DTW-GO Based Microarray Time Series Data Analysis for Gene-Gene Regulation Prediction

Andy C. Yang and Hui-Huang Hsu

Abstract. Microarray technology provides an opportunity for scientists to analyze thousands of gene expression profiles simultaneously. Due to the widely use of microarray technology, several research issues are discussed and analyzed such as missing value imputation or gene-gene regulation prediction. Microarray gene expression data often contain multiple missing expression values due to many reasons. Effective methods for missing value imputation in gene expression data are needed since many algorithms for gene analysis require a complete matrix of gene array values. In addition, selecting informative genes from microarray gene expression data is essential while performing data analysis on the large amount of data. To fit this need, a number of methods were proposed from various points of view. However, most existing methods have their limitations and disadvantages.

To estimate similarity between gene pairs effectively, we propose a novel distance measurement based on the well-defined ontology structure for genes or proteins: the gene ontology (GO). GO is a definition and annotation for genes that describe the biological meanings of them. The structure of GO can be described as a directed acyclic graph (DAG), where each GO term is a node, and the relationships between each term pair are arcs. With GO annotations, we can hence acquire the relations for the genes involved in the experiment. The semantic similarity of two genes within biological aspect can be identified if we perform some quantitative assessments on the gene pairs with their GO annotations.

In this chapter, we first provide the reader with fundamental knowledge about microarray technology in Section 1. A brief introduction for microarray experiments will be given. We then discuss and analyze essential research issues about microarray in Section 2. We also present a novel method based on k-nearest neighbor (KNN), dynamic time warping (DTW) and gene ontology (GO) for the analysis of microarray time series data in Section 3. With our approach, missing value imputation and gene regulation prediction can be achieved efficiently. Section 4 introduces a real microarray time-series dataset. Effectiveness of our method is shown with various experimental results in Section 5. A brief conclusion is made in Section 6.

Andy C. Yang · Hui-Huang Hsu
Department of Computer Science & Information Engineering,
Tamkang University, Taipei, 25137, Taiwan R.O.C.

M. Biba and F. Xhafa (Eds.): Learning Structure and Schemas from Documents, SCI 375, pp. 247–274.
springerlink.com

1 Introduction

Content of this section tends to bring essential knowledge for the reader to understand the process of microarray technology. The importance of this technology is also mentioned. This section ends with the description of microarray data processing and its relation to the ontology structure: the gene ontology (GO).

1.1 What Is Microarray?

Microarray is a widely-used biological experimental approach in this decade. It makes it possible to perform large amounts of gene or protein data experimental operations at the same time. The concept of microarray is based on the differential reactions acted by each sample on the microarray gene chip relative to the experimental conditions. Generally, microarray technology is divided into two aspects: cDNA microarray and Affymetrix microarray. In cDNA microarray, controlled and experimental samples are dyed with two different colors and then hybridized to generate various experimental results. Affymetrix microarray is chosen while biologists or associations need to perform tests on huge amounts of data that are previously cloned and manufactured by Affymetrix microarray producers. Applications of cDNA microarray are more common in several biological research laboratories because one can produce cDNA microarray chips with data of interest more easily. On the other hand, costs of Affymetrix microarray are much more expensive than cDNA microarray. Therefore, we focus on cDNA microarray in this chapter.

1.2 Importance of Microarray Technology

Traditionally, biologists need to perform the same operation for biological experiments due to the limitation of instruments. For example, if we want to experiment on one gene sample and observe its reactions, we have to prepare the sample for several copies. This pre-processing task is critically time-consuming, not to mention much more time needed during the process of molecular or biological experiments. With the development of microarray technology, performing experiments on genes of interest becomes much easier because biologists can now retrieve enough amounts of data they need with little time required. Due to this high throughput biological technology, numerous gene expression data are generated simultaneously. In the meanwhile, the large amounts of data provide us great challenges of analysis. Retrieving meaningful information hidden in these data is essential to facilitate the development of drugs, or the discovery of diseases.

1.3 Microarray Data Processing

Procedures of microarray experiments can be described as following steps:

- Prepare cDNA data for a certain gene which is going to be experimented.
- Print the cDNA data of interest onto microarray chips.

- Design the suitable probes consisting of two cDNA or mRNA samples: One controlled sample and one experimented sample.
- Label the two different probe samples with red (experimented sample) and green (controlled sample) fluorescent dyes.
- Hybridize probes to the microarray chip, and clean up the chip.
- Scan the hybridized microarray chip with computer instruments and save the quantified data for subsequent analyses.

As listed above, quantified data are generated after scanning the hybridized microarray chip. These data represent different degrees of reactions for each gene sample. In other words, we can identify whether a gene tends to act as controlled or treated samples by calculating the ratio of red and green colors in the quantified data for this gene. For example, if the quantified data of two genes are with four and two for their red-green ratio respectively, it means the gene with larger red-green ratio acts like the treated sample than the other gene. For observation convenience, these data are usually transformed into the logarithm format with base two. These logarithmic data for genes on a gene chip are called microarray gene expression data.

Microarray time series data are matrix-like collections of gene expression values that represent reactions for each gene at different time slots as shown in Table 1. Each row in the microarray time-series data stands for a gene ORF profile, while each column in the matrix represents the specific time point. Different kinds of microarray time series data are with different time slots due to distinct gene sampling time and frequency. Gene expression values in the microarray time-series data may be positive or negative numbers. Positive gene expression values of some gene samples on the chip show that these genes are induced with treated sample, and negative values mean repressed reactions. The task is to analyze these gene expression values in different time slots and find the correlations between genes for the inferring of gene-gene interactions.

Table 1 Microarray time series data

Gene	**Time Slot 1**	**Time Slot 2**	**…**	**Time Slot n**
Gene #1	0.56	0.80		0.90
Gene #2	-0.24	-0.1		0.60
Gene #3	0.12	0.24		0.50
…	…	…		…
Gene #n	0.78	-0.14		-0.56

1.4 Microarray and Gene Ontology

Gene ontology (GO) is a biological definition and annotation for genes that describes the biological meanings of each gene. Generally, most known genes have

specific annotations (terms) in GO structure within three independent domains: molecular function (MF), biological process (BP), and cellular component (CC). Terms within three above domains record and represent various molecular or biological meanings for each annotated gene from different aspects respectively. Molecular function considers the biological or biochemical activity at the molecular level. Biological process consists of many molecular functions that are involved in a related biological activity or reaction. It denotes a biological objective which genes contribute to. Cellular component records the place in cells where a gene product is active. One gene may have more than one annotation in each domain. These annotations provide hidden information for corresponding genes from the biological aspect.

The main task in microarray gene expression data analysis is to identify the gene pairs or groups that are highly co-expressed under individual experimental conditions. Usually, various distance measurements or classification / clustering operations are performed on gene expression values in microarray data. However, these kinds of procedures only take gene expression values into consideration so that they lack biological explanations and are not effective, either. With proper usage of gene ontology, this task can be done more efficiently and accurately. Detail descriptions about gene ontology and its importance in microarray data analysis are shown in Section 3.2.

2 Research Issues in Microarray Time-Series Data

Microarray technology is getting more and more popular due to its high throughput for biological data. Research on microarray technology or relative data analysis can be categorized into various aspects. In this section, we discuss three major research issues about microarray including missing value imputation, gene regulation prediction, and gene clustering or statistical operations. Brief descriptions and literature review are presented for these three issues.

2.1 Missing Value Imputation

Before further analysis of microarray data, one critical issue must be addressed: missing value imputation. Microarray time series data usually consist of multiple missing values. Certain portions of gene expression values that do not exist in microarray gene expression raw data are called missing values. It is necessary to effectively estimate and impute these missing values for subsequent analysis of microarray gene expression data. Acuna and Rodriguez discuss the reason why missing values occur in [1]. These values possibly resulted from inaccuracy of experimental operations, or unobvious reaction at several time slot points of certain genes. Fig. 1 illustrates the missing value problem in microarray time series data. If there is a particular gene I with one missing value at time slot J, then YIJ is used to represent the target missing value. For example, $G_{3,3}$ in Fig. 1 stands for a missing value of gene 3 at the third time point of that gene.

