Unraveling complex temporal associations in cellular systems across multiple time-series microarray datasets

Wenyuan Li, Min Xu, Xianghong Jasmine Zhou *

Molecular and Computational Biology, Department of Biological Sciences, University of Southern California, California, CA 90089, USA

ARTICLE INFO

Article history:
Received 7 June 2009
Available online 18 January 2010

Keywords:
Time-series microarray data
Complex temporal association

ABSTRACT

Unraveling the temporal complexity of cellular systems is a challenging task, as the subtle coordination of molecular activities cannot be adequately captured by simple mathematical concepts such as correlation. This paper addresses the challenge with a data-mining approach. We introduce the novel concept of a “frequent temporal association pattern” (FTAP): a set of genes simultaneously exhibit complex temporal expression patterns recurrently across multiple microarray datasets. Such temporal signals are hard to identify in individual microarray datasets, but become significant by their frequent occurrences across multiple datasets. We designed an efficient two-stage algorithm to identify FTAPs. First, for each gene we identify expression trends that occur frequently across multiple datasets. Second, we look for a set of genes that simultaneously exhibit their respective trends recurrently in multiple datasets. We applied this algorithm to 18 yeast time-series microarray datasets. The majority of FTAPs identified by the algorithm are associated with specific biological functions. Moreover, a significant number of patterns include genes that are functionally related but do not exhibit co-expression; such gene groups cannot be captured by clustering algorithms. Our approach offers advantages: (1) it can identify complex associations of temporal trends in gene expression, an important step towards understanding the complex mechanisms governing cellular systems; (2) it is capable of integrating time-series data with different time scales and intervals; and (3) it yields results that are robust against outliers.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

The cell is a dynamic system, where diverse biological processes are coordinated in precise and complex ways to maintain temporal coherence [1]. Understanding the rules governing such temporal complexity is an extremely challenging task. A possible first step is to extract frequent association rules, e.g., a set of genes which frequently exhibit the same activities. Recent progress in microarray has made it possible to simultaneously measure time-dependent expression profiles of all genes in an entire genome. Clustering algorithms can be used to identify groups of genes that exhibit similar temporal expression patterns [2–4]; time-warping approaches can be used to identify shifted gene expression correlations [5–7]. Neither approach, however, is up to the challenge of identifying more complex temporal expression associations (i.e., those involving more than simple co-expression or shifted co-expression).

An example of complex temporal association during a cell cycle is displayed in Fig. 1. The expression profiles of two yeast genes, CLN2 and SIC1, are plotted across two consecutive cell cycles based on a microarray study [8]. It is known that CLN2 is the major activator of SIC1 phosphorylation, an essential step for the cell cycle to progress through the G1 phase to the S phase [9]. There is no direct or shifted correlation between the two genes in either cycle. Nevertheless, their expression data contain distinct patterns that remain consistent across the two cell cycles (the solid points and lines in Fig. 1). In particular, the solid-line pattern of CLN2 and that of SIC1 show coordinated change repeatedly across the two cell cycles. Such periodic coupling suggests the existence of a non-trivial interaction between the two genes, a process beyond simple activation or inhibition at the expression level.

Naturally, such complex patterns are difficult to extract from the genome-wide expression profiles unless they occur frequently within a single dataset, or across multiple datasets. As time-series microarray datasets are generally quite short (more than 80% of all time-series expression datasets contain fewer than 9 time points) [10], we focus on identifying patterns that are repeated across multiple datasets. The rapid accumulation of publicly available time-series microarray data has made the task of identifying frequent temporal associations possible.

Integrating multiple microarray time-series datasets is a non-trivial task for several reasons. Firstly, the gene expression
values generated by different platforms are not directly comparable. Even using the same platform, alternative experimental parameters may result in systematic variations between datasets. These effects often lie beyond the capability of statistical normalization to remove [11–13]. Secondly, the time scale of the time-series datasets may vary (e.g., from minutes to hours). Thirdly, biological processes may unfold at different rates in response to experimental conditions, the organisms studied and even among individuals [5]. In other words, we can expect systematic variations in the magnitudes of gene expression values across datasets, as well as in the time scale of the variation and the units employed.

