國立交通大學

生物資訊所

碩士論文

基於人類與小鼠微陣列基因表現圖譜識別功能性基因群

Identifying functional gene clusters based on microarray gene expression profiles in human and mouse genomes

研 究 生：洪瑞鴻

指導教授：黃憲達　教授

中華民國九十六年六月

基於人類與小鼠微陣列基因表現圖譜識別功能性基因群

Identifying functional gene clusters based on microarray gene expression profiles in human and mouse genomes

研究生：洪瑞鴻	Student：Jui-Hung Hung
指導教授：黃憲達	Advisor：Hsien-Da Huang

國立交通大學
生物資訊所
碩士論文

A Thesis
Submitted to Institute of Bioinformatics
College of Biological Science and Technology
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Bioinformatics

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

基於人類與小鼠微陣列基因表現圖譜識別功能性基因群

學生：洪瑞鴻　　　　指導教授：黃憲達

國立交通大學生物資訊所碩士班

摘　　要

跨物種的基因表現圖譜分析能提供在天擇的進程中被保留的基因的功能和其參與機制的訊息，保留在物種間的基因群所扮演的生化功能極可能具有不易被取代的重要功能。尋找這樣的功能性基因群能加速基因療法中候選基因的發現和藥物的開發。為此，本研究提出一個新穎的計算處理架構試圖找出保留於人類和小鼠的基因群，利用奇異值分解（singular value decomposition） 和分群演算法分析基因表現，並且利用同源連結（ortholog linkage）和時間規整演算法(time warping algorithm)使來自不同物種（異質性）的微陣列基因表現圖譜時間序列可以比較並依此建議同源基因群。同時，我們實作模糊最近聚類（fuzzy nearest-cluster）方法來預測這些可能在生物過程（bioprocess）扮演極重要的角色的同源（orthologous）基因群的功能併施行統計檢定找出具有生物意義、保留在物種間影響細胞週期的基因群。

簡而言之，這個研究結合序列和時間序列圖譜層級的相似性來建議跨物種的功能性基因群，提供基因療法實驗的候選基因並希望能貢獻在跨物種基因分析的卓越進展。

最後，為了讓整個流程方便應用在未來的相關研究上，整個架構被模組並程式化成一個獨立的可執行套件並結合視覺化的呈現。

Identifying functional gene clusters based on microarray gene expression profiles in human and mouse genome

student：Jui-Hung Hung Advisors：Dr. Hsian-Da Huang

Institute of Bioinformatics

National Chiao Tung University

ABSTRACT

Cross-species gene expression analysis provides information of gene functions and involving mechanisms, which conserved in evolutionary process. Gene groups conserved in species are very likely to play irreplaceable biochemical functions. Searching for this kind of functional gene groups can accelerate the discovery of candidate genes in gene therapy and development of drug design. For this, our research proposes a novel computational scheme to figure out the genes having important biochemical functions, especially targets on the genes which are conserved in human and mouse. These genes conserved across evolutionary history would be most likely to reveal fundamental biochemical functions. This work utilizes singular value decomposition (SVD) and clustering techniques to analyze gene expression, and exploits orthologous linkage and time warping algorithm making microarray time-series gene-expression profiles of different species (heterogeneous profiles) comparable to suggest orthologous gene groups. In the meanwhile, in order to make the results more promising, we use fuzzy nearest cluster method to predict the functions of orthologous genes which might play important roles in the bioprocess

and perform statistical test according to our annotation of predicated gene function to find the genes having biological significance among these orthologous genes.

In brief, this research combines sequence- and time-series expression- levels ortholog to suggest functional genes among multiple species, provides materials for candidate gene therapy experiments and hopes to contribute remarkable advancement in cross-species orthologous gene analysis. In the end, in order to let the whole process be utilized in further application, the scheme is modeled and programmed in a standalone executable package with visualized presentation.

目 錄

表　目　錄

圖 目 錄

Chapter 1 Introduction

1.1 Overview of the scheme

Since the high-throughput microarray assay has been widely used, gene function prediction is shown to be reliable by classifying their gene expression profile similarity. In some of frontier research, including experimental drug and gene therapy, understandings of orthologous genes can be helpful accelerating the progress of the discovery[1].

Research about orthologous gene functional groups searching is not rare but limited.

This research considers both sequence homology and time-series expression profile pattern similarity to search for the orthologous functional gene groups among multiple species, and is able to deal with different time point number and interval, and it integrates gene functional predication to annotate and suggest their biological meanings which make the result more reliable.

In sum, this work presents a novel scheme to discover orthologous time-series gene expression profiles in multiple species. With this scheme, it is now easier to observe and disclose the important functional genes conserved in evolution process by time-series microarray profiles.

1.1.1 Gene expression time series

A gene expression profile is the result of microarray analyses, which give the breakdown of the switching on and off of certain genes (Figure 1.1). Gene expression profile is an important asset, especially for scientifically understanding biological processes from the expression of gene. DNA microarrays, oligonucleotide arrays and all other high throughput assays for gene activity give biologists the chance to view the global mRNA profile systematically[1]. There are two types of experiments, static and time series experiments[2]. In static expression experiments, only a snapshot of the expression of genes in different samples is measured[3]. On the other hand, when the profile containing the information of time intervals, it reveals the expression of genes with cell cycle stage, development stage, or any time related pattern. Gene expression time series is a list of expression data for a gene along a number of different experimental time intervals and would correspond to a row in the representation (Figure 1.2). Through the variation of mRNA expression level with time, which enables further investigations of the gene regulation networks[4, 5], functional groupings of genes, distinction of cell cycles[6], tissue-specific profiling, etc, scientists are now unraveling the mechanism of bioprocesses efficiently.

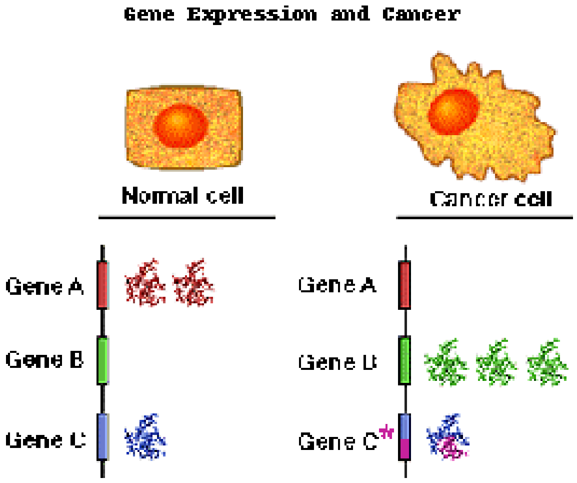


Figure 1.1 Gene expression also varies within a certain type of cell at different points in time. For example, the gene expression profiles of an organ might differ between normal and cancerous states, as shown here. [http://www.ncbi.nlm.nih.gov/Class/MLACourse/Modules/MolBioReview/gene_expression_prostate.html]

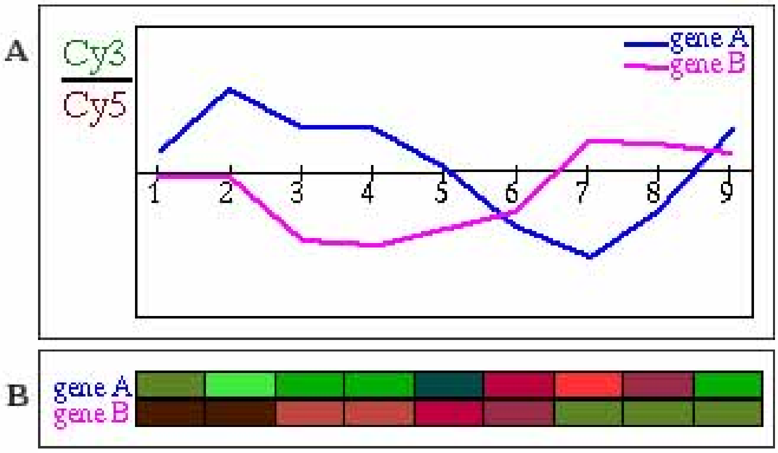


Figure 1.2 Different representations of gene expression profiles: (A) pattern, (B) color scale [http://gepas.bioinfo.cipf.es/cgibin/tutoX?c=clustering/clustering.config]

1.1.2 Gene function prediction

Determining the functions of genes is an essential problem in biology, which is fundamental to realize the molecular and biochemical processes, identify and validate new drug targets and develop reliable diagnostics. Recent advances in genomic sequencing have generated an astounding number of new putative genes and hypothetical proteins whose biological function remains a mystery. On average, there are 70% of the genes in a genome having poorly known or no known functions. There are two typical techniques that can be used on gene expression data for gene function annotation or predication. The first technique is clustering, such as hierarchical clustering, k-means clustering, SVD, and PCA, while the second is classification, such as Hidden Markov Model (HMM), Support Vector Machine (SVM), and Neural Networks (NN).

1.1.3 Data clustering

Data clustering or clustering algorithms is an approach to group data by categories.

The primary aim of clustering is to figure out several clusters and centroids or prototypes and using these centroids to represent the original enormous data.

In brief, data clustering is more likely to attempt to group data in smaller set. However, some clustering approaches can be used as classifiers as well, and it is needless to predefine classes (unsupervised learning). Clustering approaches are feasible to be utilized in gene expression analysis, since the genes are numerous and the interactions are complex.

Hierarchical clustering: In hierarchical clustering, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object[7]. Hierarchical Clustering is subdivided into agglomerative methods, which proceed by series of fusions of the n objects into groups, and divisive methods, which separate n objects successively into finer groupings. One of the simplest agglomerative hierarchical clustering methods is single linkage, also known as the nearest neighbor technique. The defining feature of the method is that distance between groups is defined as the distance between the closest pair of objects, where only pairs consisting of one object from each group are considered.

The minimum value of these distances is said to be the distance between two clusters. At each stage of hierarchical clustering, the clusters whose distance is minimal are merged. See Figure 1.3 for example.

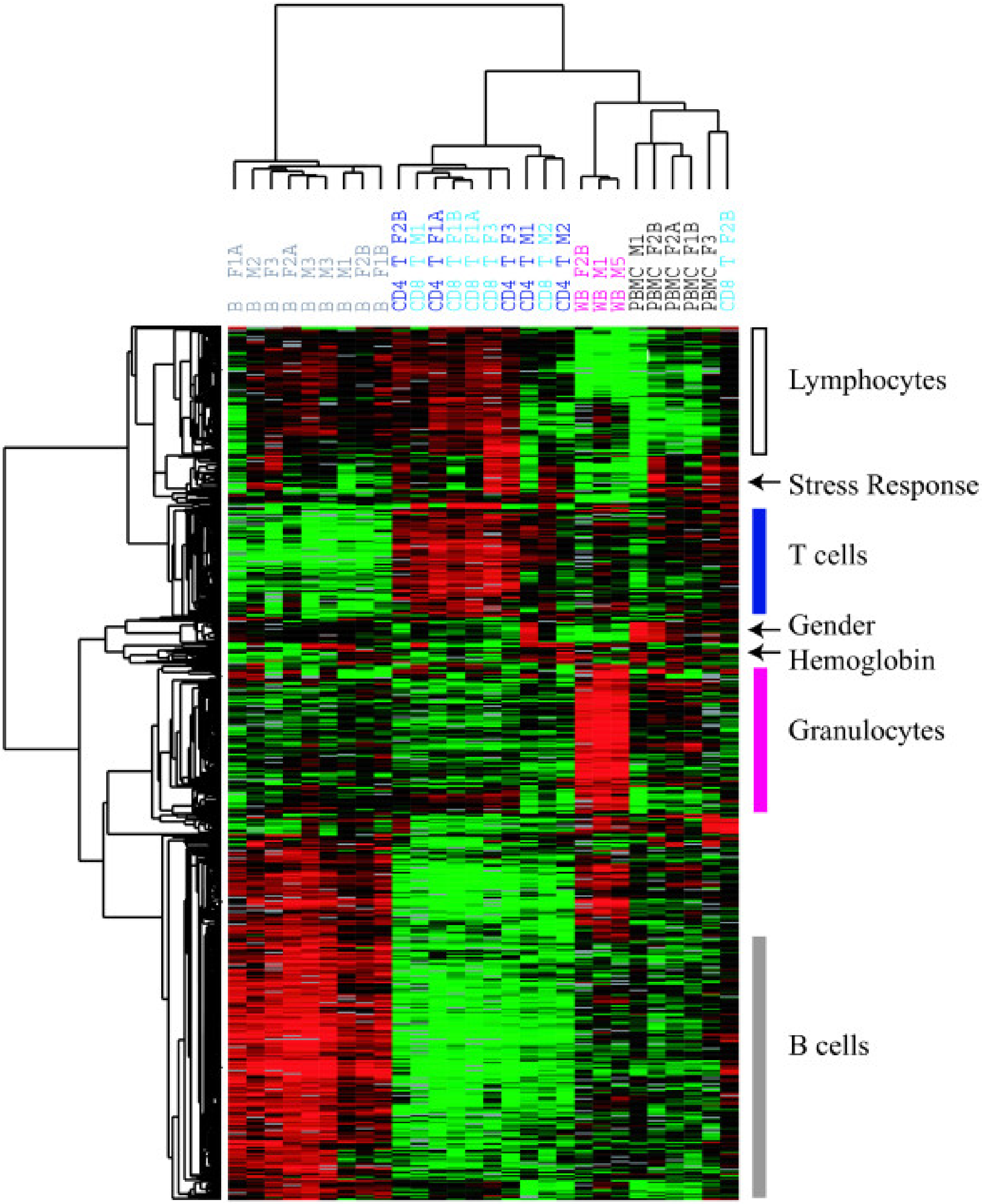


Figure 1.3 Overview of hierarchical clustering of all samples. Genes and blood samples are organized by hierarchical clustering based on overall similarity in expression patterns. Expression levels are represented by a color key in which bright red represents the highest levels and bright green represents the lowest levels, and less saturated shades represent intermediate levels of expression. [http://www.biomedcentral.com/1471-2164/7/115/figure/F1]

K-means clustering: This nonhierarchical method initially takes the number of components of the population equal to the final required number (K) of clusters[8]. In this step itself the final required number of clusters is chosen such that the points are mutually farthest apart. Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance. The centroid's position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.

K-Means Training starts with a single cluster with its center as the mean of the data. This cluster is split into two and the means of the new clusters are iteratively trained. These two clusters are again split and the process continues until the specified number of clusters is obtained. If the specified number of clusters is not a power of two, then the nearest power of two above the number specified is chosen and then the least important clusters are removed and the remaining clusters are again iteratively trained to get the final clusters.

1.1.4 Distance functions

There are two main families of distances to measure how closely related are two groups of genes:

Euclidean: this kind of distance strategy calculates the length of two separate points in n-directional space by their absolute differences[9]. For example, Euclidean distance is measure by following definition:

For two points A= (a_1, a_2… a_n), and B= (b_1, b2… b_n), Euclidean distance = $\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$

Correlation: Contrasting to Euclidean, this type of strategy accounts for the trends within both profiles[9]. For example, Pearson correlation measures the similarity in shape between two profiles by the following formula:

For two points A= (a_1, a_2… a_n), and B= (b_1, b2… b_n), Pearson correlation distance

$$= 1 - \frac{1}{n}\sum_{i=1}^{n}\left(\frac{a_i - \bar{a}}{\sigma_a}\right)\left(\frac{b_i - \bar{b}}{\sigma_b}\right)$$

where $\bar{a}$, $\bar{b}$ are the mean of a_i and b_i, and σ_a, σ_b are the standard deviation of a_i and b_i

These two kinds of distance strategies will lead to different clustering results. Please see Figure 1.4 for illustration.

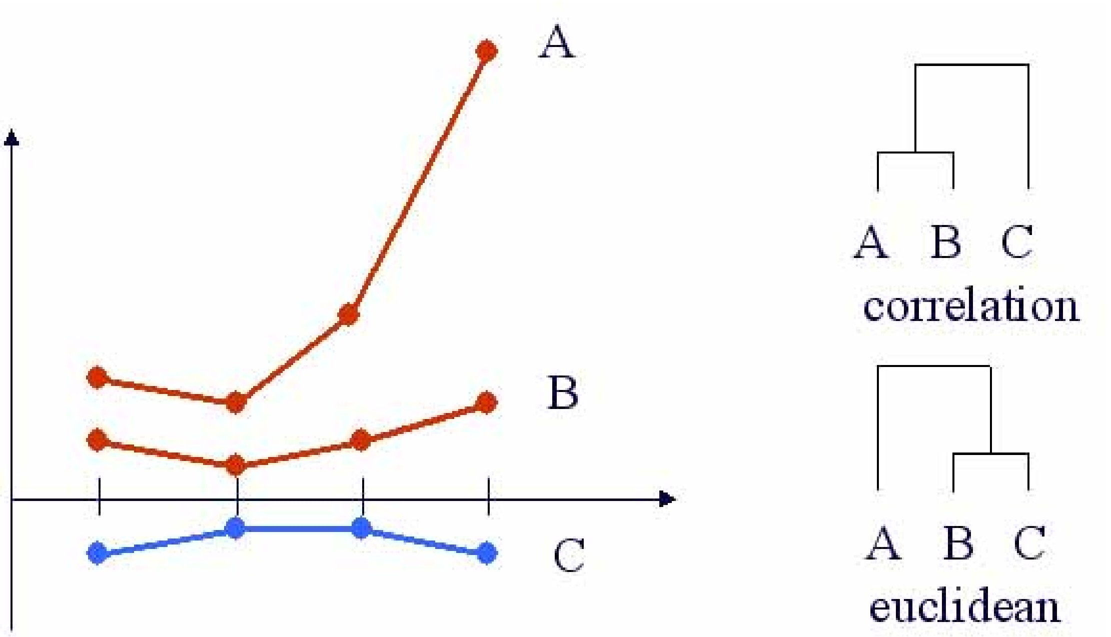


Figure 1.4 Different distances will render different classifications because we are asking for grouping based on different features (trends in the case of correlation and absolute differences in the case of Euclidean distances) [http://gepas.bioinfo.cipf.es/cgibin/tutoX?c=clustering/clustering.config]

1.1.5 Comparative analysis of different species

Comparing genomic properties of different organisms is of fundamental importance in the study of biological and evolutionary principles[10]. Although differences among organisms are often attributed to differential gene expression, genome-wide comparative analysis thus far has been based primarily on genomic sequence information.

By miscellaneous gene function predication techniques, biologists are now more interesting in orthologous gene searching among different species. Since comparative analysis of the expression data among two or more model organisms promises to enhance fundamental understanding of the universality as well as the specialization of molecular biological mechanisms. It also may prove useful in medical diagnosis, treatment, and drug design. Comparisons of the DNA sequence of entire genomes already give insights into evolutionary, biochemical, and genetic pathways. Considering that gene expression profiling gives more information of genes' biochemistry functional roles, comparative analysis based on microarray data is now a blossomed area.

1.2 Motivation

Recently, Microarray expression analysis has become an important technique for evaluating gene expression level in genomic scale. In addition, due to the profound progress in gene sequencing, considerable number of genes are predicted and found. However, there are only 30% of genes are explicitly analyzed and understood the functional roles they playing in biological process[1]. It had been shown that using microarray gene expression analysis to predict the functions of genes is an important and efficient means[5]. However, one of the defects of conventional approaches is that the number of sampling points and growth rate of cells (affected by experiment conditions) should be unified, which means that each experiment should be carefully designed to provide comparable samplings, and this is not practical in most of the cases. Therefore, mostly, searching for functional gene clusters is constrained in mono-species model and parallel experiment.

Despite so, by compiling the information of gene expression profile from diverse organisms, tissues, and conditions, scientists are now capable of dissecting more advanced topics. For instance, Grigoryev D. N., who proposed his renowned research on Genome Biology, introduced a multi-species model using gene expression profile to find orthologous gene-expression genes of lung cells suffered from ventilator-associated lung injury among human, mouse, and dog. Grigoryev also suggested these genes are potential candidate genes of acute lung injury remedy in the future, and inferred these genes are conserved among the evolution process because they play crucial protection functions after lung injury.

Applying the idea of utilizing cross-species or -tissue orthologous gene-expression profile to search important gene groups and their biological functions to other topics like cell cycle, is a general concept, which needs further investigation and development.

Nevertheless, when analyzing microarray time-series gene-expression profile, scientists have to face the difficulty of coordinating different growth rate of cells and the number and time intervals of sampling in each independent experiment, especially of distinct organisms, which is now becoming a pressing issue to let cross-species gene-expression profiles comparable.

1.3 Goals

This dissertation proposes a novel computational scheme to search and analyze the orthologous gene-expression profiling genes conserved in evolution process and involved in certain bioprocess. We try to contribute to experiment verifying and treatment development, and answer following questions:

1. How to combine and take advantage of both sequence- and expression-level orthologous gene predication?
2. How to build up the mapping relationships between genes of multi-species?
3. How to deal with noise or experimental artifacts?
4. How to let different time-series profile be comparable when the experiment conditions, growth rate, sampling time point number are different?
5. How to make our predication convincing enough?

This research hopes to propose a novel scheme to solve these related tasks on the basis of other research with integration and improvement, and contributes remarkable advancement in cross-species orthologous gene analysis.

Chapter 2 Related Works

Some of the existing research had given answers to parts of the questions we devote to solve; however, these solutions are still not sufficient to resolve our problems completely. Furthermore, most of them avoid the questions that how to let different time-series profile be comparable when the experiment conditions, growth rate, and especially sampling time point number are different

2.1 Cross species analysis with static profiling

2.1.1 Genome-wide expression data of six organisms [10]

S. Bergmann et al. present a comparative study of large datasets of expression profiles from sic evolutionarily distant organisms: *S. cerevisiae*, *C. elegans*, *E. coli*, *A. thaliana*, *D. melanogaster*, and *H. sapiens*. They use genomic sequence information to connect these data and compare global and modular properties of the transcription programs. Linking genes whose expression profiles are similar, functionally related sets of genes are frequently coexpressed in multiple organisms. Bergmann integrates the expression data with genomic sequence information to address three biological issues. First, we verify that coexpression is often conserved among organisms and propose a method for improving functional gene annotations using this conservation. Second, we compare the regulatory relationships between particular functional groups in the different organisms using the iterative signature algorithm (ISA), giving initial insights into the extent of conservation of the gene regulatory architecture. See Figure 2.1

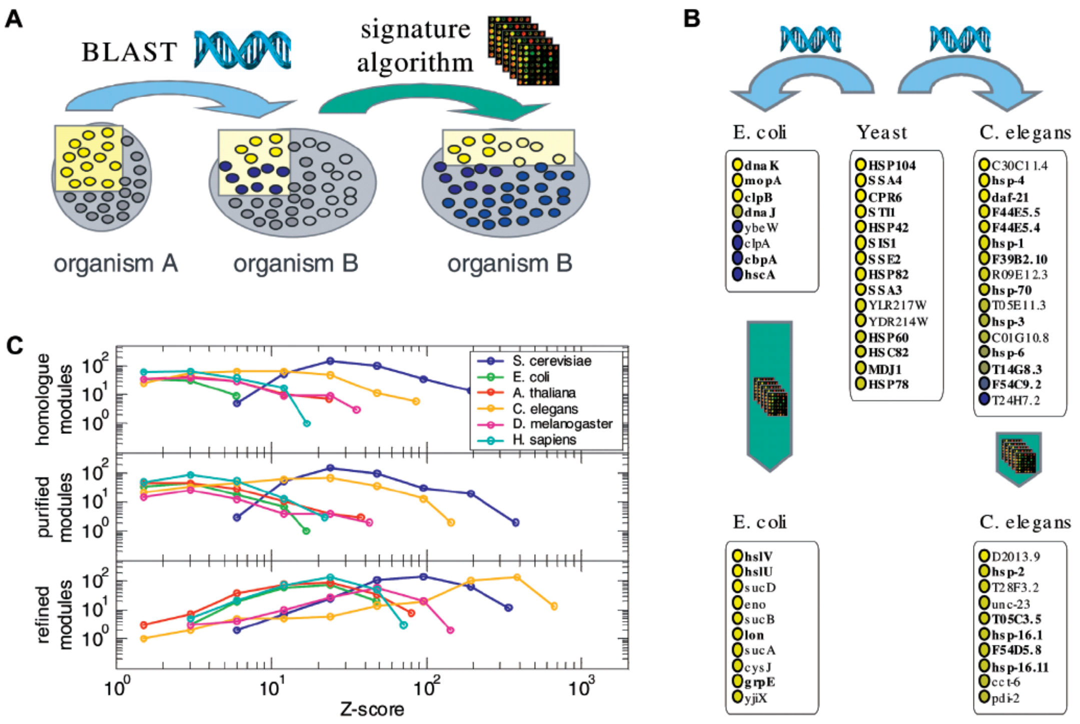


Figure 2.1 Starting from a set of coexpressed genes associated with a particular function in organism A, they first identify the homologues in organism B using BLAST. Only some of these homologues are coexpressed while others are not. The signature algorithm selects this coexpressed subset and adds further genes that were not identified based on sequence.

This approach didn't consider data with time series. Also, they linked data of different species by BLAST, which can provide sequence level homology, but they use ISA to extend genes they linked to more genes co-expressed in the same species. In order words, genes they found in the end only have ortholog in expression-level. Moreover, although they try to tell the functions of genes they found, but not with clear evidence and inference.

2.1.2 Orthologous expression profiling in multi-species models[11]

Conventional techniques perform and analyze gene-expression profiling by using species-specific Affymetrix GeneChips to search for candidate genes related VALI (ventilator-associated lung injury). The individual analysis of species-specific arrays produced large lists of candidate genes and several challenges, with the most notable being an excessive number of genes for candidate gene selection. While meta-analysis strategies exist for narrowing candidate gene selection from multiple experimental systems, this analysis can only be applied to the same species cross-platform array comparison, to use this approach for analysis of experiments involving diverse species we speculated that multispecies gene0expression profiles could be linked using RESOURCERER[12], which is based on EGO database and contains information for all commercially available Affymetrix Genechips.

D. N. Grigoryev speculated that overlapping responses to mechanical stretch in orthologous genes across species might reveal candidate genes involved in an evolutionarily conserved defense mechanism to lung injury that might be triggered by ventilator-induced lung injury.

This research first calculated gene-expression changes for each tested species and linked expression values obtained for orthologous genes. Orthologous genes exhibiting similar patterns of expression across all species were selected as VALI-related candidates under the assumption that gene-expression responses conserved across evolutionary history would be most likely to reveal fundamental biological responses to VALI. See Figure 2.2 .

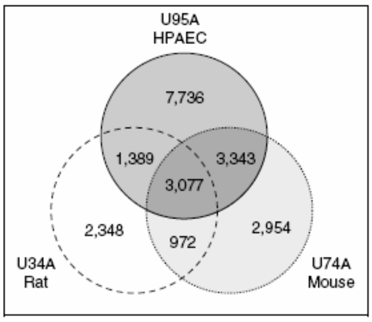


Figure 1
Overlaps between rat (U34A GeneChip), mouse (U74A GeneChip) and human (U95A GeneChip) Affymetrix array platforms based on the human (U133A GeneChip) ortholog assignments. The sum of numbers inside each circle represents the total number of ortholog pairs formed with reference genes on the U133A GeneChip by corresponding arrays (see also Table 1). The reference genes formed 3,077 pairs with corresponding orthologs that were represented on all depicted arrays.

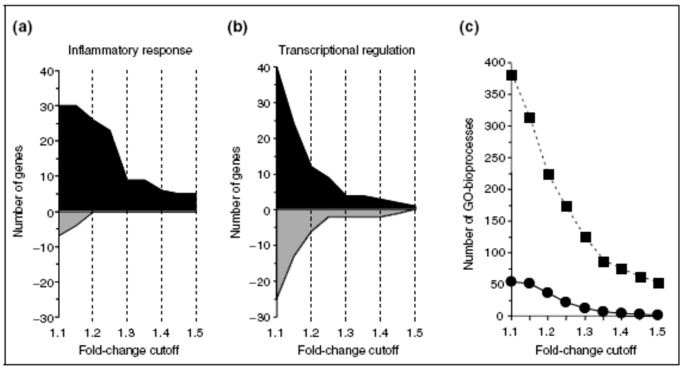


Figure 5
Distribution of co-regulated and inversely regulated biological bioprocesses identified by linkage to GO. (a) Genes involved in a co-regulated bioprocess (inflammatory response; GO 6954) and (b) an inversely regulated bioprocess (DNA-dependent regulation of transcription; GO 6355). Solid areas under the curve represent upregulated genes and gray areas under the curve represent downregulated genes. (c) A summary of all co-regulated (top curve) and inversely regulated (bottom curve) GO bioprocesses identified by MAPPFinder corresponding to the increment in the fold-change cutoff.