Ouyang et al. find that there are about 5% to 90% missing values existing in various available microarray gene expression time series datasets respectively [2]. Studies also argue that simply ignoring or removing missing values from the raw data could lose meaningful information of these genes [3][4]. For these large amounts of data, it is required to first impute these missing values with effective methods. Without imputation for missing values, further analysis cannot be performed. To date, many imputation methods for handling missing values in microarray time series data have been developed. Troyanskaya et al. summarize and implement three methods: singular value decomposition based method (SVD-impute), weighted k-nearest neighbor (KNN-impute), and row average imputation [5]. The results in the paper show that the KNN imputation outperforms SVD-impute and naive methods such as zero or row average imputation. The most suitable number of parameter k in KNN method is also proved to be set between 10 and 20 in the paper. Afterward, several imputation methods are proposed based on KNN. For example, Kim et al. develop a new cluster-based imputation method called sequential k-nearest neighbor (SKNN) method [6]. The method imputes the missing values sequentially from the gene having least missing values, and uses the imputed values for the later imputation. The study is typically an example showing the effectiveness of KNN with some improvements on it.

	E1	E2	E3	…	…	E5
Gene1	-0.3	0.5	0.1	0.4	-0.6	0.1
Gene2	0.4	$G_{2,2}$	-0.4	$G_{2,4}$	-1.1	0.9
Gene3	-0.2	0.3	$G_{3,3}$	0.5	-0.7	0.2
…	0.6	0.5	0.1	$G_{4,4}$	$G_{4,N-1}$	$G_{4,N}$
…	-0.5	$G_{N-1,2}$	0.3	0.4	-0.6	0.1
GeneN	0.7	0.1	$G_{N,3}$	-0.3	0.2	0.5

Fig. 1 Missing values in microarray time series data

In addition to KNN or KNN-based imputation methods, there are still other imputation methods proposed from different standpoints. Oba et al. propose an estimation method for missing values based on Bayesian principal component analysis (BPCA) [7]. The method combines mathematical theorems with parameters that need not to be complicated. The results of BPCA outperform the KNN and SVD imputations according to the authors' evaluations. Moreover, an imputation method based on the local least squares (LLS-impute) formulation is proposed to estimate missing values in the gene expression data [8]. Both KNN and LLS imputations need to find similar genes for a target gene while imputing gene missing

values. Other proposed methods take various points of view into consideration. Regression modeling approaches are also used to solve the missing value imputation problem despite it is difficult to determine the parameters used for regression models [9-11].

For existing imputation methods, BPCA is shown to outperform others. But it is not easy to determine the number of principal axes, either [12][13]. Among all related works and published research, existing methods for microarray missing value imputation mainly utilize k-nearest neighbor (KNN) or KNN-like approaches to estimate the missing values. When applying KNN to impute missing values, we have to choose k similar genes without missing entries at the time slot point as the target missing value. This issue is discussed in the following subsection.

2.2 Gene Regulation Prediction

In the gene cell cycle or in a biological process, the expression level of one gene is usually regulated by other genes. There might be one-to-one or many-to-one regulatory relations. If one gene regulates other genes, it is called an input gene. On the contrary, if one gene is a regulated target, it is called an output gene [14]. For transcriptional regulations among all genes, there are two sorts of situations, activation and inhibition. In activation regulations, the expression of the output gene is increased with the presence of the input gene, and vice versa. In other words, an activator gene regulates the activatee gene in the biological process so that the gene expression level of the two genes forms the trend of positive correlations. On the contrary, a trend of negative correlations results from the inhibition regulations.

Typically, microarray time series data analysis aims to observe and find out pairs of genes with highly-correlated relations as above-mentioned. This kind of issue is called gene regulation prediction. Research on this issue has been performed for these years, and a variety of approaches have been proposed. The most commonly-used distance measurement is Euclidean distance or statistical calculations such as Pearson correlation coefficient (PCC). However, these kinds of distance measurements have many disadvantages. For example, Euclidean distance of two sequences is very sensitive to the points on the sequences that are far away from other points or the mean. These points are so-called outliers and they often occur in many domains. The existence of outliers influences a lot while measuring the similarity of genes [15]. PCC is a statistical measurement to identify whether two sequences are relative to each other or not. But PCC is not suitable here because for microarray time series data we have to focus on the local similarity but not the global correlation of two genes. The reason is that even genes with known regulations may have reaction time delay and offsets on the time axis [16]. As a result, comparing local similarity is more important than comparing the distance of whole time slot points while identifying similarity of two genes. Moreover, gene pairs in microarray data are often of different length. This reduces the practicability of gene regulation prediction methods requiring time sequences of the same length in real microarray datasets.

Other commonly proposed solutions include similarity analysis [17][18] or Bayesian networks [6][19]. Yeung et al. aim to find potential regulatory gene pairs by finding dominant spectral component of gene pairs [20]. Results of our approach for regulatory gene prediction are compared with Yeung's work because datasets and effectiveness assessment we use are the same. Among above regulatory genes prediction methods, some of them may have success for the analysis of microarray time series data, but their effectiveness is very limited. The most important reason is that these methods take only gene expression values into consideration and they lack external or biological information of genes. External information such as gene ontology for genes themselves is regarded as a hint to increase the accuracy of distance measurements between gene pairs [21][22]. This kind of external information for genes is proved to be helpful. As a result, it is necessary to apply a distance measurement that not only has the capability of pointing out local similarity but is also effective even with certain existing outliers in microarray time-series data. Furthermore, gene ontology information for genes should also be taken into consideration.

2.3 Gene Clustering and Statistical Operations

Clustering analysis and statistical operations are also used while dealing with microarray time series data [23][24]. Clustering is grouping similar genes into a finite set of separate clusters. This concept aims to group genes into several sets so that genes falling in the same group tend to have similar reactions to experimental conditions or genes themselves. Hierarchical clustering is the most commonly used clustering approach for microarray time series data. Genes with similar biological functions or reactions are found and collected step by step. Eventually, gene groups are constructed that provide some information for biologist to perform further analysis. However, clustering analysis can be taken as the extension of gene-gene regulation prediction. This is because we still need to identify the distance of gene pairs when we are going to build clusters for genes.

Statistical analysis for microarray time series data is performed from different standpoints. Several techniques such as t-test, p-test, or some hypotheses are used to predict whether genes have similar reactions or not. Nevertheless, statistical analysis usually requires large amounts of data sets and time-consuming calculations. Accordingly, we do not leave space for this issue. We mainly focus on microarray data analysis for gene-gene regulation prediction in this chapter.

3 DTW-GO Based Microarray Data Analysis

In order to find the distance between gene pairs for missing value imputation and further gene regulation prediction, we propose a novel and effective approach. Our approach takes both gene expression values and external biological information for genes into account. Dynamic time warping (DTW) algorithm is used in our approach as the substitution for commonly-used Euclidean distance while estimating

distance between gene pairs. This is because the importance of finding whether there exist subsequences with highly similar relations is emphasized while analyzing whole microarray time series data [8][16]. We then try our method with several variants of DTW to further improve its efficiency and accuracy.

Moreover, we also add gene ontology (GO) information for genes themselves into our approach to make the distance measurement more accurate. GO is a definition and annotation for genes that describe the biological meanings of them. Each known gene has a specific annotation (term) in GO structure within three independent domains: molecular function (MF), biological process (BP), and cellular component (CC). Terms within three above domains consider different aspects respectively. One gene may have more than one annotation in each domain. These GO terms are quite informative because they provide biological meanings for genes. In our approach, GO terms are taken as the external information for genes while estimating the distance between gene pairs.

Finally, we combine our approach with the k-nearest neighbor (KNN) method to first impute missing values. After missing values are estimated, the prediction of regulatory gene pairs can be done with our approach as the distance measurement. This section briefly describes the DTW algorithm and the GO structure, followed by defining our approach that combines the above algorithm and information for genes with the KNN method for missing value imputation and the eventual prediction of regulatory gene pairs.

