123
AT3G04320	0.2549	0.3926	0.0123
AT1G73260	0.2549	0.3926	0.0123
AT1G76960	0.2549	0.3926	0.0123
AT1G67330	0.2549	0.3926	0.0123
245449_at	0.2353	0.3606	0.0090

Table B.20: *GO:0006950[response to stress]*

gene	DV	M	SD
AT4G37070;AT4G37060	0.2353	0.3606	0.0090
AT5G25280	0.2353	0.3606	0.0090
AT5G46500	0.2353	0.3606	0.0090
AT5G44570	0.2353	0.3606	0.0090
AT5G39120;AT5G39150;AT5G39180;AT5G39110	0.2353	0.3606	0.0090
AT5G24313	0.2353	0.3606	0.0090
AT5G23840	0.2353	0.3606	0.0090
AT5G03350	0.2353	0.3606	0.0090
AT3G56410	0.2353	0.3606	0.0090
AT3G52470	0.2353	0.3606	0.0090
AT4G22080;AT4G22090	0.2353	0.3606	0.0090
AT1G19180	0.2353	0.3606	0.0090
AT1G16850	0.2353	0.3606	0.0090
AT3G12540	0.2353	0.3606	0.0090
AT3G20590	0.2353	0.3606	0.0090
AT3G29670	0.2353	0.3606	0.0090
AT1G72450	0.2353	0.3606	0.0090
AT1G07600;AT1G07590	0.2353	0.3606	0.0090
AT1G14870	0.2353	0.3606	0.0090
AT1G59930;AT1G59920	0.2353	0.3606	0.0090
AT2G31880;AT2G31890	0.2353	0.3606	0.0090
AT2G15890	0.2353	0.3606	0.0090
AT2G05510	0.2353	0.3606	0.0090
AT2G01520	0.2353	0.3606	0.0090

Table B.20: *GO:0006950[response to stress]*

gene	DV	M	SD
AT2G30930	0.2353	0.3606	0.0090
265974_at	0.2157	0.3307	0.0095
AT5G64870	0.2157	0.3307	0.0095
AT5G62280	0.2157	0.3307	0.0095

Appendix C

Pseudocode

This chapter shows the pseudocode for generating the TP vs. PP plot (Section C.1), generating the prediction table (Section C.2), generating the precision vs. PP plot (Section C.3), model intersection based on discriminant values (Section C.4), model averaging and model stacking (Section C.5).

C.1 The TP vs. PP Plot

Algorithm C.1.1 shows how to generate the cloudy and mean TP vs. PP plot for a given classification method, M. With method M, cross-validation on the training data is performed 100 times, yielding 100 discriminant values for each gene. As thresholds, another top 100 discriminant values are generated by applying the classifier C (learned from the training data using method M) back to the training data. For each of the top 100 discriminant values, 100 TP and 100 PP are computed using the cross-validation results. Also, for each of the top 100 discriminant values, the mean of the 100 TP and the mean of the 100 PP are computed for generating the mean TP vs. PP curve.

Algorithm C.1.1: Plot_TP_VS_PP(*method* : M)

```
Train a classifier, C, using method M.
Apply C to the training data to generate a discriminant value for each gene.
Let $DV_1, DV_2, ..., DV_{100}$ be the top 100 such discriminant values.
Begin plot
  1. Generate 100 randomizations, $\mathcal{R}$, of the training data.
     for each randomization $\in \mathcal{R}$
     do { Do 20-fold cross-validation using method M on the training data.
          Store the cross-validation results (the discriminant value for each gene).
  2.
     for each $DV_i \in \{DV_1, DV_2, ..., DV_{100}\}$
     do { for each randomization $\in \mathcal{R}$
              do { Compute TP and PP from the cross-validation results          (i)
                   (PP is the number of predicted positives whose discriminant
                   values are greater than $DV_i$, and TP is number of
                   the predicted positives that are true).
                   plot(PP,TP).
          Compute the average TP (mean(TP)) and average PP (mean(PP))
          over all the randomizations.
          plot(mean(PP),mean(TP)).
End plot
```

Algorithm C.1.2 shows how to generate the cloudy and mean TP vs. PP plot for the permutation data, given a classification method, M. This procedure is similar to Algorithm C.1.1 except that the GOBP data are permuted for each of the 100 randomizations.