Figure 2.2 Different distances will render different classifications because we are asking for grouping

The basic concept of linking differentially-expressed gene with other species is similar to linking co-expressed gene group in our approaches. Even regardless of the inability of dealing time-series profiles, their scheme need a common experiment condition (in their case VALI) among samples of all species, which means that the experiment should be carefully designed and executed. This constraint makes this approach unpractical in many situations. Although they did do some experiment to prove genes they found is associate with VALI, however, their scheme unable to give a global view of gene function in genomic scale.

2.2 Cross species analysis with time series

2.2.1 GSVD for comparative analysis of expression data sets of two different organisms[13]

GSVD (generalized singular value decomposition) provides a comparative mathematical framework for two genome-scale data sets from the two-genes X arrays spaces to two reduced and diagonalized “genelets” X “arraylets” spaces. The genelets are shared by both data sets. Each genelet is expressed only in the two corresponding arraylets, with a corresponding "angular distance" indicating the relative significance of this genelet, i.e. its significance, in one data set relative to that in the other (see Figure 2.3).

O. Alter shows that mathematical reconstruction of gene expression in a subset of genelets may simulate experimental expression in subset of genelets may simulate experimental observation of only the process that these genelets are inferred to represent. By using GSVD, the framework enables comparative reconstruction and classification of the genes and arrays of both data sets and the comparison of yeast and human cell-cycle expression data sets are illustrated (see Figure 2.4).

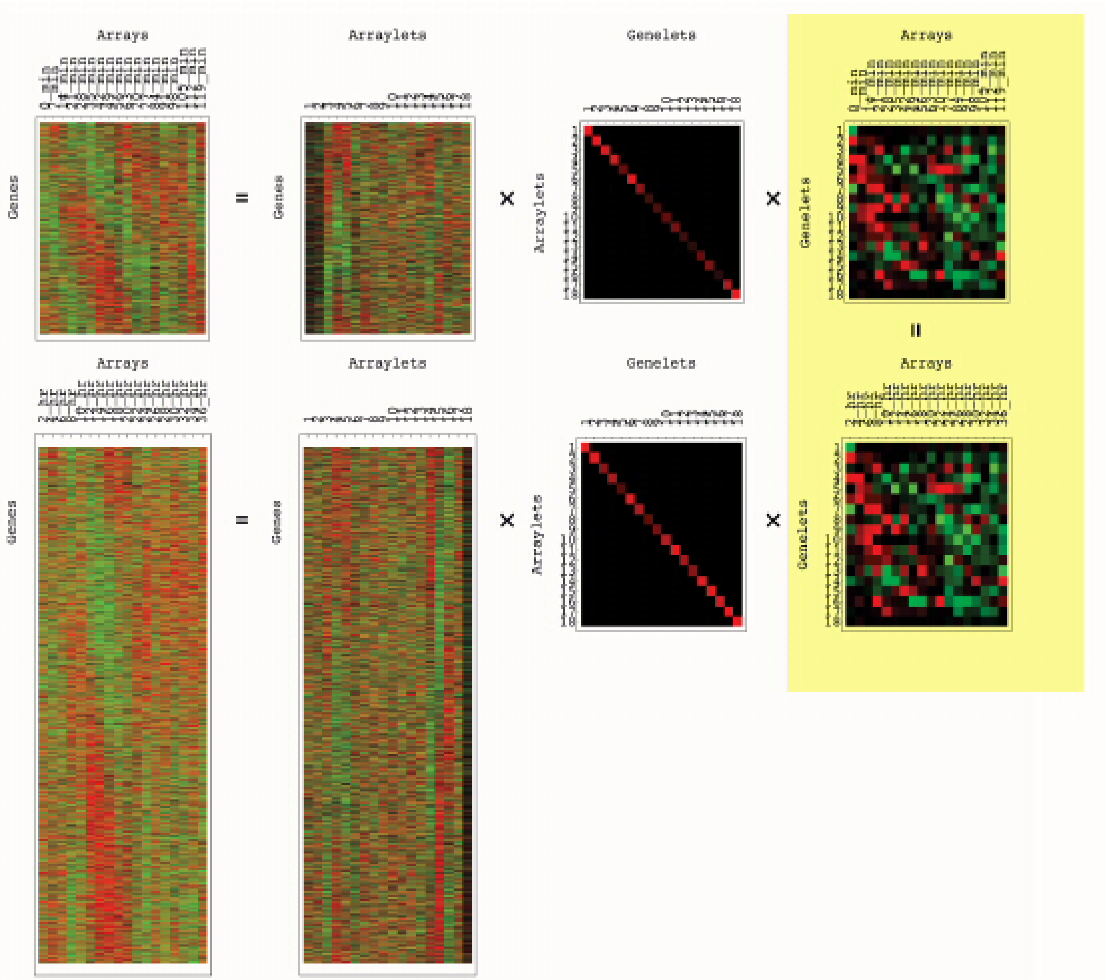


Figure 2.3 Illustration of GSVD

GSVD relies on the strong basis of mathematical theory and suggest a general approach analyzing two data sets. However, the major improvement can be categorized in three points: flexibility of multi-species model, limitation of data reduction, and incapableness of heterogeneous data sets.

First, GSVD provides useful framework to analyzing data set from two species. However, it is not suitable for models consisted of more than two species. Also, data sets from two species should be in same vector space, i.e. their dimension—the number of time points—should be the same, which is unpractical in most of the case. GSVD restrict the data to a small subset of similar conditions, such as time points along the cell cycle, which drastically reduces the size of the dataset and limits the scope of comparison[10].

Third, data sets from two species should be in same vector space, i.e. their dimensions—the number of time points—should be the same, which is unpractical in most of the case.

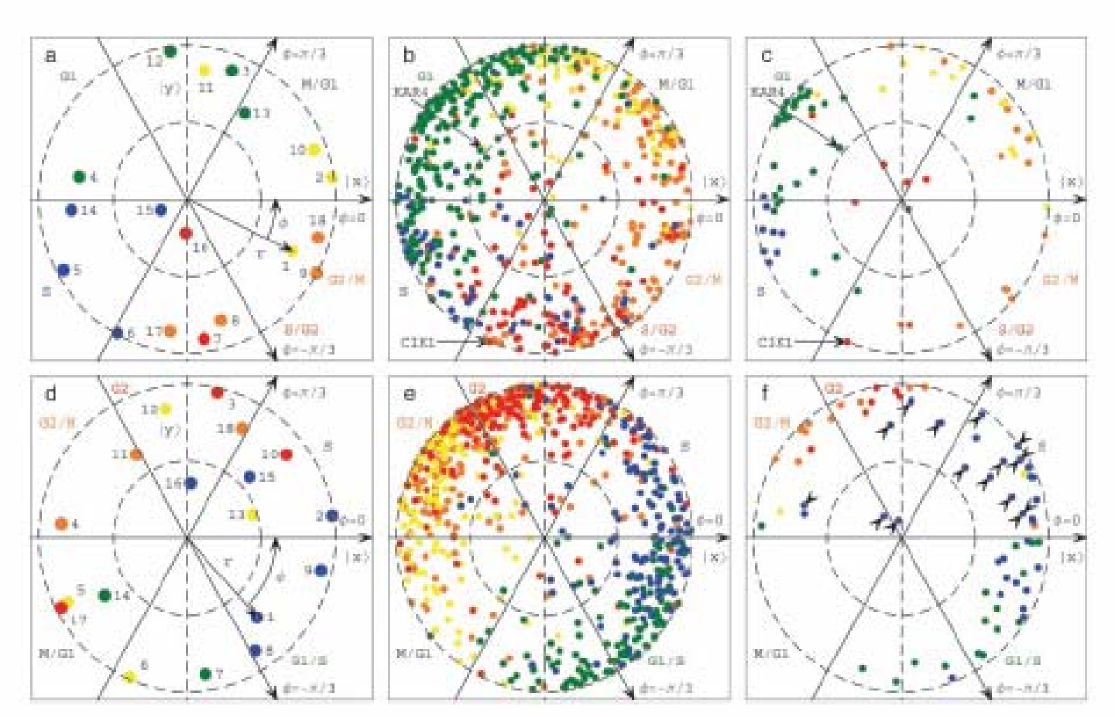


Figure 2.4 Yeast and human expression reconstructed in the six-dimensional cell-cycle subspaces approximated by two-dimensional subspaces.

2.2.2 Continuous representation of time-series expression profiles[2]

Z. Bar-Joseph et al. present a general algorithm to detect genes differentially expressed between two nonhomogeneous time-series data sets. Their algorithm overcomes these difficulties by using a continuous representation for time-series data and combining a noise model for individual samples with a global difference measure. They introduce a corresponding statistical method for computing the significance of this differential expression measure. They used their algorithm to compare cell-cycle dependent gene expression in wild type and knockout yeast strains. Their experiments suggest additional roles for the transcription factors Fkh1 and Fkh2 in controlling cellular activity in yeast.

They use cubic splines to represent gene expression curves. Cubic splines are a set of piecewise cubic polynomials and are frequently used for fitting time series and other noisy data.

$$y(t) = \sum_{i=1}^{n} C_i S_i(t) \qquad t_{min} \leq t < t_{max}$$

Using splines, we can use a linear warping function to obtain an optimal alignment by adjusting shift and stretch parameters to minimize a global error function. See Figure 2.5.

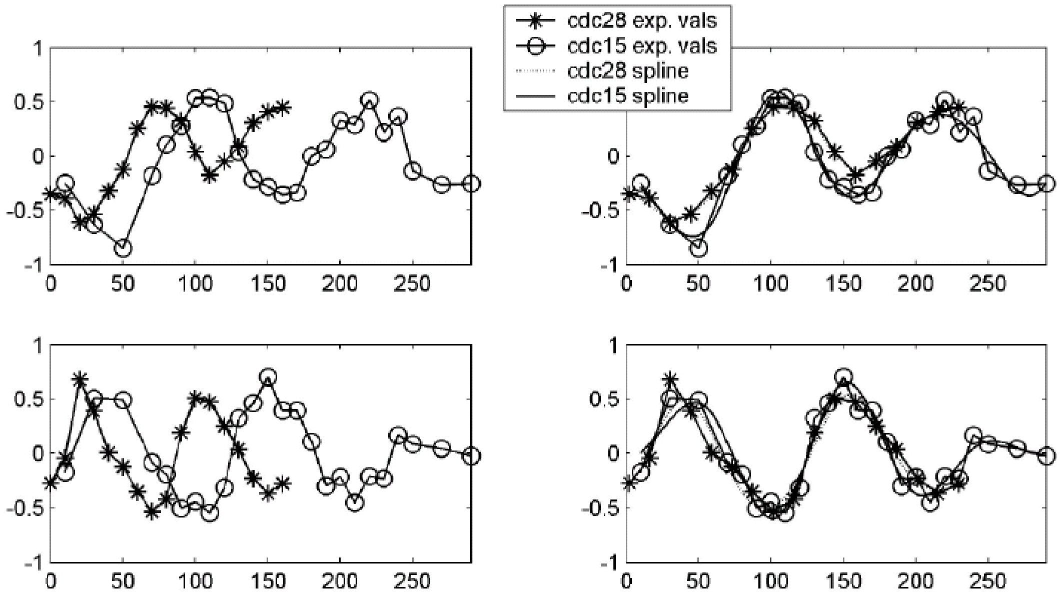


Figure 2.5 Alignment of genes for the cdc28DS to cdc15DS.

In this work, they used B-splines, a type of spline that is mathematically convenient for data approximation. B-splines are described as a linear combination of a set of basis polynomials. This approach has shown to be useful in many cases. It considers heterogeneous data sets and gives the solution by Cubic spline algorithm. However, their design is not suitable dealing with data sets need to be mapped by semi-global alignment—one of the data sets is in fact only former or later part of another. However, the EM algorithm nature they adopted in their approach let their fitting process slower. And it is a shame that their statistically analysis did not cover function annotation.

2.2.3 Aligning gene expression time series with time warping algorithms [1]

Biological processes have the property that multiple instances of a single process may unfold at different and possibly non-uniform rates in different organisms, strains, individuals, or conditions. For instance, different individuals affected by a common disease may progress at different and varying rates. Increasingly, biological processes are being studied through time series of RNA expression data collected for large numbers of genes. Because common processes may unfold at varying rates in different experiments or individuals, methods are needed that will allow corresponding expression states in different time series to be mapped to one another. John Aach and George M. Church present implementations of time warping algorithms applicable to RNA and protein expression data and demonstrate their application to published yeast RNA expression time series.

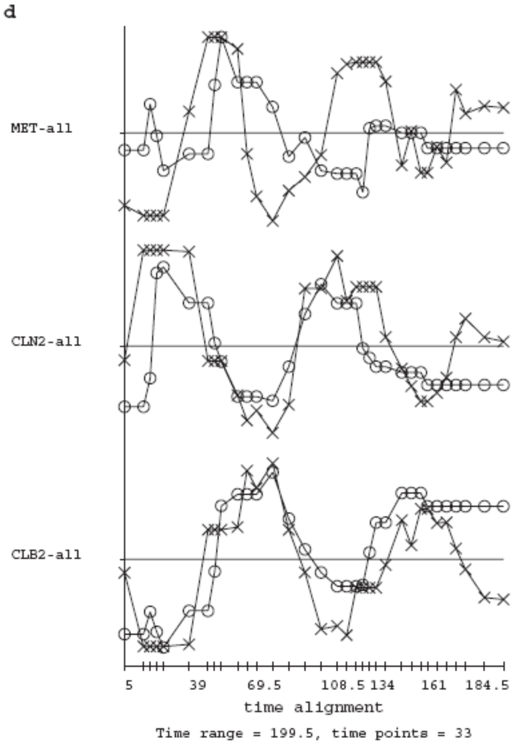


Figure 2.6 Time warping result.

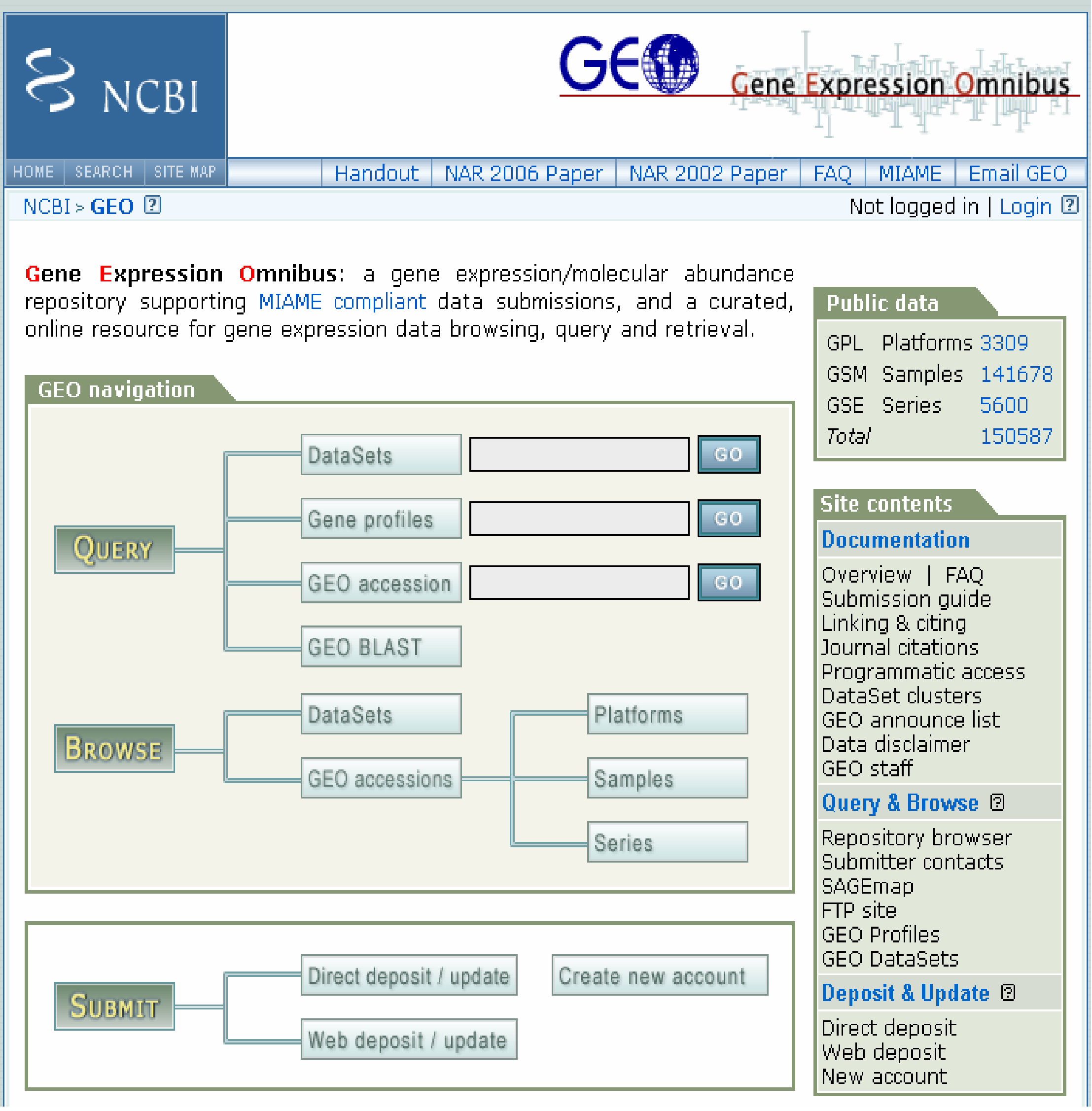
They show time warping to be superior to simple clustering at mapping corresponding time states. Depending on the domain of application, these might include cell-specific parameters such as average cell size or physiological parameters such as blood pressure or temperature. The relative contributions of such parameters to alignment score calculations can be adjusted using feature weight parameters already supported by the programs. The alignment programs can also be used not only to align RNA and protein expression series individually, but series that combine both RNA and protein data. Finally, the programs can also be applied to aligning non-temporal series such as expression profiles for cells over a range of concentrations of compounds (concentration warping).

Chapter 3 Materials and Methods

3.1 Materials

3.1.1 Datebases

GEO: a curated, online resource for gene expression data browsing, query and retrieval[14]. GEO contains 141678 sampling data, which provide us enormous experimental gene expression profiles. See Figure 3.1. Each dataset is fully annotated and completely normalized. But the comprehensive data collection, our gene function predication can be supported by adequate experiments under all kinds of conditions and treatments, which makes our predication more reliable.



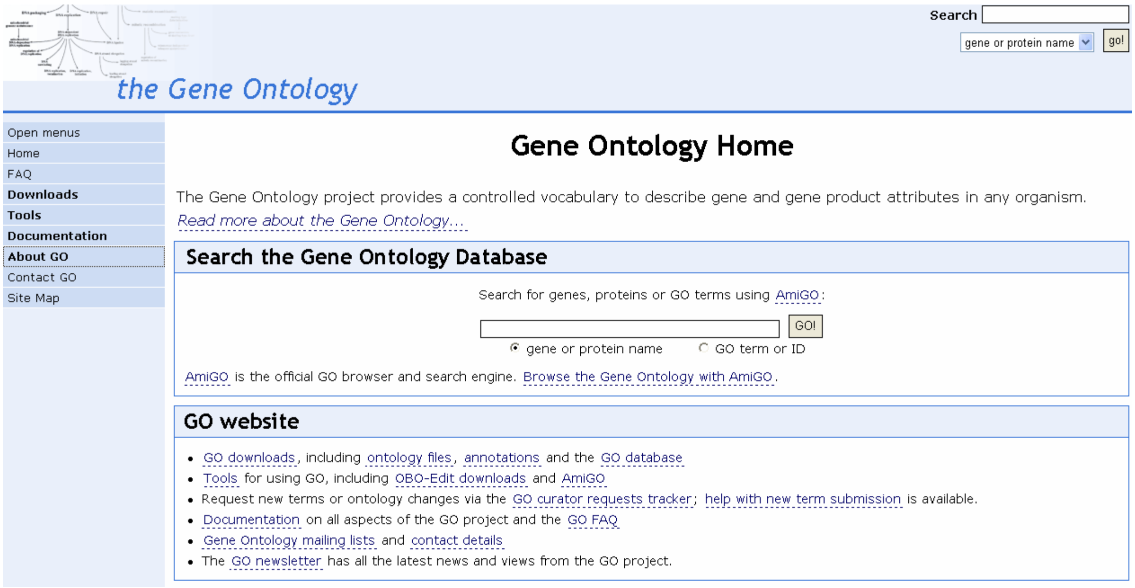
Gene Expression Omnibus: a gene expression/molecular abundance repository supporting MIAME compliant data submissions, and a curated, online resource for gene expression data browsing, query and retrieval.

Public data

GPL	Platforms	3309
GSM	Samples	141678
GSE	Series	5600
Total		150587

Figure 3.1 Web page of GEO. [http://www.ncbi.nlm.nih.gov/geo/]

GO: provides a controlled vocabulary to describe gene and gene product attributes in any organism[15]. See Figure 3.2.



Gene Ontology Home

The Gene Ontology project provides a controlled vocabulary to describe gene and gene product attributes in any organism.
Read more about the Gene Ontology...

Search the Gene Ontology Database

Search for genes, proteins or GO terms using AmiGO:

GO!
● gene or protein name ○ GO term or ID

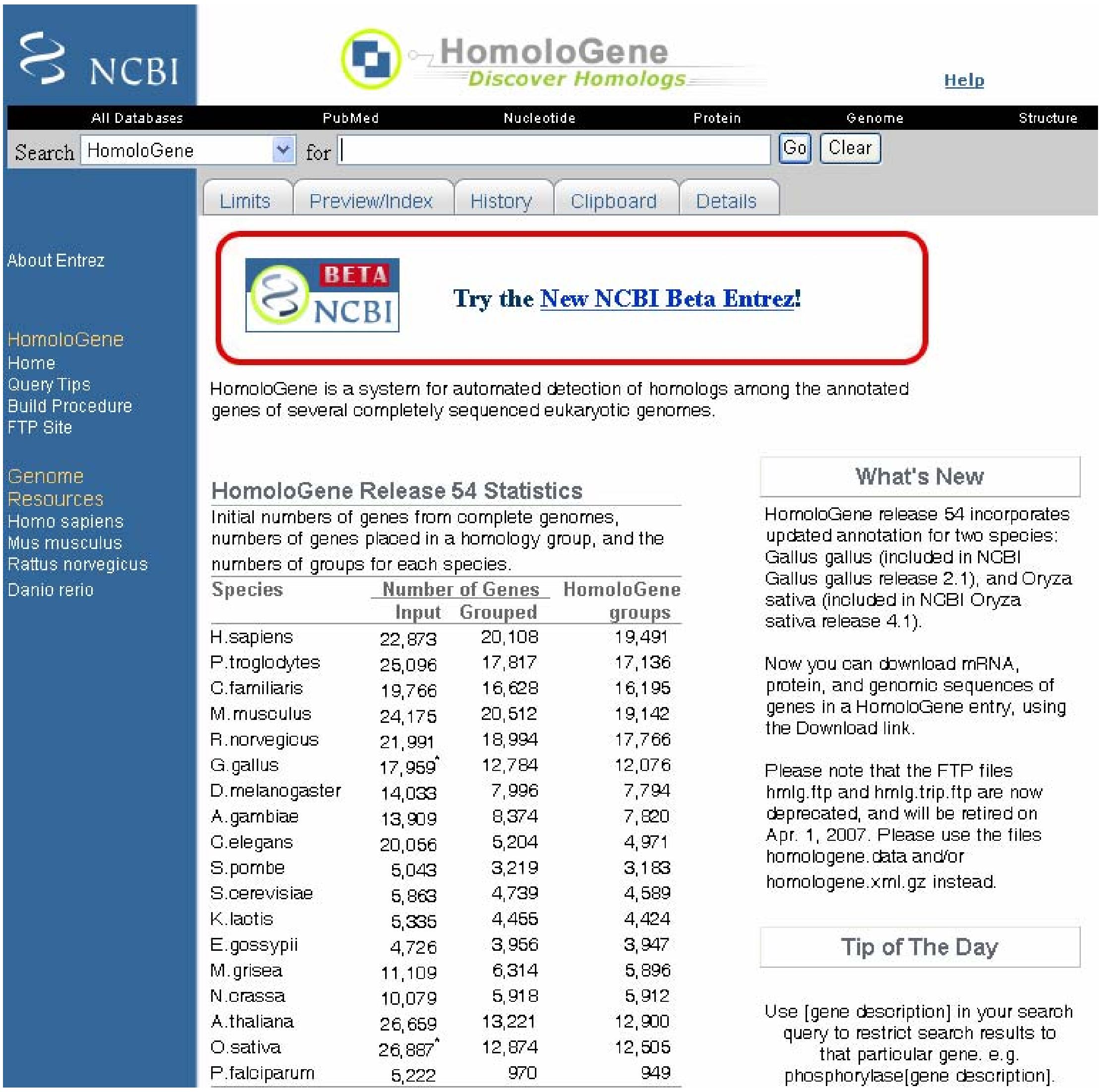
AmiGO is the official GO browser and search engine. Browse the Gene Ontology with AmiGO.

GO website

- GO downloads, including ontology files, annotations and the GO database
- Tools for using GO, including OBO-Edit downloads and AmiGO
- Request new terms or ontology changes via the GO curator requests tracker; help with new term submission is available.
- Documentation on all aspects of the GO project and the GO FAQ
- Gene Ontology mailing lists and contact details
- The GO newsletter has all the latest news and views from the GO project.

Figure 3.2 Web page of GO. [http://www.geneontology.org/]

HomoloGene: a system for automated detection of homologs among the annotated genes of several completely sequenced eukaryotic genomes. See Figure 3.3.



Try the New NCBI Beta Entrez!

HomoloGene is a system for automated detection of homologs among the annotated genes of several completely sequenced eukaryotic genomes.

HomoloGene Release 54 Statistics

Initial numbers of genes from complete genomes, numbers of genes placed in a homology group, and the numbers of groups for each species.

Species	Number of Genes		HomoloGene
	Input	Grouped	groups
H.sapiens	22,873	20,108	19,491
P.troglodytes	25,096	17,817	17,136
C.familiaris	19,766	16,628	16,195
M.musculus	24,175	20,512	19,142
R.norvegicus	21,991	18,994	17,766
G.gallus	17,959*	12,784	12,076
D.melanogaster	14,033	7,996	7,794
A.gambiae	13,909	8,374	7,820
C.elegans	20,056	5,204	4,971
S.pombe	5,043	3,219	3,183
S.cerevisiae	5,863	4,739	4,589
K.lactis	5,335	4,455	4,424
E.gossypii	4,726	3,956	3,947
M.grisea	11,109	6,314	5,896
N.crassa	10,079	5,918	5,912
A.thaliana	26,659	13,221	12,900
O.sativa	26,887*	12,874	12,505
P.falciparum	5,222	970	949

What's New

HomoloGene release 54 incorporates updated annotation for two species: Gallus gallus (included in NCBI Gallus gallus release 2.1), and Oryza sativa (included in NCBI Oryza sativa release 4.1).

Now you can download mRNA, protein, and genomic sequences of genes in a HomoloGene entry, using the Download link.

Please note that the FTP files hmlg.ftp and hmlg.trip.ftp are now deprecated, and will be retired on Apr. 1, 2007. Please use the files homologene.data and/or homologene.xml.gz instead.

Tip of The Day

Use [gene description] in your search query to restrict search results to that particular gene. e.g. phosphorylase[gene description].

Figure 3.3 Web page of homologene. [http://www.ncbi.nlm.nih.gov/HomoloGene/]

3.1.2 Data set of Experiments and results

We take three types of data to test the proposed scheme: artificial, homogeneous, and heterogeneous data sets. We will discuss this in chapter result.

3.1.3 Date set of gene funcation predication

The microarray sample was fetched from NCBI GEO in order to be the materials of FNC[16] (a predication algorithm, we will discuss it in latter session) training and prediction. GEO stores many precious gene expression profiles. We took advantage of GEO’s comprehensive collection of published dataset, and extracted experiments which consisted of several time points and are suitable to be compared with the time series. Time series samples generated on Affymetrix GeneChip platform was considered firstly. The samples of each time point are combined by averaging and converting to log2 ratio by the mean of the expression level in all time point. That is, for the gene G_1 in a profile with 7 time points T_1, T_2, …, T_7 and each time point have two samples E_{t1a}, E_{t1b}, E_{t2a}, E_{t2b}, …, E_{t7a}, E_{t7b}. Next, we do the process of **averaging**, **converting to log2 ratio, and standardization** as shown below. We arranged the dataset as a matrix which represented as following **Matrix**.