3.1 Dynamic Time Warping

It has been shown in many domains that dynamic time warping (DTW) algorithm works well on finding the similarity for a pair of time series data [25][26]. In general, DTW is widely-used in voice and pattern recognition because it obtains a precise matching along the temporal axis, and it maximizes the number of pointwise matches between two curves [27][28]. If two series with time slot points are given as input, the DTW algorithm can discover the best possible alignment between them by calculating the minimum sum of whole matched points on the two time series.

DTW is a recursive algorithm that starts with matching each point-to-point pair from the first element to the last element on the two input sequences. In Fig. 2, if we are going to align two sequences that are similar with observation, the application of Euclidean distance or Pearson correlation coefficient on these two sequences may be ineffective because of shifts on time axis. With DTW mapping method, local similarity can be found as the best mapping path within the two sequences to be aligned. As a result, if two genes with similar gene expression values at certain time slot points in microarray time series data are analyzed by DTW, it is more precise to determine the similarity between these two genes. This is because DTW can discover their similarity that cannot be identified with other distance measurements.

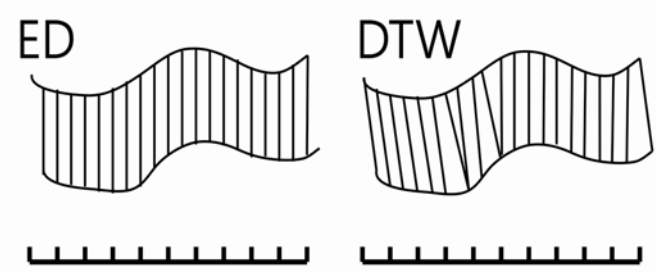

Fig. 2 Time series sequence similarity measurement

Equations of DTW algorithm are as follows:

Distance of two time slot points:

The distance between the elements of the two time series is computed as:

$$dis(i, j) = \left| x_i - y_i \right| \tag{1}$$

Base Conditions:

$$\begin{aligned} &e(0,0) = 0; \\ &e(1,1) = dis(x_1,y_1)*W_D; \\ &e(\mathrm{i},0) = \infty \text{ for } 1 \leqq \mathrm{i} \leqq \mathrm{I}; \\ &e(0,\mathrm{j}) = \infty \text{ for } 1 \leqq \mathrm{j} \leqq \mathrm{J}; \end{aligned} \tag{2}$$

where W_D is the weighted value for the paths in the diagonal direction.

Recursive Relation:

$$e(i, j) = \min \begin{cases} e(i, j-1) + dis(x_i, y_j) * W_V \\ e(i-1, j-1) + dis(x_i, y_j) * W_D \\ e(i-1, j) + dis(x_i, y_j) * W_H \end{cases} \tag{3}$$

where Wv, W_D, and W_H denote the weighted value for the paths in the vertical, diagonal, and horizontal directions respectively.

Output: DTW distance for two sequences X and Y:

$$\mathrm{DTW(X,Y)} = \frac{1}{n+m} * e(i, j) \tag{4}$$

where length of X and Y are n and m respectively.

3.1.1 Refinement of the DTW Algorithm

To further increase the efficiency and accuracy of our approach, we survey and analyze some variants of DTW and try to add them in our approach. Variants of DTW are usually divided into two aspects: speeding up and accuracy increasing. In the following subsections, we describe these two sorts of refinements for our approach.

3.1.1.1 Computational efficiency of DTW

DTW has a critical disadvantage: high computational cost. Typically, time complexity of the traditional DTW algorithm is O(n*m) for two input sequences with length n and m respectively. As we will show in Section 4, we use the Spellman's dataset to perform missing value imputation with totally 6178 genes in the dataset. If we naively use original DTW algorithm to calculate DTW distance of the whole 6178 genes, the computational time cost is awfully amazing that reduces the practicability of the algorithm. To solve this problem, several methods are proposed to speed up the calculation of DTW. Among all existing methods, we find the most useful one called FastDTW algorithm proposed by Salvador & Chan [29]. The authors propose their algorithm that has only linear time and space complexity. FastDTW uses a multilevel approach with three following operations:

(1) **Coarsening:** Coarsening means that FastDTW shrinks a time series into a smaller one which represents the same curve as accurately as possible with fewer data points.

(2) **Projection:** After FastDTW performs the coarsening step, it will find a minimum-distance warping path at a lower resolution, and use the path to guess another minimum-distance warping path in a higher resolution.

(3) **Refinement:** Finally, FastDTW refines each warping path in every resolution projected from a lower resolution with local adjustments.

If there are 32 points in an original time series, FastDTW cuts the data of points needed from 32 with two-times reducing rate (32->16->8->4->2). However, according to our experiment, we find that coarsening with three-times reducing rate performs better than coarsening with two-times reducing rate in terms of the dataset involved. This is because the datasets we use contain only 18 or 17 time points and need not too many coarsening operations. As a result, we modify the FastDTW algorithm and set the coarsening rate as three-times.

3.1.1.2 Accuracy of DTW

Except computational cost, there's the other attractive issue for the original DTW algorithm called the singularity problem [30]. In some cases, DTW generates unintuitive alignments where a single point on one time series is mapped onto a large subsection of the other time series. This kind of unexpected alignment is the

singularity problem. When the two sequences to be aligned are basically similar but with only slightly different amplitude at the peaks or valleys mapped on the two sequences, DTW will perform a one-to-many mapping for the time points. This kind of mapping will easily fail to find obvious and intuitive alignments for sequences. Therefore, it is essential to mitigate the singularity problem of DTW.

We survey and analyze several adjustments aiming to reduce the singularity problem of DTW and choose four of them to implement in our approach. In the following paragraph, we give a brief description about the four adjustments.

(1)Windowing: Berndt and Clifford proposed a restricted version of DTW so that the allowable paths for the DTW algorithm are limited with a warping window : $|i-j| \leqq w$, where w is a positive value [31]. This constraint may mitigate the seriousness of singularity problem but it is not able to prevent it.

(2)Slope weighting: Kruskall and Liberman proposed a modification of DTW so that the recursive equation in original DTW algorithm is replaced by r(i,j) = d(i,j) +min{r(i-1,j-1) , X*r(i-1, j), X*r(i, j-1)}, where X is a positive real number [32]. With this constraint, the warping path is increasingly biased toward the diagonal if the weighted value X gets larger. This modification of DTW takes the weighted value into consideration and it tries to slightly encourage the warping path to go in the diagonal direction to reduce singularity.

(3)Step patterns (Slope constraint): Itakura proposed a permissible step for the warping path with r(i,j) = d(i,j) +min{r(i-1,j-1) , r(i-1, j-2), r(i-2, j-1)} [33]. With this constraint, the warping path is forced to move one diagonal step if the previous step goes in the parallel direction to an axis.

(4)Derivative Dynamic Time Warping: Keogh and Pazzani introduced a modification of DTW called Derivative Dynamic Time Warping (DDTW) [34]. The authors consider only the estimated local derivatives of gene expression values in sequences instead of using the whole gene expression values themselves. The estimation equation is as follows:

Distance for two time points in two sequences:

$dis(i, j) = | E(X_i) - E(Y_i)|^2$

where $E(X_i) = \{ (X_i - X_{i-1}) + [(X_{i+1} - X_{i-1}) / 2] \} / 2$,

and $E(Y_i) = \{ (Y_i - Y_{i-1}) + [(Y_{i+1} - Y_{i-1}) / 2] \} / 2$ (5)

DDTW takes moving trends of certain subsequences into account in order to identify the distance of the two sequences.

These methods tend to form some constraints to force the warping path not to go along the horizontal or vertical direction too much. For the four variants of DTW, we consider slope weighting should bring the best results for imputation because it is more flexible and slightly encourages the warping path to go to the diagonal. Forcing the warping path to go to the diagonal too much may mitigate the singularity problem, but it is also at the risk of filtering the most suitable alignment of two genes.