Algorithm C.1.2: Plot_TP_VS_PP_Permuted(*method* : M)

```
Train a classifier, C, using method M.
Apply C to the training data (GOBP data permuted) to generate a discriminant
value for each gene.
Let $DV_1, DV_2, ..., DV_{100}$ be the top 100 such discriminant values.
Begin plot
  1. Generate 100 randomizations, $\mathcal{R}$, of the training data.
     for each randomization $\in \mathcal{R}$
     do { Permute the GOBP data.
          Do 20-fold CV using method M on the training data.
          Store the cross-validation results (the discriminant value for each gene).
  2.
     for each $DV_i \in \{DV_1, DV_2, ..., DV_{100}\}$
     do { for each randomization $\in \mathcal{R}$
            do { Compute TP and PP from the cross-validation results (see (i)
                 in Algorithm C.1.1).
                 plot(PP,TP).
          Compute the average TP (mean(TP)) and average PP (mean(PP))
          over all the randomizations.
          plot(mean(PP),mean(TP)).
End plot
```

C.2 The Prediction Table

Algorithm C.2.1 shows how to generate the prediction table. A classifier, C, learned from the training data, is applied to the prediction data to predict the discriminant value of each gene. For each of the top 100 predictions (with the top 100 discriminant values), the mean precision and the standard deviation of the precision are computed using the cross-validation results on the training data.

Algorithm C.2.1: Generate_Prediction_Table(*method* : M)

```
% Obtain the cross-validation results and 100 randomizations, $\mathcal{R}$ (see Algorithm C.1.1).
Plot_TP_VS_PP(M).
Train a classifier, C, using method M.
Apply classifier C to the prediction data to generate a discriminant value for
each gene.
Let $DV_1, DV_2, ..., DV_{100}$ be the top 100 such discriminant values.
for each $DV_i \in \{DV_1, ..., DV_{100}\}$
    do {
        for each randomization $\in \mathcal{R}$
            do {
                Compute TP and PP from the cross-validation results (see
                (i) in Algorithm C.1.1).
            }
        Compute average TP (mean(TP)) and average PP (mean(PP))
        over all randomizations.
        Let mean = mean(TP)/mean(PP).
        Let sd = standard deviation of the precision (see Formula 3.18 in Chapter 3).
        print(gene $i$, $DV_i$, mean, sd).
    }
```

C.3 The Precision vs. PP Plot

Algorithm C.3.1 shows how to generate the (cloudy) precision vs. PP plot for a given classification method, M. A classifier C is learned from the training data using method M. The discriminant value of each gene in the prediction data is predicted using C. The top 100 genes (with the top 100 discriminant values) are selected as predictions. For each prediction, 100 precision estimates and the mean of them are computed using the cross-validation results on the training data. Finally, a theoretical precision that the random classifier could achieve is added.

Algorithm C.3.1: Plot_Precision_VS_PP(*method* : M)

```
% Obtain the cross-validation results and 100 randomizations, $\mathcal{R}$ (see Algorithm C.1.1).
Plot_TP_VS_PP(M).
Train a classifier, C, using method M.
Apply classifier C to the prediction data to generate a discriminant value for
each gene.
Let $DV_1, ..., DV_{100}$ be the top 100 such discriminant values.
Begin plot
  1.
    for each $DV_i \in \{DV_1, DV_2, ..., DV_{100}\}$
    do {
        for each randomization $\in \mathcal{R}$
        do {
            Compute TP and PP from the cross-validation results (see (i)
            in Algorithm C.1.1).
            Let $\hat{prec}$ = TP/PP.
            plot(PP, $\hat{prec}$).
        }
        Compute average TP (mean(TP)) and average PP (mean(PP))
        over all randomizations.
        Let mean = mean(TP)/mean(PP).
        PP = $i$.
        plot(PP, mean).
    }
  2. % Plot the theoretical worst-case curve.
    RP = #positives in training data / #training samples.
    Add a horizontal line with height RP for the random classifier.
End plot
```

C.4 C-AND

Algorithm C.4.1 shows how we intersect two classification methods, M_1 and M_2, and how we assess them. The basic idea is to compare the top n predictions of M_1 with the top n predictions of M_2, and to keep only those predictions that are in both sets. This is done for n from 1 to N.