Averaging:

$$A_{11} = \frac{(E_{t1a} + E_{t1b})}{2}, A_{12} = \frac{(E_{t2a} + E_{t2b})}{2}, ..., A_{17} = \frac{(E_{t7a} + E_{t7b})}{2}; A_{1mean} = \frac{\sum_{k=1}^{7} A_{1k}}{7}$$

Log2 ratio: $N_{11} = \lg\left(\frac{A_{11}}{A_{1mean}}\right), N_2 = \lg\left(\frac{A_{12}}{A_{1mean}}\right), ..., N_7 = \lg\left(\frac{A_{17}}{A_{1mean}}\right)$

Standardization: $\mathrm{S}_{11} = \frac{\mathrm{N}_{11} - \mathrm{N}_{1\mathrm{mean}}}{\sigma_1}$

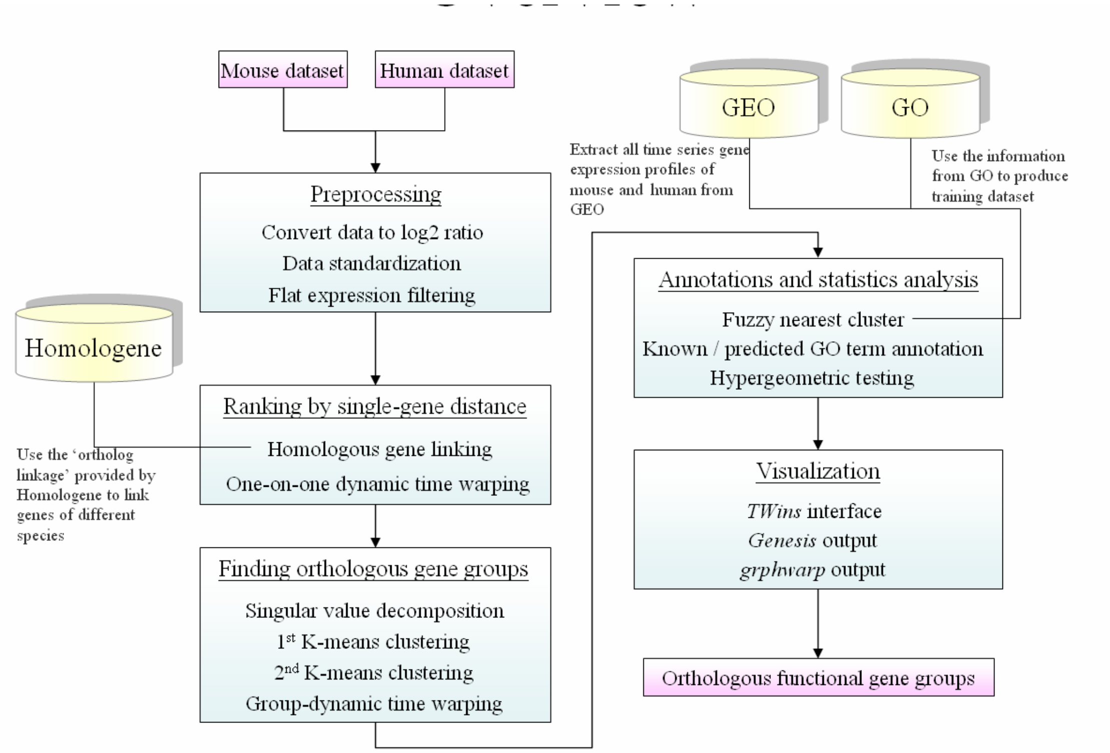
Matix:

$$\begin{array}{c|cccc} & T_1 & T_2 & \cdots & T_7 \\ \hline G_1 & \mathrm{S}_{11} & \mathrm{S}_{12} & \cdots & \mathrm{S}_{17} \\ G_2 & \mathrm{S}_{21} & \mathrm{S}_{22} & \cdots & \mathrm{S}_{27} \\ G_3 & \mathrm{S}_{31} & \mathrm{S}_{32} & \cdots & \mathrm{S}_{37} \\ G_4 & \mathrm{S}_{41} & \mathrm{S}_{42} & \cdots & \mathrm{S}_{47} \end{array}$$

Experiment performed on human and mouse using Affymetrix GeneChips were extracted from GEO and annotated their functional category by GO. We compiled the entire GO (released at Dec. 2006) combining with our prediction and constructed the mapping relation of Gene symbol name between SwissProt ID (because part of the data store in GO is specified by SwissProt) from SwissProt. By this, we can then annotate the GO term to each gene, and calculate the p-value to suggest biological significance of the occurrence of this GO term.

3.2 Methods overview

The flowchart of the scheme is presented below (Figure 3.4):



Mouse dataset
Human dataset
GEO
GO
Extract all time series gene expression profiles of mouse and human from GEO
Use the information from GO to produce training dataset
Preprocessing
Convert data to log2 ratio
Data standardization
Flat expression filtering
Homologene
Use the 'ortholog linkage' provided by Homologene to link genes of different species
Ranking by single-gene distance
Homologous gene linking
One-on-one dynamic time warping
Finding orthologous gene groups
Singular value decomposition
1st K-means clustering
2nd K-means clustering
Group-dynamic time warping
Annotations and statistics analysis
Fuzzy nearest cluster
Known / predicted GO term annotation
Hypergeometric testing
Visualization
TWins interface
Genesis output
grphwarp output
Orthologous functional gene groups

Figure 3.4 system flow of the scheme

The whole scheme consists of five components: *preprocessing*, *ranking by single-gene distance*, *finding orthologous gene groups*, *annotation and statistics analysis and visualization*.

The process first takes formatted human and mouse datasets as input. In preprocessing stage, the datasets are normalized, standardized, and filtered out genes with flat expressions, which implies these genes are not response dynamically toward the perturbation in the experiment along with time and therefore lack of referable meanings in our time-series profile analysis.

After the preprocessing, the remaining genes in human and mouse datasets are cross-species linked by ortholog linkage provide by Homologene database, which builds the linkages based on the sequence similarity of genes among species. Each linked gene pair is calculated its single-gene distance by dynamic time warping algorithm. All gene pairs are ranked according to their distance. The scheme selects top *T* gene pairs for further analysis, since these genes share both sequence and expression similarity as responding to experimental perturbation.

Next, in order to further infer the functional roles among these genes, we group these *T* genes into smaller clusters. Before doing so, we act singular value decomposition to filter out the noise in expression. We don't do SVD in preprocessing stage for several reasons, which will be discussed in following chapters. Then, K_1-means clustering was performed on one of the dataset (in this research, human), and after that, the scheme acts second time K_2-means clustering on another dataset within each firstly-clustered group into even smaller clusters, which generates K_1*K_2 clusters. The scheme acts group-dynamic time warping to suggest unified warping path of this group of genes pairs. Now, by above processes, the parallel essence in sequence and expression among each group is sufficient to suggest that these groups of genes play important roles in certain bioprocess, and conserve in evolution process. In the fourth stage of the scheme, we try to suggest the functional roles of these gene groups. By the help of GO, GEO and fuzzy nearest clustering algorithm, we can annotate the known and predicted GO term of each gene. We use statistical testing to judge and recognize the candidate genes with biological significance and proposed the inference that these genes play essential roles in certain bioprocess during the evolution process.

In the end, to further visualize of result, the scheme which had been carefully programmed will generate abundant and useful information of the results. The whole scheme is embedded in our web sever TWins, and the output files can be fed in to Genesis and grphwarp program for further visualization and analysis.

3.3 Algorithm

3.3.1 Dynamic Time Warping (DTW) algorithm

DTW (see figure 3.12), which is similar to the sequence alignment used in computational biology, are firstly introduced in speech recognition. By compression and expansion operations, multiple time points with calculated weight coefficient can be aligned to a single time point. DTW considers the warping distance according to the vectors in feature space, and the distance can be evaluated by simple Euclidean distance, Pearson correlation coefficient, or more complicated functions in which the distance is sensitive to position in the feature space[1].

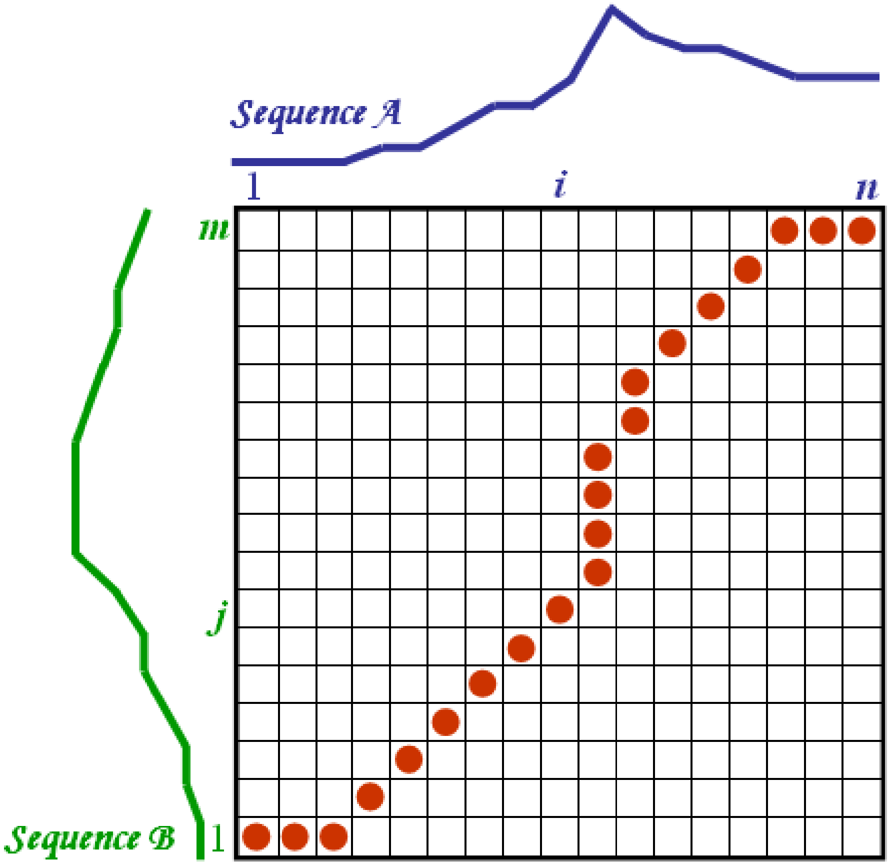


Figure 3.5 An illustration of DTW algorithm.

The basic idea of time warping is that replications of nominally the same trajectory will trace out approximately the same curve (expression profiles pattern), but with varying time patterns. To minimize the warping distance between two observed profiles, a recursion to find the minimal distance is the main part of the calculation. This program adopts conventional DTW, see the equation below, where τ , μ are time points, and a, b are the expression values of two time series:

$$D_{i,j} = \begin{cases} D_{i-1,j-1} + \dfrac{\tau+\mu}{2} \cdot |a_i - b_j| \\ D_{i-1,j} + \dfrac{\tau}{2} \cdot |a_i - b_j| \\ D_{i,j-1} + \dfrac{\mu}{2} \cdot |a_i - b_j| \end{cases}$$

Relative to the a series, a second time series b for a different instance of the process may contain a set of time points τ and μ . The sample points may come from a trajectory that traces through different regions of k-space or traces through the same regions at different rates[1] (Figure 3.6). Simple time warping uses dynamic programming to find the mapping between two series that minimizes a weighted sum of the k-space distances between the corresponding sample points, subject to constraints of order preservation and globality. The mapping identifies an optimal time alignment of the two series. The task of finding it is set up as a dynamic programming problem by placing the time points of each series along the axes of a grid, representing alignments as paths through the grid cells, and finding the path with minimum accumulated weighted distance score.

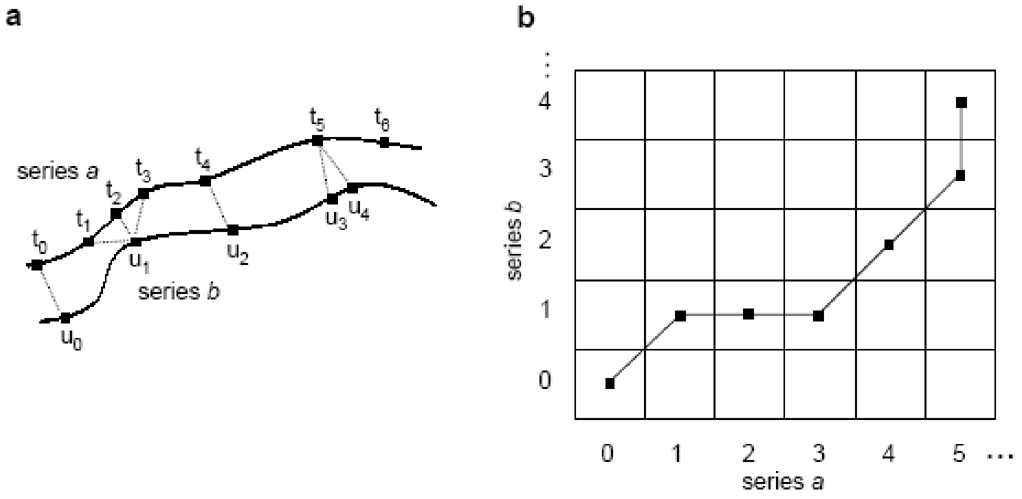


Figure 3.6 Two time series in a two-dimensional feature space containing sample points from a continuous process, with sample points of each series mapped to each other by simple time warping.

The mappings of the optimal path identify places where multiple time points of one series correspond to a single time point of the other. Where measurement time intervals are comparable between the series, these may represent situations in which the instance of the biological process measured by one series moves quickly through a phase of the process relative to the instance measured by the other series. We call such situations compression/expressions and they are analogous to the insertion / deletions considered in sequence alignment algorithms. Time warping algorithm maps two time series in a way that compensates for varying relative rate differences in gene expression levels moving along similar expression trajectories[17].

3.3.2 Singular value decomposition (SVD) and Clustering approaches

SVD is a common technique for analysis of multivariate data, and gene expression data are well suited to analysis using SVD. In the literature the number of components that results from SVD is sometimes associated with the number of underlying biological processes that give rise to the patterns in the data[18].

Let X denotes an m x n matrix of real-valued data and rank r. In the case of microarray data, x_{ij} is the expression level of the ith gene in the jth assay. The elements of the ith row of X form the n-dimensional vector g_i, which we refer to as the transcriptional response of the ith gene. Alternatively, the elements of the jth column of X form the m-dimensional vector a_j, which we refer to as the expression profile of the jth assay.

The equation for singular value decomposition of X is the following:

$$X = USV^T$$

where U is an m x n matrix, S is an n x n diagonal matrix, and V^T is also an n x n matrix. The columns of U are called the left singular vectors (eigengenes), $\{u_k\}$, and form an orthonormal basis for the assay expression profiles, so that $u_i \cdot u_j = 1$ for i = j, and $u_i \cdot u_j = 0$ otherwise. The rows of V^T contain the elements of the right singular vectors (eigenarrays), $\{v_k\}$, and form an orthonormal basis for the gene transcriptional responses. The elements of S are only nonzero on the diagonal, and are called the singular values.

In systems biology applications, we generally wish to understand relations among genes. The signal of interest in this case is the gene transcriptional response g_i. The SVD equation for g_i is

$$g_i = \sum_{k=1}^{r} u_{ik} s_k v_k \text{ , where } i : 1,\ldots,m$$

which is a linear combination of the eigengenes $\{v_k\}$.

SVD is a linear transformation of the expression data from the n-genes x m-arrays space to the reduced r-eigenarrays x r-eigengenes space[19]. See Figure 3.7 for illustration.

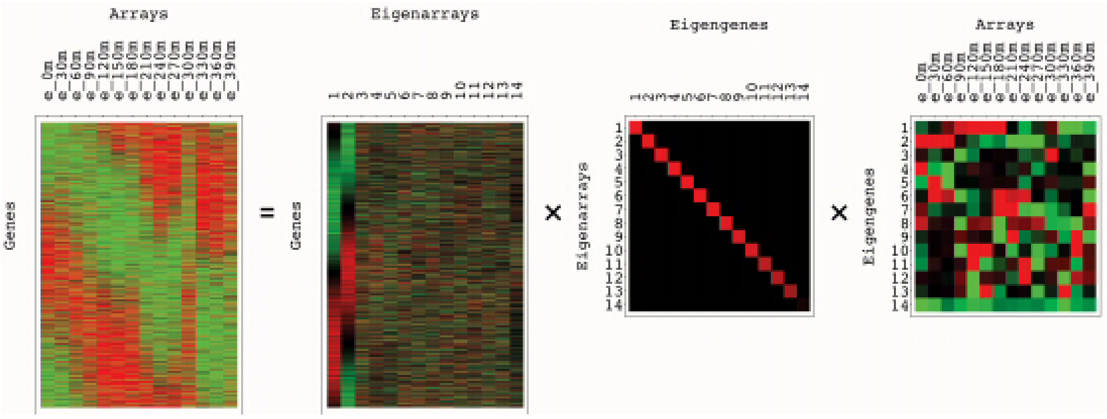


Figure 3.7 SVD for genome-scale expression data analysis. [http://genome-www.stanford.edu/SVD/]

Relation to principal component analysis: There is a direct relation between PCA and SVD in the case where principal components are calculated from the covariance matrix. The matrix US then contains the principal component scores, which are the coordinates of the genes in the space of principal components.

Even though each component on its own may not necessarily be biologically meaningful, SVD can aid in the search for biologically meaningful signals[18]. The height of each singular value indicates its importance in explaining the data. More specifically, the square of each singular value is proportional to the variance explained by each singular vector. The relative variances are often plotted (See Figure 3.8). If the original variables are linear combinations of a smaller number of underlying variables, combined with some low-level noise, the plot will tend to drop sharply for the singular values associated with the underlying variables and then much more slowly for the remaining singular values. One approach is to ignore components beyond where the cumulative relative variance or singular value becomes larger than a certain threshold, usually defined upon the dimensionality of the data. Everitt and Dunn[20] propose an alternate approach based on comparing the relative variance of each component to 0.7/n[18]. By normalizing the data and filtering out those eigengens and eigenarrays (i.e. substituting zero for the singular value lower than 0.7/n) that are inferred to represent noise or experimental artifacts, SVD can reconstruct the original data as a matrix which contain only significant signals

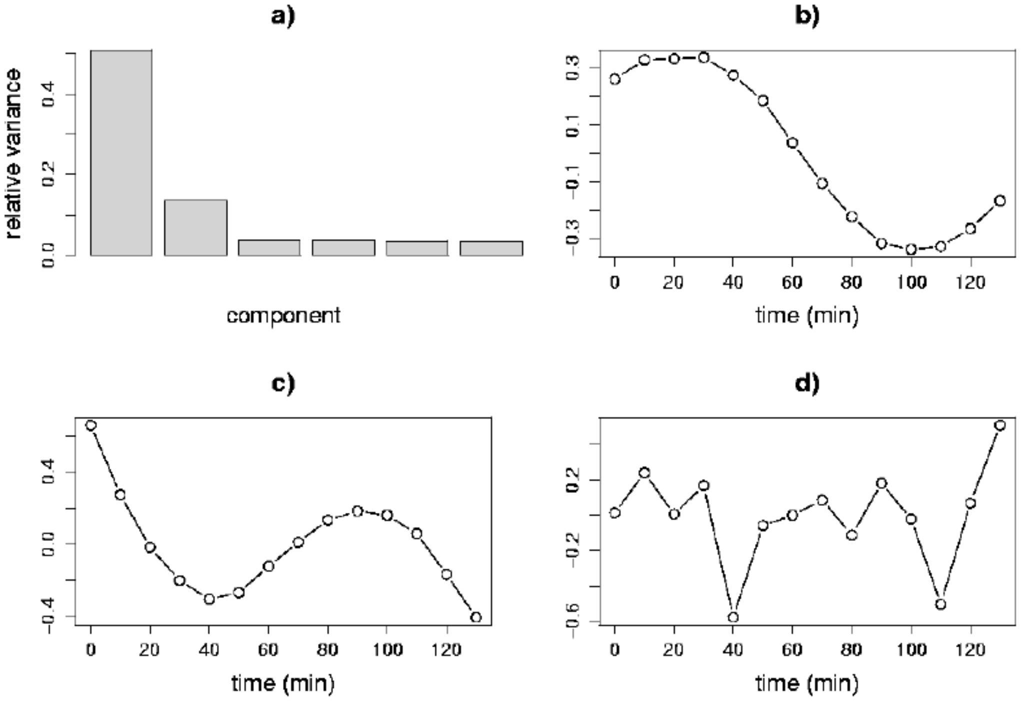


Figure 3.8 Visualization of the SVD of cell cycle data. (a) Plots of relative variance; (b) the first eignegen is shown; (c) the second eignegene is shown. (d) The third eigengen lacks the obvious cyclic structure of the first and second.[18]

K-means clustering takes the matrix as input and genes are grouped according to the value in the row they represented. In our experiment we took Pearson correlation distance to evaluate how close two genes are.

3.3.3 Guilt-by-association (GBA) principle

GBA infers uncategorized items by the close similarity to known items which can be judged by evaluating the distance[21]. GBA principle is widely applied in biological function prediction and candidate gene discovery. In gene expression profile analysis, uncategorized genes can be grouped together with known genes by the distance or the correlation of their expression pattern.

3.3.4 Fuzzy Nearest-Cluster (FNC)[21]

FNC utilizes the advantages of both clustering and classification. It contains two parts: (1) mining by unsupervised approach, hierarchical clustering algorithm; (2) prediction category of unclassified items by classification methods using GBA principle. See Fig. 3.9.

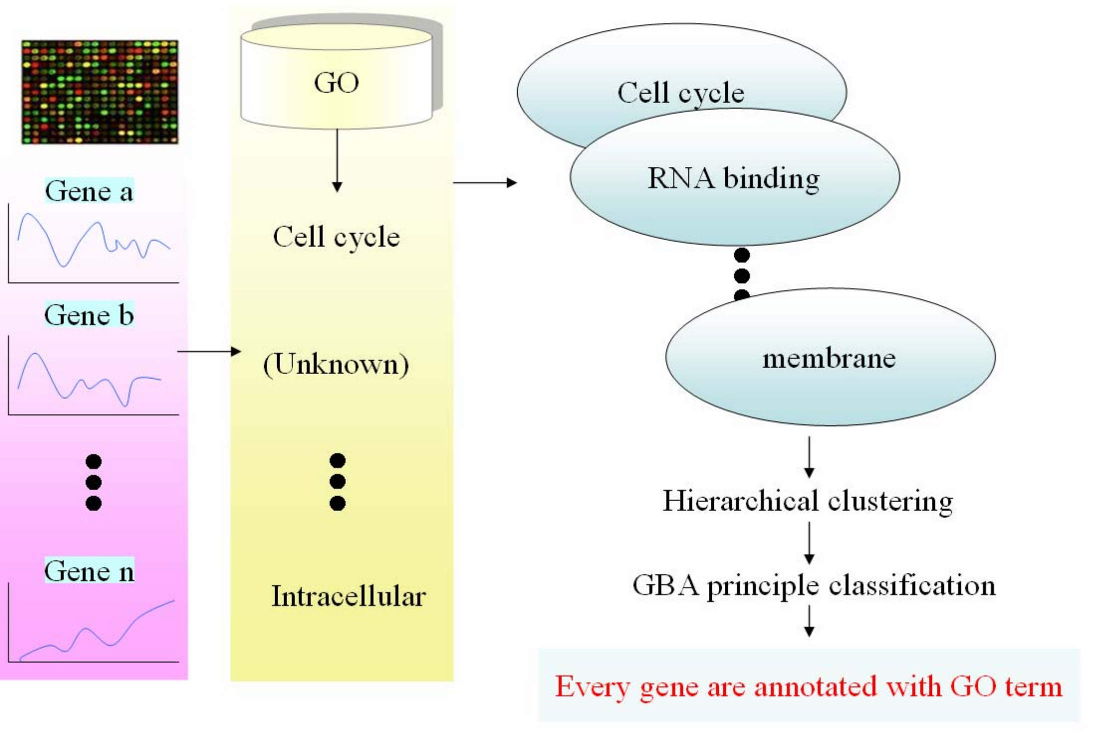


Figure 3.9 The system flow of FNC.

Figure 3.10 and 3.11 details the clustering algorithm for mining step. In the algorithm, the FNC focuses on grouping gene expression profiles, in order to mining co-expressed subgroups within a functional class.

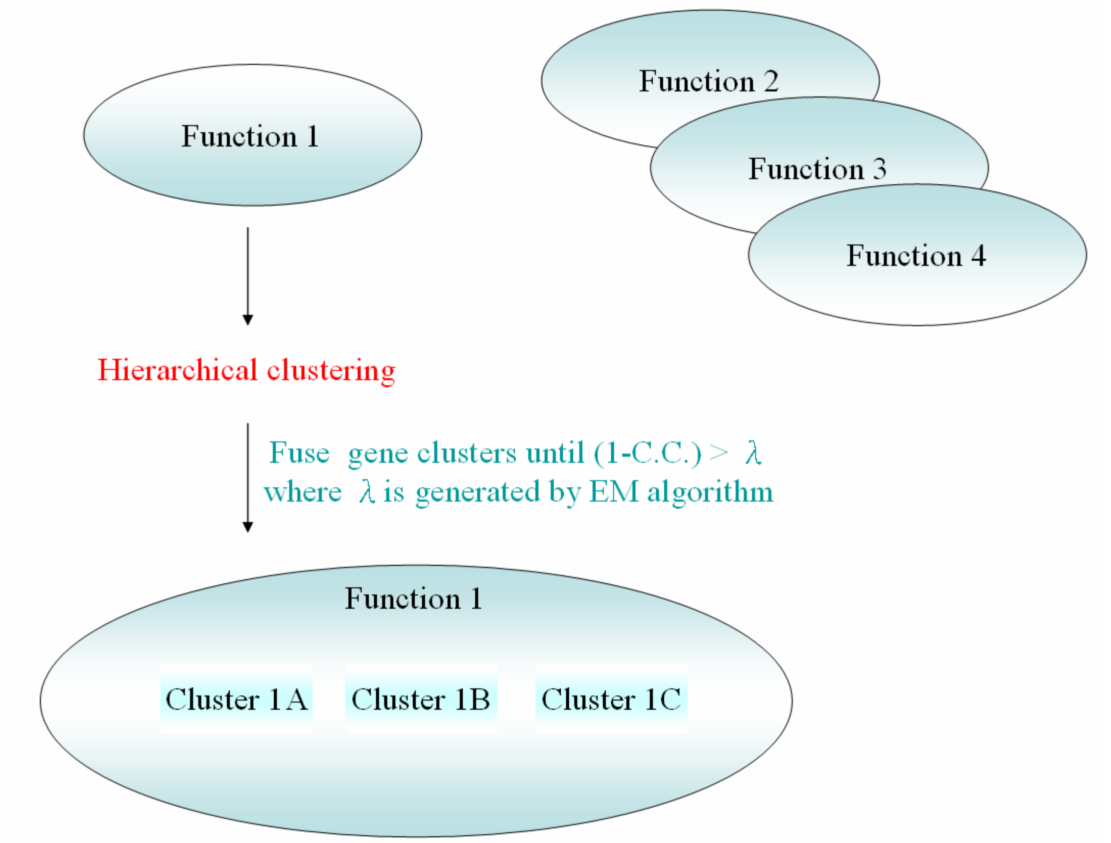
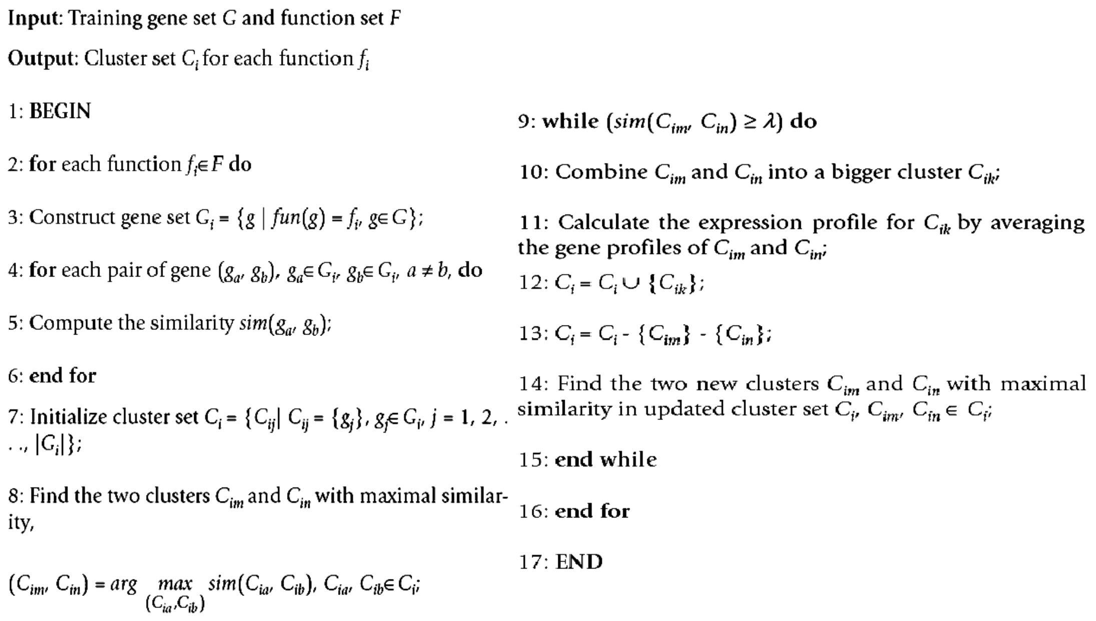


Figure 3.10 Algorithm of mining co-expressed subgroups within each function.



```
Input: Training gene set $G$ and function set $F$
Output: Cluster set $C_i$ for each function $f_i$
BEGIN
  for each function $f_i \in F$ do
    Construct gene set $G_i = \{g \mid fun(g) = f_i, g \in G\}$;
    for each pair of gene $(g_a, g_b)$, $g_a \in G_i$, $g_b \in G_i$, $a \neq b$, do
      Compute the similarity $sim(g_a, g_b)$;
    end for
    Initialize cluster set $C_i = \{C_{ij} | C_{ij} = \{g_j\}, g_j \in G_i, j = 1, 2, . . ., |G_i|\}$;
    Find the two clusters $C_{im}$ and $C_{in}$ with maximal similarity,
    $(C_{im}, C_{in}) = arg \max_{(C_{ia}, C_{ib})} sim(C_{ia}, C_{ib}), C_{ia}, C_{ib} \in C_i$;
    while ($sim(C_{im}, C_{in}) \geq \lambda$) do
      Combine $C_{im}$ and $C_{in}$ into a bigger cluster $C_{ik}$;
      Calculate the expression profile for $C_{ik}$ by averaging the gene profiles of $C_{im}$ and $C_{in}$;
      $C_i = C_i \cup \{C_{ik}\}$;
      $C_i = C_i - \{C_{im}\} - \{C_{in}\}$;
      Find the two new clusters $C_{im}$ and $C_{in}$ with maximal similarity in updated cluster set $C_i$, $C_{im}, C_{in} \in C_i$;
    end while
  end for
END
```

Figure 3.11 Pseudo code of mining co-expressed subgroups.