We aim to retrieve suitable modifications of DTW to make our approach the best distance measurement. To fit this need, we implement the four above modifications for DTW in our approach. We also compare the imputation effectiveness resulted from of these modifications of DTW in order to improve the accuracy of our approach while applied in missing value imputation. Experimental results show that performing slope weighting brings the best result. The detail is discussed in Section 5.

3.2 Gene Ontology

The structure of GO can be viewed as a directed acyclic graph (DAG), where each GO term is a node, and the relationships between each term pair are arcs. Nodes with parent-children relations imply they are similarly defined within biological functions or reactions, while the children nodes are more specific. In other words, GO is a hierarchical structure, where child terms are more specialized and parent terms are less specialized. Each node in GO can have several parent nodes and several children nodes just in case that relations between each node do not form a cycle. The most commonly-used relations in GO are "is-a" relations and "part of" relations. For example, if the relation "term A is a term B" exists in GO, it means term A is a subtype of term B. By contrast, if the relation "term A is part of term B" stands, it means all children terms of term A with term A itself belong to term B. Each term in GO has one unique GO id for it, but the number of GO id does not represent the similarity between terms. These terms are used to annotate (describe) each gene to identify possible biological functions of it.

Fig. 3 illustrates an example of GO. For instance, GO id 0015749 shown in Fig. 3 denotes a term "monosaccharide transport", which has the relation "is-a" with its parent term (GO id 0008643). Equally, the parent-children relation between terms at consequent levels starting from one specific node can be traced level by level to the root node. If we start from the term GO: 0015749, we can trace the path from the selected node to the root as "GO: 0015749->GO: 0008643->GO: 0006810->GO: 0051234->GO: 0008150". With this directed acyclic graph structure, we can easily query the GO annotation terms of each gene in microarray gene expression time series data to give a general view of the biological activities of the genes involved.

Since each gene may have totally different terms in the three independent domains, deciding which domain we are focusing on is hence very important. Besides, one gene may be annotated by more than one term even in the same domain. Moreover, each term can have more than "one-to-one" relation with its parent term or children term. This forms various complicated reticular relations for annotated genes. Typically, a completed tracing path of GO annotation terms for one gene from the root to the leaf nodes is complex. Therefore, the way how we can use gene ontology differs from the involved data themselves and the algorithm we are applying. Sometimes it can also depend on which kind of analysis we are performing. With GO tern annotation, each gene can have a uniform representation across biological databases. As a result, GO annotations for genes can be taken as their external information while determining distance among them. For more

information about gene ontology, please refer to the Gene Ontology website. With GO annotations, we can hence acquire the relations for the genes involved in the experiment. The distance of two genes within biological aspect can be identified if we perform some quantitative assessments on the gene pair with their GO annotations.

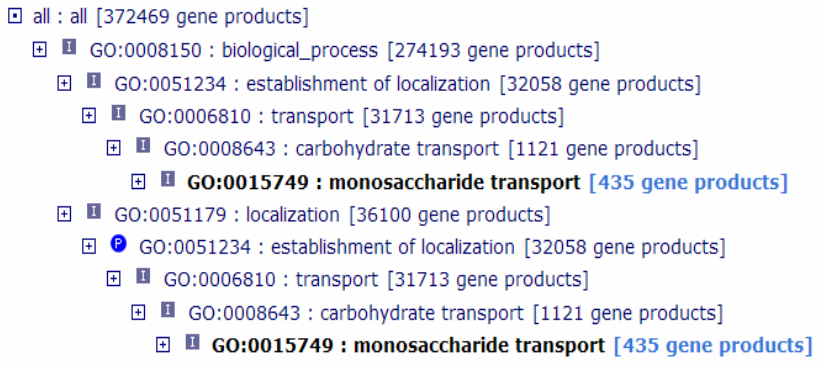

Fig. 3 Example of gene ontology

3.2.1 Application of Gene Ontology

The main task in microarray gene expression data analysis is to identify the gene pairs or groups that are highly co-expressed under individual experimental conditions. The most common procedure is performing distance measurement or classification and clustering on gene expression values. Nevertheless, with the usage of gene ontology, this task can be done more efficiently and accurately. Lord et al. investigate the validity of using GO information as semantic distance for gens compared with using traditional distance measurement [35]. Effectiveness of taking GO annotations as external information for genes is proved in the work. The authors also recommend choosing the "is_a" relation between gene pairs when determining distance for genes because "is_a" relation occupies almost 90% of all relations recorded. Consequently, we take the "is_a" relation into account in our approach because it is the most used relation in GO structure. Another example of using gene ontology is in [36]. In the study, GO terms are used as the information content. Semantic closeness is defined if the most immediate parent node is shared by two annotation terms. The authors also merge various GO-based distance measurement algorithms that consider intra and inter ontological relations by translating each relative term into a hierarchical relation within a smaller sub-ontology.

To our best knowledge, Tuikkala et al. propose the first method that uses gene ontology [37]. Operations of the algorithm proposed by the authors can be briefly descried as follows:

- First find the sets of GO ids for each pair of genes being identified.
- Create a table recording the tracing path of all terms annotated for both genes.

- Calculate the probability of the occurrence of each term in the table.
- Estimate all the parent-children relations of each term in the path-tracing table to determine whether the two genes have common ancestors.
- For genes that have shared parent nodes in the GO tracing path, calculate the mean probability of occurrence of all their matched GO term combinations.
- The mean probability of occurrence is taken as the distance between gene pairs.

This study proposes a typical approach that combines GO with gene expression values into processing to retrieve better missing value imputation results. The method proposed by the authors utilizes the information content for each annotation term in GO structure. According to the authors, probability of occurrence of each node has to be first calculated. This numeric value of each GO term is called the p value and it represents how informative this term is. If a term has a larger p value, it is often visited by most tracing paths in GO structure. As a result, the informative degree of this term is hence reduced. If the p values of GO terms that annotate one gene pair are large, these two genes tend to be less related.

Our approach takes the concept in Tuikkala et al.'s work into account. However, the authors in the work only find the minimum p value of shared ancestors of GO terms for two genes. This operation is insufficient because in GO structure two GO terms that are used to annotate different genes may have several relations. Having ancestors in common is only one of the relations for these two GO terms. We have to consider whether GO term pairs that annotate gene pairs form the parent-children relation, or they are even the same one. Theoretically, two genes with GO terms in common tend to be more relative than two genes having GO terms that only have shared ancestors. As a result, our approach gives different weighted values while calculating p values for the three term-term relations: the same terms, parent-children relation, and ancestor- sharing relation. The three relations are marked as case1, case2, and case3 in order as shown in Fig. 4. The star symbol in Fig. 4 illustrates the closest shared ancestors or two GO terms A and B. To find the best weighted values for these three relations, we implement several parameters and the results will be discussed in Section 5. Finally, the mean p value of all GO term pair combinations for two genes is used to the further semantic distance measurement of these two genes.

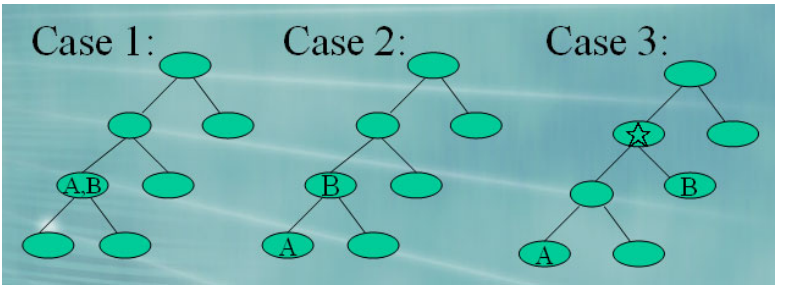

Fig. 4 Three relations between GO terms

3.3 DTW-GO Based Microarray Data Analysis

Our approach aims to provide an accurate distance measurement that takes both gene expression values and external information for genes into account. To fit this need, we survey and analyze existing assessments that provide distance measurements for gene pairs. Tuikkala et al. propose a distance measurement combining both distance for gene expression values and GO information for these genes as the semantic distance. Accuracy for the method is validated in the paper. The equation of the distance measurement for two genes (g_x, g_y) in Tuikkala et al.'s work is as follows:

$$DIS(g_x, g_y) = D^{GO}(g_x, g_y)^{\alpha} * D^{EXP}(g_x, g_y) \tag{6}$$

where D^{GO} is the average p value of all GO term pairs used to annotate g_x and g_y, α is a positive weighted parameter that controls how much the semantic dissimilarity value contributes to the combined distance. D^{EXP} is Euclidean distance of (g_x, g_y).