Algorithm C.4.1: C-AND($method$: M_1, M_2)

```
1.
  Generate 100 randomizations, $\mathcal{R}$, of the training data.
  for each randomization $\in \mathcal{R}$
    do { Using 20-fold cross-validation, estimate a discriminant value for
         each training sample using $M_1$.
         Sort the training samples by discriminant values.
  for each randomization $\in \mathcal{R}$
    do { Using 20-fold cross-validation, estimate a discriminant value for
         each training sample using $M_2$.
         Sort the training samples by discriminant values.
2.
  Let $N = 100$.
  Apply method $M_1$ to the training data to get a classifier, $C_1$.
  Let $a_1, ..., a_N$ be the top $N$ discriminant values of $C_1$ on the training samples.
  Likewise for $b_1, ..., b_N$. (Note: This is the training step.)
  Let $(a_1, b_1), (a2, b2), ..., (a_N, b_N)$ be the $N$ pairs of discriminant values.
  for each randomization $\in \mathcal{R}$
    do { for each $(a_i, b_i)$
           do { Using the sorted training samples, compute PP and TP (PP is
                the number predicted positives, and TP is number of true
                positives predicted by both $M_1$ and $M_2$).
                plot(PP,TP).
```

C.5 Model Averaging and Model Stacking

Algorithm C.5.1 shows the code for model averaging. For each of the basic classification method, M$_j$ ($j = 1, 2, ..., 5$), a basic classifier, C$_j$, is learned from the training data. C$_j$ is applied back to the training data to generate a discriminant value for each gene. Thus, each gene has a new feature vector consisting of the discriminant values from the five basic classifiers. These feature vectors are input to a combining classification method, C, to create an “averaged classifier.” The averaged classifier linearly combines the five basic classifiers using five weights.

Algorithm C.5.1: Model_averaging($method : \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4, \mathrm{M}_5$)

```
1. Training:
   Let C be the combining classifier.
   Train a basic classifier, $\mathrm{C}_j$, using method $\mathrm{M}_j$ $(j = 1, 2, ..., 5)$.
   Apply $\mathrm{C}_j$ to all of the training data to generate a discriminant value for each
   gene.
   Let $D_{ij}$ be the discriminant value of sample $\mathbf{x}_i$ using classifier $\mathrm{C}_j$.
   So, for each gene, $i$, we get a vector of discriminant values, $(D_{i1}, D_{i2}, ...D_{i5})$.
   Create an "averaged classifier" by performing C on these new feature vectors
   (using same GOBP).
   This averaged classifier produces a discriminant value for each gene.
2. Prediction:
   for each gene in the prediction data
   do {
       for each method $\mathrm{M}_j \in \{\mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4, \mathrm{M}_5\}$
          do { Compute a "basic" discriminant value.
   }
   for each gene in the prediction data
   do {
       Apply the averaged classifier to the gene's basic discriminant values
       to compute an averaged discriminant value.
   }
3. Cross-validation:
   Divide the training data into 20 folds.
   for each fold, F
   do {
       Train all the basic classifiers on the other 19 folds.
       Train the averaged classifier on the other 19 folds using C.
       Test the averaged classifier on fold F.
   }
```

Algorithm C.5.2 shows the code for model stacking. For each of the five basic classification methods, M$_j$ ($j = 1, 2, ..., 5$), cross-validation is used to generate a discriminant value for each gene in the training data. Thus, each gene has a new feature vector consisting of the five discriminant values from the cross-validation results. These new feature vectors are input to a combining classifier, C, to create a "stacked classifier," which linearly combines the five basic classifiers using five weights.

