1-1-2005

Application of gene ontology and rough sets theory in predicting molecular functions and biological processes for microarray gene expression data

Jian Gong
Iowa State University

Application of gene ontology and rough sets theory in predicting molecular functions and biological processes for microarray gene expression data

by

Jian Gong

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Julie A. Dickerson, Major Professor
Ranjan Maitra
Daniel Berleant

Iowa State University

Ames, Iowa

2005

Graduate College
Iowa State University

This is to certify that the master's thesis of

Jian Gong

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

—

—

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to give my sincerely thanks to my major professor, Dr. Julie Dickerson, for any suggestion, advices, help, encouragement, and patience you have given to me, either on my study or on my life.

It is my pleasure to have Dr. Berleant and Dr. Maitra being my committee members. Thank you for your patience and understanding.

I would also thank Dr. Lishuang Shen, Pan Du and Xiaoyun Tang for your suggestions and help in dealing with technical problems. Thanks should also go to Ling Li, for your kind help on validating the prediction results on Arabidopsis data. You are all my good teachers.

Thanks also are due to my dear parents, my sister, and my brother who are in China. Thank you for your support so many years, I love you. Special thanks go to my husband, Jianqiang Xin and my daughter Ray Xin, thank you for your support and love, you make my life colorful.

ABSTRACT

The functions of most plant genes are unknown even in the best-studied organisms. Finding out or estimating what functions or biological processes a gene involves can help interpret and understand the biological metabolic pathways. It is necessary to generate hypotheses about functions or biological processes for unknown genes to help design more meaningful experiments.

The traditional methods of microarray data analysis are based on the assumption that genes with similar expression profiles share the similar functions or biological processes, or genes with similar functions or biological processes share the similar expression profiles. In fact, genes with different functions or in different processes may have the similar expression profiles, and genes with similar profiles may have totally different functions or involve in different processes. To avoid using this assumption, supervised methods will be used.

Unlike the commonly used clustering methods, which start the analysis directly with the expression profiles, we used both the background knowledge (Gene Ontology annotation) and expression profiles during the analysis. First, gene expression data was annotated by some broad Gene Ontology (GO) terms, according to their positions in Directed Acyclic Graph (DAG) of GO. Then Rough sets theory was applied to generate rules that characterize every class so that the classifier can classify unknown genes or unclear genes into those broad GO classes. At last, the trained classifier will predict the unknown genes. This method gave reasonable results either on yeast cell cycle data set or Arabidopsis time-course data set.

ACRONYMS

Acronyms	Meanings
GO	Gene Ontology
MF	Molecular Function
BP	Biological Process
CC	Cellular Component
DAGs	Directed Acyclic Graphs
IS	Information System
DS	Decision System
TAIR	The Arabidopsis Information Resource

CHAPTER 1 GENERAL INTRODUCTION

1. Problem Statement

Finding out or estimating what biological processes a gene involves can help interpret and understand the biological metabolic pathways. A biological process, such as cell death, aging is a process that occurs in living organisms (http://www.thefreedictionary.com/biological%20process). The interactions between biological processes are very complicated. One biological process may require involvement of hundreds of genes. One gene may also be involved in many biological processes. Many genes have been studied and their biological processes have been found; however there are still a lot of genes within biological processes whose involvement is unknown, even in well-studied organisms. Ancillary into, such as DNA microarray techniques and Gene Ontology annotations can assist in predicting biological processes for unknown genes.

Usually unsupervised clustering analysis methods, such as PCA, CART, are used to address this problem. Unsupervised analysis finds genes with similar gene expression profiles and then generates hypotheses about the unknown genes in a cluster by looking at the functions of the known genes. Unsupervised clustering employs correlation or distance metrics to reflect the similarity of gene expression profiles among genes. Based on the assumption that genes with similar expression profiles share the similar functions or biological processes, the functions or biological processes of these unknown genes are then hypothesized. There are two shortcomings of this analysis: 1), genes having similar expression profiles do not necessarily occur in the same pathway. 2), Genes in the same

pathway do not have similar profiles. 3) This method did not include any background knowledge, such as the results from previous studies.

2. Proposed Solution

There are two disadvantages in unsupervised methods.

First, unsupervised methods, like clustering, are purely syntactical in the sense that it does not take advantage of the existing knowledge in the learning process [11]. The unsupervised techniques are entirely based on the numerical expression data, without showing biologically relevant information on the clustering results [9]. To combine biologically relevant information during analysis, we can use Gene Ontology annotations, which will be described in Chapter 2.

Second, unsupervised methods are based on the assumption that genes with similar expression profiles share the similar gene functions or biological processes. However, this assumption is not correct. Genes with different functions or in different processes may have the similar expression profiles, and genes with similar profiles may have totally different functions or involve in different processes. To avoid using this assumption, supervised methods will be used. Among supervised methods, Rough sets theory has been studied and applied to microarry data analysis, medicine data analysis widely in recent years [7] [8] [10] [11]. Rough Sets theory can be used without assuming independence between attributes, and the algorithm is built on the basis of discernibility theory, which will be discussed in Chapter 2 in detail.

We will use Rough sets theory and Gene Ontology annotation to predict molecular functions or biological processes from gene expression profiles.

According to their gene ontology annotations, the genes with GO molecular function (MF) or biological process (BP) annotations will be put into a training set. The genes with unknown GO MF or BP annotations are defined as unknown. The goal is to generate hypotheses about the MF or BP annotations for the unknown genes using a classifier based on rough sets theory. This classifier will be used to classifier those genes with unknown GO MF or BP annotations. Finally we will generate hypotheses of those unknown genes and verify those hypotheses.

3. Thesis Outline

Chapter 2 gives the background needed in this thesis, including microarray data analysis, Gene Ontology, and rough sets theory. Chapter 3 describes how Rough sets Theory and Gene Ontology information can be applied to yeast data for predicting the gene functions in yeast. Chapter 4 applies the same methodology to Arabidopsis data. Chapter 5 discusses the results and gives the conclusions.

CHAPTER 2 BACKGROUND

1. Microarray Data Analysis

The advent of large-scale DNA microarray technology provides biologists with benefits and challenges. One benefit is that tens of thousand genes can be studied simultaneously instead of one by one. The challenge is how to find useful information from this high dimensional gene expression data. For instance, the main question that is often asked is what similar/different patterns in the gene expression profiles really mean?

To address this challenge, many gene expression data analysis techniques and tools have recently emerged. There are two main methods in gene expression data analysis: unsupervised methods and supervised methods. Clustering is the most popular unsupervised method, which uses standard statistical algorithms to arrange genes according to similarity pattern of gene expression [1] [18] [19]. Supervised methods, such as Support Vector Machine (SVM) [20] [21] [22], are usually used to predict gene functions or sample tissue types.

2. Gene Ontology

For the increasing demand of data exchange and experiment comparison, Gene Ontology Consortium [2] made the Gene Ontology (GO), which aims to produce a shared, dynamic and controlled vocabulary that covers all organisms. The Gene Ontology consists of three categories: molecular function, biological process, and cellular component. Using the

ontology, high-quality annotations for many model organisms are now available [3]. The GO consortium is working continuously on updating GO terms and corresponding annotations, and all this information is stored in the GO database, which can be downloaded or accessed from http://www.geneontology.org.

The existence of GO provides us with both a controlled vocabulary, and also paves another method for function prediction, clustering interpretation and validation.

3. Unsupervised methods

For unsupervised analysis, which is shown in Figure 2.1, GO is used in cluster validation, gene function prediction, and biological interpretation of gene clusters as a complementary meaning to many statistical techniques. Kim et al [9] used a graph-theoretic model to interpret gene clusters in the GO space. Each GO term in a cluster was assigned a GO code, which is related to the depth of GO term in a GO DAG graph. The distance between each two GO terms in one cluster is the distance metric, called principal distance:

$$Pd(v_1, v_2) = \begin{cases} 0, & \text{if } a_i = b_i \text{ for all } i, \\ W(L), & \text{otherwise.} \end{cases},$$

where two GO codes $v_1 = a_1 a_2 ... a_H$, and $v_2 = b_1 b_2 ... b_H$ with $a_i, b_i \in N_0$, $L = \max_{1 \le i \le H} \{i | a_i = b_i\}$, $W(L) = 150 - 10(L-1)$. Finally the distance between all the GO terms in one gene cluster can be calculated as the average principal distance of all pairs of GO terms. From the result, they can interpret the gene functions or biological processes for one gene cluster by finding the representative GO code. Recently, Lord et al [13] presented a measurement called semantic similarity to measure how close two GO terms are to each other in the DAG graph. The semantic similarity measurement is based on the information

theory: more often one GO term occurs, less information it has. Thus the root of the DAG has no information, and the leaf nodes have more information. The minimum subsumer was defined as the common parents for two GO terms in DAGs with biggest information volume. The potential usage of semantic similarity is to make a search engine that can find genes or proteins with similar function, given a specific gene or protein.

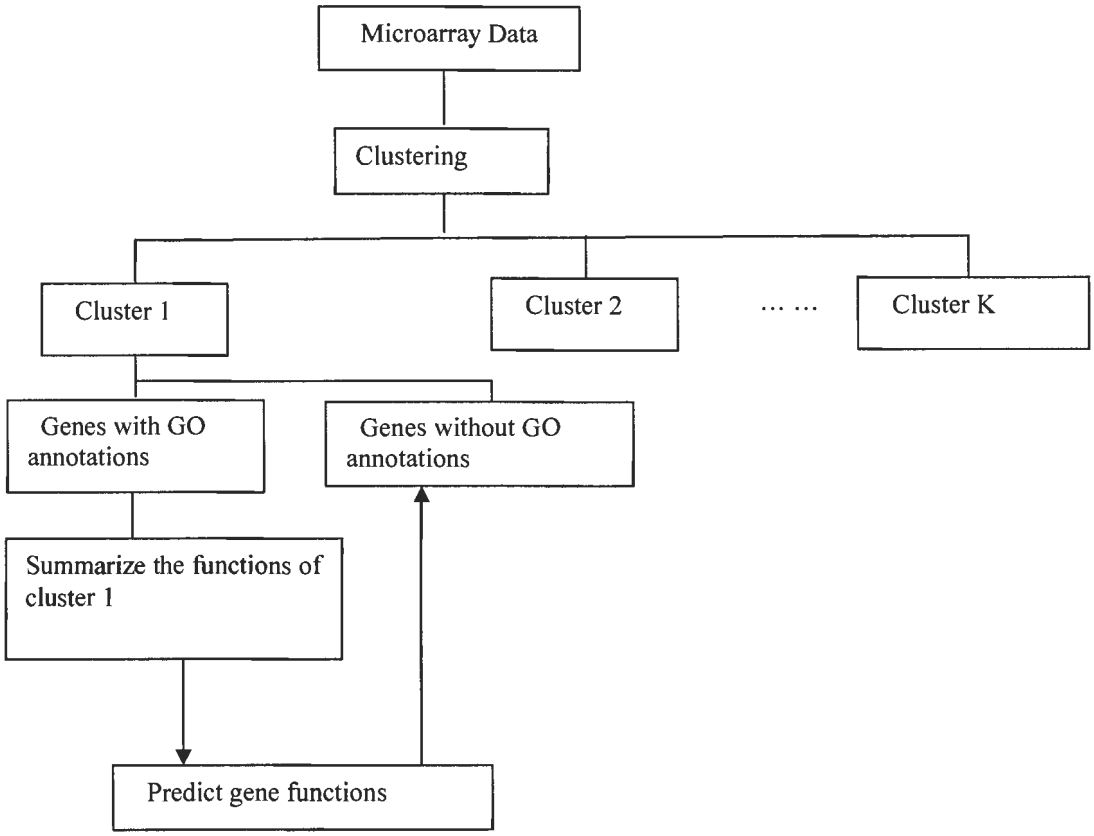

Figure 2.1: Microarray data analysis using unsupervised method

4. Supervised methods

In the analysis of microarray data, unsupervised methods can mine through data; extract relevant information without the presence of a class label, or teacher signal. However, supervised methods use class label, or teacher signal to extract information.