We then modify equation (6) as follows:

$$DIS(g_x, g_y) = D^{GO\_NEW}(g_x, g_y)^{\alpha} * D^{DTW}(g_x, g_y) \tag{7}$$

where D^{GO_NEW} is our estimation of p values of all GO term pairs used to annotate g_x and g_y as mentioned in Section 3.2.1,$\alpha$is the positive weighted parameter as shown in equation (6). D^{DTW} is the DTW distance of (g_x, g_y). In equation (7), we replace Euclidean distance with DTW distance, and replace original p value estimation with our approach. This is because we consider that DTW is more suitable than Euclidean distance while calculating distance between gene expression values. Equally, we use our new estimation for semantic distance between gene pairs to retrieve higher accuracy. After defining our distance measurement for gene pairs, the way we apply our distance measurement in missing value imputation and gene regulation prediction are described in coming subsections.

3.3.1 Missing Value Imputation

In this subsection, we propose a novel missing value imputation approach combining our distance measurement with the k-nearest neighbor (KNN) method. The KNN method selects genes with expression values similar to those genes of interest to impute missing values. For example, if we consider gene G that has one missing values at experiment time slot T, KNN would find K other genes that have a value at experiment time slot T, but with expression values most similar to Gene G in experiments time slot points except for T. A weighted average of values at experiment time slot T from the chosen K closest genes is then used as the estimation for the missing value in gene G. The weighted value of each gene in the K closest similar genes is given by the distance of its expression to that of gene G. Euclidean distance is commonly used to determine the k closet genes which are similar to the target gene G with missing values to impute. Here we use our

distance measurement as the estimation to determine the closeness of gene pairs. The steps of our approach for missing value imputation are as follows:

1. In order to impute the missing value G_{IJ} for gene I at time slot J, the KNN-impute algorithm chooses k genes that are most similar to the gene I and with the values in position k not missing.
2. The missing value is estimated as the weighted average of the corresponding entries in the selected k expression vectors:

$$G_{IJ} = \sum_{i=1}^{k} W_i \times e_{iJ} \tag{8}$$

3. The weighted value

$$W_i = \frac{1}{DIS(g^*, g_i) \times \Delta} \tag{9}$$

$$\text{where } \Delta = \sum_{i=1}^{k} [1/(Sim(g^*, g_i)] \tag{10}$$

and g* denotes the set of k genes closest to g_i, DIS(g*, g_i) is our distance measurement as shown in equation (7). Missing values for the target gene are hence imputed with our approach.

When applying the KNN-based method for the imputation of missing values, there are no constant criteria for selecting the best k-value. Choosing a small k value produces poorer performance after imputation. On the contrary, choosing a large neighborhood may include instances that are significantly different from those containing missing values. However, one study shows that setting k-value between 10 and 20 brings the best results for KNN imputation [5]. KNN can be an effective and intuitive imputation method if it works with a proper distance measurement for genes such as our approach.

3.3.2 Gene Regulation Prediction

After missing values are imputed with our imputation approach, we will then perform gene regulation prediction. Our approach first calculates and records the distance for all gene pairs with equation (7). The mean of numeric distance for all gene pairs is then calculated, assume DISmean. Gene pairs with distance less then DISmean are retained and recorded as potential regulatory gene pairs. These recorded gene pairs are subsequently compared with the known regulatory gene pairs called Filkov's datasets for validation. Detailed information for Filkov's datasets will be given in Section 4. Afterward, the number of mapping gene pairs between the validation datasets and gene pairs found based on our distance measurement is gathered. Theoretically, potential regulatory gene pairs should have shorter distance compared with the others in all gene pair combinations. The detail algorithm of our approach for gene regulation prediction is described as follows:

Algorithm for the proposed approach to identify regulatory gene pairs:

1. For all gene pair combinations, calculate the distance of each gene pair with equation (7).
2. Calculate the mean distance of all gene pair combinations, assume ***DISmean***.
3. Record gene pairs with distance less than ***DISmean***, assume S_{SIM}.
4. Compare S_{SIM} with Filkov's datasets. Count the number of matched gene pairs.

4 Datasets and Performance Assessment

In order to evaluate the effectiveness of our approach for missing value imputation and gene regulation prediction, we evaluate it on a real microarray dataset. In this section, we first give a brief description about the dataset used in our experiments. Subsequently, we introduce general performance assessment for missing value imputation and gene regulation prediction respectively.

4.1 Real Microarray Dataset

In this chapter, the microarray dataset we used is proposed by Spellman et al. and Cho et al. [38][39]. The data were obtained for genes of Yeast Saccharomyces cerevisiae cells with four synchronization methods: alpha-factor, cdc15, cdc28, and elutriation. Spellman's dataset is widely used as the real dataset in microarray research [5][7][8]. These four subsets of the dataset contain totally 6178 gene ORF profiles with their expression values across various amounts of time slots. In the dataset, the alpha sub-dataset contains 18 time points with seven minutes as the time interval, while the cdc28 sub-dataset contains 17 time points with ten minutes as the time interval. Here we choose alpha and cdc28 sub-datasets in Spellman's microarray datasets as the testing data because these two sub-datasets contain more non-missing gene expression values. Alpha sub-dataset contains missing values with nearly uniform distribution, while cdc28 sub-dataset contains a great portion of missing values occurring almost at some time points. These four kinds

H	I	J	K	L	M	N	O
alpha0	alpha7	alpha14	alpha21	alpha28	alpha35	alpha42	alpha49
-0.15	-0.15	-0.21	0.17	-0.42	-0.44	-0.15	0.24
-0.11	0.1	0.01	0.06	0.04	-0.26	0.04	0.19
-0.14	-0.71	0.1	-0.32	-0.4	-0.58	0.11	0.21
-0.02	-0.48	-0.11	0.12	-0.03	0.19	0.13	0.76
-0.05	-0.53	-0.47	-0.06	0.11	-0.07	0.25	0.46
-0.6	-0.45	-0.13	0.35	-0.01	0.49	0.18	0.43
-0.28	-0.22	-0.06	0.22	0.25	0.13	0.34	0.44
-0.03	-0.27	0.17	-0.12	-0.27	0.06	0.23	0.11
-0.05	0.13	0.13	-0.21	-0.45	-0.21	0.06	0.32
-0.31	-0.43	-0.3	-0.23	-0.13	-0.07	0.08	0.12
0.02	-0.33	-0.49	-0.3	-0.15	-0.24	0.4	0.53
-0.36	-0.19	0	-0.32	-0.27	-0.12	0.04	0.17
-0.1	-0.15	-0.01	-0.25	-0.16	-0.13	0.06	0.19
0	-0.01	0.12	-0.23	-0.13	0.25	0.3	-0.27
0.06	0.01	0.17	-0.14	0.01	-0.24	0.15	-1.34
-0.4	-0.22	0.19	-0.2	-0.09	0.41	0.13	-0.05
0.46	0.28	0.16	-1.72	0.33	0.05	0.22	0.3
-0.24	-0.95	-0.23	0.12	-0.02	0.23	-0.11	0.11
-0.02	-0.29	-0.07	-0.22	-0.06	-0.07	0.2	0.2
-0.11	-0.17	-0.16	0.04	0.1	-0.02	0.08	0.13
-0.36	-0.42	0.29	-0.14	-0.19	-0.52	0.04	0.04

Fig. 5 Spellman's yeast dataset

of sub-datasets record the gene expression reactions during different phases in cell cycle. With in these sub-datasets, empty values at certain time slot points are the missing values that we are going to impute and estimate. Fig. 5 illustrates the format of Spellman's dataset.