Algorithm C.5.2: Model_stacking(*method* : M_1, M_2, M_3, M_4, M_5)

```
1. Training:
   Divide the training data into 20 folds.   for each fold, F
   do { for each method $M_i \in \{M_1, M_2, M_3, M_4, M_5\}$
        do { Train a classifier on the other 19 folds.
             Using the classifier, compute a discriminant value for each gene in fold F.
   % We now have a feature vector of five discriminant values for each gene.
   Create a "stacked classifier" by training the combining classifier, C, on
   these new feature vectors (using same GOBP).
   This stacked classifier produces a discriminant value for each gene.
2. Prediction:
   for each basic method, $M_i \in \{M_1, M_2, M_3, M_4, M_5\}$
   do { Train a basic classifier, $C_i$, on all the training data (for use in prediction).
   for each gene in the prediction data
   do { for each basic classifier, $C_i$
        do { Compute a "basic" discriminant value.
   for each gene in the prediction data
   do { Apply the stacked classifier to the gene's basic discriminant values
        to compute a stacked discriminant value.
3. Cross-validation:
   Divide the training data into 20 folds.   for each fold F of the 20 folds
   do { Train a stacked classifier on the other 19 folds (note that this involves
        dividing the 19 folds into training and validation folds for training the
        stacked classifier).
        Test the stacked classifier on fold F.
```

Bibliography

[1] U Alon, N Barkai, DA Notterman, K Gish, S Ybarra, D Mack, and AJ Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.

[2] MP Brown, WN Grundy, D Lin, N Cristianini, CW Sugnet, TS Furey, M Jr Ares, and D Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of National Academy of Sciences of the United States of America*, 97(1):262–267, 2000.

[3] The Gene Ontology Consortium. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[4] C Dennis and C Surridge. *A. thaliana* genome. *Nature*, 408:791, 2000.

[5] MB Eisen, PT Spellman, PO Brown, and D Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Sciences of the United States of America*, 95(25):14863–14868, 1998.

[6] TS Furey, N Cristianini, N Duffy, DW Bednarski, M Schummer, and D Haussler. Support vector machine classification of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[7] R Guthke, W Schmidt-Heck, D Hahn, and M Pfaff. Gene expression data mining for functional genomics using fuzzy technology. In *Advances in Computational Intelligence and Learning: Methods and Applications*, pages 475–487, 2002.

[8] I Guyon, J Weston, S Barnhill, and V Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1/3):389–422, 2002.

[9] TK Hartigan, A Lægreid, J Komorowski, and E Hoving. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28, 2001.

[10] T Hastie, R Tibshirani, and J Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, New York, 2001.

[11] H Hishigaki, K Nakai, T Ono, A Tanigami, and T Takagi. Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*, 18(6):523–531, 2001.

[12] TR Hughes, MJ Marton, AR Jones, CJ Roberts, R Stoughton, CD Armour, HA Bennett, E Coffey, H Dai, YD He, MJ Kidd, AM King, MR Meyer, D Slade, PY Lum, SB Stepaniants, DD Shoemaker, D Gachotte, K Chakraburtty, J Simon, M Bard, and SH Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

[13] TR Hvidsten, J Komorowski, AK Sandvik, and A Laegreid. Predicting gene function from gene expressions and ontologies. In *Pacific Symposium on Biocomputing*, pages 299–310, 2001.

[14] J Khan, JS Wei, M Ringner, LH Saal, M Ladanyi, F Westermann, F Berthold, M Schwab, CR Antonescu, C Peterson, and PS Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679, 2001.

[15] OD King, RE Foulger, SS Dwight, JV White, and FP Roth. Predicting gene function from patterns of annotation. *Genome Research*, 13(5):896–904, 2003.

[16] M Kuramochi and G Karypis. Gene classification using expression profiles: A feasibility study. In *2nd IEEE International Symposium on Bioinformatics and Bioengineering*, pages 191–200, 2001.

[17] A Lægreid, TR Hvidsten, H Midelfart, J Komorowski, and AK Sandvik. Predicting gene ontology biological process from temporal gene expression patterns. *Genome Research*, 13(5):965–979, 2003.

[18] Y Lee and CK Lee. Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19(9):1132–1139, 2003.

[19] T Li, S Zhu, Q Li, and M Ogihara. Gene functional classification by semi-supervised learning from heterogeneous data. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 78–82, 2003.

[20] Y Lu and J Han. Cancer classification using gene expression data. *Information Systems Special Issue: Data Management in Bioinformatics*, 28(4):243–268, 2003.