We propose a novel approach to overcoming the above problems, based on the concept of a “frequent temporal association pattern (FTAP)”. A typical example is illustrated in Fig. 2, which plots the expression profiles of three genes observed in three different datasets. In each dataset, the Pearson correlations between the expression profiles are very low (<0.3) in each dataset. However, subtle associations emerge when we investigate the three datasets simultaneously. Simply put, the rising trend in Gene 1 is always associated with a “down-up” trend in Gene 2 and a “up-down” trend in Gene 3. An FTAP exhibits associations in two dimensions: (1) each gene in the pattern exhibits the same trend in multiple datasets (“frequent association”); (2) within each dataset all trends in the pattern occur on the same subset of time points (“temporal association”). In Fig. 2, each gene exhibits the same trend in all three datasets, and all three genes go through their respective trends in each dataset at the same four selected time points. Such a complex association of three genes cannot be extracted from any single dataset.

To identify the FTAPs present in multiple time-series datasets, we designed a two-stage algorithm: (1) for each gene, we identify trends that occur frequently in the available datasets; and (2) we identify sets of genes that simultaneously exhibit their respective trends in multiple datasets. This approach has several important advantages: (a) it identifies complex and subtle temporal expression associations that cannot be captured by co-expression analysis; (b) it can integrate multiple time series with different time scales and intervals; and (c) patterns can be reliably identified even in the presence of outliers, since we allow the trends to skip over a
limited number of points in the series. Our algorithm is deterministic in the sense that it produces an exhaustive list of all FTAPs in the data. It is also scalable, since many of its operations can be performed in parallel.

We applied this algorithm to 18 yeast time-series microarray datasets. The method successfully identified FTAPs that are associated with specific biological functions. A significant number of these FTAPs include genes that are functionally related in a manner that goes beyond co-expression, patterns which could not have been captured by any clustering algorithm. With the rapid accumulation of microarray data in public repositories (a threefold increase per year) [14], we believe that our method can provide an atlas of complex temporal expression associations that will facilitate more advanced mechanistic studies.

2. Related work

Pattern extraction from time-series data are an active research area in data mining. Recently developed methods include motif discovery [15], discord identification [16], longest common subsequence matching, sequence averaging, segmentation [17] and indexing [18]. However, almost all these algorithms analyze time series from a single experiment rather than the varied experiments.

Clustering and comparing gene expression time series have attracted a lot of attentions recently. Aligning time series is a conventional method of evaluating the similarity between two time series, which is known as curve alignment in the biology literature, curve registration in statistics, and time warping in engineering. Two recent examples are the clustered alignment method proposed by Smith et al. [19] and Bayesian hierarchical curve registration by Telesca and Inoue [20]. These alignment methods, however, are also limited to the analysis of time series from the same experimental source (with the same time scale and unit). In contrast, the FTAP method proposed in this paper is specifically designed for investigating time series from different experimental sources with different time scales and intervals. A number of clustering algorithms have been applied to time course gene expression data, as surveyed in the review by Androulakis et al. [21]. However, most gene expression time series contain very few time points, posing a significant challenge to clustering analysis. By integrating multiple datasets, the FTAP method to some degree overcomes the difficulties associated with limited number of time points in single expression profiles. More importantly, FTAP analysis reveals subtle patterns (e.g., that illustrated in Fig. 2) which distinguishes from what any existing time-series methods identify: curves or shapes of time series are locally similar to each other, not within the single dataset, but across multiple datasets. Therefore, such groups of genes may not be discovered by time-series clustering algorithms [10,22].

In recent years, data streams have received considerable attention (a data stream is a particular kind of time series). The topic of mining multiple data streams is closely related to our work. Papadimitriou et al. [23] developed an algorithm called SPIRIT that discovers correlations and hidden variables in multiple data streams with the same time stamps and representing the same types of measurements. However, SPIRIT is not applicable to heterogeneous time series.