4.2 Assessment of Imputation Accuracy

For assessment of imputation accuracy, genes with missing values in microarray gene expression data are first filtered to generate a complete matrix. There are 3422 and 835 genes in the complete matrix for alpha and cdc28 sub-datasets, respectively. Missing values with different missing rates ranging from 1%, 5%, 10%, 15% and 20% in the complete matrix are deleted at random to create testing datasets. Afterward, we impute missing values in the generated testing datasets with our approach and other methods to recover the deleted missing values for each data set. The estimated values are compared to the original values in the complete matrix. For numeric accuracy assessment of missing value imputation, the commonest way is to calculate the Normalized Root Mean Square (NRMS) error. Equation for NRMS error is as follows:

$$\text{NRMS} = \sqrt{mean[(y_{predict} - y_{known})^2]} \Big/ std[y_{known}] \tag{11}$$

where $y_{predict}$ and y_{known} are estimated values and known values in the complete matrix respectively, and $std[y_{known}]$ is the standard deviation of known values. An imputation method is said to outperform others if the NRMS error of it is less than that of other imputation methods.

4.3 Accuracy of Gene Regulation Prediction

Filkov et al. review related literature and collect all known gene regulations of alpha and cdc28 subsets in Spellman's yeast cell dataset [40]. They also build a database to record all known gene regulations. In our evaluation, the known gene regulations recorded in Filkov's database are taken as the validation datasets. In the database, the number of recorded gene activations and inhibitions for alpha subset is 343 and 96 respectively, while for cdc28 subset is 469 and 155 accordingly. All these regulations are in the format of A (+) B that denotes gene A is an activator that activates gene B. Similarly, C (-) D represents an inhibitor gene C which inhibits gene D. Among these regulations recorded in Filkov's database, one gene could be the activator or inhibitor for more than two other genes. For example, gene ABF1 stands for the activator for totally eight different genes in cdc28 subset. Nevertheless, gene names in Filkov's database are denoted as the gene standard name, while the gene systematic names are used in Spellman's dataset. The systematic and standard names of a gene are like two kinds of aliases for this gene. As a result, a mapping procedure between gene standard name and systematic name is required. For this purpose, we designed a program to perform

this operation. The reference database for this phase is the Saccharomyces Genome Database (http://www.yeastgenome.org/) database. The SGD database acts as a platform for biologists to refer and query yeast gene information including the gene standard name and systematic name. During the process of gene name mapping, we find that some of the gene standard name in Filkov's database cannot be found in Spellman's dataset due to the different naming conventions. For example, the mapping systematic name for gene with standard name STA1 cannot be found in the SGD database. Consequently, regulations with gene STA1 are filtered that causes the decrease of gene activations in cdc28 subset from 469 to 466. Therefore, the pre-processing of the raw data is necessary. First, we parse all regulations of alpha and cdc28 sub-datasets in Filkov's database and retrieve unrepeatable involved genes. The parsing result is shown in Table 2. Involved genes in alpha and cdc28 sub-datasets are 295 and 357 respectively.

Table 2 Parsing result for Gene Regulations

Dataset	**Content**			
	No. of Genes	***No. of Activations***	***No. of Inhibitions***	***Total***
alpha	295	343	96	439
cdc28	357	466	155	621

Theoretically, the number of pairwise gene combinations for alpha subset is C(295,2) which equals to 43365, and the number of pairwise gene combinations for cdc28 subset is C(357,2) which equals to 63546. Known regulations in Filkov's database are marked as the validation measurement to estimate the accuracy of the gene regulation prediction methods. Finally, we apply our approach on these gene pairwise combinations and count the number of potential regulatory gene pairs found by our approach that are also listed in Filkov's database. Regulations of activations and inhibitions are summed up separately. The results are shown and discussed in Section 5.

5 Experimental Results and Discussion

This section presents the way we design our experiments for missing value imputation and gene regulation prediction, following by results of our experiments and discussions about them.

5.1 Design of Experiments

To apply our approach, we need to determine several conditions and parameters used in the equations of our approach. These include which DTW adjustment and

corresponding parameters that can produce the best results, the weighted value α in equation (7) that controls how much the semantic distance contributes to the combined distance, the selection of the proper GO domain, and the decision of weighted values of the three relations for GO terms. First, we set the weighted value α of our distance measurement as zero to focus on expression values themselves to test the effect of imputation performed by the four adjustments for DTW. We combine KNN method and DTW algorithm modified with FastDTW, along with four adjustments on DTW to impute missing values in alpha and cdc28 testing datasets. NRMS errors are then calculated as the assessment to determine whether an imputation method is effective or not. We choose the adjustment method for DTW that generates the best results as the distance measurement for gene expression values used in our approach. Subsequently, we try different combinations of parameters for weighted value α, GO terms used within the three GO domains, and various weighted values for three relational cases. The parameter set which brings the best results for our distance measurement is chosen. The comparison for NRMS errors of our approach and other methods is made. Due to space limitations, parts of experimental data are not listed. The number of K for KNN is set from 10, 15, 20, 50, and 100. DTW with weighting value ranges from 1.2 to 1.8 because we find that the effectiveness is reduced if the weighting value is larger than 1.8. DTW with windowing parameter ranges from 2 to 5 for the same reason. For each experiment, we run 10 times and calculate the average value to reduce the randomness. Finally, we apply our approach to predict potential regulatory gene pairs and count the number of matched pairs with Filkov's data set as the validation of our prediction approach.

5.2 Results and Discussion

In this subsection, we present our experimental results and discussions on the effect of DTW adjustments, effect of parameters used in GO, accuracy of missing value imputation, and practicability of gene regulation prediction in order.

5.2.1 Effect of DTW Adjustments

We find that the best result is achieved when we apply our proposed method with FastDTW-based modification and slope weighting with weighted value between 1.5 and 1.8. This indicates that DTW works well with slightly weighted values that force the warping path not to form the "one-to-many" mappings. Only with proper variants of DTW such as slope weighting can the imputation results be further improved. Therefore, we use slope weighting with weight value 1.8 as the adjustment for our approach.

5.2.2 Effect of Parameters Used in GO Similarity for Our Approach

After choosing slope weighting with weighted value 1.8 as the adjustment of DTW for our approach, we have to discover the best parameters for conditions and

parameters used for the GO part of our approach. For the three GO term-term relations, we try several combinations of the parameters and find that the best parameter for case2 is near the double as the parameter for case1. Similarly, parameter for case3 should be slightly less than the double of parameter for case2. The arrangement for these parameters conforms to the concept that if two GO terms are close or even the same in GO structure, the similarity for these terms is higher. For the validation of choosing the best parameters, we experiment different parameter values. Due to the space limitation, here we only propose the best parameter for the three GO term relations: the same terms, parent-children relation, and ancestor- sharing relation as case1 = 1, case2 = 2.4 and case3 = 4.5.