For supervised analysis, the main idea is illustrated in Figure 2.2 and it can be described as: (1) build classes from the known GO annotation; (2) extract useful features from the expression data; (3) induce rule model based on the previous features; (4) evaluate the classifier and apply the rule model to the unknown genes. One key issue in supervised analysis is how to build meaningful classes from the known GO annotation. Midelfart et al [7] used a threshold to limit class size, so that they can avoid the situation of overfitting when too few genes are in one class. Another key issue is how to generate the rules for. Idelfart et al also developed a rule-based classification algorithm on the fibroblast serum response time series data, using Rough sets theory. The validation results showed that supervised method could predict the gene function very well. Inductive Logic Programming (ILP) was also used to induce functional discrimination rules to classify the three subtypes of adenocarcinoma of the lung [12].

The main disadvantage of supervised method is that usually a gene is associated with only one main function, although a gene may have multiple functions. The other disadvantage is that, to avoid over-fitting, each gene will be labeled using more general gene ontology term, this caused the details of gene function are lost in some sense. The supervised methods are attractive, as they combine the background knowledge (from GO) with gene expression to make rules; unsupervised method, clustering is purely based on the similarity of gene expression without considering any background information. GO is only used after clustering to validate and interpret the clusters. From this point of view, the supervised method is more promising in giving reliable predictions.

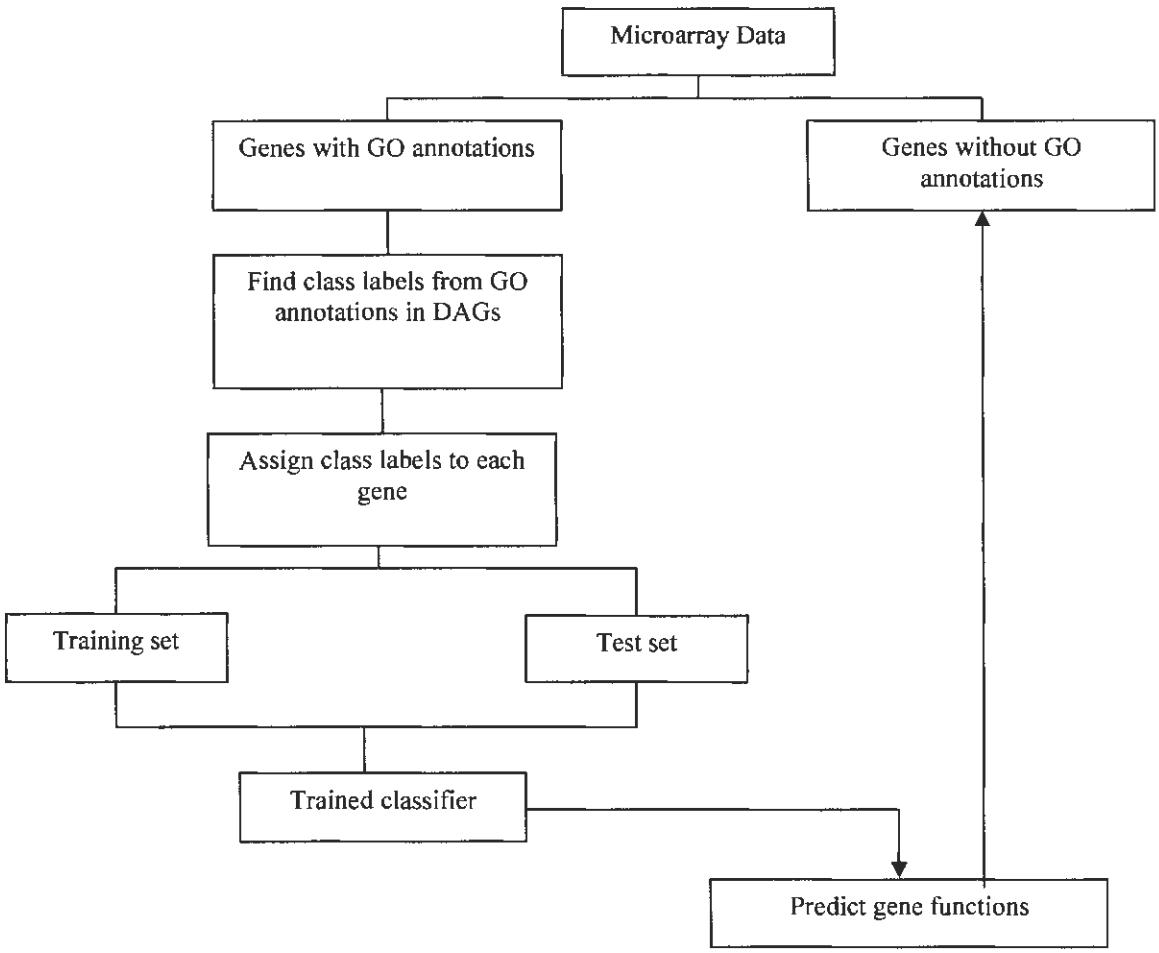

Figure 2.2: Microarray data analysis using supervised method

5. GO tools

Some GO tools are for querying and visualizing gene ontology, such as AmiGO [23], which can display the tree-structured relationship of GO terms for a gene product. It is a very convenient and popular GO tool. Some tools are for implementing automated gene ontology annotation, such as GoFigure, which accepts an input DNA or protein sequence, and uses BLAST to identify homologous sequences in the GO annotated databases, and then gives the output as a clickable graph of those GO terms [6]. Based on the assumption that genes

sharing similar functions behave similarly in an experiment, people have obtained many groups of genes using clustering methods. Gene ontology is used to find the shared functions or processes of a cluster of genes. FatiGO [5] is a web tool that either extracts important GO terms for a list of genes or compares two sets of genes and then gives those GO terms which are differently distributed in two groups, using multiple tests.

In Bioconductor [24], AnnBuilder builds up-to-date GO packages, which contain all the gene ontology information, such as, GO id, GO term, where the GO term comes from, and which aspect the GO term belongs to. The ontoTools package provides some functions for working on GO package, such as visualizing the DAG (directed acyclic graph) graph, computing the parent-child matrix, etc.

6. Rough sets theory

Rough sets theory was introduced by Pawlak in the early 1980's to deal with a vague description of objects. Rough sets allow for representing sets approximately in terms of the available context knowledge [15]. It is usually used for two main purposes: prediction and description. Prediction is concerned with predicting unknown values of some data using available information, while description means to identify important patterns in the data, and present them to the user in an understandable way [18]. An advantage of the methodology is that no assumptions about the independence of the attributes are necessary. Rough sets theory maybe good for microarray data because microarray data is usually high dimensional and the attributes are not independent.

6.1. Information system

An information system is basically a flat table or view [15]. Each row is an object/observation, and each column is an attribute. We define an information system Λ by a pair *(U, A)*, where *U* is a non-empty, finite set of objects and A is a non-empty, finite set of attributes.

$$\Lambda = (U, A) \tag{2.1}$$

Every attribute $a \in A$ of an object has a value. An attribute's value must be a member of the set V_a which is called the value set of attribute a.

$$x_1 a : U \rightarrow V_a \tag{2.2}$$

6.2. Decision System

Decision systems are a special kind of information system that label the objects of *A* to form classes. These classes can then be modeled using rough sets theory analysis. The classification labels are the target attribute for learning. An information system can be converted to a decision system by simply adding a decision attribute $d \notin A$ to Λ:

$$\Lambda' = (U, A \cup \{d\}) \tag{2.3}$$

For example, the data in Table 2.1 is an information system,

$\Lambda = (U, \{Hair, Height, Weight, Lotion\})$, and the corresponding decision system is

$\Lambda' = (U, \{Hair, Height, Weight, Lotion\} \cup \{Sunburn\})$.

Table 2.1: Sunburn: An example of decision system [16]

Name	Hair	Height	Weight	Lotion	Sunburn
Sarah	Blonde	average	Light	no	sunburned
Annie	Blonde	short	average	no	sunburned
Emily	Red	average	Heavy	no	sunburned
Pete	Brown	tall	Heavy	no	none
John	Brown	average	Heavy	no	none
Dana	Blonde	tall	average	yes	none
Alex	Brown	short	average	yes	none
Katie	Blonde	short	Light	yes	none

6.3. Indiscernibility

In many cases, objects in an information system are indistinguishable by a set of attributes $B \in A$. Such a set of objects are said to be indiscernible from each other and therefore, comprise an equivalence class on the set of attributes B. Such a set is denoted as $[x]_B$ or $IND_A(B) = \{(x, x') \in U^2 \mid \forall a \in B\, a(x) = a(x')\}$. Objects x and x' are said to be indiscernible from each other by attributes from B if $(x, x') \in IND_A(B)$. Take Table 1 for example, when $B = \{Hair\}$, we can see that $Sarah, Annie, Dana$ all have the same hair color: blond, so that they are indiscernible from each other with respect to Hair and they belong to the same equivalence class. For some possible B, we have

$$IND_A(\{Hair\}) = \{\{Sarah, Annie, Dana, Katie\}, \{Emily\}, \{Pete, John, Alex\}\}$$

$$IND_A(\{Height\}) = \{\{Sarah, Emily, John\}, \{Annie, Alex, Katie\}, \{Pete, Dana\}\} \quad (2.4)$$

$$IND_A(\{Weight, Lotion\}) = \{\{Sarah\}, \{Annie\}, \{Emily, Pete, John\}, \{Dana, Alex\}, \{Katie\}\}$$

6.4. Set approximation

The universe can be defined as the collection of all the objects. From the above, we can see that the universe can be divided into several subsets, given a set of attributes. For

example, for different values of Weight and Lotion, the universe can be partitioned as five subsets, $\{Sarah\}, \{Annie\}, \{Emily, Pete, John\}, \{Dana, Alex\}, \{Katie\}$. For a prediction problem, the most interesting subsets are those with the same decision attribute values. The problem is how to describe the features of one subset with the same decision value, or with the same class label. In practice most sets cannot be determined unambiguously and hence have to be approximated. This is the basic idea of rough sets. In an Information System with $\Lambda = (U, A)$ and $B \subseteq A$, we approximate decision class X using the information contained by the attribute set of B. The lower and upper approximations are defined:

$$X : \underline{B}X = \{x | [x]_B \subseteq X\} \tag{2.5}$$

$$X : \overline{B}X = \{x | [x]_B \cap X \neq 0\} \tag{2.6}$$

The lower approximation contains all objects that are definitely members of X on the basis of knowledge of B, while the objects in the set of the upper approximation are possible members of X on the basis of knowledge of B. The boundary region is defined as the difference between the upper and the lower approximation. This is the set of objects which cannot be unambiguously attributed to any set X :

$$X : BN_B(X) = \overline{B}X - \underline{B}X \tag{2.7}$$

The set of objects above the upper approximation consists of all objects that are certainly non-members of X. A set is rough if $BN_B(X) \neq 0$. Rough sets can also be characterized numerically by the following coefficient:

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|} \tag{2.8}$$

where $|X|$ denotes the cardinality of set X. Clearly, $0 \le \alpha_B(X) \le 1$. If $\alpha_B(X) = 1$, X is said to be crisp with respect to B, and otherwise, if $\alpha_B(X) < 1$, X is said to be rough with respect to B. To have any idea about how much an object x in some equivalent class belongs to X we define rough membership. The rough membership function quantifies the degree of relative overlap between the set X and the equivalence class to which x belongs.

$$\mu_X^B(x) : U \to [0,1] \; and \; \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \tag{2.9}$$

When all the objects in an equivalent class belong to set X, $\mu_X^B = 1$.