When the subgroups of each function are classified, each function-unknown gene would be predicted its function according to fuzzy k-nearest clusters algorithm described in figure 3.12 and 3.13.

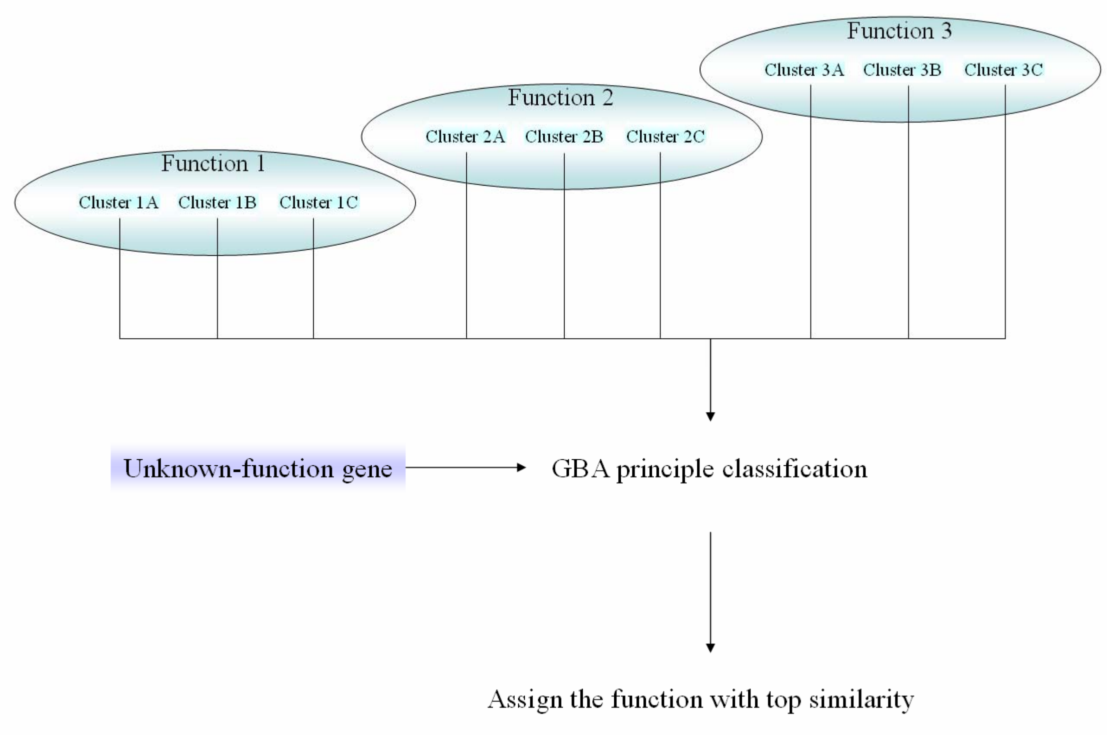
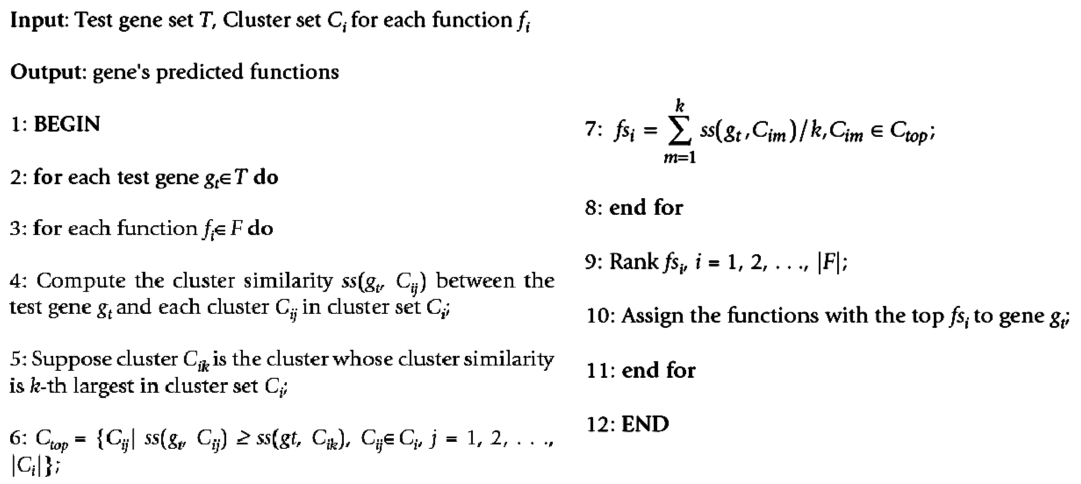


Figure 3.12 Algorithm of fuzzy k-nearest clusters for functional prediction.

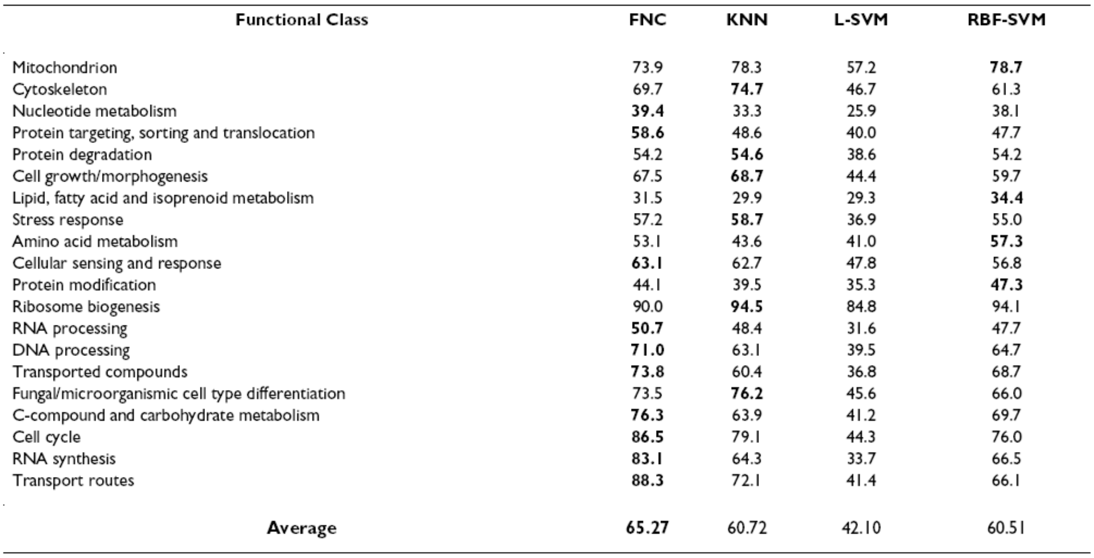


```
Input: Test gene set $T$, Cluster set $C_i$ for each function $f_i$
Output: gene's predicted functions
BEGIN
for each test gene $g_t \in T$ do
    for each function $f_i \in F$ do
        Compute the cluster similarity $ss(g_t, C_{ij})$ between the test gene $g_t$ and each cluster $C_{ij}$ in cluster set $C_i$;
        Suppose cluster $C_{ik}$ is the cluster whose cluster similarity is $k$-th largest in cluster set $C_i$;
        $C_{top} = \{C_{ij} | ss(g_t, C_{ij}) \geq ss(gt, C_{ik}), C_{ij} \in C_i, j = 1, 2, ..., |C_i|\}$;
        $fs_i = \sum_{m=1}^{k} ss(g_t, C_{im}) / k, C_{im} \in C_{top}$;
    end for
    Rank $fs_i$, $i = 1, 2, ..., |F|$;
    Assign the functions with the top $fs_i$ to gene $g_t$;
end for
END
```

Figure 3.13 Pseudo code of fuzzy k-nearest clusters for functional prediction.

FNC had shown to be more competent than the existing techniques in ranking the true functional classes in its top-ranked perditions. The classification results in listed in Table 3.1.

Table 3.1 Classification result (%) for largest 20 functional classes. Values in bold indicate the top performance in each row.



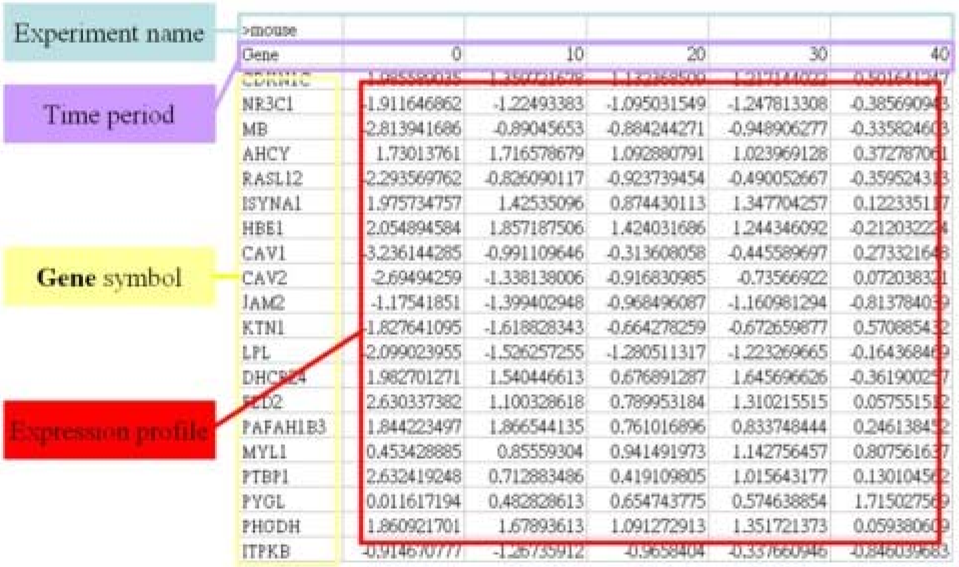
Functional Class	FNC	KNN	L-SVM	RBF-SVM
Mitochondrion	73.9	78.3	57.2	**78.7**
Cytoskeleton	69.7	**74.7**	46.7	61.3
Nucleotide metabolism	**39.4**	33.3	25.9	38.1
Protein targeting, sorting and translocation	**58.6**	48.6	40.0	47.7
Protein degradation	54.2	**54.6**	38.6	54.2
Cell growth/morphogenesis	67.5	**68.7**	44.4	59.7
Lipid, fatty acid and isoprenoid metabolism	31.5	29.9	29.3	**34.4**
Stress response	57.2	**58.7**	36.9	55.0
Amino acid metabolism	53.1	43.6	41.0	**57.3**
Cellular sensing and response	**63.1**	62.7	47.8	56.8
Protein modification	44.1	39.5	35.3	**47.3**
Ribosome biogenesis	90.0	**94.5**	84.8	94.1
RNA processing	**50.7**	48.4	31.6	47.7
DNA processing	**71.0**	63.1	39.5	64.7
Transported compounds	**73.8**	60.4	36.8	68.7
Fungal/microorganismic cell type differentiation	73.5	**76.2**	45.6	66.0
C-compound and carbohydrate metabolism	**76.3**	63.9	41.2	69.7
Cell cycle	**86.5**	79.1	44.3	76.0
RNA synthesis	**83.1**	64.3	33.7	66.5
Transport routes	**88.3**	72.1	41.4	66.1
Average	**65.27**	60.72	42.10	60.51

3.4 Methods

In this section we will disclose the proposed scheme in five steps.

3.4.1 Preprocessing

File format description: The applicable file format consists of three parts: *experiment name, time periods***,** and *expression data with gene symbol*. *Experment name* needs to be notified by a '>' at the beginning of the first line. *Time periods* should start with "Gene" and follow with time points, which are separated by tab. Next, each line of the *expression data* are named by its gene symbol as the beginning of the line, and followed by corresponding time-series expression data separated by tab. gene symbol name can be substituted by any other ID, however, any other type of gene name will not be able to map to gene ontology category defined by GO.



>mouse					
Gene	0	10	20	30	40
CDKN1C	1.085580035	1.350721678	1.132685099	1.217144022	0.501641247
NR3C1	-1.911646862	-1.22493383	-1.095031549	-1.247813308	-0.385690943
MB	-2.813941686	-0.89045653	-0.884244271	-0.948906277	-0.335824603
AHCY	1.73013761	1.716578679	1.092880791	1.023969128	0.372787061
RASL12	-2.293569762	-0.826090117	-0.923739454	-0.490052667	-0.359524313
ISYNA1	1.975734757	1.42535096	0.874430113	1.347704257	0.122335117
HBE1	2.054894584	1.857187506	1.424031686	1.244346092	-0.212032224
CAV1	-3.236144285	-0.991109646	-0.313608058	-0.445589697	0.273321648
CAV2	-2.69494259	-1.338138006	-0.916830985	-0.73566922	0.072038321
JAM2	-1.17541851	-1.399402948	-0.968496087	-1.160981294	-0.813784039
KTN1	-1.827641095	-1.618828343	-0.664278259	-0.672659877	0.570885432
LPL	-2.099023955	-1.526257255	-1.280511317	-1.223269665	-0.164368469
DHCR24	1.982701271	1.540446613	0.676891287	1.645696626	-0.361900257
FZD2	2.630337382	1.100328618	0.789953184	1.310215515	0.057551512
PAFAH1B3	1.844223497	1.866544135	0.761016896	0.833748444	0.246138452
MYL1	0.453428885	0.85559304	0.941491973	1.142756457	0.807561637
PTBP1	2.632419248	0.712883486	0.419109805	1.015643177	0.130104562
PYGL	0.011617194	0.482828613	0.654743775	0.574638854	1.715027569
PHGDH	1.860921701	1.67893613	1.091272913	1.351721373	0.059380609
ITPKB	-0.914670777	-1.26735912	-0.9658404	-0.337660946	-0.846039683

Figure 3.14 Input file format.

Convert data to log2 ratio: When the formatted data set is inputted to the system, it will be converted to log2 ratio according to the mean expression of each row. Please see 3.1.3 for details.

Data standardization: Since our algorithm considers Euclidean distance as our distance function, each profile should therefore be standardized, or the distance will be affected by the expression level of profile, and the trend of pattern will be missing or ignore by the algorithm. In statistics, a standard score (also called z-score or normal score) is a dimensionless quantity derived by subtracting the population mean from an individual (raw) score and then dividing the difference by the population standard deviation. The z-score reveals how many units of the standard deviation a case is above or below the mean. So, by standardization, the pattern will be substitute to a trend according to original mean, which is suitable for our Euclidean distance function.

Flat expression filtering: A level expression pattern is not helpful throughout our analysis, since this kind of pattern reveals no information of differential expression, which hinders our time warping algorithm to distinguish the warping path and following warping distance. So before entering next stage that performing time warping algorithm, flat expression should be filtered.

We denote a flat expressed gene as the gene of which expression in each time points is oscillating in the range of 1.3*(mean of the expression in all samples) and 0.7*(mean of the expression in all samples). In other word, if there are more than one expression of all time points higher than the upper bound or lower than the lower bound, this gene will be retained and submitted to next stage. (We do not perform noise filtering in this stage for several reasons. We will disclose them in discussion chapter.)

3.4.2 Ranking by single-gene distance

For the reason that we are finding genes that share both sequence- and expression-level similarity, we have to select genes linked by sequence homology and with relatively low distance which implies their expression pattern is similar to achieve our requirement.

Homologous gene linking: One means of searching orthologous gene-expression profile in multi-species models we refer to is proposed by Grigoryev et al. in 2004[11]. Although their research is not suitable for time series expression profiling which is strongly influenced by cell growth rate, their idea to associate genes of multi-species is examined to be very useful. Grigoryev used ortholog links which are identified by RESOURCERER[12, 22] between the most commonly used Affymetrix rat, mouse, and human GeneChips for multi-species cross-platform gene-expression analysis to build the linkage of different species. NCBI Homologene is also competent providing gene linkage between species.

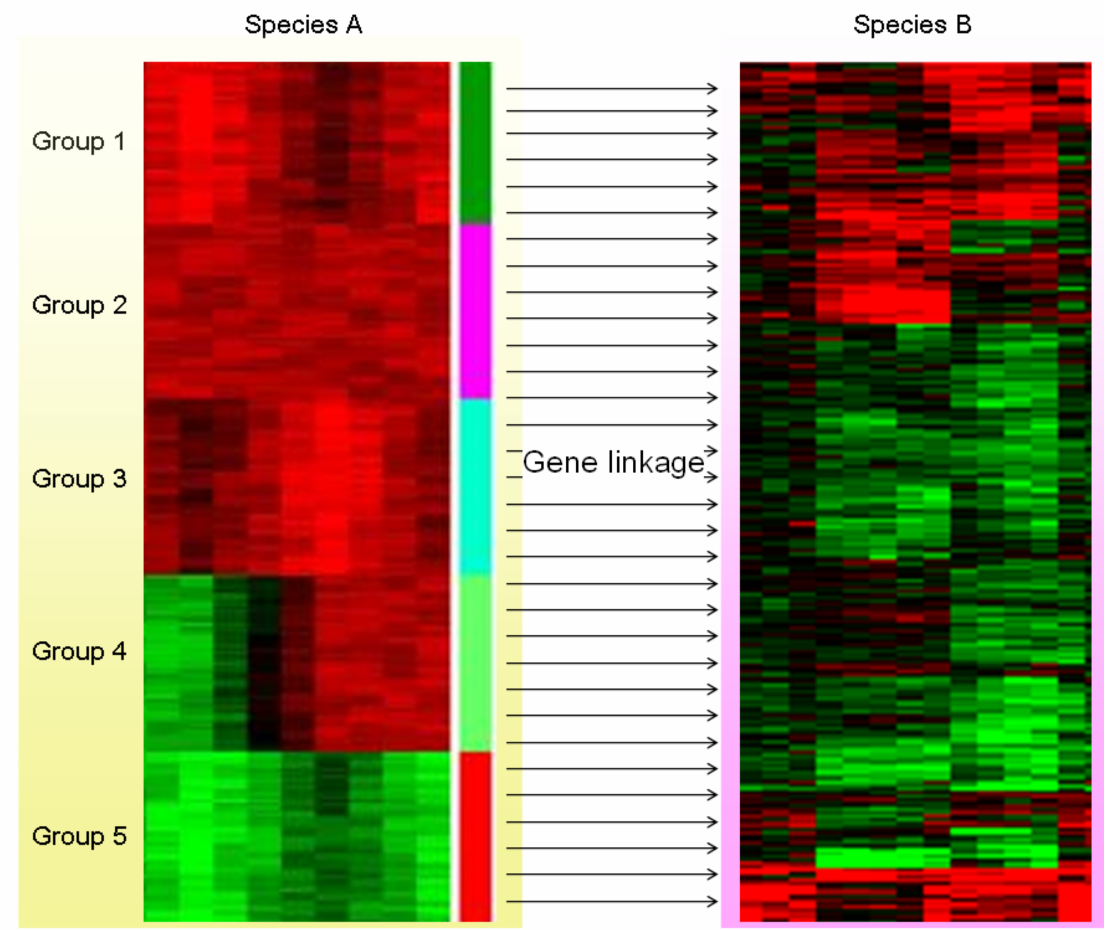


Figure 3.15 Build the gene linkage between each gene of groups and the gene of another species.

Since each Affymetrix GeneChip has full annotation, the gene symbol of the probe set is indicated. NCBI Homologene provides great mapping information of gene symbols of different organisms. Homologene built the mapping relationship according to their homology on sequence level. Since genes with similar sequence probably share similar role of biochemical functions, combining sequence homology and expression profiles to suggest their common functions is intuitionally more reliable.

After the linkage is constructed, each gene can be mapped with its corresponding gene of another species (see Figure 3.15). Next, we perform single gene time warping to calculate and rank the warping distance to tell the genes with similar expression pattern.

One-on-one dynamic time warping: When the expression profile are sampled every single hour from normal cells of human and mouse, the different growth rate affected the profiling which make direct one-on-one mapping unreliable. Aach's research[1] indicated even in the same processes the unfold rates of different experiments or individuals are different. Dynamic time warping (DTW) algorithms are proposed to make different time series to be comparable by finding their corresponding expression states. When two time series expressed in the same pattern however in different rate, DTW is able to find the optimal warping path which aligns two time series yielding shortest distance.

Therefore, we leverage the advantage of the DTW to estimate how close two genes are considering their time difference. By doing so, we can rank all genes by their warping distance. The top ranking genes show relatively resembling expression, which means these genes are both sequence and expression analogical.

After collecting these genes, we are going further to specify their function. This entails the helps of following two stages.

3.4.3 Finding orthologous gene groups

Orthologous genes are likely to share similar pattern of expression. The co-expressed genes can be inferred to be coding for proteins that partake in common biological function[21]. We therefore find the gene cluster that share parallel expression pattern by unsupervised learning approach, which assumes no prior knowledge about the prospective candidate genes.

Singular value decomposition (noise filtering): Before been grouped, SVD was performed to normalize the data by filtering out the eigengenes and eigenarrays that are inferred to represent noise or experimental artifacts enables meaningful comparison of the expression of different genes across different arrays in different experiments. However, SVD has its mathematical limitation; we will discuss this in chapter discussion. SVD filters eigengenes of which relative variance lower than 0.7/n (n is the number of time points) and recombines the matrix again. After SVD, we will retain only significant information which contains less noise. Expression profile without observable noise can lead to better clustering result. See Figure 3.16 .

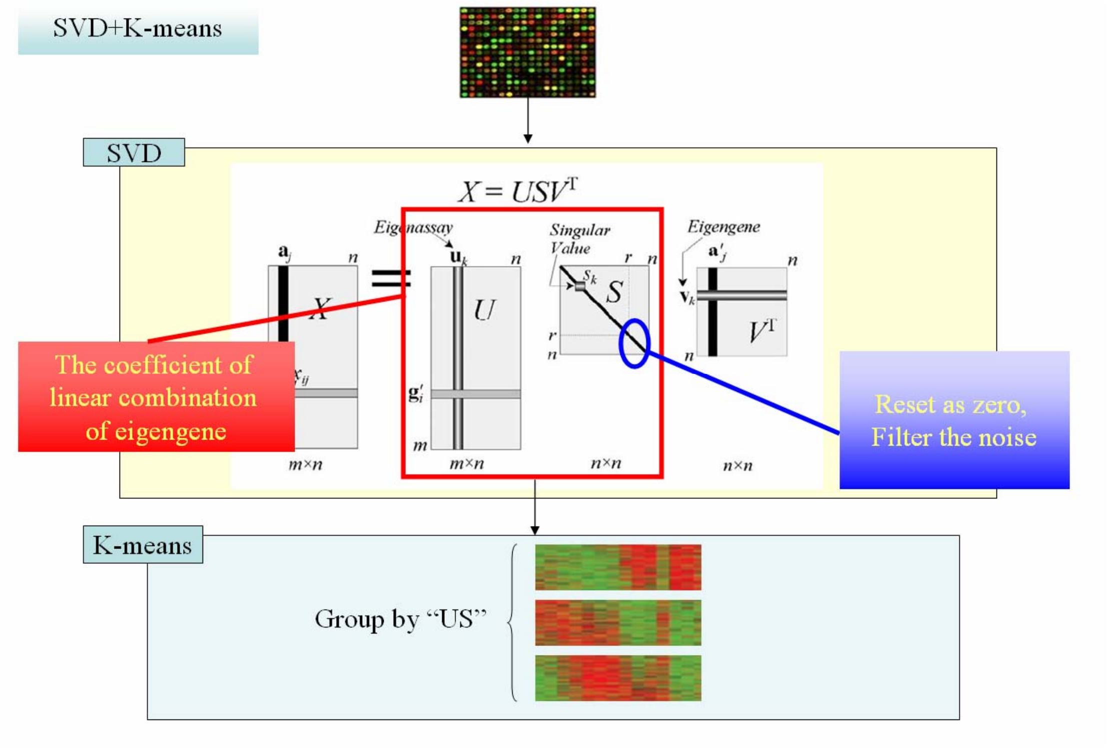


Figure 3.16 The flow chart of stage Finding Orthougous Gene Groups.

The challenge we faced next is how to determine which group of genes has the common expression among both species. Although each of them has already shown good expression resemblance, it is not guarantee that all of the genes share parallel pattern. Moreover, we have to cluster genes two times according to both species which can assure the groups we found are conserved in both species.

1st K-means clustering: There are numerous techniques can be used for searching co-expression gene cluster, for example, hierarchical clustering, k-means clustering, diametrical clustering, etc. These unsupervised learning methods are especially useful when we try to search candidate genes from huge amounts of genes expression, such as genome-wide microarray, since the entire gene reaction toward the conditions, treatments, development stages are not yet completely understood. We decide to use K-means clustering in our scheme. See Figure 3.17 (a), we groups gene in species A (in our case, human). By this, human genes in the same group show great pattern homology, however, not in mouse, which can be solved by doing 2nd K-means clustering.

2nd K-means clustering: We perform K-means clustering algorithm again on linked genes belongs to another species (in this case, mouse) and therefore divided the groups, which were firstly grouped and linked, into more specific and conserved parts (see Figure 3.17 (b)).

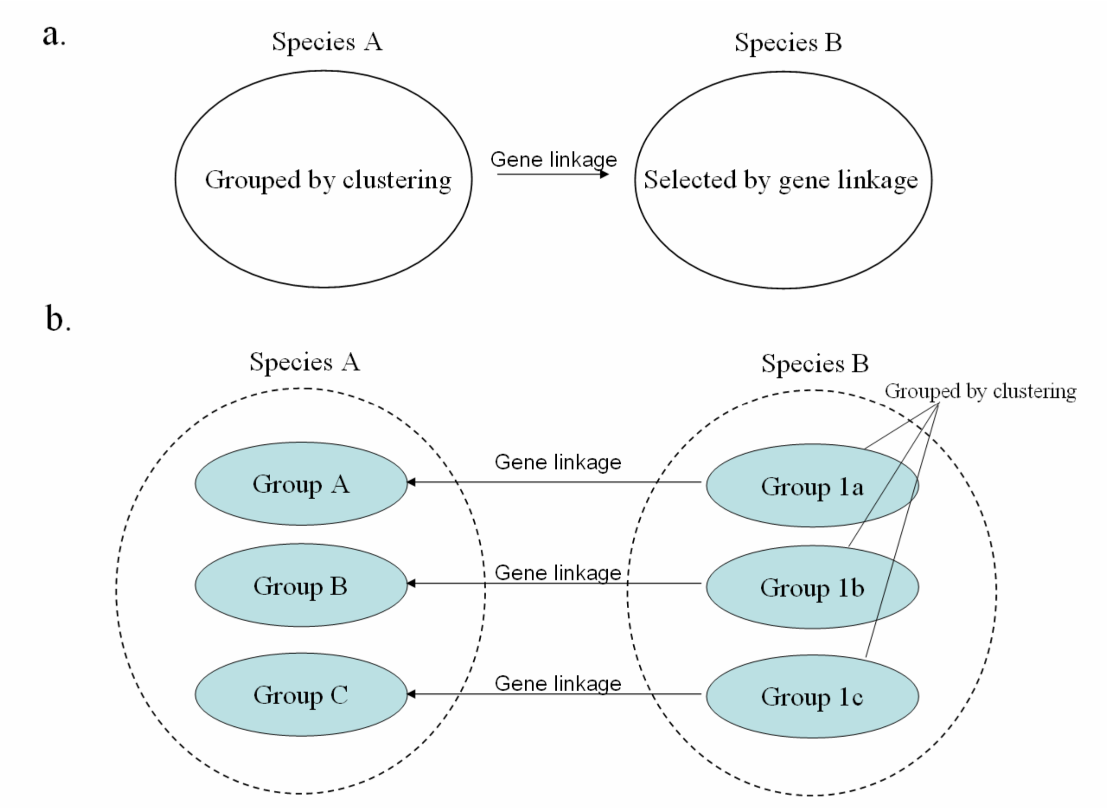


Figure 3.17 (a) Group the genes in species A to find genes share similar expression pattern and select the genes in species B by the mappings according to gene linkage. (b) For each group of genes found in (a), perform clustering again on species B in order to divide the selected genes of species B into smaller groups which share resembling pattern. Then follow the gene linkage again to re-construct the relationship.