Table 3 NRMS values for different parameters of GO similarity in alpha and cdc28 dataset

Value of α \ GO domain		**Molecular Function**	**Biological Process**	**ALL**
0.25	alpha	0.74894	0.93786	0.63011
	cdc28	0.82003	1.00207	0.71102
0.50	alpha	0.73745	0.92661	0.62369
	cdc28	0.81060	0.99125	0.70331
0.75	alpha	0.74048	0.94236	0.66014
	cdc28	0.82358	1.10559	0.74224
1.00	alpha	0.75984	0.94713	0.69971
	cdc28	0.82276	1.13171	0.77186
1.25	alpha	0.76688	0.95967	0.73014
	cdc28	0.82053	1.13705	0.81677
1.50	alpha	0.78104	0.95140	0.74144
	cdc28	0.82201	1.14055	0.82746
1.75	alpha	0.79860	0.96237	0.75324
	cdc28	0.82555	1.14442	0.82738
2.00	alpha	0.79925	0.97145	0.75984
	cdc28	0.83850	1.15595	0.81154
2.25	alpha	0.80934	0.98366	0.76658
	cdc28	0.83339	1.15542	0.81108
2.50	alpha	0.81479	0.99147	0.77897
	cdc28	0.86462	1.16412	0.81766
2.75	alpha	0.82369	1.00647	0.79471
	cdc28	0.86146	1.16831	0.82707
3.00	alpha	0.83471	1.02169	0.80036
	cdc28	0.88596	1.17059	0.82822
3.25	alpha	0.85901	1.03004	0.80996
	cdc28	0.90022	1.18341	0.83748
3.50	alpha	0.86971	1.05526	0.82748
	cdc28	0.91826	1.18641	0.84589

Another task is to determine which GO domain produces the best results. For this experiment, we separate the GO terms for genes within the three domains: biological process (BP), molecular function (MF), and cellular component (CC). We then experiment the imputation results with GO terms within these three domains respectively, compared with the imputation results with the combinations of them. The result of CC is simply removed because the number of GO terms in CC is much less than terms in BP and MF so that it provides very little information for genes. Experimental results show that using all GO terms in the three domains produces the best results. This makes sense because GO information for genes is not sufficient without enough GO terms provided.

Besides, the weighted value αin equation (7) is also needed to be determined. Largerαvalues mean that the semantic distance is strongly emphasized to our distance measurement. We also try various values set between 0.25 to 3.5 as literatures suggest and record the corresponding imputation results. For experimental convenience, we focus on the testing data set with missing rate = 20% because missing rate of the involved real microarray data is close to the rate. Experimental results for determining these parameters are listed in Table 3. The results show that setting αas 0.5 brings the best imputation result.

5.2.3 Accuracy of Missing Value Imputation

After all parameters for our approach are determined, we then perform missing value imputation on alpha and cdc28 sub-datasets with our approach. We also implement several existing methods such as the KNN method, BPCA, and LLS for comparison. Experimental results are shown in Fig. 6 and Fig. 7 for alpha and cdc28 sub-datasets respectively. We observe and compare the results above and hence make some summaries. As shown in Fig. 6, the imputation method that only utilizes KNN with FastDTW achieves better results than using KNN. This proves that taking DTW distance as the distance measurement is more suitable than taking Euclidean distance while handling microarray time series data. BPCA and

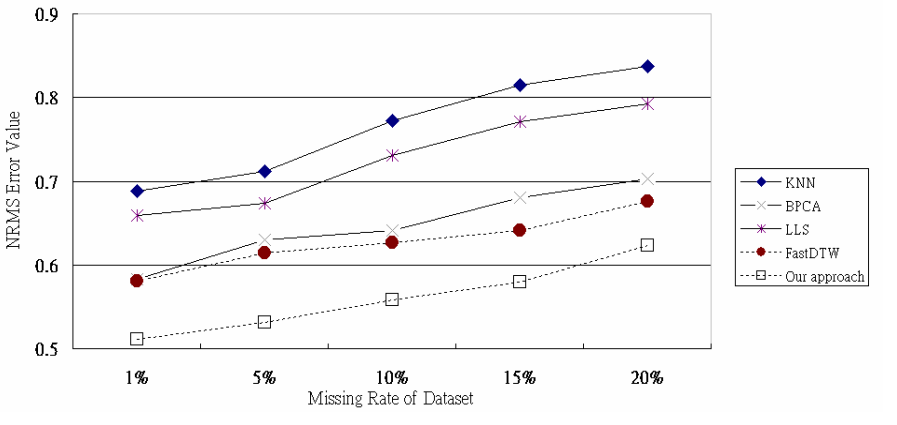

Fig. 6 Imputation results of alpha dataset

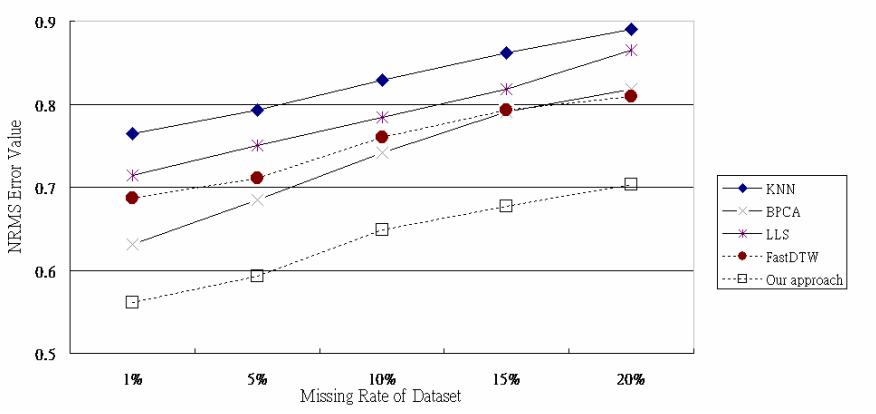

Fig. 7 Imputation results of cdc28 dataset

LLS seem to outperform KNN. Our approach is the most effective method when using FastDTW with slope weighting and proper parameters for GO distance measurement. Sequences of effectiveness of these imputation methods may change a little in certain percentage of missed data. This may result from the randomness while deciding which values to be removed in the complete matrix. We also experiment on the effectiveness of our semantic distance measurement based on GO with that of Tuikkala et al.'s work. Experimental results show that the NRMS error of our approach is about 0.4 less than that of Tuikkala et al.'s work. We do not list the whole experimental results due to space limitation.

Fig. 7 illustrates almost the same situation as Fig. 6. Basically results of all imputation methods are worse than results in alpha sub-dataset. This is because the cdc28 sub-dataset contains more missing values than the alpha sub-dataset. Theoretically, NRMS error increases when there are many missing values in the dataset. Furthermore, even using FastDTW brings better results than BPCA when the missing rate is larger than 15%. This shows the weakness of BPCA while dealing with microarray time series dataset with a large portion of missing values. To summarize, using our approach with suitable parameters can retrieve the best imputation results. Besides, we find that methods relative to KNN including KNN, FastDTW, and FastDTW with adjustments retrieve the best results when the number of K is set between K =10 and K = 20. This stands for Troyanskaya's research in 2001. As a result, while applying KNN or KNN-liked methods to impute missing values in microarray time series data, setting the number of K between 10 and 20 generates the best result empirically. Assigning the value of K less than 10 or more than 20 will not bring a better result.

5.2.4 Practicability of Gene Regulation Prediction

After missing values are imputed with our approach, we then perform gene regulation prediction. Yeung et al. propose their work for similar aim of regulatory gene

prediction [20]. Table 4 shows the experimental results of our approach and Yeung et al.'s method.

Table 4 Number of identified regulatory gene pairs

Dataset / # of Known gene pairs	Method			
	PCC	Yeung's method	DTW	Our approach
alpha(+)/ 343	36	223	215	297
alpha(-)/ 96	5	55	56	66
cdc28(+)/ 469	66	N/A	287	380
cdc28(-)/ 155	14	N/A	87	101

In Table 4, activation regulations and inhibition regulations from Filkov's database are separated. The four numbers lying in the first column denote the known gene regulations from Filkov's database for alpha and cdc28 sub-datasets. The numbers of mapping gene pairs found by the four methods, including Pearson correlation coefficient (PCC), Yeung et al.'s method, distance measurement with only DTW, and our approach are listed in the corresponding grids of the table. Gene pairs are said to be similar if their PCC values are larger than 0.5 according to Yeung et al.'s work. We can see that PCC can only find very few mapping known regulatory gene pairs, while Yeung et al.'s method than PCC. However, Yeung et al. only experiment alpha sub-dataset. Therefore we mark the result of cdc28 sub-dataset of Yeung et al.'s method with N/A. Obviously, with our method we can find much more known regulatory gene pairs compared with other methods. In alpha activation regulations, we can even find almost 297/343 = 86% of known regulatory gene pairs and 380/469 = 81% of known regulatory gene pairs in cdc28 activation regulations. The results show that our approach is not only accurate for missing value imputation but also effective for regulatory gene prediction.