[21] A Mateos, J Dopazo, R Jansen, Y Tu, M Gerstein, and G Stolovitzky. Systematic learning of gene functional classes from DNA array expression data by using multilayer perceptrons. *Genome Research*, 12(11):1703–1715, 2002.

[22] DW Meinke, JM Cherry, C Dean, SD Rounsley, and M Koornneef. *Arabidopsis thaliana*: A model plant for genome analysis. *Science*, 282(5389):662–682, 1998.

[23] H Midelfart, A Lægreid, and J Komorowski. Classification of gene expression data in an ontology. In *Proceedings of the 2nd International Symposium on Medical Data Analysis*, pages 186–194, 2001.

[24] S Mnaimneh, AP Davierwala, J Haynes, J Moffat, WT Peng, W Zhang, X Yang, J Pootoolal, G Chua, A Lopez, M Trochesset, D Morse, NJ Krogan, SL Hiley, Z Li, Q Morris, J Grigull, N Mitsakakis, CJ Roberts, JF Greenblatt, C Boone, CA Kaiser, BJ Andrews, and TR Hughes. Exploration of essential gene functions via titratable promoter alleles. *Cell*, 118(1):31–44, 2004.

[25] DV Nguyen and DM Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50, 2002.

[26] C Niehrs and N Pollet. Synexpression groups in eukaryotes. *Nature*, 402(6761):483–487, 1999.

[27] P Pavlidis, J Weston, J Cai, and WN Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the 5th International Conference on Computational Molecular Biology*, pages 242–248, 2001.

[28] S Ramaswamy, P Tamayo, R Rifkin, S Mukherjee, CH Yeang, M Angelo, C Ladd, M Reich, E Latulippe, JP Mesirov, T Poggio, W Gerald, M Loda, ES Lander, and TR Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of National Academy of Sciences of the United States of America*, 98(26):15149–15154, 2001.

[29] M Schena, D Shalon, RW Davis, and PO Brown. Quantitative monitoring of gene expression patterns with a DNA microarray. *Science*, 270(5235):467–470, 1995.

[30] D Shalon, SJ Smith, and PO Brown. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, 6(7):639–645, 1996.

[31] H Shatkay, S Edwards, WJ Wilbur, and M Boguski. Genes, themes and microarrays: Using information retrieval for large-scale gene analysis. In *Proceedings of the*

International Conference on Intelligent Systems for Molecular Biology, volume 8, pages 317–328, 2000.

[32] P Tamayo, D Slonim, J Mesirov, Q Zhu, S Kitareewan, E Dmitrovsky, ES Lander, and TR Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of National Academy of Sciences of the United States of America*, 96(6):2907–2912, 1999.

[33] S Tavazoie, JD Hughes, MJ Campbell, RJ Cho, and GM Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22(3):281–285, 1999.

[34] K Toufighi, SM Brady, R Austin, E Ly, and NJ Provart. The Botany Array Resource: E-northerns, expression angling, and promoter analyses. *The Plant Journal*, 43(1):153–63, 2005.

[35] M Trochesset and A Bonner. Clustering labeled data and cross-validation for classification with few positives in yeast. In *Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics (BioKDD)*, 2004.

[36] V Walbot. A green chapter in the book of life. *Nature*, 408:794–795, 2000.

[37] M West, C Blanchette, H Dressman, E Huang, S Ishida, R Spang, H Zuzan, JA Jr Olson, JR Marks, and JR Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of National Academy of Sciences of the United States of America*, 98(20):11462–11467, 2001.

[38] LF Wu, TR Hughes, AP Davierwala, MD Robinson, R Stoughton, and SJ Altschuler. Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nature Genetics*, 31(3):255–265, 2002.

[39] W Zhang, QD Morris, R Chang, O Shai, MA Bakowski, N Mitsakakis, N Mohammad, MD Robinson, R Zirngibl, E Somogyi, N Laurin, E Eftekharpour, E Sat,

J Grigull, Q Pan, WT Peng, N Krogan, J Greenblatt, M Fehlings, van der D Kooy, J Aubin, BG Bruneau, J Rossant, BJ Blencowe, BJ Frey, and TR Hughes. The functional landscape of mouse gene expression. *Jounral of Biology*, 3(5):21, 2004.