Using example in Table 2.1, let $B = \{Lotion\}$, and $X = \{x | Sunburn = none\}$, so $\underline{B}X = \{Dana, Alex, Katie\}$, $\overline{B}X = \{Dana, Alex, Katie, Pete, John\}$, $\underline{B}N_B(X) = \overline{B}X - \underline{B}X = \{Pete, John\}$, and $U - \overline{B}X = \{Sarah, Annie, Emily\}$. Since $\underline{B}N_B(X)$ is not empty, X is a rough set, with $\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|} = \frac{3}{5} = 0.6$.

6.5. Feature Selection

An important practical issue is whether some of the attributes in an information system are redundant with respect making the same object classifications as with the full set of attributes *A*. [25]

Feature reduction can occur if we find some subset of attributes, which preserve the same partition, or the same set approximation. The rest of the attributes are redundant since they don't have any effect on the performance of the partition or the classification. Such subsets of attributes are usually not unique, and those with minimal number of attributes are called reducts. Given an information system $\Lambda = (U, A)$, a reduct of A is a minimal set of attributes $B \in A$ such that $IND_{\Lambda}(B) = IND_{\Lambda}(A)$.

Searching for reducts is computationally expensive when the number of attributes increases. If the number of attributes is not too large, some existing methods based on genetic algorithm can compute the reducts in reasonable time.

6.6. Discernibility matrix

If Λ is an information system with n objects, the discernibility matrix of Λ is a symmetric $n \times n$ matrix with entries c_{ij} . Each entry consists of the set of attributes upon which objects x_i and x_j differ [16].

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \text{ for } i, j = 1,...,n \tag{2.10}$$

If the number of attributes is m, the attributes of Λ are denoted as: $a_1, a_2, ..., a_m$. The corresponding Boolean variables for $a_1, a_2, ..., a_m$ are $a_1^*, a_2^*, ..., a_m^*$. The discernibility function f_{Λ} for an information system Λ is defined as below,

$$f_{\Lambda}(a_1^*, a_2^*, ..., a_m^*) = \cap\{\cup c_{ij}^* \mid 1 \leq j \leq i \leq n, c_{ij} \neq empty\}, where\ c_{ij}^* = \{a^* \mid a \in c_{ij}\} \tag{2.11}$$

The resulting set, f_Λ gives the set of the reduct of Λ, which is also a minimal set of attributes that have the same classification performance as all attributes.

Still using the Sunburn example, our goal is to find the minimal set of attributes, which discern objects in the same ways as the full set of attributes.

Let $a_1 = Hair, a_2 = Height, a_3 = Weight, a_4 = Lotion$ and let the corresponding Boolean variables $a_1^* = Ha, a_2^* = He, a_3^* = W, a_4^* = L$. The discernibility function for Λ is:

$$
\begin{aligned}
f_\Lambda(Ha, He, W, L) = {} & (He \cup W) \cap (Ha \cup W) \cap (Ha \cup He \cup W) \cap (Ha \cup W) \\
& \cap (He \cup We \cup L) \cap (Ha \cup He \cup W \cup L) \cap (He \cup L) \\
& \cap (Ha \cup He \cup W) \cap (Ha \cup He \cup W) \cap (Ha \cup He \cup W) \cap (He \cap L) \cap (Ha \cup L) \cap (W \cup L) \\
& \cap (Ha \cup He) \cap (Ha) \cap (Ha \cup He \cup W \cup L) \cap (Ha \cup W \cup L) \cap (Ha \cup He \cup W \cup L) \\
& \cap (He) \cap (Ha \cup W \cup L) \cap (He \cup W \cup L) \cap (Ha \cup He \cup W \cup L) \\
& \cap (Ha \cup He \cup W \cup L) \cap (He \cup W \cup L) \cap (Ha \cup He \cup W \cup L) \\
& \cap (Ha \cup He) \cap (He \cup W) \\
& \cap (Ha \cup W)
\end{aligned}
\tag{2.12}
$$

The set of all prime implicants [16] of f_Λ determines the set of all reducts of information system Λ. An implicant of a Boolean function F is any conjunction of variables such that if the values of these variables are true then the values of the function F is also true no matter what other variables are. A prime implicant is a minimal implicant. From (2.11), we can see if He, Ha, W and L are all true, the value of f_Λ is also true, so {He,Ha,W,L} is one implicant. Also we can see {He,Ha,W}and {He,Ha,L} are also implicants. No other implicants exist for (2.11). So the prime implicants are {He,Ha,W} or {He,Ha,L}.

Table 2.2: The object-relative discernibility matrix

	Sarah	*Annie*	*Emily*	*Pete*	*John*	*Dana*	*Alex*	*Katie*
Sarah								
Annie	He,W							
Emily	Ha,W	Ha,He,W						
Pete	Ha,He,W	Ha,He,W	Ha,He					
John	Ha,W	Ha,He,W	Ha	He				
Dana	He,W,L	He,L	Ha,He,W,L	Ha,W,L	Ha,He,W, L			
Alex	Ha,He,W,L	Ha,L	Ha,He,W,L	He,W,L	He,W,L	Ha,He		
Katie	He,L	W,L	Ha,He,W,L	Ha,He,W, L	Ha,He,W, L	He,W	Ha,W	

If the operation is restricted to only run over a column j in the discernibility matrix, instead of all the columns, it will yield the j- relative discernibility function. The results will be the reducts (set of attributes) upon which x_j is differ from all other objects.

6.7. Decision rules

Usually, we are more interested in how to use the available attributes to describe the objects in different classes, so that we can use those descriptions (or rules) to classify those unknown objects.

For a decision system $\Lambda' = (U, A \cup \{d\})$, the universe can be partitioned into m subuniverses, where $m = r(d)$ and $r(d)$ is the cardinality of the value collection of d, called rank of d. If we take the jth column as all the columns with the same decision value, we can get a special definition for the j-relative discernibility matrix, called decision relative matrix $M^d(\Lambda) = (c_{i,j}^d)$. Where $c_{i,j}^d = \begin{cases} empty, & if\ d(x_i) = d(x_j) \\ c_{i,j} - \{d\}, & otherwise \end{cases}$

Take the Sunburn example again, in Table 2.1, the objects are ordered according to their decision value. The corresponding discernability matrix is shown in Table 2.3:

Table 2.3: The decision-relative discernibility matrix

	Sarah	*Annie*	*Emily*	*Pete*	*John*	*Dana*	*Alex*	*Katie*
Sarah	Sunburned							
Annie								
Emily								
Pete	Ha,He,W	Ha,He,W	Ha,He	None				
John	Ha,W	Ha,He,W	Ha					
Dana	He,W,L	He,L	Ha,He,W,L					
Alex	Ha,He,W,L	Ha,L	Ha,He,W,L					
Katie	He,L	W,L	Ha,He,W,L					

In table 2.3, if the discernibility function is restricted to only run over the columns in one class, the result is reducts that can distinguish objects in this class from all other classes. The reduct is also {He, Ha, W} or {Ha, L}. In theory, the reducts obtained by running on classes are smaller in size than the reducts obtained by running on objects. With the reducts available, the rules could be generated, as in Table 2.4.

Table 2.4: Decision rules for Sunburn example

Rule 1	(hair=blonde)&(lotion=no)=>(sunburn=sunburned) 2
Rule 2	(hair=blonde)&(height=average)=>(sunburn=sunburned) 1
Rule 3	(height=average)&(weight=light)=>(sunburn=sunburned) 1
Rule 4	(hair=blonde)&(height=short)&(weight=average)=>(sunburn=sunburned) 1
Rule 5	(hair=red)=>(sunburn=sunburned) 1
Rule 6	(hair=brown)=>(sunburn=none) 3
Rule 7	(lotion=yes)=>(sunburn=none) 3
Rule 8	(height=short)&(weight=light)=>(sunburn=none) 1
Rule 9	(height=tall)=>(sunburn=none) 2

7. Rosetta system

Rosetta is a free software system that consists of a set of software components for discernibility-based tabular data analysis. It implements rough sets theory analysis. Usually, rough sets deal with classificatory analysis of tabular data. The data values are often discrete.

The goal of rough sets is to approximate the sets that cannot be described precisely by using the present attributes. There are several steps to finish one classification task:

1. Data formatting and preprocessing: to make sure the data can be loaded into Rosetta properly; we have to make data as flat, two-dimensional tables. For training data, the last column should be class label. Also all the missing values in the table should be dealt with, and there are several methods to do this.
2. Data transformation: Data discretization is often used on numerical attributes. It involves converting the exact observations into intervals or ranges of values. It results a coarser view of the world, and a reduction on the value set size for observations.
3. Generating rules: If-then rules are then generated by using the minimal attribute subsets, also called reducts.
4. Evaluation of rules: by classifying new objects using the above rules, we can evaluate the classificatory performance of the rules. After we get a set of good rules, we can apply them onto those objects without class knowledge, so new hypothesis can be generated.

CHAPTER 3 METHODS AND RESULTS ON YEAST CELL CYCLE DATA

1. Yeast cell cycle data

The yeast cell cycle experiment was conducted in 1998 [27] and the object was to find genes whose transcript levels vary periodically within the cell cycle. The experiment used DNA microarrays and the samples were from yeast cultures synchronized by three independent methods: factor arrest, elutriation, and arrest of a cdc15 temperature-sensitive mutant [27]. The yeast cell cycle data consists of 5 time course data sets. The measurements in each data set showed the expression ratios of more than 6,000 genes across around 20 time points. Previous research found that 800 genes were cell cycle regulated. However, around half of these genes have unknown functions.