Some of the concepts used in this paper will sound familiar to those versed in the literature, but there are important differences to keep in mind. Firstly, out of a “trend” is different from that found in recent papers [24,25], where trends must consist of consecutive time points. As Fig. 2 illustrates, we allow a trend to pass over one or few time points. Therefore, several common techniques such as sliding time windows are not applicable to our problem. We use the concept of rank vectors which can be easily expanded and enumerated, to abstract and represent trends. We note that the longest increasing subsequence (LIS) problem [26], a classic of algorithm problems, also allows inconsecutive time points. In fact, the LIS is a specific example of the trends that our algorithm can identify. Secondly, the concept of “association” differs from that appearing in the literature on association rule mining [27], where an association rule has the form [gene A], [gene B] → (7 min)[gene C]. Our association definition describes a correlation of genes across multiple datasets on a subset of time points. The genes involved in a FTAP may exhibit more complicated patterns than the simple up-or-down trends of association rules.

3. The frequent temporal association model

In this section, we formally define the concept of a FTAP. We then decompose the problem of finding FTAPs into two subproblems for efficient mining.

3.1. Trend abstraction and representation

Our first task is to represent a temporal trend using its rank vector. The formal definition is given below.

**Definition 1.** Rank vector of a trend

Let \( \mathbf{x} \equiv [x_1, x_2, \ldots, x_k] \in \mathbb{R}^{1 \times k} \) be an expression vector consisting of \( k \) time points. The rank vector of \( \mathbf{x} \) is defined as \( \mathbf{r} \equiv [r_1, r_2, \ldots, r_k] \), where \( r_i \) is the rank of \( x_i \) if the elements of \( \mathbf{x} \) were sorted in ascending order.

An example of how to derive the rank vector of a trend is shown in Fig. 3(a). The rank vector summarizes the essential structure of a trend, bypassing systematic variations in expression values generated by different microarray platforms. Using rank vectors, all possible trends of length-(\( k + 1 \)) can easily be constructed from the trends of length-\( k \). This is accomplished simply by inserting the number \( (k + 1) \) into any position of the length-k rank vector. An example of this process is shown in Fig. 3(b), and its formal defini-
tion is given in Property 1. Iterating this process automatically generates all possible trends without duplicates, as presented in Property 2 and illustrated in Fig. 4. This property will prove very important to the efficiency of our method. Throughout the paper, trends are represented solely by their rank vectors \( r \).

**Property 1 (Trend extension or super-trend).** A trend \( r = [r_1, r_2, \ldots, r_k] \) can be extended to the trend \( r' \) with length \((k + 1)\) by placing the number \((k + 1)\) before or after any element of \( r \). That is, the possible vectors \( r' \) that can be obtained by extending \( r \) are \([k + 1, r_1, r_2, \ldots, r_k], [r_1, k + 1, r_2, \ldots, r_k], [r_1, r_2, \ldots, r_k, k + 1], \ldots, [r_1, r_2, \ldots, r_k, k + 1] \). The vectors \( r' \) are called super-trends of \( r \). This relationship is denoted \( r' \supset r \).

**Property 2 (Completeness of trend extension).** All trends with length greater than \( k \) can be obtained by extending all trends of length-\( k \), according to the process described in Property 1. No duplicate trends are generated during this process.

#### 3.2. Trend occurrences/substantiation

Before defining trend occurrence, we introduce the concept of a time-series sub-vector.

**Definition 2 (Time-series sub-vector).** Let \( x = [x_1, x_2, \ldots, x_n] \) be a time-series vector. A sub-vector of \( x \) is defined as \( y = [x_i, x_{i+1}, \ldots, x_j] \), where \( 1 \leq i \leq j \leq n \).

For the rest of the paper, we use the symbol “\( T \)" to represent the \( i \)th time point associated with \( x_i \) in a time-series vector \( x \).