Group-dynamic time warping: When genes are grouped into k_1*k_2 clusters by clustering technique, the next step is to discover the same expression pattern among multi species. After grouping genes, each group of gene is regarded as a pair (see Figure 3.18). Take all of the pairs as the input of group-DTW algorithm and rank each pair with their warping distance. The concept is similar to one-on-one DTW, however this time we are doing group-DTW, which align two groups of genes. Two groups of genes, each group belong to a species, if there expression profiles are followed certain rules or patterns, even though their sampling time points and growth rate are different, DTW would consider all the time points alignment possibility and let two time series comparable by the best warping path. According to the information provides by DTW, we can figure out which group of genes shows better orthologous expression.

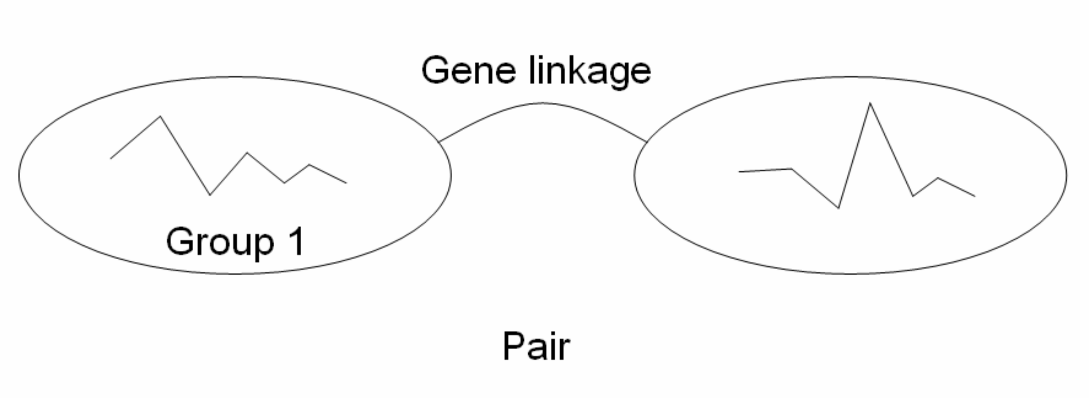


Figure 3.18 Pair contains two set of genes associated by gene linkage. Each set belongs to a species and builds the corresponding relation by gene linkage.

Next, we are going to assign functions to these groups of genes.

3.4.4 Annotations and statistics analysis

Although in some situations, biologists already know a subset of genes involved in certain biological pathway of interest; however, as shown in Yi's research[23], the gene annotated with GO terms[15] ranging from 25.1% in *Danio rerio* to 96.2% in *Saccharomyces cerevisiae*, also, the gene annotations by MIPS[24] or Biomax is ranging from 44.3% in *Thermoplasma acidophilum* to 73.1% in *Bacillus subtilis* 168[25], the functional roles of each gene are not thoroughly studied. So, we are not supposed to only focus on these annotated genes. Many useful approaches show great accuracy when predicting gene functions. Scientists use hierarchical clustering, Hidden Markov models, SVM, FNC, and so on, to predict the functions of genes.

Fuzzy nearest cluster: We chose FNC to be the prediction approach of our scheme, since it combines both clustering and classification and display great prediction accuracy. We use FNC to predict gene functions on both species. Every GDS

Affymatrix gene chip data set of human and mouse is fetched from GEO and processed (please see 3.1.3). We search each genes in GO to know their functional term. Any gene without GO term will be predicted by FNC throughout thousands of experimental data set. The final predication will count on the voting of these thousands data set. Top three GO terms will be annotated to this gene.

By doing functional predication in human and mouse, we are now having the ability to annotate and analyze the orthologous gene group found by previous steps.

Known/predicted GO term annotation: Now, genes in each cluster, which share both sequence and expression similarity can be annotated by known GO term and predicted GO term by FNC. We have know that the co-expressed genes can be inferred to be coding for proteins that partake in common biological function. It is important to understand that these genes are not just co-expressed, they also conserved in sequence and expression in evolution path. By the annotation, we can infer the functions of orthologous gene groups according to the statistical test, which makes the inference more reliable and promising.

Hypergeometric testing: To examine the biological significance of the pairs (ortholgous gene groups); the known and predicted GO term annotation is take into account. Genes will be calculated the p-value of GO terms by hypergeometric distribution:

$$p = 1 - \sum_{i=0}^{k-1} \frac{\begin{pmatrix} M \\ i \end{pmatrix}\begin{pmatrix} N-M \\ n-i \end{pmatrix}}{\begin{pmatrix} N \\ n \end{pmatrix}}$$

In this equation, N is the total number of gene product [15] of platform species, can be regarded as background. M is the number of genes within the background which are annotated to interested GO term. n is the number of genes within the same specified group.

A pair with relatively low p-value implies that these genes gathering together shows significant biological meanings, which suggest that the function of this group is annotated by this specific GO term.

3.4.5 Visualization

In order to provide better understanding and analysis of the results, we design a web sever and implement several useful functions in our program package. We generate many kinds of files containing useful information, including warping path, expression level plot, p-value, and so on. The package gives many delicate illustrations (Figure 3.19). We also demonstrate great ability leveraging other tools, like Genesis and grphwarp by generating files for the software.

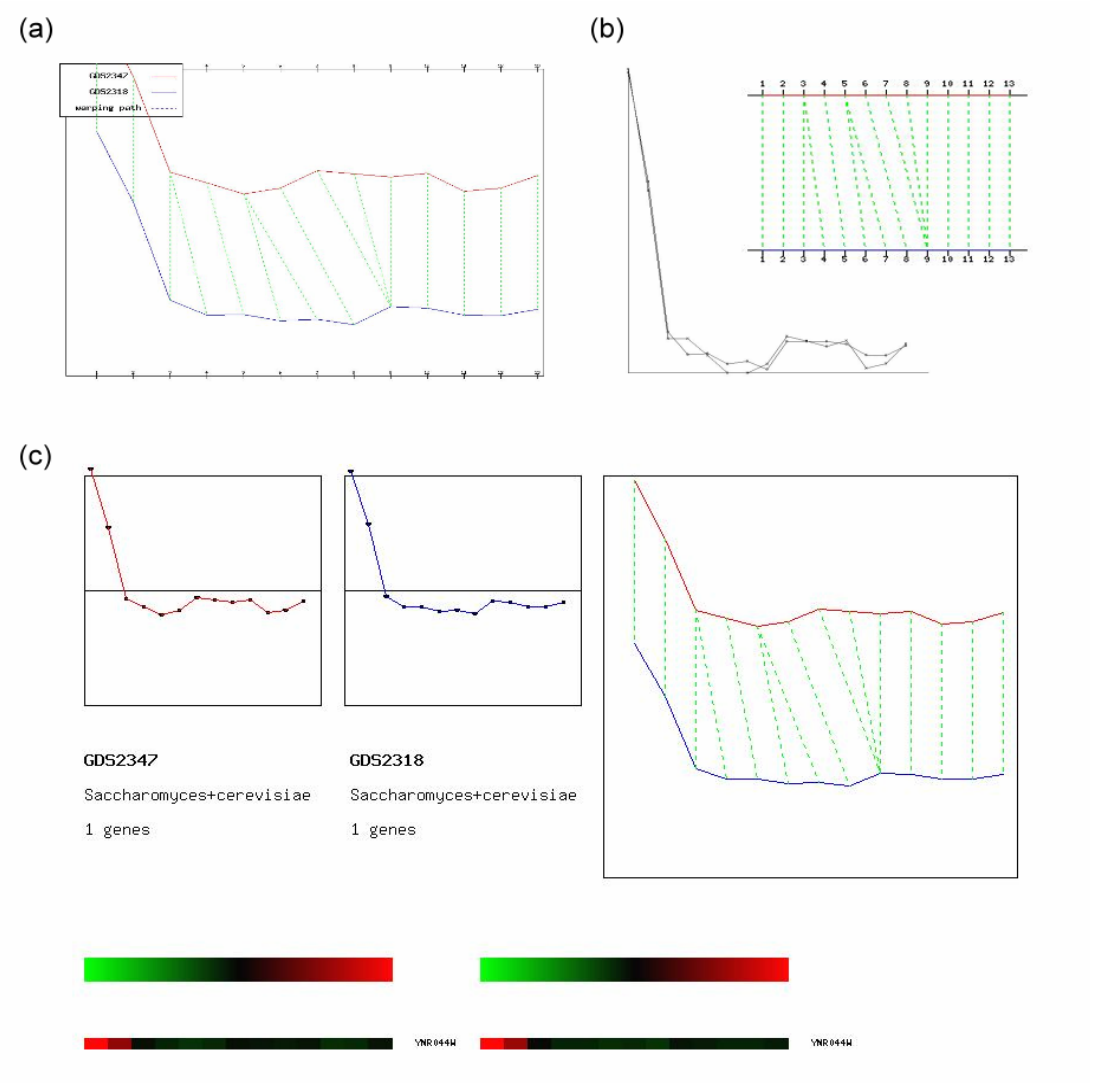


Figure 3.19 (a) Warping path display. (b) Another way to understand warping path. (c) An overview containing profile patterns and the expression level.

Chapter 4 Implementation

This scheme is developed carefully into a package containing binary and source code, which is easy to reconstruct in all platform. This package is available on our website.

4.1 Database

All of the data retrieved from public domain, are reconstructed and organized in sever and manage by MySQL database system.

4.1.1 GEO dataset

We fetched the whole dataset from GEO, including 1790 human dataset and 1623 mouse dataset and 5519 dataset for all species. Our GEO database schema is described in figure 4.1.

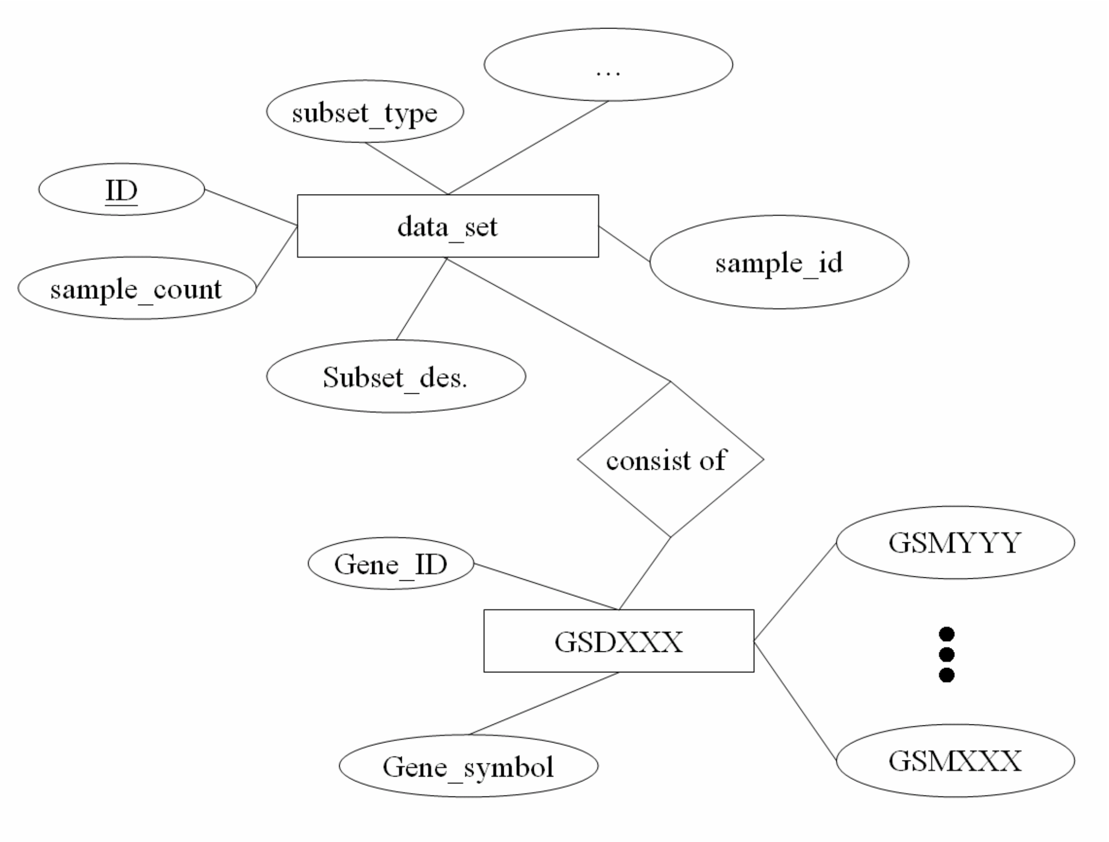


Figure 4.1 Database scheme of mirror GEO.

The program assesses each expression profile by querying mysql database with statement that specified its GDS ID storing in attribute sample_id of data_set. By this schema, FNC can easily scan through every entry in GEO dataset. Each expression profile are formatted and standardized during the process of FNC and Time warping.

4.1.2 GO dataset

GO had already provided organized database dumping file for users to download and import to their servers. In our scheme, we focused on the GO term at the level of four including cell-cycle category, since cell-cycle category is also defined in MIPS, which let our prediction comparable to other approaches.

4.1.3 Homologene dataset

Homologene database curated mapping information between genes of different species. The linkage information of human and mouse is presented in the form of text file, which can be easily access by file I/O operation in computer language.

4.2 Implementation of time warping

4.2.1 DTW core

The core program of DTW is adapted from BTW (Boltzmann Time Warping) web server. We take off the Boltzmann pair probabilities estimation sub-routine and maintaining group and one-on-one DTW algorithm in it. DTW is implement like sequence alignment considering weight function according to time periods. We also implement the semi-global alignment algorithm into the core, which just is a modification of normal DTW with different initiation condition. See Figure 4.2.

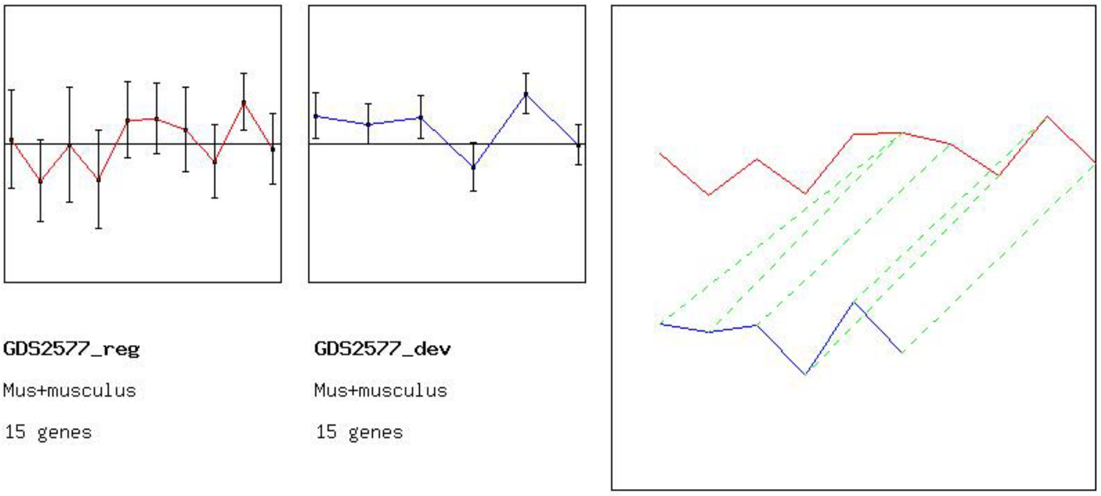


Figure 4.2 Demonstration of semi-global DTW.

However, because of the dynamic programming nature of the algorithm, the distance function should be able to be calculated in this means (dynamic programming). We will discuss this in next section.

4.2.2 Group-Euclidean distance function

Considering original formula of Euclidean distance function, and there are two vectors, A (a$_1$, a$_2$, a$_3$, a$_4$) and B (b$_1$, b$_2$, b$_3$, b$_4$), the distance between A and B is $\sqrt{\sum_i (b_i - a_i)^2}$. However, this sort of formula is not suit for dynamic programming nature of DTW algorithm. The formula should be modified as $\sum_i \sqrt{(b_i - a_i)^2}$. It is obvious that the distance is no longer Euclidean distance. However, in single-gene warping, this is not a problem, if we do not sparing the distance when calculating single gene distance, so $\sum_i (b_i - a_i)^2 = \sum_i (b_i - a_i)^2$. This problem can be more serious when calculating group-Euclidean distance. That is, the distance function design for DTW has its limitation; we will discuss this in chapter discussion.

4.3 Implementation of SVD

4.3.1 SVD core

The function doing SVD is based on a routine by Forsythe et al., which is in turn based on the original routine of Golub and Reinsch[26], found, in various forms, in Wilkinson and Reinsch, in LINPACK, and elsewhere. These references include extensive discussion of the algorithm used. In our implementation, the parameter of the function is adapted to our specific usage.

SVD is a series of matrix operations. Here, we take a low-dimension example as our demonstration materials. Consider matrix A = $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$. The first step doing SVD is computing the eigenvalue of matrix A^TA. So we implement three matrix operations: transpose, multiplex, and Gauss-Jordan elimination (containing elementary row operations) to complete this step. In this example, we got three eigenvalue of A^TA, λ_1=3, λ_2 = 3, and λ_3 = 3. Therefore, matrix A has singular value $\sigma_1 = \sqrt{\lambda_1}$ = $\sqrt{3}$, σ_2 = $\sqrt{3}$, and σ_3 = $\sqrt{3}$. So, we got matrix S = $\begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \\ 0 & 0 & 0 \end{bmatrix}$. With the information of eigenvalue, we can compute corresponding eigenvectors V(λ) which are the basis of *ker*(A^TA- λ). In our example, V = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Next, the most complicate part is computing matrix U, since u_i = $\frac{1}{\sigma_1}$ A v_i, we still only need mutiplex operation to solve this part. Here, u_1= $\begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{-1}{\sqrt{3}} \\ 0 \end{bmatrix}$.

Because we only have three singular values, but U has four dimension in this case, so we need to compute an additional orthonormal basis. In our implementation, a simple Gauss-Jordan elimination can help to generate a solution; remember that this basis is not unique. By these operations, SVD can be done in polynomial time.

An expression profile which had been normalized and standardized is taken as input, and the diagonal matrix S is the only thing we concern in following step doing noise filtering.

4.3.2 Noise reduction

The diagonal values of S make up the singular value spectrum. The height of any one singular value is indicative of its importance in explaining the data. More specifically, the square of each singular value is proportional to the variance explained by each singular vector. The relative variances are often plotted. The approach we used is proposed by Everitt and Dunn, the approach based on comparing the relative variance $S_k{}^2(\sum_i S_i{}^2)^{-1}$ of each component to 0.7/n, where n is the number of time point of the expression profile.

The S matrix outputted by our SVD function, was calculated the relative variance and filter out any eigengene with relative variance lower than 0.7/n.

4.4 Implementation of FNC

All of the programs are developed under Linux system using C++ with STD library. The program automatically fetched human expression profile one at a time, and FNC on it. The prediction results were stored and summed up, after thousands of prediction on the same genes but different samples, we can get a more reliable result of the function predication.

4.4.1 Mining by unsupervised approach

The algorithm of this part is described in previous chapter. We maintained a class to store entire expression profile, which includes also kind of operations, such as standardization and normalization. We leveraged the function in Cluster3[27] to do the hierarchical clustering according to known genes' GO annotations and prune the edges between clusters, of which have relatively low correlations to other clusters just like the algorithm says.

4.4.2 Predicting category by classification methods

When each GDS profile has been processed by the approach described in last section, the program will maintain a cluster profile to represent the average expression pattern of each cluster of each GO term.

Expression profiles of unknown-function genes are read and calculate the Pearson Correlation with every average expression pattern of each cluster of all functions. Correlation coefficient higher than λ will be recorded and summed up after all 1790 dataset were predicted by FNC. The top 3 predicted functions of each gene will be regarded as the final prediction of the FNC.

4.5 Statistical analysis

4.5.1 Hypergeometric testing

The routine calculating p-value of hyergeometric testing is implemented based on binomial coefficient subroutine and garmma function. Since the factorial function used in binomial coefficient subroutine is just a gamma function but offset by one. By gamma function, the time complexity computing binomial coefficient is reduced to O(n).

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

In hypergeometric formula shown above, N is now the total number of genes with known and predicted function, M is the number of genes including predicted genes which are annotated to interested GO term. n is the number of genes within the group. By doing so, the p-value can demonstrate the biological significance of this GO term in the group.

4.6 Implementation of visualization

4.6.1 GD library[28]

GD library is a C language library providing convenient function generating graph according to common file formatted, like JPEG and GIF. By the help of GD library, our package can directly generate graphs to demonstrate simple clustering and warping result.

4.6.2 Genesis and graphwarp

Genesis and graphwarp provide even more information about the clustering, and support more analysis. Our package generate corresponding file format for these two programs. The result is shown in Figure 4.3.

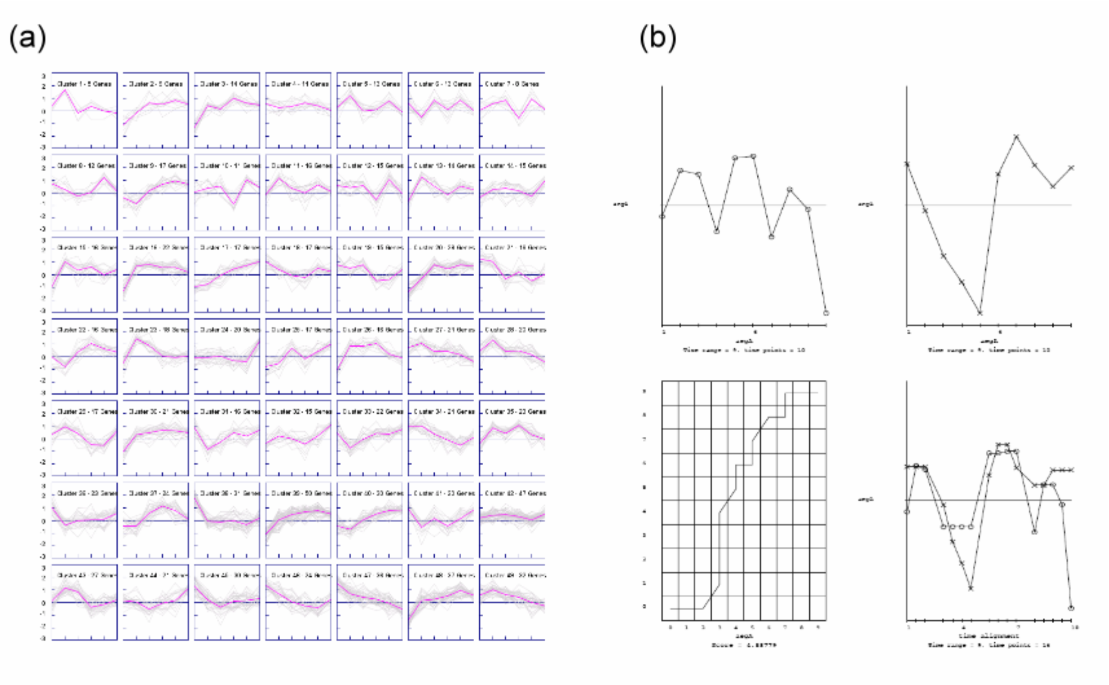


Figure 4.3 Visualization of (a)Genesis (b) graphwarp.

Chapter 5 Results

In this chapter, we will demonstrate the ability of our scheme by both artificial and real data sets and tabulate the predication of FNC in following sections. The real data set contains both homogeneous (both yeast) and heterogeneous (human and mouse) data set.

5.1 Artificial data set

Before we perform our scheme on real experiments, we design a series of artificial expression data to test the essential functions of our package. In our first experiment, we artificially generate three kinds of patterns, see Figure 5.1, the first and second pattern is strictly increasing and strictly decreasing pattern respectively. The third one, we simulate a cell-cycle-regulation-like pattern including two cell cycles (two peaks, in order words). Each of them contains only single pattern and 100 genes. In these three cases, all of them display great performance in our algorithm.

Next, we compare a special case of the third pattern, we eliminate second time points in each peak, which demonstrate a phenomenon of early entering each S stage (we assume that genes are regulated and highly expressed in this stage), and call it pattern mutant-three (Figure 5.2).

Furthermore, pattern mutant-three is compared with pattern three by our scheme, you can see the DTW capture this kind of time shift and present their warping path suggesting that the sampling time in these two time point should be adjusted or modified.

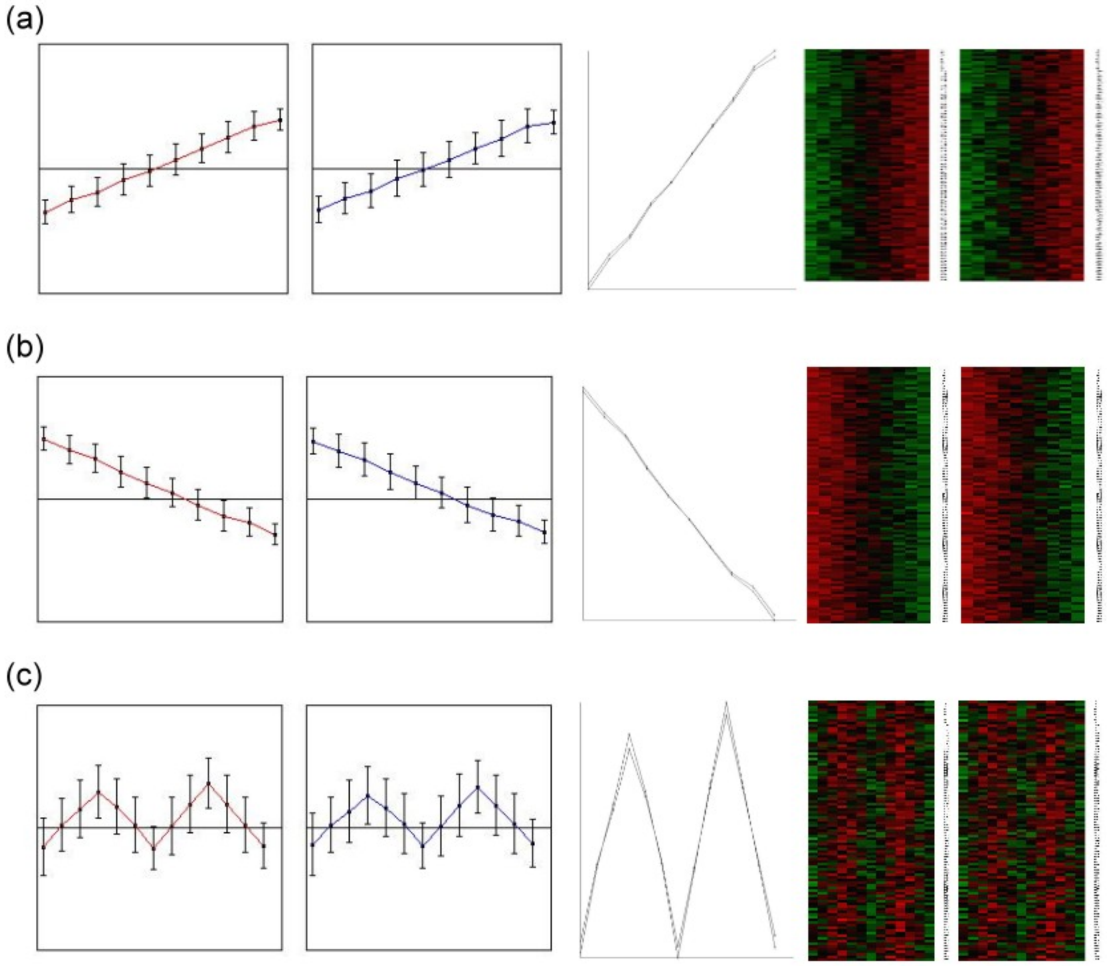


Figure 5.1 Results of (a) pattern one (increasing) (b) pattern two (decreasing) (c) pattern three (two peaks)

Also, if we enable the noise reduction by SVD, you can observe easily that the deviation of expression is dramatically reduced. Although the deviation is a random artifact that we generate on purposed, this demonstration shows how good the SVD can do in noise reduction. See Figure 5.2 (b)

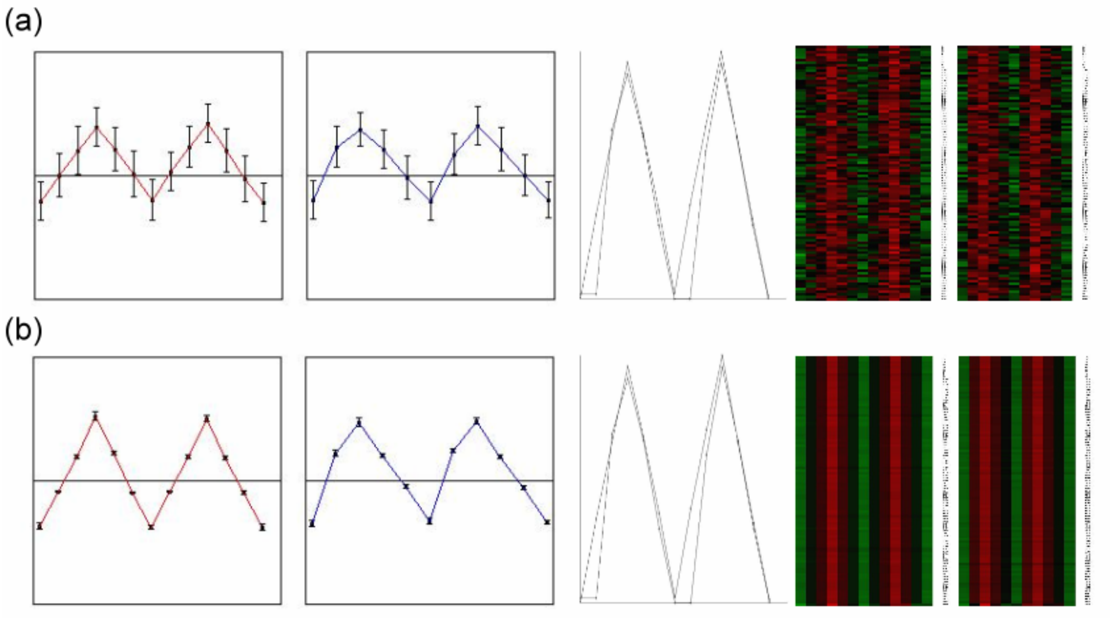


Figure 5.2 (a) Results show a time shift in the beginning of each peak. (b) SVD filters lots of noise.