2. Preprocessing

The data set used in this thesis was one of the five data sets. It had 24 time points and more than 6000 genes. Since the experiment was conducted on cDNA microarrays, the measurements in the data were ratios. There were a lot of missing values in the data, so preprocessing was focused on dealing with the missing values. The preprocessing could be stated as:

1. Check each gene across the 24 time points;
2. If the gene has no missing values, go to next gene;
3. If the gene has missing values,

a). If there is no three or more continuous non-missing values, this gene will be deleted;

b). If there is single missing value, linear interpolations will be implemented using the values of neighbors;

c). If there are two or more continuous missing values, the algorithm will standardize each gene profile to 0 mean and 1 variance, and find genes with the most similar profiles with this gene (correlation coefficient > 0.95). Then the missing values in this gene will be filled with the mean values of the similar genes. Resume the gene profile to original scale.

4. Repeat 1, 2, and 3 until there are no missing values over the data set.

3. Constructing training data

Among the 800 recognized cell cycle regulated genes, 36 genes were deleted during the preprocessing (See appendix A for the list of genes), leaving 764 genes. According to SGD's GO annotations, 444 out of 764 genes were annotated, and 320 genes were not annotated with Gene Ontology. The goal was to predict gene functions for those 320 genes using the information in the 444 genes. The data including these 444 genes was called training data. However this training data had no decision attributes (class labels) so far, so the decision attributes needed to be added to build a complete training data.

First, each gene was annotated with GO terms as specific as possible in the DAG (Directed Acyclic Graphs). In this problem, we need to find out the broader annotations so that we could construct the training data with limited number of classes. In the DAG graph, we moved up the GO tree from the specific terms used to annotate the genes in a list to find

GO parent terms that the genes have in common. The SGD Gene Ontology Slim Mapper tool [29] was used to annotate this list of 800 genes to their more general parent GO terms. The results are shown in Table 3.1.

Table 3.1: The results from GO Slim Mapper for 800 cell cycle regulated yeast genes

GO Terms	Number of Genes
Catalytic activity	252
Transporter activity	65
Transcriptor regulator activity	38
Structural molecule activity	35
Other	111
Molecular function unknown	319

According to the above results, we constructed training data set with class labels as catalytic activity (1), transporter activity (2), transcriptor regulator activity (3), structural molecule activity (4), other (5), and molecular function unknown, where "other" class consists of several GO terms with small number of genes. Table 3.2 lists the class labels and relative information. The goal was to predict molecular functions for genes in 'molecular function unknown' class.

Table 3.2: Class label information

Class labels	Meaning of class labels	Class label codes
Catact	Catalytic activity	1
Tranpact	Transporter activity	2
Tranregact	Transcriptor regulator activity	3
Strmolact	Structural molecule activity	4
Other	Other	5

4. Classifier training

To predict gene functions for 'molecular function unknown' class, we trained a classifier using training data in Rosetta system.

After the training data set was loaded into Rosetta system, we randomly split the training data into two sets: training set and testing set with split factor = 0.6. So the training set contained 80% (355 genes) of the training data, and the testing set contained 20% (89 genes) of the training data. However, at this point the data was still numeric, so we needed to discretize data. The equal frequency bin method was applied to discretize the training data, and then the testing data was discretized using the same cuts.

Genetic algorithms were applied to get the minimal attribute set, which was also called reducts. Table 3.3 shows the 10 reducts from training set.

Table 3.3: Reducts from training set

	Reducts
1	{ cdc15.10.min, cdc15.50.min, cdc15.90.min, cdc15..120.min, cdc15..160.min, cdc15..190.min, cdc15..220.min, cdc15..250.min }
2	{ cdc15.10.min, cdc15.50.min, cdc15.80.min, cdc15.90.min, cdc15..130.min, cdc15..160.min, cdc15..190.min, cdc15..240.min }
3	{ cdc15.10.min, cdc15.50.min, cdc15.70.min, cdc15.80.min, cdc15.90.min, cdc15..130.min, cdc15..160.min, cdc15..190.min }
4	{ cdc15.10.min, cdc15..30.min, cdc15.50.min, cdc15.90.min, cdc15..130.min, cdc15..150.min, cdc15..190.min, cdc15..230.min, cdc15..240.min }
5	{ cdc15.10.min, cdc15..30.min, cdc15.50.min, cdc15.90.min, cdc15..110.min, cdc15..130.min, cdc15..190.min, cdc15..230.min, cdc15..240.min }
6	{ cdc15.10.min, cdc15..30.min, cdc15.90.min, cdc15..110.min, cdc15..130.min, cdc15..170.min, cdc15..190.min, cdc15..230.min, cdc15..240.min }
7	{ cdc15.10.min, cdc15..30.min, cdc15.50.min, cdc15..130.min, cdc15..160.min, cdc15..170.min, cdc15..190.min, cdc15..210.min, cdc15..240.min }
8	{ cdc15.10.min, cdc15.50.min, cdc15.80.min, cdc15.90.min, cdc15..130.min, cdc15..190.min, cdc15..200.min, cdc15..220.min, cdc15..240.min }
9	{ cdc15..30.min, cdc15.50.min, cdc15.70.min, cdc15..120.min, cdc15..140.min, cdc15..160.min, cdc15..220.min, cdc15..230.min, cdc15..250.min } 1.0
10	{ cdc15.10.min, cdc15.50.min, cdc15.90.min, cdc15..120.min, cdc15..160.min, cdc15..190.min, cdc15..230.min, cdc15..250.min }

Based on these reducts, classification rules were generated. See Table 3.4 to find the examples. There were totally 4510 rules generated based on training set.

Table 3.4: Examples of rules

Rule 1	*(cdc15..30.min="(-Inf,-1.005)")&(cdc15.70.min="(-Inf,-0.7250)")=>(class.label=1[7])*
Rule2	*(cdc15.10.min="(-1.115,-0.595)")&(cdc15..210.min="(-Inf,-0.325)")&(cdc15..250.min="(-0.11,0.225)")=>(class.label=4[1])*

The generated rules then were applied to the testing set and the classification results were obtained. The confusion matrix is shown in Table 3.5.

Table 3.5: Confusion matrix

	Predicted					
Actual		*Catact*	*tranpact*	*tranregact*	*strmolact*	*Other*
	catact	20	5	2	1	2
	tranpact	2	15	1	2	1
	tranregact	2	0	6	0	2
	strmolact	1	1	0	8	0
	Other	3	0	0	3	11
	True Positive Rate	0.714	0.714	0.67	0.57	0.69

Using the trained classifier, the molecular functions of the unknown genes were predicted. See results in Appendix B.

CHAPTER 4 RESULTS ON ARABIDOPSIS DATA

1. Experiment and Data

The data used here is from experiment "Gene expression and carbohydrate metabolism through the diurnal cycle". The object of this experiment was to gain insight into the synthesis and functions of enzymes of starch metabolism in leaves of Arabidopsis. Affymetrix GeneChips were used to analyze the transcriptase throughout the diurnal cycle. This experiment is a courtesy of the Nottingham Arabidopsis Stock Center's microarray database. The data was obtained from NASC's Affymetrix service and offered for public access at BarleyBase [32].

The experiment has involved sampling leaves at eleven different time points as follows: 0, 1, 2, 4, 8, 12, 13, 14, 16, 20, and 24 h (where time 0 is the onset of dark and 12 h is the onset of light, h means hours). The 24 h time point is a repeat of 0 h. Two biological replicates were used. The raw data indicated the expression values of 22,840 genes on 22 chips. And data was normalized using RMA (Robust Microarray Analysis).

2. Preprocessing

First, ANOVA test was applied to find out genes that had changed during diurnal cycle. 777 genes were found with p value < 0.01. Then for each time point, average of two replicates was kept for the convenience of computation. So now, the data contains 777 rows (genes) and 11 columns (time points).

3. Extract class labels from GO annotations

According to the Arabidopsis annotation from TAIR, 444 out of the 777 genes are annotated with Gene Ontology (GO). In total, 1934 GO ids were used to annotate 444 genes, because one gene may have annotation from different categories, and sometimes, there are more than one GO ids in one categories. Among 1934 GO ids, 735 are from the category of biological process and the number of unique GO ids in these 735 GO ids is 202. The annotation also includes the evidence code that shows the source of annotation. For example, TAS means traceable author statement, and IEA means inferred from electronic annotation, for more details, see Table 4.1. Different evidence codes indicate different levels of reliability of the annotation. Among these types of evidence codes, some are reliable, and some are weaker. According to their reliability, we gave different weights to different evidence codes in Table 4.1.

Table 4.1: Evidence codes and their weights[33]

EVIDENCE CODE	MEANING OF THE EVIDENCE CODE	EVIDENCE WEIGHT
IDA	Inferred from direct assay	1.0
TAS	Traceable author statement	1.0
IMP	Inferred from mutant phenotype	0.9
IGI	Inferred from genetic interaction	0.9
IPI	Inferred from physical interaction	0.9
IEP	Inferred from expression pattern	0.8
ISS	Inferred from structural similarity	0.8
NAS	Non-traceable author statement	0.7
IEA	Inferred from electronic annotation	0.6
Other		0.5

For each unique GO term, the weight of GO term can be calculated as $W_{GO} = \sum_{i=1}^{K} w_{evi}(i)$, where K is the total occurrence of this GO term.

4. Build DAGs

Having the weight information, we took all the GO terms in the annotations of 202 unique GO ids as leaves to build the directed acyclic graphs (DAGs) for this cluster the ontology of biological process (BP). The weight of each node is computed by adding up the weights of its children. So the closer is the node to the root, the bigger weight it has.

At last, by the aid of FCModeler [26], we can view the DAGs for this list of 202 GO terms below. Since the graph is very big, only a small part of the DAGs is showed here. We are interested in those nodes with big weights and are far from the root, since those nodes will represent the meaning of the graph with reasonable specifics. Figure 4.1 illustrates the structure of DAG.

The depth of a GO term refers to the distance between this GO term and the root, which is 'GO: 0003674 : molecular_function'. For instance, in Figure 4.2, 'GO:0016209 : antioxidant activity' is located the second layer of DAG graph, its distance to the root is 1 layer, so its depth is 1 and 'GO:0004362' has the depth of 2. The depth information of all the GO terms can be found in this way.

Besides the evidence weights and depth information, the frequency of use for a certain GO term is also important. For each unique GO id, which occurs in the annotations, we can count its occurrence as the summation of occurrence of all its children.