6 Conclusions

In this chapter, we introduce a novel approach that provides an effective distance measurement for genes based on gene ontology (GO) annotations. GO is the structural definition for genes that provides biological information about genes or proteins. With the application of GO terms, external information such as biological functions for genes can be exploited so that the effectiveness of microarray data analysis is improved. We then perform missing value imputation by taking our

approach as the distance measurement for gene pairs combined with the KNN method. We also analyze and implement modifications of DTW both for efficiency increasing and accuracy improvement to achieve better imputation results. After missing values are imputed, our approach is then used to predict potential regulatory gene pairs. Experimental results show that our approach with specific adjustments outperforms other methods not only for missing value imputation, but also for gene regulation prediction. Our approach facilitates analysis for microarray time series data.

References

[1] Acuna, E., Rodriguez, C.: The treatment of missing values and its effect in the classifier accuracy. In: Proceedings of the Classification, Clustering XE Clustering and Data Mining Applications, pp. 639–648 (2004)

[2] Ouyang, M., Welsh, W.J., Georgopoulos, P.: Gaussian mixture clustering and imputation of microarray XE microarray data. Bioinformatics 20(6), 917–923 (2004)

[3] Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M.: Distinct types of diffuse large B-cell lymphoma identified by gene expression XE gene expression profiling. Nature 403, 503–511 (2000)

[4] Chen, L.C., Lin, Y.C., Arita, M., Tseng, V.S.: A novel approach for handling missing values in microarray XE microarray data. In: Proceedings of the International Computer Symposium, pp. 45–50 (2008)

[5] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarray XE microarrays. Bioinformatics 17(6), 520–525 (2001)

[6] Kim, S., Imoto, S., Miyano, S.: Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression XE gene expression data. Biosystems 75, 57–65 (2004)

[7] Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S.: A bayesian missing value estimation method for gene expression XE gene expression profile data. Bioinformatics 19(16), 2088–2096 (2003)

[8] Kim, H., Golub, G.H., Park, H.: Missing value estimation for DNA microarray XE gene expression XE gene expressiondata: local least squares XE local least squares imputation. Bioinformatics 21(2), 187–198 (2005)

[9] Choong, M.K., Charbit, M., Yan, H.: Autoregressive-model-based missing value estimation for DNA microarray XE microarray time series data. IEEE Transactions on Information Technology in Biomedicine 13(1), 131–137 (2009)

[10] Choong, M.K., Levy, D., Yang, H.: Study of microarray XE microarray time series data based on forward–backward linear prediction and singular value decomposition XE singular value decomposition. International Journal of Data Mining and Bioinformatics 3(2), 145–159 (2009)

[11] Shan, Y., Deng, G.: Kernel PCA regression for missing data estimation in DNA microarray XE microarray analysis. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1477–1480 (2009)
[12] Wang, X., Li, A., Jiang, Z., Feng, H.: Missing value estimation for DNA microarray XE microarray gene expression XE gene expression data by support vector regression imputation and orthogonal coding scheme. BMC Bioinformatics 7, 1–10 (2006)
[13] Wong, D.S.V., Wong, F.K., Wood, G.R.: A multi-stage approach to clustering and imputation of gene expression XE gene expression profiles. Bioinformatics 23, 998–1005 (2007)
[14] Liu, J., Ni, B., Dai, C., Wang, N.: A simple method of inferring pairwise gene interactions from microarray XE microarray time series data. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp. 3346–3351 (2005)
[15] Yang, A.C., Hsu, H.H., Lu, M.D.: Outlier filtering for identification of gene regulations in microarray XE microarray time-series data XE time-series data. In: Proceedings of the Third International Conference on Complex, Intelligent and Software Intensive System, pp. 854–859 (2009)
[16] Tseng, V.S., Chen, L.C., Chen, J.J.: Gene relation discovery by mining similar subsequences in time-series microarray XE microarray data. In: Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 106–112 (2007)
[17] Vlachos, M., Kollios, G., Gunopulos, G.: Discovering similar multidimensional trajectories. In: Proceedings of the Eighteenth International Conference on Data Engineering, pp. 673–684 (2002)
[18] Lee, M.S., Liu, L.Y., Chen, M.Y.: Similarity analysis of time series gene expression XE gene expression using dual-tree wavelet transform. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. I-413–I-416(2007)
[19] Friedman, N., Linial, M., Nachman, I., Péer, D.: Using Bayesian network to analyze expression data. In: Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, pp. 601–620 (2000)
[20] Yeung, L.K., Yan, H., Liew, A.W.C., Szeto, L.K., Yang, M., Kong, R.: Measuring correlation between microarray XE microarray time series data using dominant spectral component XE dominant spectral component. In: Proceedings of the Second Asia-Pacific Bioinformatics Conference, vol. 29, pp. 309–314 (2004)
[21] Mohammadi, A., Saraee, M.H.: Estimating missing value in microarray XE microarray data using fuzzy clustering and gene ontology XE gene ontology. In: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, pp. 382–385 (2008)
[22] Xiang, Q., Dai, X.: Proving missing value imputation in microarray XE microarray data by using gene regulatory information. In: Proceedings of the Second International Conference on Bioinformatics and Biomedical Engineering, pp. 326–329 (2008)
[23] Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. National Academy of Science 95, 14863–14868 (1998)

[24] Kalpakis, K., Gada, D., Puttagunta, V.: Distance measures for effective clustering of ARIMA time-series. In: Proceedings of the IEEE International Conference on Data Mining, pp. 273–280 (2001)
[25] Myers, C., Rabiner, L., Roseneberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Transactions On Acoustics, Speech, and Signal Processing ASSP-28, 623–635 (1980)
[26] Rabiner, L., Rosenberg, A., Levinson, S.: Considerations in dynamic time warping algorithms for discrete word recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing ASSP-26, 575–582 (1978)
[27] Furlanello, C., Merler, S., Jurman, G.: Combining feature selection and DTW for time-varying functional genomics. IEEE Transactions on Signal Processing 54(6), Part 2, 2436–2443 (2006)
[28] Yu, H.M., Tsai, W.H., Wang, H.M.: Query-by-Singing system for retrieving karaoke music. IEEE Transactions on Multimedia 10(8), 1626–1637 (2008)
[29] Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 11(5), 561–580 (2007)
[30] Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing ASSP-26, 43–49 (1978)
[31] Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of the Workshop on Knowledge Discovery in Databases (1994)
[32] Kruskall, J.B., Liberman, M.: The symmetric time warping algorithm: from continuous to discrete. Time Warps, String Edits, and Macromolecules: The theory and Practice of String Comparison (1983)
[33] Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-23, 52–72 (1975)
[34] Keogh, E., Pazzani, M.: Derivative dynamic time warping. In: Proceedings of the First SIAM International Conference on Data Mining, Chicag, Illinois (2001)
[35] Lord, P.W., Stevens, R.D., Brass, A., Goble, C.A.: Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics 19, 1275–1283 (2003)
[36] Sanfilippo, A., Baddeley, B., Beagley, N., Gopalan, B.: Enhancing automatic biological pathway generation with GO-based gene similarity. In: Proceedings of the International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, pp. 448–453 (2009)
[37] Tuikkala, J., Elo, L., Nevalainen, O.S., Aittokallio, T.: Improving missing value estimation in microarray XE microarray data with gene ontology XE gene ontology. Bioinformatics 22, 566–572 (2006)
[38] Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K.M., Eisen, B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray XE microarray hybridization. Molecular Biology of the Cell 9, 3273–3297 (1998)
[39] Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D.: A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell 2, 65–73 (1998)

[40] Filkov, V., Skiena, S., Zhi, J.: Analysis techniques for microarray XE microarray time-series data XE time-series data. In: Proceedings of the Fifth Annual International Conference on Computational Molecular Biology, pp. 124–131 (2001)
[41] Website: Gene ontology XE Gene ontology website, http://www.geneontology.org/ (last accessed on March 1, 2011)
[42] Website: Saccharomyces Genome Database XE Saccharomyces Genome Database, http://www.yeastgenome.org/ (last accessed on March 1, 2011)