Finally, we try more complicate data sets, which are composed by two patterns. See Figure 5.3 (a), we denote two data sets, one is data set A, composed by pattern three (cell-cycle-like) and pattern one (strictly increasing), another one is data set B, consisting of a "V" pattern (pattern four), which is similar to the valley between the peaks in pattern three, and pattern one. We set our k=2, and enable semi-global alignment, and hope our system can capture all these difference.

As we expected, the scheme shows great result again, these two patterns are separate automatically (see Figure 5.3 (a)), and semi-global alignment DTW successfully map the "V" pattern to the valley.

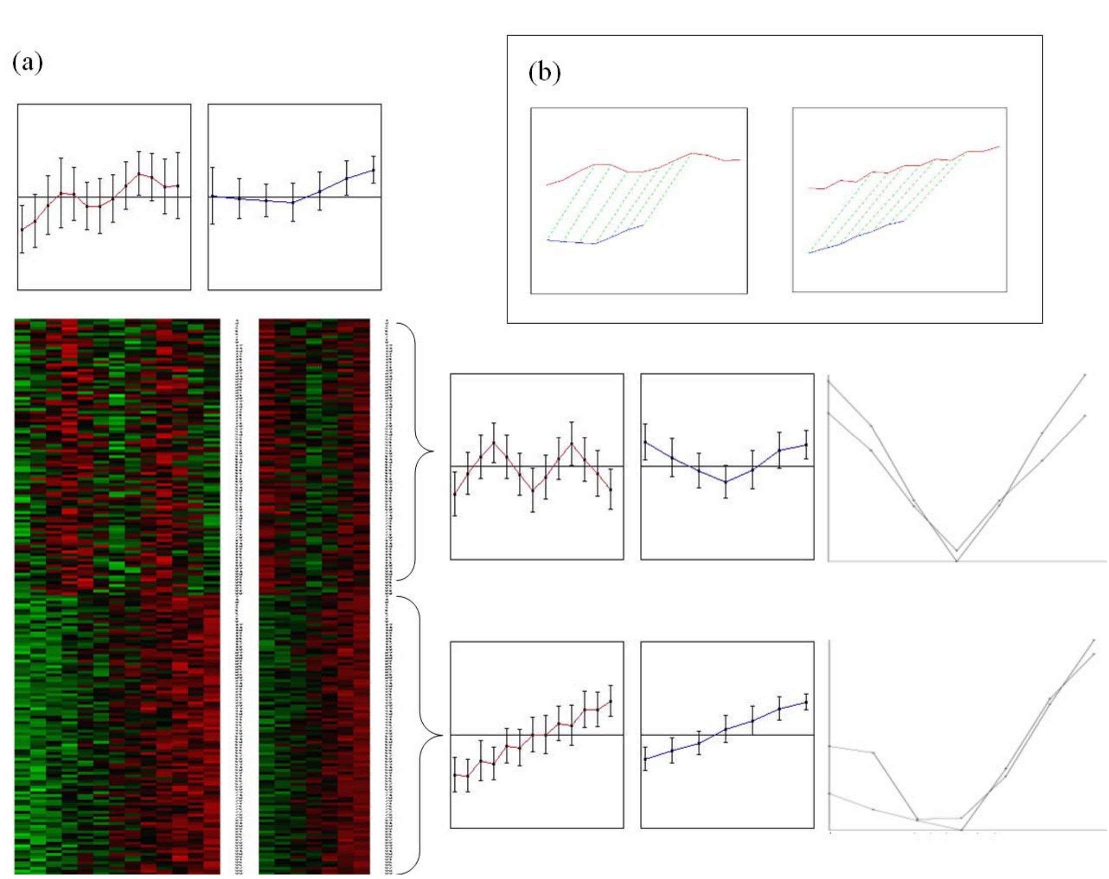


Figure 5.3 (a) Results of mixed patterns. (b) semi-global DTW works well in this case.

These simple artificial data sets, although not covering every patterns in gene expression, has already testified the ability of our core algorithms, DTW, SVD, and k-means clustering.

In the next section, we will try our scheme on a homogeneous data set and considering the whole process we proposed, including the five stages we described in former chapter.

5.2 Yeast data set

The yeast data set is fetch from GDS2318 (yhp1 double mutant across two cell cycles) and GDS2347 (wild type). Comparing these two data sets provides insight into the role of Yox1 and Yhp1 in early cell cycle box-dependent transcription. We analysis them by our package, we choose top 500 genes in single-gene distance ranking, and set K1= K2=4.

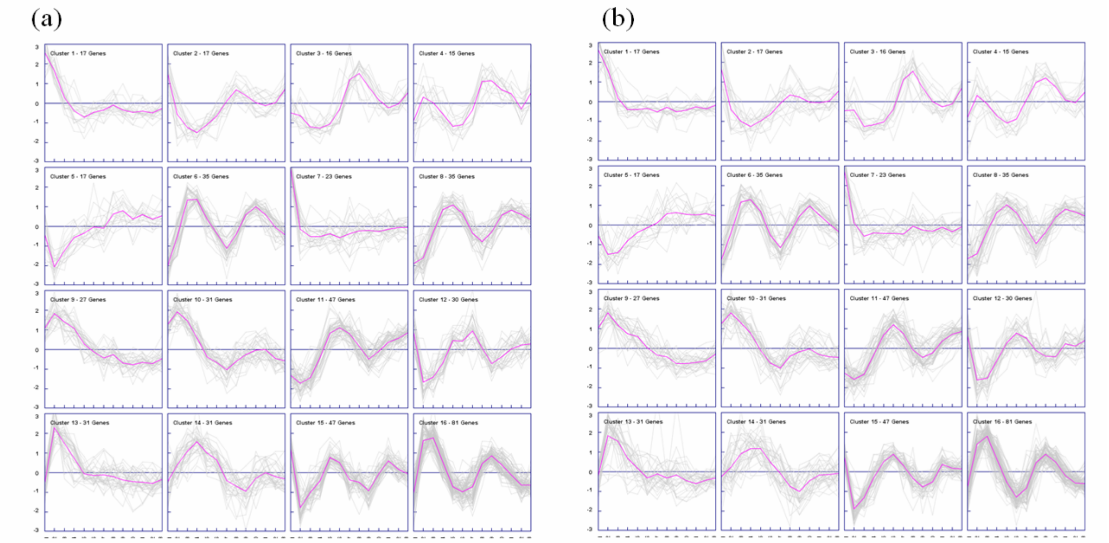


Figure 5.4 Gene clustering of (a) normal Yeast cell, and (b) knock-out Yeast.

Each corresponding cluster shows almost parallel pattern without any time shifting. However, although we have discovered the same changes of cell-cycle in gene CDC20 described in literature (see Figure 5.5), most of the genes are still unaffected toward the knockout of Yox1 and Yhp1. In our analysis, knockout of Yox1 and Yhp1 do affect and regulate some genes, however do not change general cell cycle. It maybe due to other secondary regulation mechanism provoked and worked as rescue and therefore the general cell cycle are not changed.

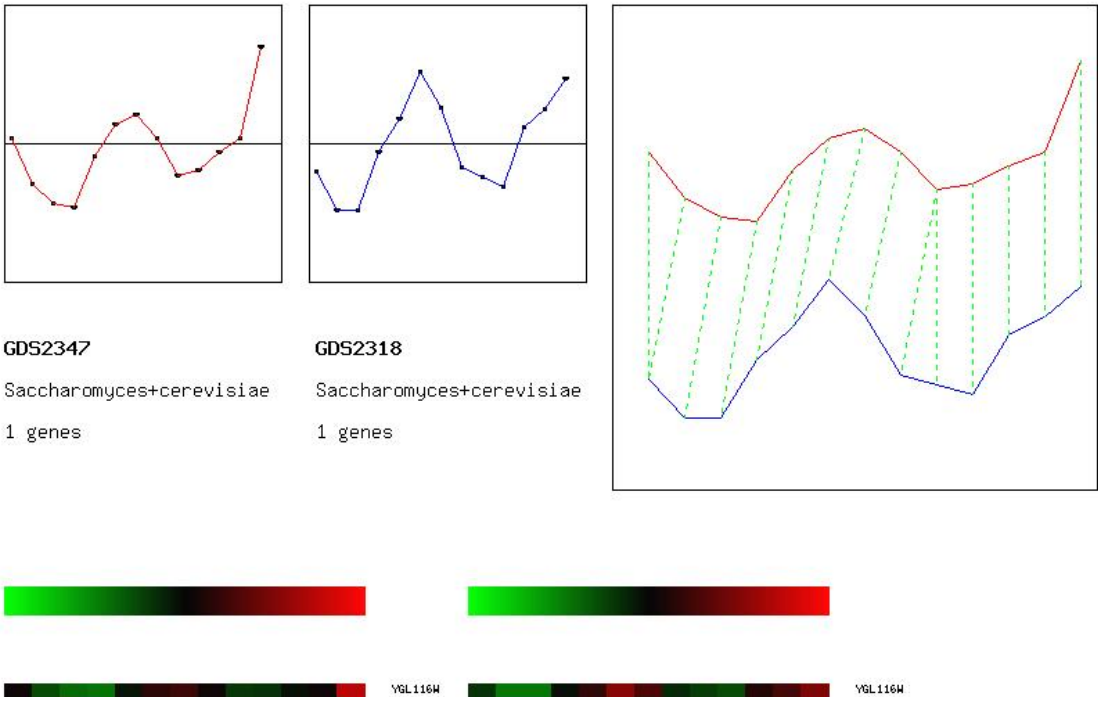


Figure 5.5 CDC20 shows time shift in knockout yeast.

Here, we have show the proposed scheme work well in homogeneous data sets. Next, which is our major purpose, we will discuss the performance when analyzing heterogeneous data sets such as experiments crossed human and mouse.

5.3 Human and mouse fetal liver data set

The data sets of human and mouse fetal liver is sampled through the development from embryonic to birth stage, contributed by Dr.Wang, Chang Gung memorial Hospital.

Development stages mapping between human and mouse is crucial in discovering the development mechanism conserved in both species. However, the cell growth rate is extremely different in this situation, so this is a great material for our scheme. We pick top 500 genes in single-gene ranking, and choose K1 = 5, and K2=4.

We list these fifteen orthologous functional gene groups with their expression profiles, warping path, and GO annotation.

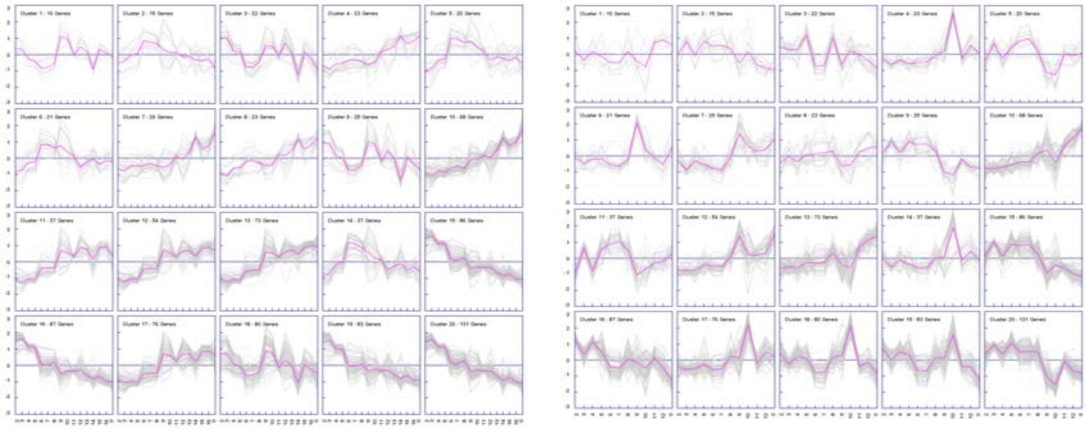


Figure 5.6 Left: Gene clustering of mouse. Right: Gene clustering of human

We at first examine the grouping result by Genesis. Most of the groups show similar expression pattern. Although the results are not as good as previous experiments, however, it has demonstrated great ability finding orthologous functional gene groups in cross-species model.

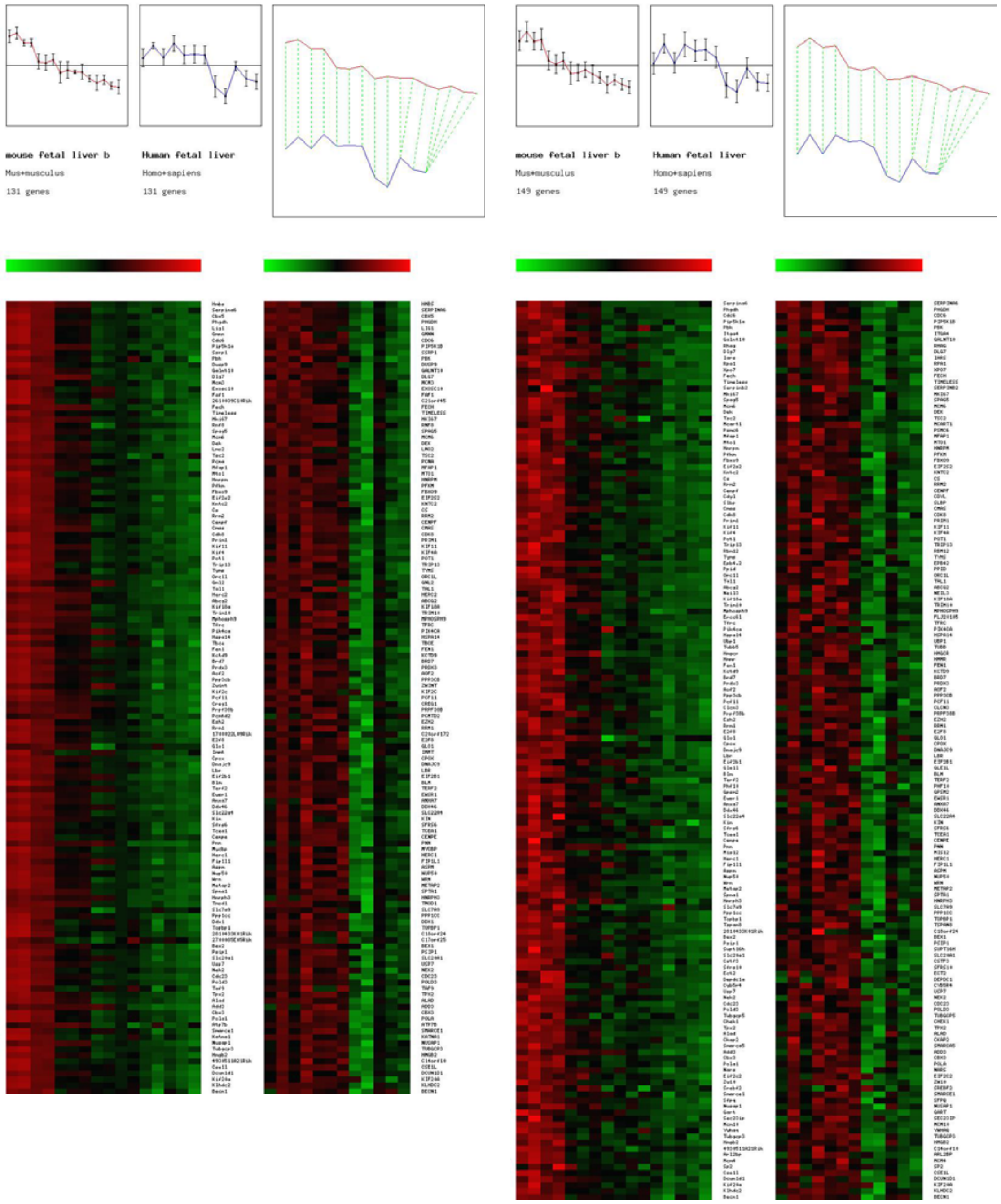


Figure 5.7 Result of un-SVD(right) genes comparing to genes with SVD(left).

In Figure 5.7, we demonstrate again the effects of SVD. If we do not enable SVD in our system, this cluster will still be gathered together (103 genes overlapped with genes in the cluster with SVD), however, you can see the deviation of expression is reduced and therefore being easier in interpreting the pattern.

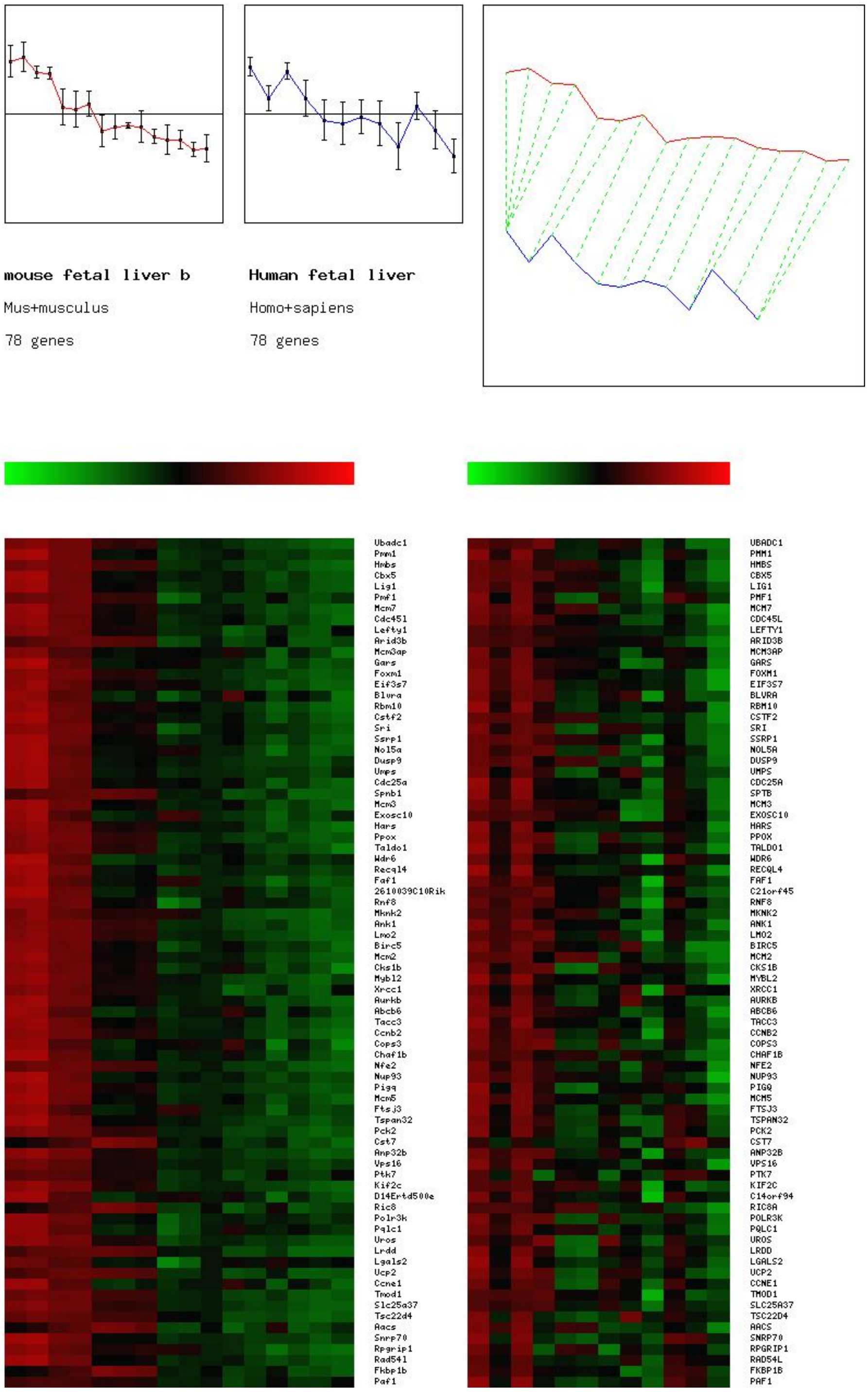


Figure 5.8 Case 1: genes highly expressed in early stages.

We will discuss two clusters that display shortest warping distance and good statistical significance. Before that, we can observe in twenty clusters that all of the warping paths suggest samplings should be extended in the beginning and the end of the human data set. In this cluster, containing 78 genes, GO terms (see table 5.1.) that show great biological significance (p-value < 0.001) are focusing on cell-generation-related functions. We can infer that these genes which are highly expressed in the early stages of development play important role in increasing the quantity of cells in liver. Their functions include cell division (11.54%) and cytoplasm (12.82%), which show many cells are generated. Also, cell cycle and DNA binding and response to DNA damage stimulus are all highly associated with this inference. Therefore, by this evidence, this functional gene cluster is highly reliable and convincing that it is an orthologous gene cluster.

Table 5.1 GO term with high biological significance in case 1.

Rank	GO term	probability	p-value
1	cell cycle	16.03%	6.49E-13
2	cell division	11.54%	2.09E-12
3	DNA binding	15.38%	3.66E-06
4	cytoplasm	12.82%	5.07E-06
5	response to DNA damage stimulus	3.85%	0.000147

Then, we try to interpret the information gives in second cluster. Genes in this cluster are highly expressed in the late stage of development. What draws one's attention is that GO terms (see table 5.2.) are suggesting differentiation and biosynthesis related functions, which implies liver is now activating and more specified in these stages. Their functions include endothelial and epithelial cell differentiation (around 5.5%), which show blood vassals are now developed. Also, biosynthesis (4.63%) means the blood starts to circulate in the liver and the liver start to trigger its functions in biosynthesis.

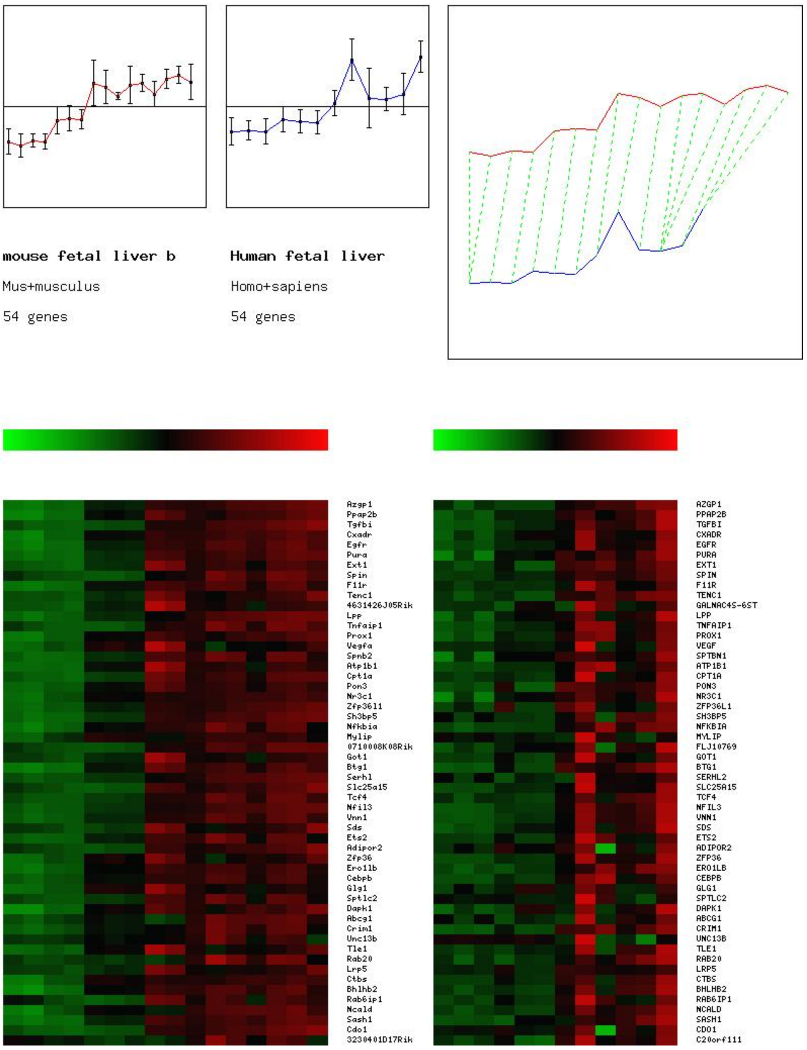


Figure 5.9 Case 2: genes highly expressed in later stages.

Table 5.2 GO term with high biological significance in case 2.

Rank	GO term	probability	p-value
1	cell proliferation	12.04%	2.15E-06
2	cytoplasm	17.59%	5.41E-06
3	DNA binding	18.52%	2.44E-05
4	biosynthesis	4.63%	4.89E-05
5	endothelial cell differentiation	1.85%	0.000147619
6	epithelial cell differentiation	3.70%	0.000150824
7	transcription corepressor activity	3.70%	0.000321868
8	calmodulin binding	4.63%	0.000858941

There are another cluster intriguing our notices, Genes in case 3 (see Figure 5.10), which are highly expressed latter than genes in case 2 display biological significance (see Table 5.3) in extracellular space (23.61%), membrane (41.66%), and immune response (13.88%). It can be infer that after liver started to trigger their biosynthesis function and bloods began to circulate in liver, the liver are now busy in outer cell signal transduction involving membrane proteins and extracellular space and developing immune system in liver.