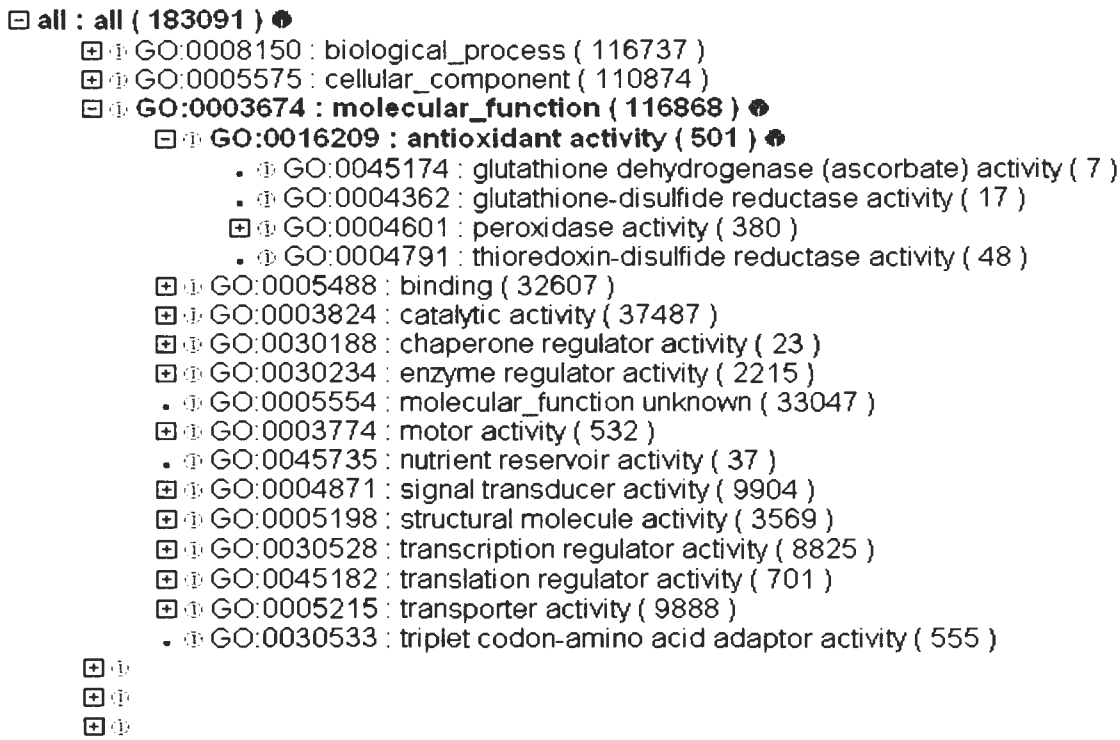

Figure 4.1 part of DAG graph (from GO browser AmiGO)

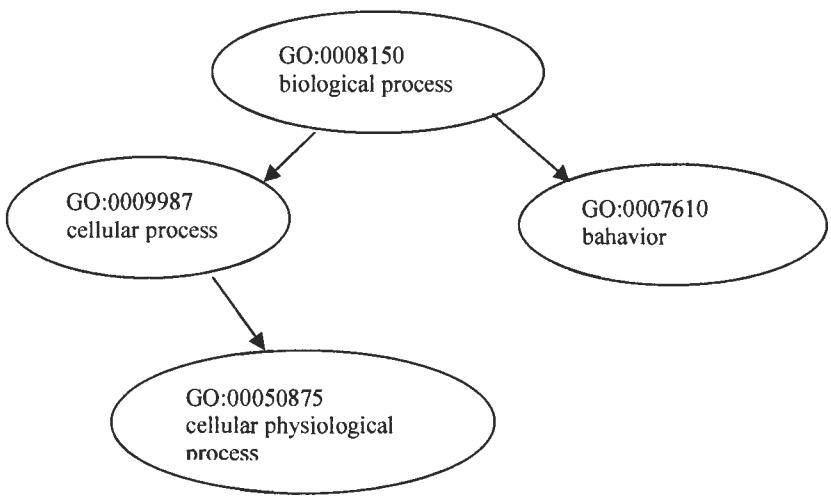

Figure 4.2: An example of GO depth

Thus we have the BP annotations for this list of genes and also we have the weights, depth and usages information. Next step is to determine how many classes we will have for training set and what they are.

To choose class labels from thesis GO id, we need to see the weight information and usage information. The criteria to choose class label is: choose GO ids with depth 3, with big weight and usages values. According to this criterion, 4 GO ids were chosen as class labels, see Table 4.2.

With the class labels available, we labeled each gene to one or more classes, according to the relationship of its GO term and the class label GO term in the DAG graph. So the training set is available. Then rough sets theory will be applied to build a classifier using these training data in Rosetta system [25].

Table 4.2 Class labels

GO id	GO term	Class label abbreviation	Number of Children	Weight
GO:0009058	biosynthesis	BioSyn	18	27.2
GO:0008151	cell grow or/and maintainence	CelGrow	34	52.1
GO:0005975	carbohydrate metabolism	CarbMet	37	61.3
GO:0006629	lipid metabolism	LipMet	11	37

5. Discretization

After the training data set was loaded into Rosetta, we randomly split the training data into two sets: training set and testing set. The split factor was set as 0.8. So the training set contains 668 objects and testing set contains 167 objects. However, at this point the data is still numeric, so we need to discretize the data. The equal frequency bin method was used to discretize the training data, and then testing data was discritized using the same methods.

6. Reduct Computation and Rule Synthesis

Genetic algorithm then was applied to get the minimal attribute subsets, which is also called reduct. Rules were generated using the reduct. After applying rules generated from training set onto testing set, we got the classifier performance, which is shown in Table 4.3.

Table 4.3 Confusion matrix

	Predicted				
		CarbMet	*BioSyn*	*CelGrow*	*LipMet*
Actual	*CarbMet*	6	1	0	2
	BioSyn	3	3	0	3
	CelGrow	0	3	18	0
	LipMet	0	0	0	3
	True Positive Rate	0.67	0.43	1	0.38

In Table 4.3, CarbMet means carbohydrate metabolism, BioSyn means biosynthesis, CelGrow means cell growth and maintenance, and LipMet means lipid metabolism. The results show the classifier can classify class CelGrow correctly, and it cannot classify LipMet from others very well. We can also notice that 3 of 9 CarbMet genes were classified as BioSyn genes. It probably is because that there are some overlaps between these two classes.

7. Analysis of results

After the classifier was trained, we used the classifier to classify new objects, which had no GO annotation for biological processes. Part of the results is shown in Table 4.4 for illustration and the complete list is located in Appendix C.

Since the unknown genes had no annotation for biological process, it is hard to validate the prediction results. However, we might find some clues from other sources, such as GO annotations for molecular functions, or pathway information. Using AtGeneSearch tool in MetNet Exchange [34], some of the results got verified, see Table 4.5.

Table 4.4 Part of the prediction results

Probe Set Name	Locus Name	Predicted biological process
267305_at	At2g30070	BioSyn
250692_at	At5g06560	BioSyn
264265_at	At1g09280	BioSyn
266830_at	At2g22810	BioSyn
249472_at	At5g39210	BioSyn
258244_at	At3g27770	CarbMet
256225_at	At1g56220	CarbMet
255325_at	At4g04210	CarbMet
250232_at	At5g13950	CarbMet
247178_at	At5g65205	CelGrow
246234_at	At4g37280	CelGrow
256173_at	At1g51730	CelGrow
249728_at	At5g24390	CelGrow
255737_at	At1g25420	CelGrow
250215_at	At5g14080	LipMet
266420_at	At2g38610	LipMet
266483_at	At2g47910	LipMet
255874_at	At2g40550	LipMet
256216_at	At1g56340	LipMet
267392_at	At2g44490	LipMet
259880_at	At1g76730	LipMet
261254_at	At1g05805	LipMet

Table 4.5 Prediction results validation

Locus_ID	Tair_annotation	Go_Biological Process	Predicted biological process
At5g17990	"Encodes the tryptophan biosynthetic enzyme phosphoribosylanthranilate transferase (PAT1, called trpD in bacteria). Converts anthranilate and phosphoribosylpyrophosphate into phosphoribosylanthranilate and inorganic pyrophosphate."	tryptophan biosynthesis	Biosynthesis
At5g01270	"double-stranded RNA-binding domain (DsRBD)-containing protein, contains Pfam profile PF00035: Double-stranded RNA binding motif"	protein complex assembly	Biosynthesis
At3g10230	"lycopene cyclase (LYC) mRNA, complete cds"	carotene biosynthesis	Biosynthesis

Table 4.5 (continued)

Locus_ID	Tair_annotation	Go_Biological Process	Predicted biological process
At5g42100	"glycosyl hydrolase family 17 protein, similar to beta-1,3-glucanase precursor GI:4097948 from (Oryza sativa)"	carbohydrate metabolism proteolysis and peptidolysis	Biosynthesis
At2g22810	"1-aminocyclopropane-1-carboxylate synthase 4 / ACC synthase 4 (ACS4), identical to gi:940370 (GB:U23481)"	ethylene biosynthesis response to auxin stimulus	Biosynthesis
At3g12290	"tetrahydrofolate dehydrogenase/cyclohydrolase, putative, similar to SP|P07245 C-1-tetrahydrofolate synthase, cytoplasmic (C1-THF synthase) (Includes: Methylenetetrahydrofolate dehydrogenase (EC 1.5.1.5); Methenyltetrahydrofolate cyclohydrolase (EC 3.5.4.9"	folic acid and derivative biosynthesis	Biosynthesis
At2g20670	"expressed protein, contains Pfam profile PF04720: Protein of unknown function (DUF506)"	phosphoenolpyruvate-dependent sugar phosphotransferase system	Cell growth and maintainence
At5g57550	xyloglucan endotransglycosylase-related protein (XTR3)	cell wall biosynthesis (sensu Magnoliophyta)	Cell growth and maintainence
At4g30850	"expressed protein, contains Pfam domain, PF03006: Uncharacterised protein family (Hly-III / UPF0073)"	cytolysis	Cell growth and maintainence
At1g75780	beta tubulin gene downregulated by phytochrome A (phyA)-mediated far-red light high-irradiance and the phytochrome B (phyB)-mediated red light high-irradiance responses	cell elongation response to light	Cell growth and maintainence
At5g15930	Encodes a putative plant adhesion molecule.	biological_process unknown	Cell growth and maintainence

Table 4.5 (continued)

Locus_ID	Tair_annotation	Go_Biological Process	Predicted biological process
At2g44670	"senescence-associated protein-related, similar to senescence-associated protein SAG102 (GI:22331931) (Arabidopsis thaliana);"	biological_process unknown	Cell growth and maintainence
At2g04850	"auxin-responsive protein-related, related to auxin-induced protein AIR12 GI:11357190 (Arabidopsis thaliana)"	electron transport	Cell growth and maintainence
At2g36250	"plastid division protein FtsZ mRNA, complete cds"	cell cycle	Cell growth and maintainence
At1g41830	"multi-copper oxidase type I family protein, similar to pollen-specific BP10 protein (SP|Q00624)(Brassica napus); contains Pfam profile: PF00394 Multicopper oxidase"	male gametophyte development microspore germination	Cell growth and maintainence
At5g49720	radial swelling mutant shown to be specifically impaired in cellulose production	cellulose biosynthesis cell elongation	Cell growth and maintainence
At2g35860	"beta-Ig-H3 domain-containing protein / fasciclin domain-containing protein, contains Pfam profile PF02469: Fasciclin domain"	cell adhesion N-terminal protein myristoylation	Cell growth and maintainence
At1g78620	"integral membrane family protein, contains Pfam domain PF01940: Integral membrane protein"	phospholipid biosynthesis	Lipid Metabolism

CHAPTER 5 CONCLUSIONS

There are several aspects of this thesis that influenced to get better classification performance.

First, the method used to build training data was based only on the Gene Ontology. For yeast data set, we used SGD Gene Ontology Slim Mapper [29] to find out the general GO terms for genes with GO annotation. We chose several terms as class labels according to how many genes it covered. Then five classes were built. However, it was not complete to have these five classes, since it is possible that the unknown genes have none of the functions described in these five classes. Also the class 'other' was too broad to define a class, and it was another reason that we didn't get the good results even for training data. The unbalanced class size might also affect the results. For example, in yeast data set, the class 'catalytic activity' contained 252 genes, however, the class 'structural molecule' contained 35 genes only.

Secondly, rough sets theory mainly deals with discrete data, and in general, gene expression data is continuous numeric data. It is necessary to discretize the data before we load the data into Rosetta system. However, the process of discretizing data will result in information loss inevitably [17].

The application of Gene Ontology and rough sets theory in predicting gene functions or biological processes is a relatively new area in gene expression analysis area. The results in this thesis show the feasibility of this technique in more experiments. The other advantage of this technique is that it is not based on the assumption that genes with similar expression profiles share the same or similar functions or processes.

REFERENCES

[1] MB Eisen, PT Spellman, PO Brown, D Botstein: Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA, 95:14863-14868, 1998.

[2] Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nature Genetics, 25(1): 25-29, 2000.

[3] GO-EBI, EMBL-EBI: The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res. 2004 January 1; 32(Database issue): D258–D261.

[4] Javier Herrero, Fátima Al-Shahrour, Ramón Díaz-Uriarte, Álvaro Mateos, Juan M. Vaquerizas, Javier Santoyo and Joaquín Dopazo: GEPAS: A web-based resource for microarray gene expression data analysis. Nucleic Acids Research, 2003, Vol. 31, No. 13 3461-3467.

[5] Al-Shahrour F, Diaz-Uriarte R, Dopazo J. FatiGO: A web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics. 2004 Mar 1; 20(4): 578-80. Epub 2004 Jan 22.

[6] Khan S, Situ G, Decker K, Schmidt CJ: GoFigure: automated Gene Ontology annotation Bioinformatics. 2003 Dec 12; 19(18): 2484-5.

[7] Herman Midelfart, astrid Lagreid, and Jan Komorowski: Classification of Gene Expression Data in an Ontology. Proc. of the Second International Symposium on Medical Data Analysis (ISMDA-2001), LNCS 2199, pages 186-194. © Springer-Verlag, 2001.

[8] Lagreid A, Hvidsten TR, Midelfart H, Komorowski J, Sandvik AK: Predicting gene ontology biological process from temporal gene expression patterns. Genome Res. 2003 May;13(5):965-79. Epub 2003 Apr 14.

[9] Lee SG, Hur JU, Kim YS: A graph-theoretic modeling on GO space for biological interpretation of gene clusters. Bioinformatics. 2004 Feb 12; 20(3): 381-8. Epub 2004 Jan 22.

[10] Hvidsten TR, Laegreid A, Komorowski J: Learning rule-based models of biological process from gene expression time profiles using gene ontology. Bioinformatics. 2003 Jun 12; 19(9): 1116-23.

[11] Hvidsten TR, Komorowski J, Sandvik AK, Laegreid A: Predicting gene function from gene expressions and ontologies. Pac Symp Biocomput. 2001; 299-310.

[12] Liviu Badea: Functional discrimination of gene expression patterns in terms of the gene ontology. Pac Symp Biocomput. 2003; 565-76.

[13] P.W.Lord, R.D.Stevens, A.Brass and G.A.Goble: Semantic Similarity Measures as Tools for Exploring the Gene Ontology. Pac Symp Biocomput. 2003; 601-12.

[14] Jerzy Stefanowski, and Alexis Tsoukias: Valued Tolerance and Decision Rules. Accessed from http://l1.lamsade.dauphine.fr/~tsoukias/papers/banffjsat.pdf on Jan 2005.

[15] Introduction to rough sets theory. Accessed from http://www.kbs.twi.tudelft.nl/Education/Cyberles/Trondheim/Rough_sets/html/rs_th_01introd.html on Oct 2004.

[16] Jan Komorowski, Adzistaw Pawlak, Lech Polkowski, Andrzej Skowron: Rough Sets: A Tutorial. In: S.K. Pal and A. Skowron (Eds.), Rough-Fuzzy Hybridization: A New Trend in Decision-Making, Springer-Verlag, Singapore, 3-9.

[17] Shirley Hui: Rough Set Classification of Gene Expression Data. Accessed from http://www.cs.uwaterloo.ca/~s2hui/RoughSetProject.pdf on Oct 2004.

[18] Ben-Dor A, Shamir R, Yakhini Z: Clustering gene expression patterns. J Comput Biol. 1999 Fall-Winter; 6(3-4): 281-97.

[19] Balasubramaniyan R, Hullermeier E, Weskamp N, Kamper J: Clustering of gene expression data using a local shape-based similarity measure. Bioinformatics. 2004 Oct 28.

[20] Wang M, Yang J, Liu GP, Xu ZJ, Chou KC: Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. Protein Eng Des Sel. 2004 Jun; 17(6): 509-516. Epub 2004 Aug 16.

[21] Mitra P, Murthy CA, Pal SK: A probabilistic active support vector learning algorithm. IEEE Trans Pattern Anal Mach Intell. 2004 Mar; 26(3): 413-8.

[22] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares Jr., and David Haussler: Knowledge-based analysis of microarray gene expression data by using support vector machines. Genetics, Vol. 97, Issue 1, 262-267, January 4, 2000.

[23] Gene ontology browser. Accessed from http://www.godatabase.org/cgi-bin/go.cgi on Jun 2004.

[24] Bioinformatics R packages. Accessed from http://www.bioconductor.org on Jan 2004.

[25] Aleksander Ohrn: Discernibility and Rough Sets in Medicine: Tools and Applications, PhD thesis, Department of Computer and Information Science, Norwegian

University of Science and Technology, Trondheim, Norway. NTNU report 1999:133, IDI report 1999:14, ISBN 82-7984-014-1. 239 pages.

[26] Fuzzy Cognitive Map tool, FCModeler. Accessed from http://clue.eng.iastate.edu/~julied/research/fcmodeler/index.html on Jun 2004.

[27] Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwannath R. Iyer, Kirk Anders, Michael B. Eisen, Partrick O. Brown, David Botstein, and Bruce Futcher: Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. Molecular Biology of the Cell 9, 3273-3297, 1998.

[28] SGD: Saccharomyces Genome Database. Accessed from http://www.yeastgenome.org on Jan 2005.

[29] Gene ontology mapping tool: SGD GO Slim Mapper. Accessed from http://db.yeastgenome.org/cgi-bin/GO/goTermMapper on Jan 2005.

[30] Gene ontology tool for finding important GO term in a list of genes. GO Term Finder. Accessed from http://www.yeastgenome.org/help/goTermFinder.html on Nov 2004.

[31] Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G: GO::TermFinder--open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Vol. 20 no. 18 2004, pages 3710–3715, bioinformatics.

[32] Lishuang Shen, Jian Gong, Rico A. Caldo, Dan Nettleton, Dianne Cook, Roger P. Wise and Julie A. Dickerson: BarleyBase—an expression profiling database for plant genomics. Nucleic Acids Research, 2005, Vol. 33, Database issue D614-D618.

[33] Du P, Gong J, Wurtele ES, Dickerson JA: Modeling Gene Expression Networks using Fuzzy Logic. August, 2004, SMC, IEEE Transaction.

[34] Metabolic Network Exchange. Accessed from http://www.public.iastate.edu/~mash/MetNet/homepage.html on April 2005.

APPENDIX

A: Genes chopped by preprocessing

1 YAL032C
2 YAL043C-A
3 YAL048C
4 YAR040C
5 YAR043C
6 YAR044W
7 YBL004W
8 YBL005W
9 YBL005W-B
10 YBL007C
11 YBL027W
12 YBL051C
13 YBL104C
14 YBR016W
15 YBR017C
16 YBR030W
17 YBR031W
18 YBR044C
19 YBR157C
20 YBR164C
21 YBR169C
22 YBR170C
23 YBR172C
24 YBR289W
25 YCL063W
26 YCR023C
27 YCR067C
28 YCR069W
29 YCR079W
30 YCR083W
31 YCR094W
32 YCR103C
33 YDL019C
34 YDL031W
35 YDL032W
36 YDL183C

B: Gene function prediction results for yeast cell cycle data

Table 3.6: Gene function prediction results for unknown genes

Gene Names	Predicted functions
YAL053W	catalytic activity
YBL009W	catalytic activity
YBL100C	transporter activity

YBL113C	catalytic activity
YBR007C	catalytic activity
YBR054W	catalytic activity
YBR071W	structural molecule activity
YBR086C	transporter activity
YBR089W	catalytic activity
YBR094W	other
YBR108W	catalytic activity
YBR138C	other
YBR157C	other
YBR204C	catalytic activity
YBR242W	transcriptor activity
YBR273C	catalytic activity
YBR287W	transporter activity
YCL012W	structural molecule activity
YCL013W	structural molecule activity
YCL014W	structural molecule activity
YCL022C	other
YCL023C	catalytic activity
YCL027W	structural molecule activity
YCL038C	transporter activity
YCL042W	catalytic activity
YCL060C	other
YCL061C	structural molecule activity
YCL062W	catalytic activity
YCL065W	structural molecule activity
YCLX09W	transporter activity
YCR018C	other
YCR041W	structural molecule activity
YDL003W	transcriptor activity
YDL018C	catalytic activity
YDL037C	catalytic activity
YDL039C	transcriptor activity
YDL048C	transcriptor activity
YDL089W	structural molecule activity
YDL096C	transporter activity
YDL105W	catalytic activity
YDL117W	transcriptor activity
YDL156W	transporter activity
YDL157C	catalytic activity
YDL169C	transcriptor activity
YDL180W	transporter activity
YDL211C	catalytic activity
YDR029W	transcriptor activity
YDR033W	catalytic activity
YDR053W	catalytic activity
YDR055W	structural molecule activity
YDR089W	catalytic activity
YDR157W	transporter activity
YDR276C	transcriptor activity
YDR307W	catalytic activity
YDR346C	catalytic activity

YDR528W	catalytic activity
YEL017W	catalytic activity
YEL025C	transporter activity
YEL040W	catalytic activity
YEL047C	catalytic activity
YEL068C	transcriptor activity
YEL075C	catalytic activity
YEL076C	catalytic activity
YEL076C-A	catalytic activity
YER032W	structural molecule activity
YER124C	catalytic activity
YER150W	transporter activity
YER152C	catalytic activity
YER189W	catalytic activity
YFL006W	structural molecule activity
YFL044C	transporter activity
YFL060C	catalytic activity
YFL064C	catalytic activity
YFL067W	catalytic activity
YFR039C	structural molecule activity
YGL060W	catalytic activity
YGL101W	catalytic activity
YGL185C	transcriptor activity
YGL195W	transcriptor activity
YGL200C	catalytic activity
YGR035C	other
YGR041W	catalytic activity
YGR042W	catalytic activity
YGR146C	transcriptor activity
YGR151C	catalytic activity
YGR153W	transporter activity
YGR176W	catalytic activity
YGR189C	catalytic activity
YGR219W	catalytic activity
YGR221C	other
YGR230W	transcriptor activity
YGR234W	catalytic activity
YGR238C	catalytic activity
YGR259C	transporter activity
YGR284C	transporter activity
YHL026C	transporter activity
YHL049C	catalytic activity
YHL050C	catalytic activity
YHR022C	other
YHR029C	catalytic activity
YHR098C	catalytic activity
YHR108W	transporter activity
YHR110W	catalytic activity
YHR113W	catalytic activity
YHR127W	catalytic activity
YHR149C	catalytic activity
YHR151C	other