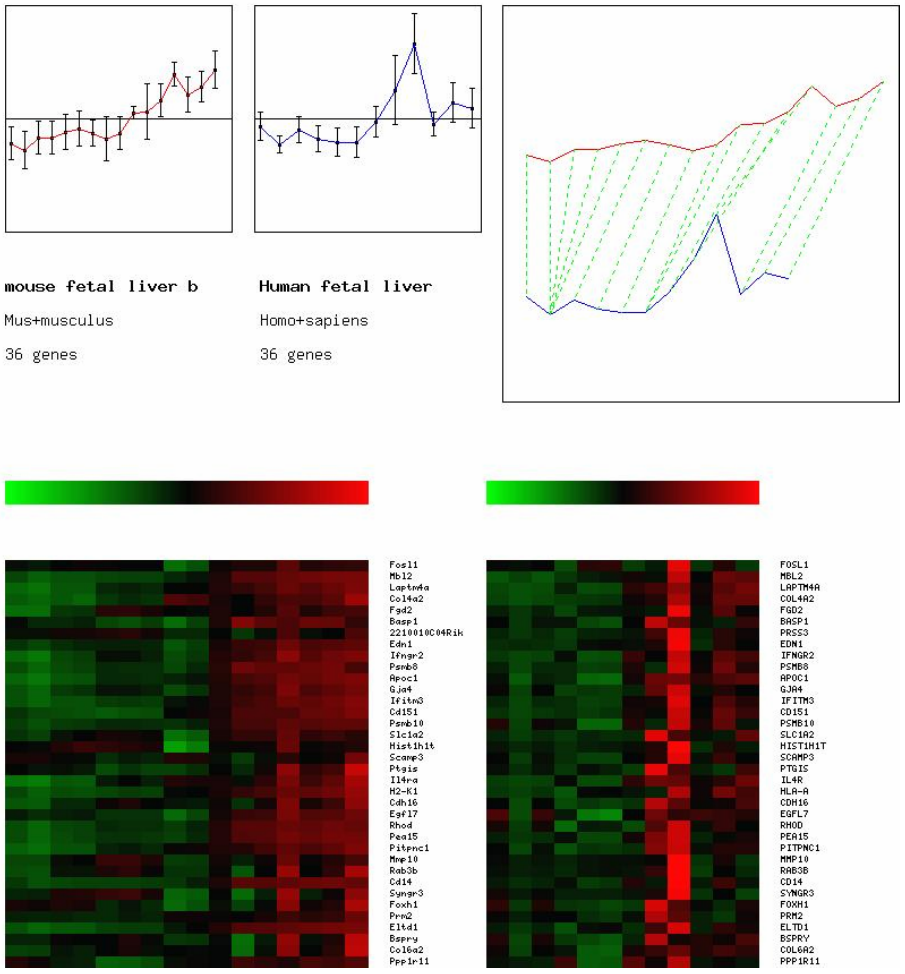


Figure 5.10 Case 3: genes highly expressed later than genes in case 2.

Table 5.3 GO term with high biological significance in case 3.

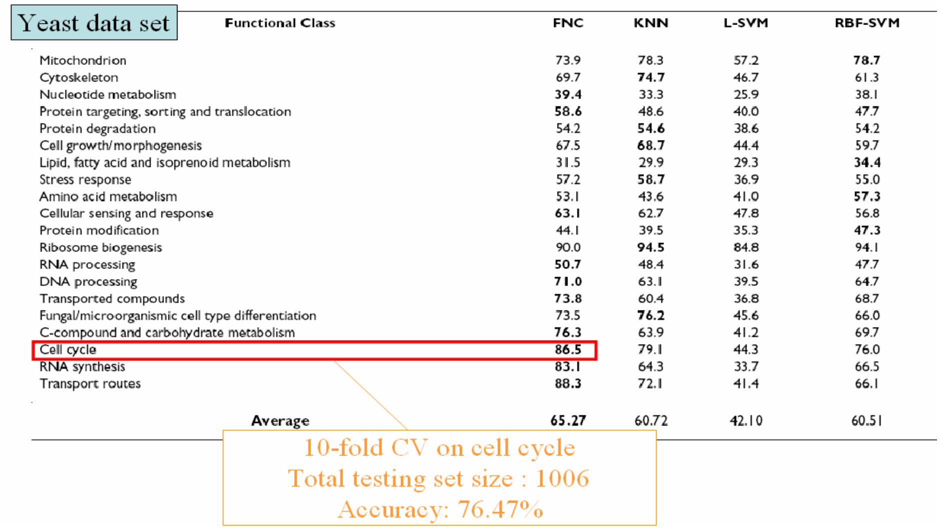
Rank	GO term	probability	p-value
1	extracellular space	0.236111	1.64E-07
2	proteasome core complex (sensu Eukaryota)	0.055556	1.26E-05
3	immune response	0.138889	2.31E-05
4	membrane	0.416667	8.11E-05
5	collagen	0.055556	0.000281
6	extracellular matrix structural constituent conferring tensile strength	0.027778	0.000912

By these case studies, it exhibits great performance in analyzing heterogeneous data sets and suggesting reliable functions to orthologous gene groups found.

5.4 Gene function prediction

This section presents the predication result of FNC, and we will discuss the result in chapter discussion.

To verify the prediction accuracy, we design a 10-fold cross-validation on GO term cell cycle from 1006 human data sets. The average accuracy is 76.47%. Although it is lower than the performance in Yeast, FNC still shows good predication ability.



Yeast data set

Functional Class	FNC	KNN	L-SVM	RBF-SVM
Mitochondrion	73.9	78.3	57.2	**78.7**
Cytoskeleton	69.7	**74.7**	46.7	61.3
Nucleotide metabolism	**39.4**	33.3	25.9	38.1
Protein targeting, sorting and translocation	**58.6**	48.6	40.0	47.7
Protein degradation	54.2	**54.6**	38.6	54.2
Cell growth/morphogenesis	67.5	**68.7**	44.4	59.7
Lipid, fatty acid and isoprenoid metabolism	31.5	29.9	29.3	**34.4**
Stress response	57.2	**58.7**	36.9	55.0
Amino acid metabolism	53.1	43.6	41.0	**57.3**
Cellular sensing and response	**63.1**	62.7	47.8	56.8
Protein modification	44.1	39.5	35.3	**47.3**
Ribosome biogenesis	90.0	**94.5**	84.8	94.1
RNA processing	**50.7**	48.4	31.6	47.7
DNA processing	**71.0**	63.1	39.5	64.7
Transported compounds	**73.8**	60.4	36.8	68.7
Fungal/microorganismic cell type differentiation	73.5	**76.2**	45.6	66.0
C-compound and carbohydrate metabolism	**76.3**	63.9	41.2	69.7
Cell cycle	**86.5**	79.1	44.3	76.0
RNA synthesis	**83.1**	64.3	33.7	66.5
Transport routes	**88.3**	72.1	41.4	66.1
Average	**65.27**	60.72	42.10	60.51

10-fold CV on cell cycle
Total testing set size : 1006
Accuracy: 76.47%

Figure 5.11 Verified the prediction accuracy of FNC in human.

5.4.1 Human gene prediction

We utilize FNC to predict the functions of each function-unknown gene on 1006 human data sets fetch from GEO. The predication result is helpful when we annotate and infer the function of the cluster we found. Please see table 5.3, 5.4.

Table 5.4 GO term predicted in human.

Rank	GO name (predicted)	# of gene (predicted)	# of gene (known)	Rank (know)
1	response to drug	2405	26	115
2	eukaryotic translation initiation factor 3 complex	1790	12	193
3	translation factor activity, nucleic acid binding	1691	10	218
4	cholesterol homeostasis	1679	10	218
5	cytokine binding	1552	9	236
6	response to oxidative stress	1518	94	48
7	pregnancy	1371	63	66
8	oligopeptide transporter activity	1367	6	297
9	cytoskeletal protein binding	1218	33	101
10	activin receptor complex	1126	2	428

Table 5.5 Known GO term in human.

Rank	GO name (known)	# of gene (known)	# of gene (predicted)	Rank (predicted)
1	membrane	4668	75	99
2	metal ion binding	2217	688	21
3	signal transduction	2081	6	307
4	intracellular	1640	873	16
5	cytoplasm	1548	7	294
6	DNA binding	1084	341	36
7	transport	1048	1367	8
8	cell cycle	1006	1	443
9	plasma membrane	753	158	62
10	immune response	733	84	91

5.4.2 Mouse gene prediction

Since this paper is focusing on human and mouse, we perform FNC on mouse data sets (1133) from GDS as well. Please see table 5.5, 5.6.

Table 5.6 GO term predicted in mouse.

Rank	GO name (predicted)	# of gene (predicted)	# of gene (known)	Rank (know)
1	receptor binding	846	82	52
2	learning and/or memory	720	9	216
3	ATPase stimulator activity	641	2	400
4	biotin binding	630	5	287
5	GTPase activator activity	562	266	19
6	oxidoreductase activity, acting on CH-OH group of donors	513	4	315
7	response to temperature stimulus	458	4	315
8	outer membrane	454	6	262
9	protein binding, bridging	442	9	212
10	positive regulation of enzyme activity	441	12	180

Table 5.7 Known GO term in mouse.

Rank	GO name (known)	# of gene (known)	# of gene (predicted)	Rank (predicted)
1	membrane	5880	185	43
2	transport	3202	239	29
3	intracellular	2280	0	--
4	metal ion binding	2203	53	113
5	DNA binding	2122	53	113
6	signal transduction	2104	40	147
7	extracellular space	2062	0	--
8	cytoplasm	1151	33	175
9	cell cycle	742	6	349
10	RNA binding	576	119	37

5.5 Web server

5.5.1 Twins overview

TWins aims to give an institutive and useful service for users to make their experiments comparable with each others. Twins provides array-wide time warping by pre-processing by k-means clustering and gives GO annotations to help infer the function of grouped genes. Fig. 5.2 demonstrates the system flow of the system of TWins. When users upload their expression profile with time series, specify the organism, and submit their request, the system will perform array-wide or conventional (depends on user selections) DTW on these two profiles. The results of DTW will be presented in a graphical interface (cooperates with grphwarp[1]) and TWins will provide grouped gene expression files for downloading. The system collected diverse cell-cycle profiles of species under different conditions from NCBI GEO, and users can upload their own data to compare with these precious experiments. Further, by the helps of mapping table extracted from SwissProt, any experiments with gene symbols names can thereby annotated by GO terms. Each GO term will be calculated its P-value by the hypergeometric distribution to suggest its biological significance. The web interface is interactive and friendly to users, provides useful functions but simple manipulations, the core program, including time warping, GO term matching, and p-value calculating procedures are all follow GNU open source copyright, the completed program package is available at the website.

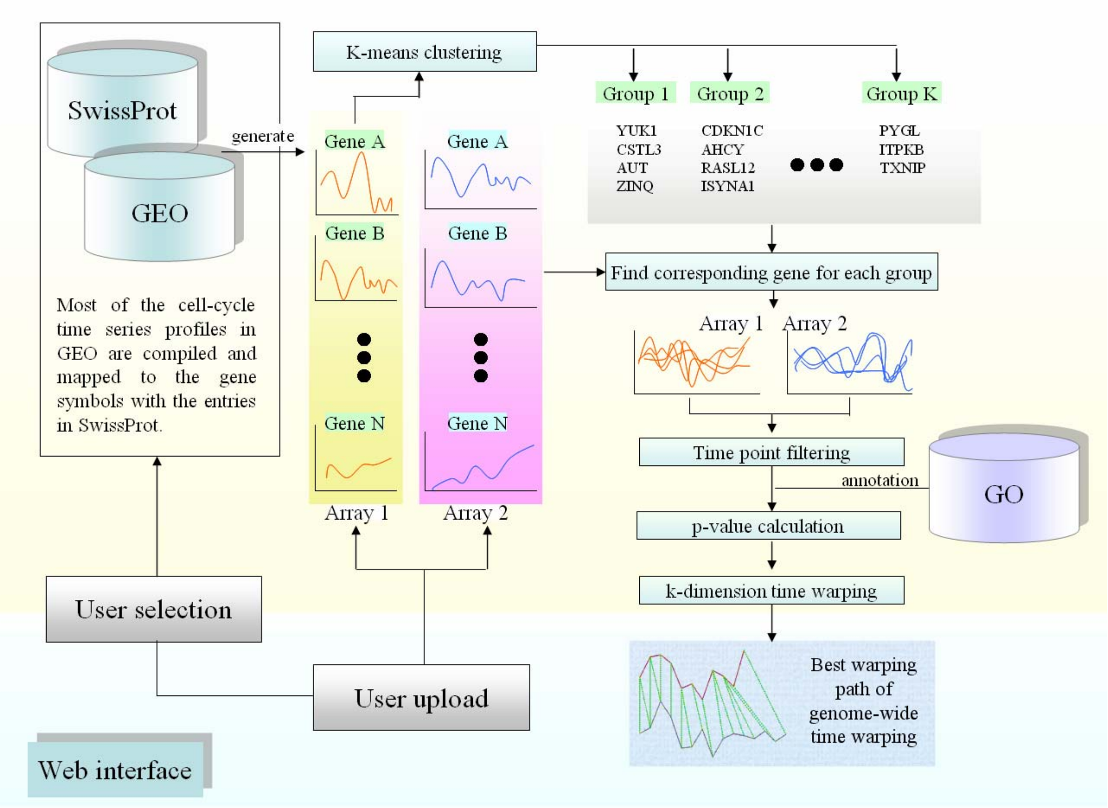


Figure 5.12 TWins system flow

Table 5.8 A List of cell-cycle profile fetched from NCBI GEO

GDS ID	Species	Condition	Time point	Platform	Feature	Reference
GDS39	Saccharomyces cerevisiae	1	14	GPL59	7680	[6]
GDS124	Saccharomyces cerevisiae	1	24	GPL62	8832	[6]
GDS400	Homo sapiens	3	4	GPL91	12651	[29]
GDS449	Homo sapiens	5	4	GPL91	12651	[30]
GDS586	Mus musculus	1	8	GPL81	12488	[31]
GDS587	Mus musculus	1	7	GPL83	11934	[31]
GDS845	Homo sapiens	3	3	GPL550	20163	[32]
GDS846	Homo sapiens	3	3	GPL550	20163	[32]
GDS847	Homo sapiens	3	3	GPL550	20163	[32]
GDS848	Homo sapiens	3	3	GPL550	20163	[32]
GDS922	Saccharomyces cerevisiae	2	3	GPL90	9335	[33]
GDS1409	Mus musculus	4	4	GPL339	22690	[34]
GDS1515	Arabidopsis thaliana	4	3	GPL198	22814	[35]
GDS1627	Homo sapiens	8	3	GPL550	20163	*
GDS1710	Homo sapiens+Mus musculus	1	3	GPL2677	5376	[36]
GDS 1875	Homo sapiens	5	9	GPL1528	22178	[37]
GDS2053	Mus musculus	2	3	GPL32	12654	[38]

* Not listed in GEO

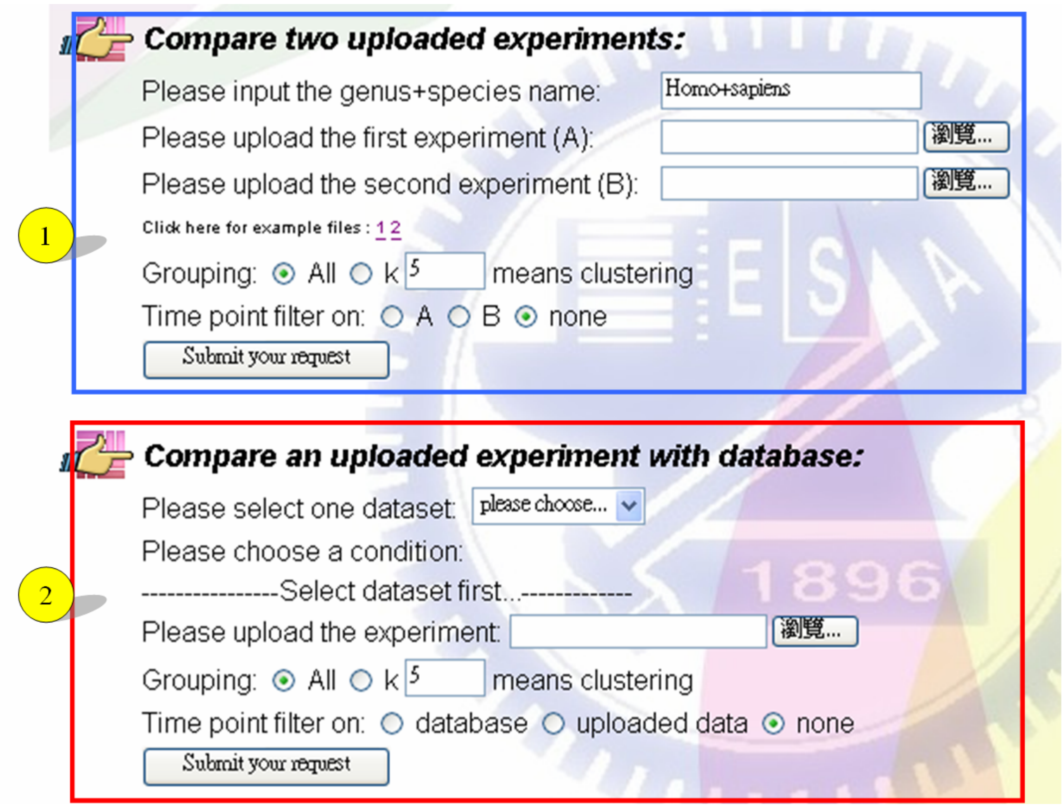
Table 5.9 Characteristics of TWins.

Comparing features	**TWins**	**genewarp**	**BTW**	**GenTxWarper**	Descriptions
Programming language	**C++**	C++	Perl & c++	JAVA	Implement by efficient language
Platform	**All**	Win32	All	All	Great compatibility
Web server	**Yes**	-	Yes	-	Provide convenient access
Database supported	**NCBI GEO**	ExpressDB[39]	Cho's[40, 41]	-	Allow users to compare their experiments with others in database
Species number	**4 (mouse, human, Arabidopsis, and yeast)**	1 (yeast)	2 (yeast and human)	-	Supporting cell-cycle profiles of diverse species
Graphic	**Web interface + grphwarp**	grphwarp	Web interface	Jave application	Provide abundant visualization
GO annotation	**Yes**	-	-	-	According to the given gene symbol name and organism name, Twins can provide GO term name for each gene
p-value	**Yes**	-	-	-	Indicate the biological significance of certain GO term happened in user's data
Feature vector	**Yes**	Yes	-	Yes	Acting DTW on more than one gene
Array-wide time warping	**Yes (by K-mean clustering)**	moderate*	-	moderate*	By K-means clustering, efficiently improve the performance of conventional DTW
Time point filtering	**Yes**	-	-	-	Filter the likely spurious time point
Open source	**Yes**	Yes	Yes	-	Give free utilization of TWins' code

* Directly act DTW on whole genome date, which might lead to unreliable alignment

5.5.2 Web interface

The web interface (see figure 5.12) allows two types of operations: (1) compare two uploaded experiments, and (2) compare an uploaded experiment with database.



Compare two uploaded experiments:

Please input the genus+species name: Homo+sapiens

Please upload the first experiment (A): 瀏覽...

Please upload the second experiment (B): 瀏覽...

Click here for example files : 1 2

Grouping: ● All ○ k 5 means clustering

Time point filter on: ○ A ○ B ● none

Submit your request

Compare an uploaded experiment with database:

Please select one dataset: please choose...

Please choose a condition:

----------------Select dataset first...-------------

Please upload the experiment: 瀏覽...

Grouping: ● All ○ k 5 means clustering

Time point filter on: ○ database ○ uploaded data ● none

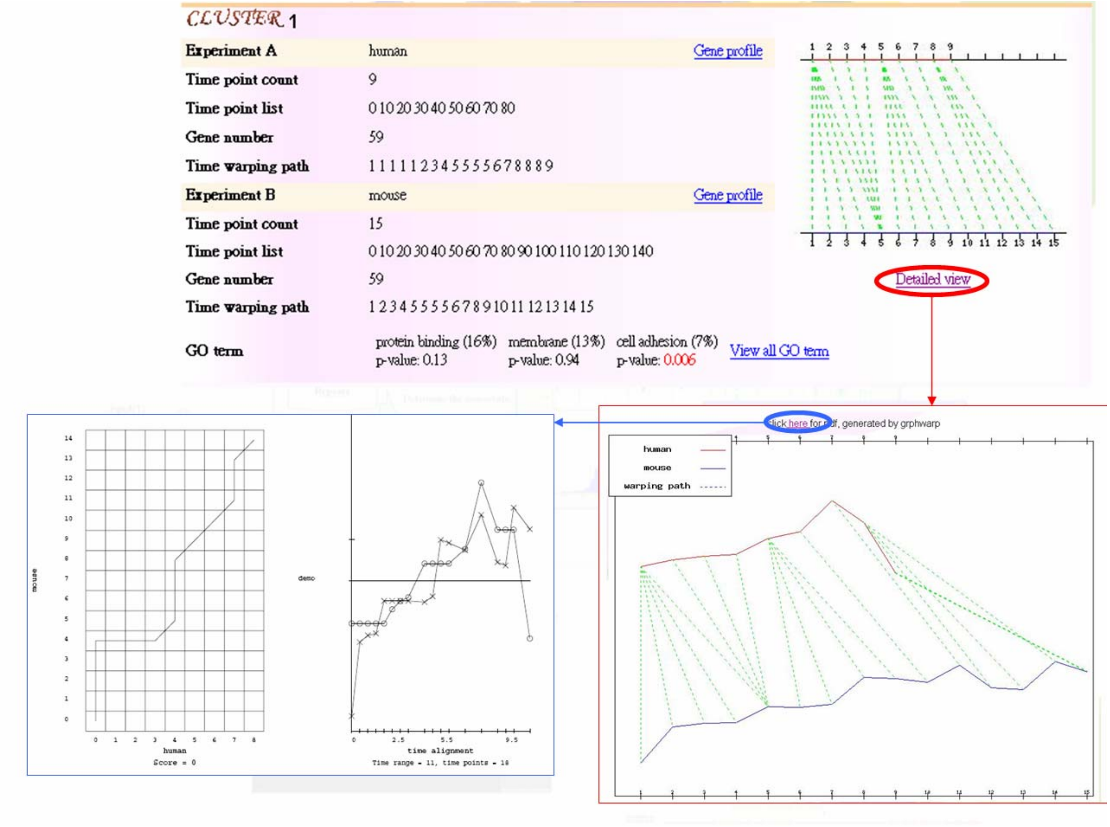
Submit your request

Figure 5.13 Two type of operations on web interface. (1) compare two uploaded experiments, and (2) compare an uploaded experiment with database

The first type of operations needs users upload two formatted files, and input the organism name. Organism name is necessary, since the system needs organism and gene symbol name to acquire the GO term and evaluates the p-value. The second type of operations need only one upload file. However, users must at first specified one dataset compiled from NCBI GEO. When the dataset are set, the webpage will present the particular experiment conditions of the dataset. For example, if dataset GDS1857 are selected, the browser will display five conditions and their descriptions, including cell-line treated by doxycycline, cell-line transfacted by HIV-1 Vpr protein and so on. Users should decide which experiment to compare with. System does not force users to input the organism name of their data, but they need to upload the profile of the

same species of the dataset they chosen, or the GO term and p-value will go wrong. Users then need to decide the grouping strategy: select “All” to directly perform conventional DTW on the datasets, or select “K means clustering” for grouping the data by this approach before acting DTW. Any big dataset is suggested to use K mean clustering in order to get better warping path. If users like the system to help screen out spurious time point, they can click on the radio button and submit their task.

After submission and waiting for system to complete users’ requests, the results page will be shown and provide detail information of your request. See figure 5.13. System will show you the warping path and p-value of GO term, every GO term having p-value lower than 0.05 will be written in red color. TWins also provides detailed warping path and output the pdf file generated by grphwarp[1] program to give more delicate graphics.



CLUSTER 1

Experiment A	human	Gene profile
Time point count	9	
Time point list	0 10 20 30 40 50 60 70 80	
Gene number	59	
Time warping path	1 1 1 1 1 2 3 4 5 5 5 5 6 7 8 8 8 9	
Experiment B	mouse	Gene profile
Time point count	15	
Time point list	0 10 20 30 40 50 60 70 80 90 100 110 120 130 140	
Gene number	59	
Time warping path	1 2 3 4 5 5 5 5 6 7 8 9 10 11 12 13 14 15	
GO term	protein binding (16%) p-value: 0.13	membrane (13%) p-value: 0.94
	cell adhesion (7%) p-value: 0.006	View all GO term

Click here for pdf, generated by grphwarp

Figure 5.14 The result of users’ request. The graph in blue square is drawn by grphwarp.

5.6 Software package

The package contains every executable routine and other utility tool. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Insitute of Bioifnormatics, NCTU. Further to the terms mentioned you should leave the copyright footers and copyright notice in the HTML headers intact, stating me as the original author.

Chapter 6 Discussion

6.1 The limitation of the distance function calculation by dynamic programming

In previous Chapter, we have mentioned that the distance function used in DTW is not exactly original Euclidean distance function but $\sum_i \sqrt{(b_i - a_i)^2}$. The group-distance version in DTW is $\sum_{i=1}^{T} \sqrt{\sum_{k=1}^{N} (b_{ki} - a_{ki})^2}$, comparing with original one $\sum_{k=1}^{N} \sqrt{\sum_{i=1}^{T} (b_{ki} - a_{ki})^2}$, where N is the number of genes in this group, and T is the number of time points. The problem of the distance function used in our algorithm is that they are not real Euclidean distance functions; they are more like functions calculating the degree of sparseness in each time points. That is, this function only works like original Euclidean distance function when the degree of sparseness in each time points is low, since $\sum_{i=1}^{T} \sqrt{\sum_{k=1}^{N} (b_{ki} - a_{ki})^2}$ will approximately equal to $N * \sqrt{\sum_{i=1}^{T} (b_{mi} - a_{mi})^2}$, where a_{mi} and b_{mi} represent the mean of a_{ki} and b_{ki}. This also supports our scheme which performs K-means clustering making the degree of sparseness in each time points lower before calculating the distance in groups, and therefore leading to result more close to real Euclidean distance.

6.2 The limitation of SVD

When we look at the algorithm of SVD closely, we will find that the rank of eigen-genes is limited to the number of time points. This is its mathematical limitation, which implies that our data in matrix A should not mix with too many kinds of patterns or the noise reduction will either not-working or filter out important information. That is why the approach proposed by O. Alter, GSVD, is said to only focus on cell-cycle regulated genes and limit genes numbers.

In our scheme, since we realized the limitation of SVD, we do SVD after the gene number is reduced by our single-gene distance ranking. In general case, we pick top hundreds of genes for SVD to process, which avoid lost of information.

6.3 Future work

We can extend our scheme to multi-species model by replace of pair-wise profile alignment by multiple-profile alignment algorithm. And therefore can analyze orthologous gene groups cross more tissue and more species. The FNC can be improved by filtering out data sets which is not well sampled or replaced with advanced technique with higher prediction accuracy. Since the verification of gene functions in always proceeding, we should update the database frequently in order to provide précised annotation and p-value analysis. The presentation support by Genesis is not very satisfying since it does not considering the warping path. This can be solved to generate this kind of clustering result by our program. We hope to keep reinforcing our package with time.

Chapter 7 Conclusions

This work proposed a scheme which at first ranks the single-gene warping distance between genes in human and genes in mouse linked by gene linkage based on their similarity on sequence, and reduced their noise by SVD, and clusters genes of a species by computing the distance between each according to their expression profiles (k-mean clustering). However, these linked genes in another species are needed to be divided into smaller groups sharing parallel expression pattern, so the clustering is acted again on each group. These small groups of genes linking with other small groups of genes are called pairs. Each pair is computed its warping distance by group-DTW algorithm. Pair with small warping distance is regarded as having similar expression pattern and sequence homology. In addition, genes are annotated with GO term and suggested with its p-value. Pairs illustrating both small warping distance and p-value are highly recommended to have similar and basic functions within multiple species.

We perform our scheme in artificial, homogeneous (yeast), and heterogeneous (human and mouse). All of them shows great outcome and demonstrate by accessible visualization program.

Although our experiment only examine two species, the scheme is reasonable to be feasible into multiple species. The genes found by this scheme are highly conserved in their sequence and expression, which suggests these genes play basic role in the functions and therefore are preserved in the evolutionary process.

This dissertation not only answer the questions listed in previous chapter, but also propose a novel scheme to solve these relevant tasks with integration and improvement, and contributes remarkable advancement in cross-species orthologous gene analysis.