YHR159W	other
YHR218W	catalytic activity
YHR219W	catalytic activity
YIL011W	catalytic activity
YIL025C	transporter activity
YIL056W	catalytic activity
YIL076W	other
YIL104C	transcriptor activity
YIL117C	catalytic activity
YIL123W	other
YIL129C	other
YIL132C	other
YIL135C	structural molecule activity
YIL140W	catalytic activity
YIL141W	catalytic activity
YIL158W	transporter activity
YIL168W	structural molecule activity
YIL177C	catalytic activity
YIR010W	structural molecule activity
YIR036C	transporter activity
YJL015C	catalytic activity
YJL018W	transcriptor activity
YJL019W	catalytic activity
YJL051W	other
YJL067W	transporter activity
YJL078C	transcriptor activity
YJL079C	catalytic activity
YJL091C	catalytic activity
YJL099W	transcriptor activity
YJL118W	other
YJL119C	structural molecule activity
YJL181W	transporter activity
YJL195C	structural molecule activity
YJL201W	catalytic activity
YJL217W	catalytic activity
YJL225C	catalytic activity
YJR003C	transcriptor activity
YJR030C	catalytic activity
YJR054W	catalytic activity
YJR154W	catalytic activity
YKL044W	transporter activity
YKL065C	catalytic activity
YKL066W	transcriptor activity
YKL069W	transporter activity
YKL108W	catalytic activity
YKL151C	transporter activity
YKL165C	catalytic activity
YKL172W	transcriptor activity
YKL177W	structural molecule activity
YKL183W	transcriptor activity
YKR010C	structural molecule activity
YKR012C	catalytic activity

YKR041W	structural molecule activity
YKR042W	other
YKR046C	transporter activity
YKR077W	catalytic activity
YKR091W	catalytic activity
YLL002W	catalytic activity
YLL012W	catalytic activity
YLL022C	catalytic activity
YLL032C	structural molecule activity
YLL066C	catalytic activity
YLL067C	catalytic activity
YLR040C	transporter activity
YLR041W	transporter activity
YLR049C	other
YLR050C	catalytic activity
YLR057W	transcriptor activity
YLR084C	structural molecule activity
YLR099C	structural molecule activity
YLR169W	transporter activity
YLR190W	other
YLR194C	catalytic activity
YLR225C	structural molecule activity
YLR235C	transporter activity
YLR236C	catalytic activity
YLR254C	other
YLR297W	transporter activity
YLR302C	catalytic activity
YLR326W	transcriptor activity
YLR353W	other
YLR373C	structural molecule activity
YLR383W	catalytic activity
YLR413W	transporter activity
YLR437C	structural molecule activity
YLR455W	other
YLR457C	transcriptor activity
YLR458W	catalytic activity
YLR462W	catalytic activity
YLR463C	catalytic activity
YLR464W	catalytic activity
YLR465C	structural molecule activity
YML012W	catalytic activity
YML020W	catalytic activity
YML033W	transcriptor activity
YML034W	catalytic activity
YML050W	other
YML052W	transporter activity
YML066C	transporter activity
YML102W	catalytic activity
YML109W	catalytic activity
YML119W	transcriptor activity
YML125C	other
YML133C	catalytic activity

YMR031C	catalytic activity
YMR048W	catalytic activity
YMR078C	catalytic activity
YMR144W	structural molecule activity
YMR163C	structural molecule activity
YMR179W	catalytic activity
YMR238W	other
YMR253C	catalytic activity
YMR278W	transcriptor activity
YNL043C	structural molecule activity
YNL056W	catalytic activity
YNL057W	other
YNL058C	transcriptor activity
YNL078W	structural molecule activity
YNL134C	catalytic activity
YNL160W	catalytic activity
YNL165W	structural molecule activity
YNL171C	structural molecule activity
YNL173C	catalytic activity
YNL176C	other
YNL181W	structural molecule activity
YNL208W	catalytic activity
YNL263C	catalytic activity
YNL273W	transcriptor activity
YNL276C	structural molecule activity
YNL300W	other
YNL326C	transporter activity
YNR009W	other
YNR066C	catalytic activity
YOL007C	transcriptor activity
YOL017W	catalytic activity
YOL019W	catalytic activity
YOL034W	catalytic activity
YOL070C	other
YOL114C	transporter activity
YOL150C	structural molecule activity
YOR018W	catalytic activity
YOR023C	other
YOR052C	structural molecule activity
YOR066W	catalytic activity
YOR073W	catalytic activity
YOR084W	catalytic activity
YOR104W	other
YOR105W	structural molecule activity
YOR114W	catalytic activity
YOR115C	catalytic activity
YOR129C	transporter activity
YOR144C	transporter activity
YOR152C	catalytic activity
YOR188W	other
YOR195W	catalytic activity
YOR235W	structural molecule activity

YOR248W	transcriptor activity
YOR256C	other
YOR258W	catalytic activity
YOR263C	transporter activity
YOR264W	transporter activity
YOR283W	catalytic activity
YOR307C	other
YOR313C	transporter activity
YOR314W	transporter activity
YOR315W	catalytic activity
YOR324C	other
YOR342C	other
YOR355W	transcriptor activity
YOR383C	transporter activity
YPL014W	transporter activity
YPL021W	transporter activity
YPL025C	transporter activity
YPL032C	catalytic activity
YPL054W	catalytic activity
YPL095C	transporter activity
YPL141C	transporter activity
YPL158C	other
YPL163C	other
YPL208W	structural molecule activity
YPL221W	catalytic activity
YPL241C	catalytic activity
YPL250C	structural molecule activity
YPL264C	transporter activity
YPL267W	catalytic activity
YPL269W	catalytic activity
YPR013C	catalytic activity
YPR018W	catalytic activity
YPR045C	transporter activity
YPR075C	other
YPR076W	catalytic activity
YPR149W	transporter activity
YPR155C	catalytic activity
YPR157W	transporter activity
YPR174C	catalytic activity
YPR202W	catalytic activity
YPR203W	catalytic activity
YPR204W	catalytic activity
YEL077C	catalytic activity

C: Biological process prediction results for Arabidopsis diurnal data

Locus ID	Predicted biological process

At5g20030	Biosynthesis
At1g65420	Biosynthesis
At3g18240	Biosynthesis
At5g17990	Biosynthesis
At4g30790	Biosynthesis
At3g11020	Biosynthesis
At5g01270	Biosynthesis
At4g18210	Biosynthesis
At3g01350	Biosynthesis
At3g10230	Biosynthesis
At3g07560	Biosynthesis
At1g57850	Biosynthesis
At5g63130	Biosynthesis
At2g45990	Biosynthesis
At5g12470	Biosynthesis
At1g03380	Biosynthesis
At4g30500	Biosynthesis
At5g42100	Biosynthesis
At2g16850	Biosynthesis
At1g48040	Biosynthesis
At2g32100	Biosynthesis
At2g30070	Biosynthesis
At5g06560	Biosynthesis
At1g09280	Biosynthesis
At2g22810	Biosynthesis
At5g39210	Biosynthesis
At1g01240	Biosynthesis
At5g41940	Biosynthesis
At5g47455	Biosynthesis
At3g12290	Biosynthesis
At3g61880	Biosynthesis
At1g02410	Biosynthesis
At2g36430	Biosynthesis
At2g02420	Biosynthesis
At1g21660	Biosynthesis
At1g15740	Biosynthesis
At5g54980	Biosynthesis
At3g04670	Biosynthesis
At1g19140	Biosynthesis
At1g72700	Biosynthesis
At3g49220	Biosynthesis
At1g18570	Biosynthesis
At2g45620	Biosynthesis
At4g14605	Biosynthesis
At1g02300	Biosynthesis
At5g39760	Biosynthesis
At5g46280	Biosynthesis
At1g73390	Biosynthesis
At5g04220	Biosynthesis
At5g50460	Biosynthesis
At5g06770	Biosynthesis
At2g25070	Biosynthesis
At1g06110	Biosynthesis
At5g20070	Biosynthesis

At5g11810	Biosynthesis
At5g56000	Biosynthesis
At1g67070	Biosynthesis
At4g02920	Biosynthesis
At1g68820	Biosynthesis
At5g16990	Biosynthesis
At3g27770	Carbohydrate Metabolism
At1g56220	Carbohydrate Metabolism
At4g04210	Carbohydrate Metabolism
At5g13950	Carbohydrate Metabolism
At1g49780	Carbohydrate Metabolism
At5g35330	Carbohydrate Metabolism
At1g17100	Carbohydrate Metabolism
At4g15140	Carbohydrate Metabolism
At3g16850	Carbohydrate Metabolism
At4g25500	Carbohydrate Metabolism
At5g49410	Carbohydrate Metabolism
At5g43080	Carbohydrate Metabolism
At1g01560	Carbohydrate Metabolism
At1g52550	Carbohydrate Metabolism
At1g28600	Carbohydrate Metabolism
At5g42820	Carbohydrate Metabolism
At1g52230	Carbohydrate Metabolism
At5g02830	Carbohydrate Metabolism
At5g23290	Carbohydrate Metabolism
At5g50450	Carbohydrate Metabolism
At1g79460	Carbohydrate Metabolism
At3g46940	Carbohydrate Metabolism
At3g18980	Carbohydrate Metabolism
At3g27420	Carbohydrate Metabolism
At2g20670	Carbohydrate Metabolism
At5g57550	Carbohydrate Metabolism
At5g25240	Carbohydrate Metabolism
At5g44110	Carbohydrate Metabolism
At5g03290	Carbohydrate Metabolism
At5g67300	Carbohydrate Metabolism
At1g59830	Carbohydrate Metabolism
At3g16350	Carbohydrate Metabolism
At2g15890	Carbohydrate Metabolism
At1g05210	Carbohydrate Metabolism
At4g21940	Carbohydrate Metabolism
At2g20740	Carbohydrate Metabolism
At2g20130	Carbohydrate Metabolism
At5g18280	Carbohydrate Metabolism
At3g56510	Carbohydrate Metabolism
At5g58600	Carbohydrate Metabolism
At1g20840	Carbohydrate Metabolism
At5g25610	Carbohydrate Metabolism
At3g14890	Carbohydrate Metabolism
At1g59700	Carbohydrate Metabolism
At1g32520	Carbohydrate Metabolism
At1g78320	Carbohydrate Metabolism
At5g48850	Carbohydrate Metabolism
At4g19140	Carbohydrate Metabolism

At3g60190	Carbohydrate Metabolism
At3g50810	Carbohydrate Metabolism
At5g26760	Carbohydrate Metabolism
At5g58950	Carbohydrate Metabolism
At2g02160	Carbohydrate Metabolism
At4g29820	Carbohydrate Metabolism
At2g37490	Cell Growth and Maintainence
At5g65205	Cell Growth and Maintainence
At4g37280	Cell Growth and Maintainence
At1g51730	Cell Growth and Maintainence
At4g12340	Cell Growth and Maintainence
At4g15430	Cell Growth and Maintainence
At2g24550	Cell Growth and Maintainence
At1g09310	Cell Growth and Maintainence
At5g24390	Cell Growth and Maintainence
At1g25420	Cell Growth and Maintainence
At2g25680	Cell Growth and Maintainence
At1g53280	Cell Growth and Maintainence
At3g44380	Cell Growth and Maintainence
At4g17110	Cell Growth and Maintainence
At2g18960	Cell Growth and Maintainence
At1g70480	Cell Growth and Maintainence
At5g15860	Cell Growth and Maintainence
At5g60170	Cell Growth and Maintainence
At1g01950	Cell Growth and Maintainence
At3g28130	Cell Growth and Maintainence
At2g43710	Cell Growth and Maintainence
At4g10030	Cell Growth and Maintainence
At1g12730	Cell Growth and Maintainence
At5g16110	Cell Growth and Maintainence
At1g32450	Cell Growth and Maintainence
At5g27380	Cell Growth and Maintainence
At3g53950	Cell Growth and Maintainence
At4g32910	Cell Growth and Maintainence
At1g48240	Cell Growth and Maintainence
At4g28310	Cell Growth and Maintainence
At3g01100	Cell Growth and Maintainence
At1g04020	Cell Growth and Maintainence
At5g67360	Cell Growth and Maintainence
At2g29360	Cell Growth and Maintainence
At4g19670	Cell Growth and Maintainence
At3g54090	Cell Growth and Maintainence
At5g38850	Cell Growth and Maintainence
At1g13210	Cell Growth and Maintainence
At4g30850	Cell Growth and Maintainence
At1g74020	Cell Growth and Maintainence
At4g33560	Cell Growth and Maintainence
At2g32730	Cell Growth and Maintainence
At1g08650	Cell Growth and Maintainence
At5g43260	Cell Growth and Maintainence
At1g03055	Cell Growth and Maintainence
At3g05000	Cell Growth and Maintainence
At2g27200	Cell Growth and Maintainence
At1g33780	Cell Growth and Maintainence

At4g36550	Cell Growth and Maintainence
At1g62560	Cell Growth and Maintainence
At1g75780	Cell Growth and Maintainence
At5g05930	Cell Growth and Maintainence
At3g54110	Cell Growth and Maintainence
At2g14750	Cell Growth and Maintainence
At5g15930	Cell Growth and Maintainence
At5g20935	Cell Growth and Maintainence
At5g15640	Cell Growth and Maintainence
At1g49750	Cell Growth and Maintainence
At1g55910	Cell Growth and Maintainence
At3g56260	Cell Growth and Maintainence
At1g04040	Cell Growth and Maintainence
At1g24340	Cell Growth and Maintainence
At1g17210	Cell Growth and Maintainence
At1g32340	Cell Growth and Maintainence
At3g09320	Cell Growth and Maintainence
At1g50460	Cell Growth and Maintainence
At2g44130	Cell Growth and Maintainence
At2g36230	Cell Growth and Maintainence
At2g34730	Cell Growth and Maintainence
At5g14690	Cell Growth and Maintainence
At2g46060	Cell Growth and Maintainence
At5g54110	Cell Growth and Maintainence
At3g50060	Cell Growth and Maintainence
At1g49230	Cell Growth and Maintainence
At5g15160	Cell Growth and Maintainence
At1g71970	Cell Growth and Maintainence
At1g18150	Cell Growth and Maintainence
At1g27760	Cell Growth and Maintainence
At4g32020	Cell Growth and Maintainence
At5g62130	Cell Growth and Maintainence
At2g37650	Cell Growth and Maintainence
At5g01790	Cell Growth and Maintainence
At5g15050	Cell Growth and Maintainence
At2g47070	Cell Growth and Maintainence
At5g49830	Cell Growth and Maintainence
At4g33010	Cell Growth and Maintainence
At5g46910	Cell Growth and Maintainence
At3g49390	Cell Growth and Maintainence
At1g73940	Cell Growth and Maintainence
At5g64040	Cell Growth and Maintainence
At2g24200	Cell Growth and Maintainence
At2g44670	Cell Growth and Maintainence
At2g06010	Cell Growth and Maintainence
At3g06330	Cell Growth and Maintainence
At5g04880	Cell Growth and Maintainence
At3g01690	Cell Growth and Maintainence
At1g29660	Cell Growth and Maintainence
At4g39730	Cell Growth and Maintainence
At1g10900	Cell Growth and Maintainence
At2g35060	Cell Growth and Maintainence
At1g13020	Cell Growth and Maintainence
At3g01770	Cell Growth and Maintainence

At4g14500	Cell Growth and Maintainence
At5g45360	Cell Growth and Maintainence
At1g01610	Cell Growth and Maintainence
At5g61290	Cell Growth and Maintainence
At2g04850	Cell Growth and Maintainence
At2g46490	Cell Growth and Maintainence
At5g20360	Cell Growth and Maintainence
At3g16280	Cell Growth and Maintainence
At2g30460	Cell Growth and Maintainence
At2g24100	Cell Growth and Maintainence
At3g16220	Cell Growth and Maintainence
At5g05100	Cell Growth and Maintainence
At5g23340	Cell Growth and Maintainence
At1g78895	Cell Growth and Maintainence
At2g43290	Cell Growth and Maintainence
At3g11100	Cell Growth and Maintainence
At5g12250	Cell Growth and Maintainence
At1g64140	Cell Growth and Maintainence
At1g19800	Cell Growth and Maintainence
At5g60540	Cell Growth and Maintainence
At5g17600	Cell Growth and Maintainence
At1g01390	Cell Growth and Maintainence
At3g46370	Cell Growth and Maintainence
At2g41800	Cell Growth and Maintainence
At4g05070	Cell Growth and Maintainence
At1g17840	Cell Growth and Maintainence
At1g17600	Cell Growth and Maintainence
At2g43920	Cell Growth and Maintainence
At1g19880	Cell Growth and Maintainence
At1g05720	Cell Growth and Maintainence
At1g01620	Cell Growth and Maintainence
At2g36250	Cell Growth and Maintainence
At5g01980	Cell Growth and Maintainence
At4g16490	Cell Growth and Maintainence
At5g27320	Cell Growth and Maintainence
At3g44450	Cell Growth and Maintainence
At1g53035	Cell Growth and Maintainence
At5g39080	Cell Growth and Maintainence
At5g18540	Cell Growth and Maintainence
At1g16260	Cell Growth and Maintainence
At5g16200	Cell Growth and Maintainence
At4g02880	Cell Growth and Maintainence
At1g71020	Cell Growth and Maintainence
At1g59590	Cell Growth and Maintainence
At1g41830	Cell Growth and Maintainence
At2g38180	Cell Growth and Maintainence
At3g59940	Cell Growth and Maintainence
At1g52540	Cell Growth and Maintainence
At1g49670	Cell Growth and Maintainence
At3g52950	Cell Growth and Maintainence
At3g48610	Cell Growth and Maintainence
At2g32980	Cell Growth and Maintainence
At3g56680	Cell Growth and Maintainence
At3g15140	Cell Growth and Maintainence

At1g09910	Cell Growth and Maintainence
At3g01440	Cell Growth and Maintainence
At5g14530	Cell Growth and Maintainence
At3g62630	Cell Growth and Maintainence
At3g56060	Cell Growth and Maintainence
At5g49720	Cell Growth and Maintainence
At5g62900	Cell Growth and Maintainence
At1g20890	Cell Growth and Maintainence
At5g13090	Cell Growth and Maintainence
At1g28070	Cell Growth and Maintainence
At3g17100	Cell Growth and Maintainence
At3g19520	Cell Growth and Maintainence
At5g05190	Cell Growth and Maintainence
At3g62010	Cell Growth and Maintainence
At1g02220	Cell Growth and Maintainence
At2g27080	Cell Growth and Maintainence
At3g52180	Cell Growth and Maintainence
At5g59010	Cell Growth and Maintainence
At5g24870	Cell Growth and Maintainence
At4g23040	Cell Growth and Maintainence
At3g58520	Cell Growth and Maintainence
At3g15630	Cell Growth and Maintainence
At2g25730	Cell Growth and Maintainence
At4g31290	Cell Growth and Maintainence
At1g26150	Cell Growth and Maintainence
At1g60430	Cell Growth and Maintainence
At1g80310	Cell Growth and Maintainence
At3g54140	Cell Growth and Maintainence
At4g38890	Cell Growth and Maintainence
At3g63000	Cell Growth and Maintainence
At1g67570	Cell Growth and Maintainence
At2g35860	Cell Growth and Maintainence
At4g00955	Cell Growth and Maintainence
At5g12010	Cell Growth and Maintainence
At1g13740	Cell Growth and Maintainence
At3g29370	Cell Growth and Maintainence
At1g19010	Cell Growth and Maintainence
At1g71030	Cell Growth and Maintainence
At3g01520	Cell Growth and Maintainence
At1g68910	Cell Growth and Maintainence
At1g71040	Cell Growth and Maintainence
At1g01490	Cell Growth and Maintainence
At3g57420	Cell Growth and Maintainence
At5g47640	Cell Growth and Maintainence
At1g65790	Lipid Metabolism
At1g31440	Lipid Metabolism
At5g14080	Lipid Metabolism
At2g38610	Lipid Metabolism
At2g47910	Lipid Metabolism
At2g40550	Lipid Metabolism
At1g56340	Lipid Metabolism
At2g44490	Lipid Metabolism
At1g76730	Lipid Metabolism
At1g05805	Lipid Metabolism

At1g53070	Lipid Metabolism
At5g64130	Lipid Metabolism
At2g16430	Lipid Metabolism
At5g19875	Lipid Metabolism
At3g27570	Lipid Metabolism
At5g54470	Lipid Metabolism
At1g33420	Lipid Metabolism
At5g39350	Lipid Metabolism
At1g28560	Lipid Metabolism
At5g15440	Lipid Metabolism
At5g58640	Lipid Metabolism
At1g17090	Lipid Metabolism
At2g40970	Lipid Metabolism
At1g59910	Lipid Metabolism
At5g39790	Lipid Metabolism
At1g06450	Lipid Metabolism
At2g18890	Lipid Metabolism
At1g32160	Lipid Metabolism
At3g05910	Lipid Metabolism
At4g35040	Lipid Metabolism
At5g64720	Lipid Metabolism
At5g28050	Lipid Metabolism
At2g01120	Lipid Metabolism
At4g17790	Lipid Metabolism
At1g13270	Lipid Metabolism
At3g55770	Lipid Metabolism
At5g35320	Lipid Metabolism
At5g55280	Lipid Metabolism
At1g78620	Lipid Metabolism