Chapter 8 References

1. Aach, J. and G.M. Church, *Aligning gene expression time series with time warping algorithms.* Bioinformatics, 2001. **17**(6): p. 495-508.
2. Bar-Joseph, Z., et al., *Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes.* Proc Natl Acad Sci U S A, 2003. **100**(18): p. 10146-51.
3. Bar-Joseph, Z., *Analyzing time series gene expression data.* Bioinformatics, 2004. **20**(16): p. 2493-503.
4. Murali, T.M., C.J. Wu, and S. Kasif, *The art of gene function prediction.* Nat Biotechnol, 2006. **24**(12): p. 1474-5.
5. de Jong, H., *Modeling and simulation of genetic regulatory systems: a literature review.* J Comput Biol, 2002. **9**(1): p. 67-103.
6. Spellman, P.T., et al., *Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.* Mol Biol Cell, 1998. **9**(12): p. 3273-97.
7. Marinova-Boncheva, V., *USING THE AGGLOMERATIVE METHOD OF HIERARCHICAL CLUSTERING AS A
DATA MINING TOOL IN CAPITAL MARKET1.* International Conference «Information Research & Applications, 2007.
8. Sung Hee Park, C.Y.P., Dae Hee Kim, Seon Hee Park and Jeong Seop Sim, *Protein Structure Abstractionand Automatic Clustering Using Secondary Structure Element Sequences*. Lecture Notes in Computer Science. 1284-1292.
9. Allison, D.B., *DNA Microarrays and Related Genomic Techniques: Design, Analysis, and Interpretation of Experiments.* 2005.
10. Bergmann, S., J. Ihmels, and N. Barkai, *Similarities and differences in genome-wide expression data of six organisms.* PLoS Biol, 2004. **2**(1): p. E9.
11. Grigoryev, D.N., et al., *Orthologous gene-expression profiling in multi-species models: search for candidate genes.* Genome Biol, 2004. **5**(5): p. R34.
12. Tsai, J., et al., *RESOURCERER: a database for annotating and linking microarray resources within and across species.* Genome Biol, 2001. **2**(11): p. SOFTWARE0002.
13. Alter, O., P.O. Brown, and D. Botstein, *Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms.* Proc Natl Acad Sci U S A, 2003. **100**(6): p. 3351-6.
14. Barrett, T., et al., *NCBI GEO: mining tens of millions of expression profiles--database and tools update.* Nucleic Acids Res, 2006.
15. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.* Nat Genet, 2000. **25**(1): p. 25-9.
16. Li, X.L., Y.C. Tan, and S.K. Ng, *Systematic gene function prediction from gene expression data by using a fuzzy nearest-cluster method.* BMC Bioinformatics, 2006. **7 Suppl 4**: p. S23.
17. kruskal, D.s.a.J., *Time warps, string edits, and macromolecles*.
18. Michael E. Wall1, Andreas Rechtsteiner1,3, Luis M. Rocha1, *Singular value decomposition and principal component analysis*. A Practical Approach to Microarray Data Analysis. 2003. 91-109.
19. Alter, O., P.O. Brown, and D. Botstein, *Singular value decomposition for genome-wide expression data processing and modeling.* Proc Natl Acad Sci U

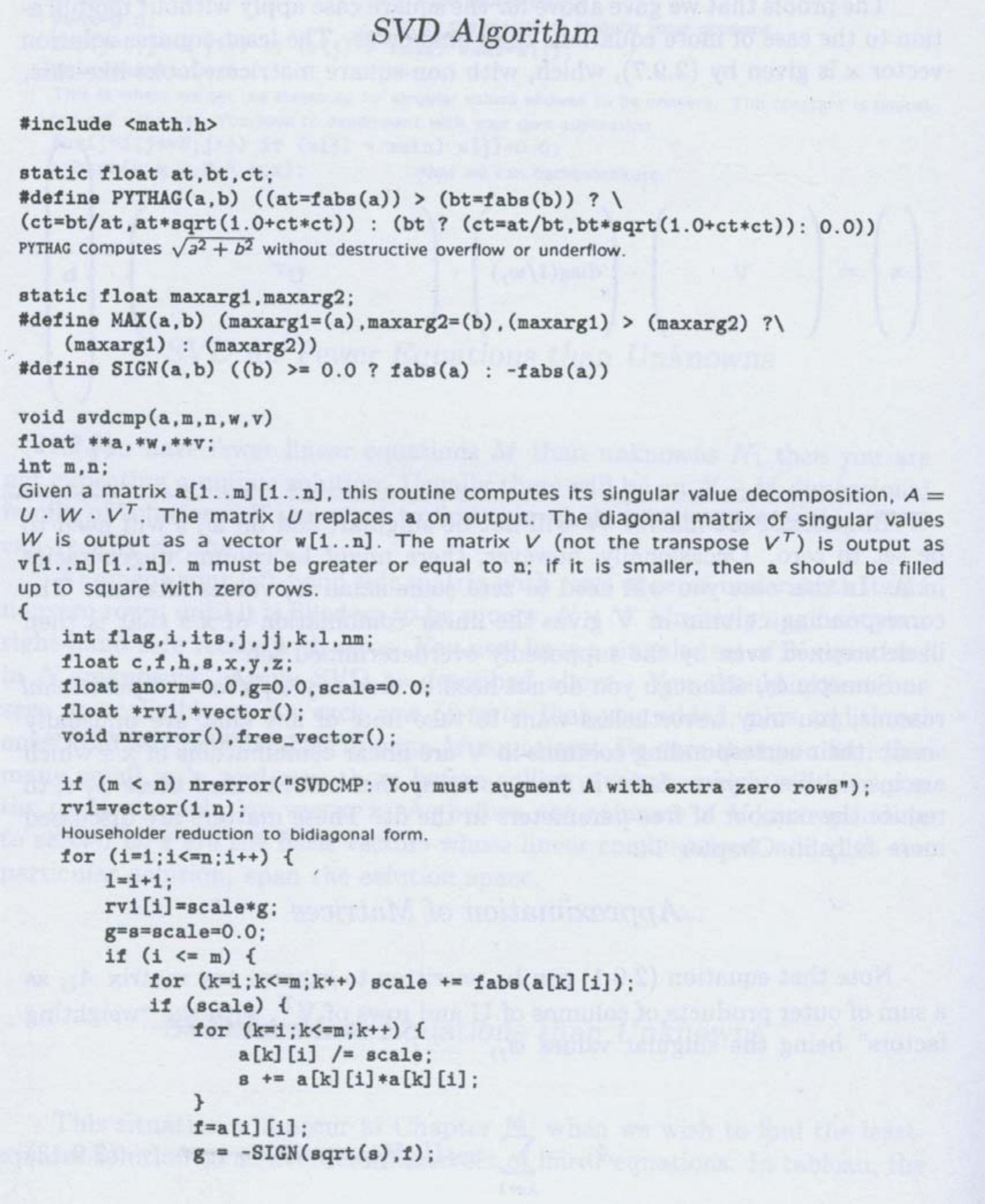
S A, 2000. **97**(18): p. 10101-6.

20. Everitt B.S., G.G., *applied multivariate data analysis*. 2001.
21. Li, X.L., et al., *Systematic gene function prediction from gene expression data by using a fuzzy nearest-cluster method.* BMC Bioinformatics, 2006. **7**(Suppl 4): p. S4-23.
22. Quackenbush, J., et al., *The TIGR gene indices: reconstruction and representation of expressed gene sequences.* Nucleic Acids Res, 2000. **28**(1): p. 141-5.
23. Yi, G., S.H. Sze, and M.R. Thon, *Identifying clusters of functionally related genes in genomes.* Bioinformatics, 2007.
24. Mewes, H.W., et al., *MIPS: analysis and annotation of proteins from whole genomes in 2005.* Nucleic Acids Res, 2006. **34**(Database issue): p. D169-72.
25. Ruepp, A., et al., *The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes.* Nucleic Acids Res, 2004. **32**(18): p. 5539-45.
26. golub, G.H., and Van Loan, Charles F., *Matrix Computations*. 1983.
27. http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/.
28. http://www.boutell.com/gd/.
29. Gentile, M., L. Latonen, and M. Laiho, *Cell cycle arrest and apoptosis provoked by UV radiation-induced DNA damage are transcriptionally highly divergent responses.* Nucleic Acids Res, 2003. **31**(16): p. 4779-90.
30. Kashanchi, F., et al., *Cell cycle-regulated transcription by the human immunodeficiency virus type 1 Tat transactivator.* J Virol, 2000. **74**(2): p. 652-60.
31. Tomczak, K.K., et al., *Expression profiling and identification of novel genes involved in myogenic differentiation.* Faseb J, 2004. **18**(2): p. 403-5.
32. Troester, M.A., et al., *Cell-type-specific responses to chemotherapeutics in breast cancer.* Cancer Res, 2004. **64**(12): p. 4218-26.
33. Martin, D.E., et al., *Rank Difference Analysis of Microarrays (RDAM), a novel approach to statistical analysis of microarray expression profiling data.* BMC Bioinformatics, 2004. **5**: p. 148.
34. Zambon, A.C., et al., *Gene expression patterns define key transcriptional events in cell-cycle regulation by cAMP and protein kinase A.* Proc Natl Acad Sci U S A, 2005. **102**(24): p. 8561-6.
35. Vanneste S, B.D.R., G. T. S. beemster, K. Ljung, I. D. Smet, G. V. Isterdael, M. Naudts, R. Iida, W. Gruissem, M. Tasaka, D. Inze, H. Gukaki, and T. Beeckman, *Cell Cycle Progression in the Pericycle Is Not Sufficient for SOLITARY ROOT/IAA14-Mediated Lateral Root Initiation in Arabidopsis thaliana.* Plant Cell, 2005. **17(11)**: p. 3035-50.
36. Bean, C., et al., *The Ankrd2, Cdkn1c and calcyclin genes are under the control of MyoD during myogenic differentiation.* J Mol Biol, 2005. **349**(2): p. 349-66.
37. Yoshizuka, N., et al., *Human immunodeficiency virus type 1 Vpr-dependent cell cycle arrest through a mitogen-activated protein kinase signal transduction pathway.* J Virol, 2005. **79**(17): p. 11366-81.
38. Miller, R.M., et al., *Dysregulation of gene expression in the 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine-lesioned mouse substantia nigra.* J Neurosci, 2004. **24**(34): p. 7445-54.
39. Aach, J., W. Rindone, and G.M. Church, *Systematic management and analysis of yeast gene expression data.* Genome Res, 2000. **10**(4): p. 431-45.
40. Cho, R.J., et al., *A genome-wide transcriptional analysis of the mitotic cell*

cycle. Mol Cell, 1998. **2**(1): p. 65-73.

41. Cho, R.J., et al., *Transcriptional regulation and function during the human cell cycle.* Nat Genet, 2001. **27**(1): p. 48-54.

Appendix 1   Pseudo code of SVD algorithm



SVD Algorithm

```
#include <math.h>

static float at,bt,ct;
#define PYTHAG(a,b) ((at=fabs(a)) > (bt=fabs(b)) ? \
(ct=bt/at,at*sqrt(1.0+ct*ct)) : (bt ? (ct=at/bt,bt*sqrt(1.0+ct*ct)): 0.0))
```

PYTHAG computes $\sqrt{a^2+b^2}$ without destructive overflow or underflow.

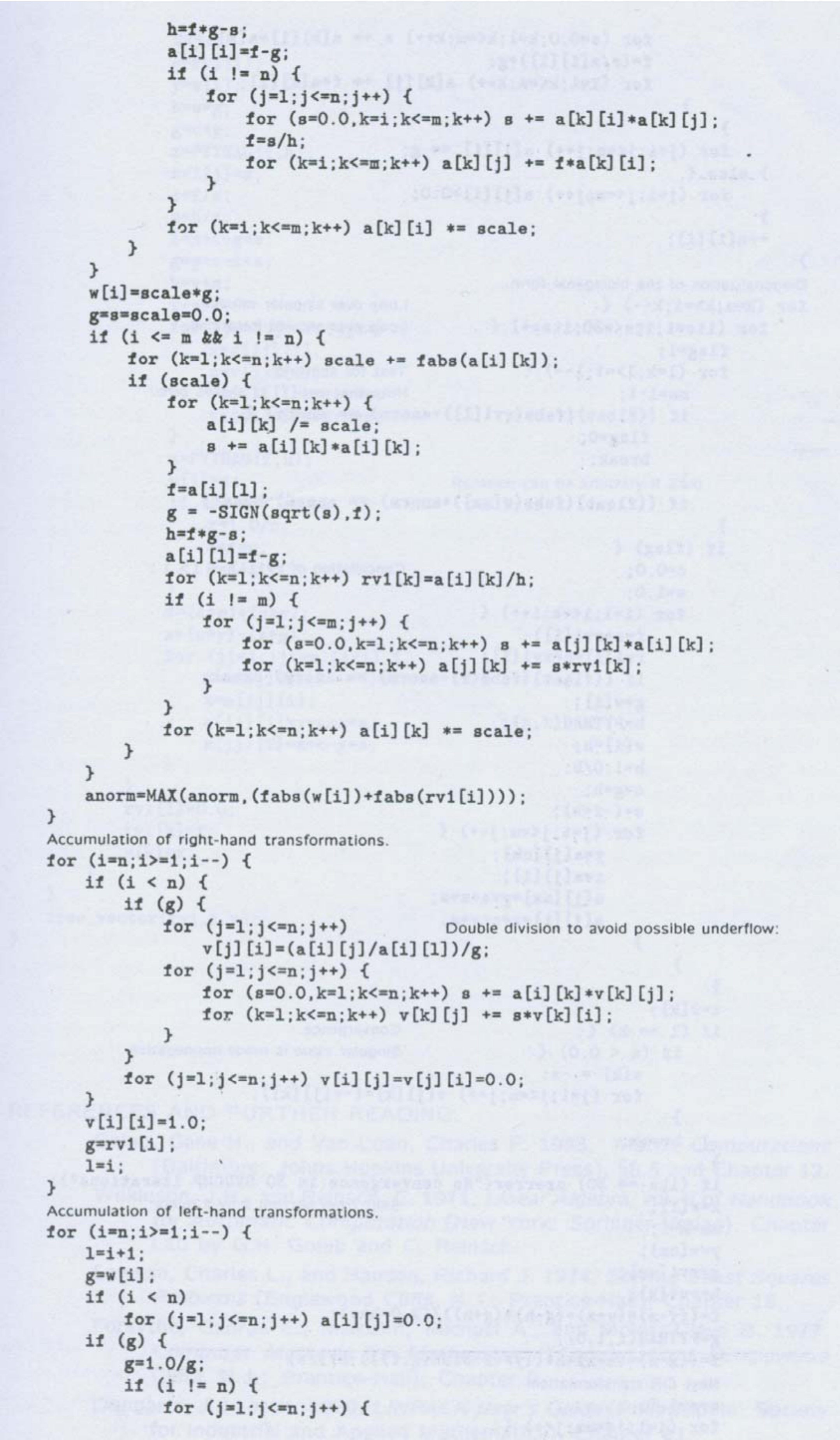
```
static float maxarg1,maxarg2;
#define MAX(a,b) (maxarg1=(a),maxarg2=(b),(maxarg1) > (maxarg2) ?\
   (maxarg1) : (maxarg2))
#define SIGN(a,b) ((b) >= 0.0 ? fabs(a) : -fabs(a))

void svdcmp(a,m,n,w,v)
float **a,*w,**v;
int m,n;
```

Given a matrix a[1..m][1..n], this routine computes its singular value decomposition, $A = U \cdot W \cdot V^T$. The matrix U replaces a on output. The diagonal matrix of singular values W is output as a vector w[1..n]. The matrix V (not the transpose V^T) is output as v[1..n][1..n]. m must be greater or equal to n; if it is smaller, then a should be filled up to square with zero rows.

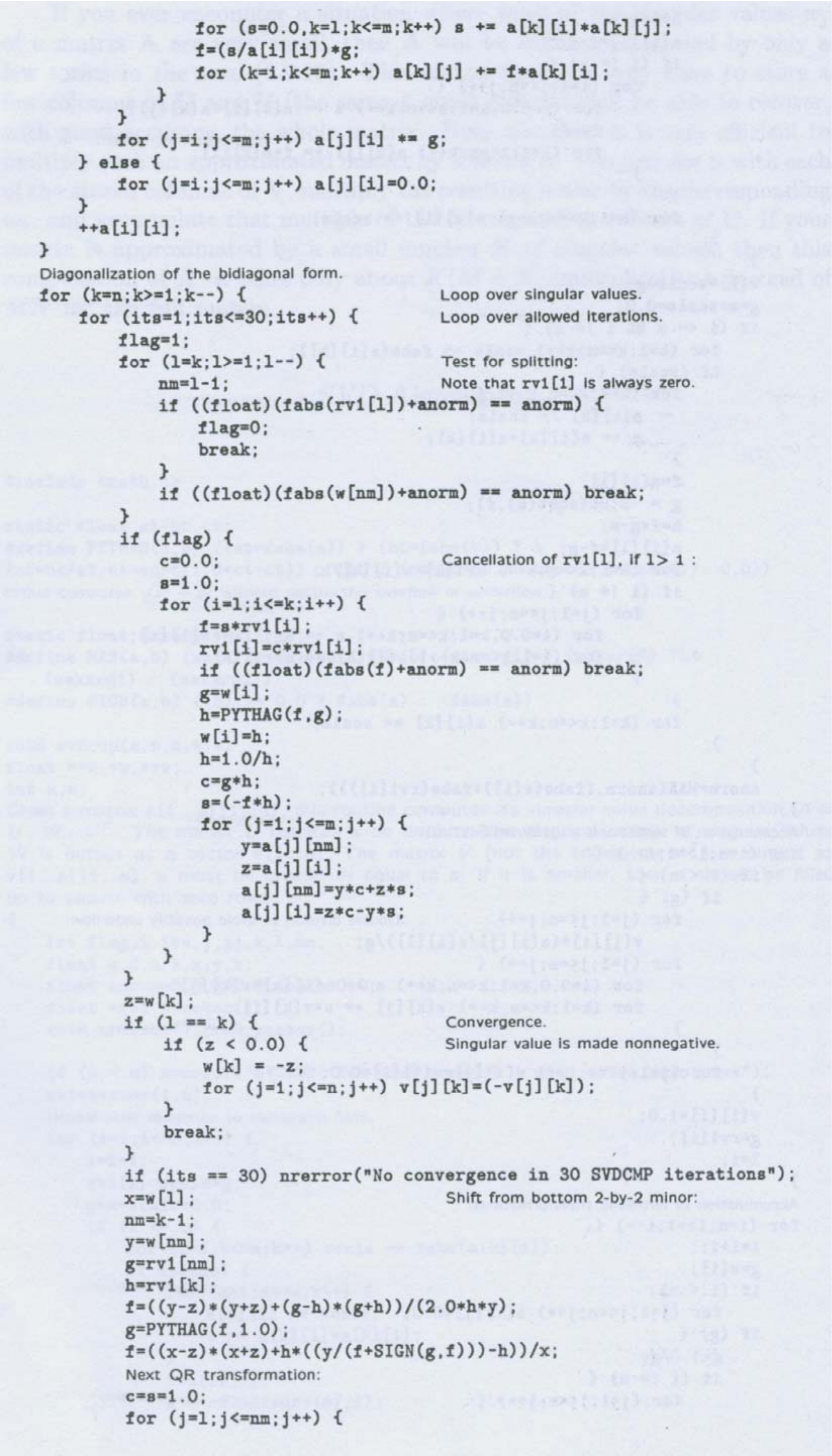
```
{
   int flag,i,its,j,jj,k,l,nm;
   float c,f,h,s,x,y,z;
   float anorm=0.0,g=0.0,scale=0.0;
   float *rv1,*vector();
   void nrerror(),free_vector();

   if (m < n) nrerror("SVDCMP: You must augment A with extra zero rows");
   rv1=vector(1,n);
```

Householder reduction to bidiagonal form.

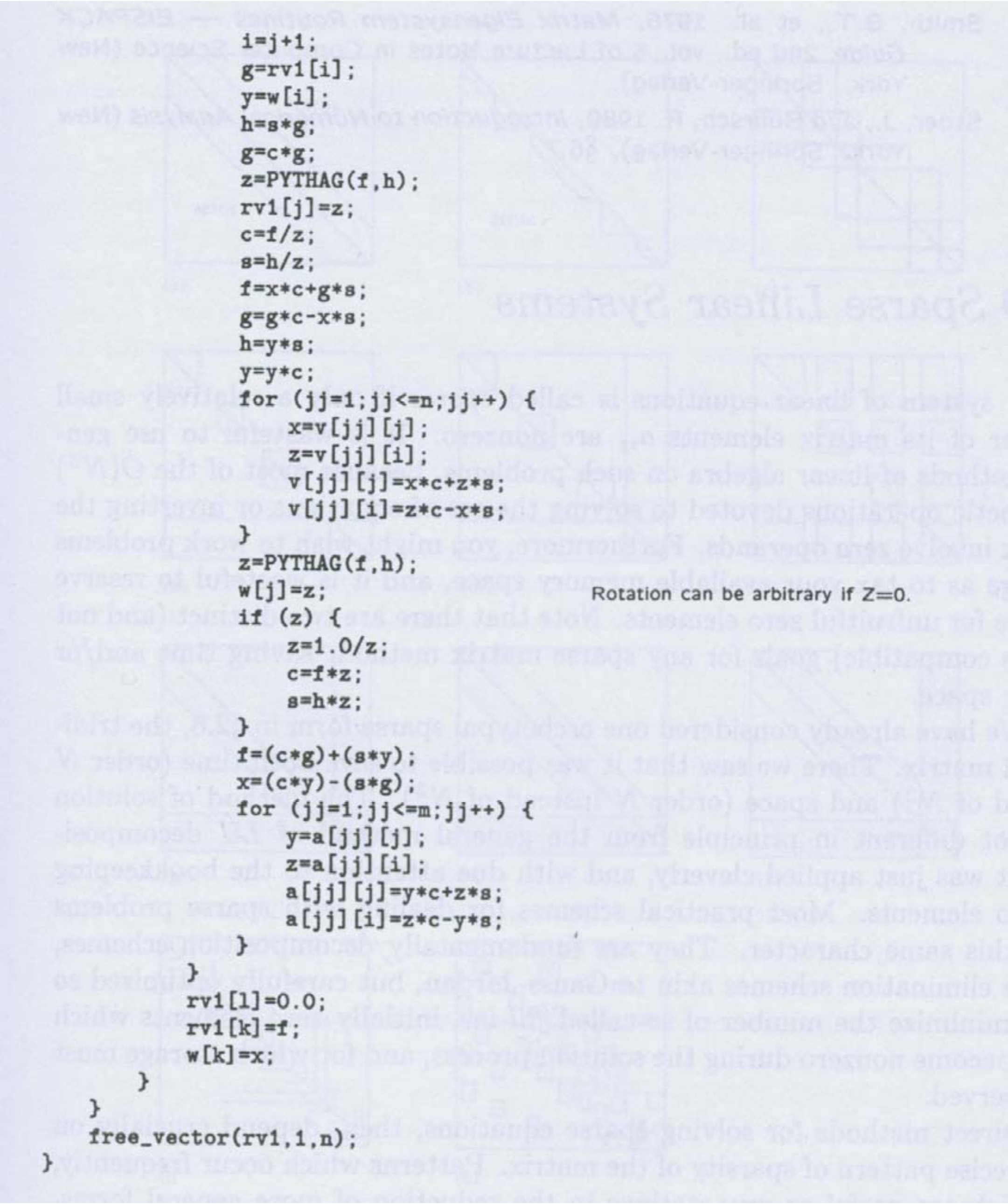
```
   for (i=1;i<=n;i++) {
      l=i+1;
      rv1[i]=scale*g;
      g=s=scale=0.0;
      if (i <= m) {
         for (k=i;k<=m;k++) scale += fabs(a[k][i]);
         if (scale) {
            for (k=i;k<=m;k++) {
               a[k][i] /= scale;
               s += a[k][i]*a[k][i];
            }
            f=a[i][i];
            g = -SIGN(sqrt(s),f);
```



```
            h=f*g-s;
            a[i][i]=f-g;
            if (i != n) {
                for (j=l;j<=n;j++) {
                    for (s=0.0,k=i;k<=m;k++) s += a[k][i]*a[k][j];
                    f=s/h;
                    for (k=i;k<=m;k++) a[k][j] += f*a[k][i];
                }
            }
            for (k=i;k<=m;k++) a[k][i] *= scale;
        }
    }
    w[i]=scale*g;
    g=s=scale=0.0;
    if (i <= m && i != n) {
        for (k=l;k<=n;k++) scale += fabs(a[i][k]);
        if (scale) {
            for (k=l;k<=n;k++) {
                a[i][k] /= scale;
                s += a[i][k]*a[i][k];
            }
            f=a[i][l];
            g = -SIGN(sqrt(s),f);
            h=f*g-s;
            a[i][l]=f-g;
            for (k=l;k<=n;k++) rv1[k]=a[i][k]/h;
            if (i != m) {
                for (j=l;j<=m;j++) {
                    for (s=0.0,k=l;k<=n;k++) s += a[j][k]*a[i][k];
                    for (k=l;k<=n;k++) a[j][k] += s*rv1[k];
                }
            }
            for (k=l;k<=n;k++) a[i][k] *= scale;
        }
    }
    anorm=MAX(anorm,(fabs(w[i])+fabs(rv1[i])));
}
Accumulation of right-hand transformations.
for (i=n;i>=1;i--) {
    if (i < n) {
        if (g) {
            for (j=l;j<=n;j++)          Double division to avoid possible underflow:
                v[j][i]=(a[i][j]/a[i][l])/g;
            for (j=l;j<=n;j++) {
                for (s=0.0,k=l;k<=n;k++) s += a[i][k]*v[k][j];
                for (k=l;k<=n;k++) v[k][j] += s*v[k][i];
            }
        }
        for (j=l;j<=n;j++) v[i][j]=v[j][i]=0.0;
    }
    v[i][i]=1.0;
    g=rv1[i];
    l=i;
}
Accumulation of left-hand transformations.
for (i=n;i>=1;i--) {
    l=i+1;
    g=w[i];
    if (i < n)
        for (j=l;j<=n;j++) a[i][j]=0.0;
    if (g) {
        g=1.0/g;
        if (i != n) {
            for (j=l;j<=n;j++) {
```



```
                for (s=0.0,k=l;k<=m;k++) s += a[k][i]*a[k][j];
                f=(s/a[i][i])*g;
                for (k=i;k<=m;k++) a[k][j] += f*a[k][i];
            }
        }
        for (j=i;j<=m;j++) a[j][i] *= g;
    } else {
        for (j=i;j<=m;j++) a[j][i]=0.0;
    }
    ++a[i][i];
}
Diagonalization of the bidiagonal form.
for (k=n;k>=1;k--) {                    Loop over singular values.
    for (its=1;its<=30;its++) {         Loop over allowed iterations.
        flag=1;
        for (l=k;l>=1;l--) {            Test for splitting:
            nm=l-1;                     Note that rv1[1] is always zero.
            if ((float)(fabs(rv1[l])+anorm) == anorm) {
                flag=0;
                break;
            }
            if ((float)(fabs(w[nm])+anorm) == anorm) break;
        }
        if (flag) {
            c=0.0;                      Cancellation of rv1[l], if l> 1 :
            s=1.0;
            for (i=l;i<=k;i++) {
                f=s*rv1[i];
                rv1[i]=c*rv1[i];
                if ((float)(fabs(f)+anorm) == anorm) break;
                g=w[i];
                h=PYTHAG(f,g);
                w[i]=h;
                h=1.0/h;
                c=g*h;
                s=(-f*h);
                for (j=1;j<=m;j++) {
                    y=a[j][nm];
                    z=a[j][i];
                    a[j][nm]=y*c+z*s;
                    a[j][i]=z*c-y*s;
                }
            }
        }
        z=w[k];
        if (l == k) {                   Convergence.
            if (z < 0.0) {              Singular value is made nonnegative.
                w[k] = -z;
                for (j=1;j<=n;j++) v[j][k]=(-v[j][k]);
            }
            break;
        }
        if (its == 30) nrerror("No convergence in 30 SVDCMP iterations");
        x=w[l];                         Shift from bottom 2-by-2 minor:
        nm=k-1;
        y=w[nm];
        g=rv1[nm];
        h=rv1[k];
        f=((y-z)*(y+z)+(g-h)*(g+h))/(2.0*h*y);
        g=PYTHAG(f,1.0);
        f=((x-z)*(x+z)+h*((y/(f+SIGN(g,f)))-h))/x;
        Next QR transformation:
        c=s=1.0;
        for (j=l;j<=nm;j++) {
```



```
                i=j+1;
                g=rv1[i];
                y=w[i];
                h=s*g;
                g=c*g;
                z=PYTHAG(f,h);
                rv1[j]=z;
                c=f/z;
                s=h/z;
                f=x*c+g*s;
                g=g*c-x*s;
                h=y*s;
                y=y*c;
                for (jj=1;jj<=n;jj++) {
                    x=v[jj][j];
                    z=v[jj][i];
                    v[jj][j]=x*c+z*s;
                    v[jj][i]=z*c-x*s;
                }
                z=PYTHAG(f,h);
                w[j]=z;                         Rotation can be arbitrary if Z=0.
                if (z) {
                    z=1.0/z;
                    c=f*z;
                    s=h*z;
                }
                f=(c*g)+(s*y);
                x=(c*y)-(s*g);
                for (jj=1;jj<=m;jj++) {
                    y=a[jj][j];
                    z=a[jj][i];
                    a[jj][j]=y*c+z*s;
                    a[jj][i]=z*c-y*s;
                }
            }
            rv1[l]=0.0;
            rv1[k]=f;
            w[k]=x;
        }
    }
    free_vector(rv1,1,n);
}
```