**Definition 3 (Occurrence of a trend).** Let \( x = [x_1, x_2, \ldots, x_n] \) be a time-series vector and \( r \) be a trend. If \( y \subseteq x \) such that \( r_y \) is equal to \( r \), we say that \( r \) occurs in \( x \), or \( x \) supports \( r \). For each \( y = [x_i, x_{i+1}, \ldots, x_j] \) satisfying this condition, \( t = [T_1, T_2, \ldots, T_k] \) is called the occurrence vector of this trend.

An example of trend occurrence is shown in Fig. 3(c). Note that our definition allows for consecutive time points, since microarray data are noisy and biological processes may unfold at different rates under different conditions. Also, a given trend may occur multiple times within a long time series.

#### 3.3. Frequent trends

In the following, we formalize the concept of frequent trend and define a related property.

**Definition 4 (Frequent trend).** Given a set of time-series vectors (which can be of different lengths) and a single trend \( r \), the support of \( r \), denoted \( \text{supp}(r) \), is defined as the number of time-series vectors in which \( r \) occurs. Given a positive integer \( \beta \) (the “support threshold”), if \( \text{supp}(r) \geq \beta \), we say that \( r \) is a frequent trend.

An important property of frequent trends is that their support function is anti-monotonic. This fact is commonly known as the Apriori property for other types of frequent patterns [28,29].

**Property 3 (Apriori property).** Given a set of time-series vectors, a single trend \( r \), its support function \( \text{supp}(r) \) and the super-trends \( r' \) of \( r \), we find that \( \forall r' \supset r, \text{supp}(r') \leq \text{supp}(r) \).

Intuitively, if a trend \( r \) does not qualify as frequent, none of its super-trends will qualify as frequent either. This property can be used to eliminate a great many candidate trends from the analysis, vastly increasing the efficiency of the frequent pattern mining algorithm. Section 4.1 describes a search algorithm based on the Apriori property.

#### 3.4. Definition of a frequent temporal association pattern

A frequent temporal association pattern has three components: genes, datasets, and time points. Let \( G = \{G_1, \ldots, G_n\} \) be a subset of genes, \( D = \{D_1, D_2, \ldots, D_n\} \) be a subset of datasets, and \( F = \{f(D_1), \ldots, f(D_n)\} \) be a set of time point sequences selected from each dataset in \( G \). That is, \( f(D_i) \) is a vector containing the indices of time points selected from the \( D_i \)th dataset. In the triple \( (G, D, F) \), each combination of elements \( (G_i, D_j, f(D_j)) \) identifies a time-series sub-vector taken from the \( G_i \)th gene in the \( D_j \)th dataset. The components of the sub-vector are associated with the time points \( f(D_j) \). We denote the sub-vector itself as \( y(G_i, D_j, f(D_j)) \), and its rank vector as \( r_y = (G_i, D_j, f(D_j)) \). For example, the association pattern shown in Fig. 2 can be expressed as a triplet \( (G, D, F) \), where \( G = \{G_1, G_2, G_3\}, D = \{D_1, D_2, D_3\}, F = \{f(D_1), f(D_2), f(D_3)\} \). The individual vectors in \( F \) are \( f(D_1) = [75, 76, 77, 78], f(D_2) = [71, 72, 73, 75], \) and \( f(D_3) = [76, 77, 78, 71] \). In these terms, a FTAP satisfies the following definition.

**Definition 5 (FTAP).** Let \((G, D, F)\) be a triplet of genes, datasets and time points, and let \( r_y = (G_i, D_j, f(D_j)) \) be the rank vector of the time-series sub-sequences defined by \( F \): \( y = y(G_i, D_j, f(D_j)) \). Then \((G, D, F)\) is a FTAP if it satisfies

\[
\forall G_i \in G, \quad r_y = (G_i, D_j, f(D_j)) = \cdots = (G_i, D_j, f(D_j))
\]

Simply put, within a given dataset the sub-vectors of all selected genes must include the same time points (temporal association), and for a given gene all the time-series sub-vectors must yield the same rank vector (frequent association).

#### 3.5. Constraints on FTAPs

Often we need to take into account constraints when mining frequent trends. Let \( r = [r_1, \ldots, r_k] \) be a trend, and let \( y = [y_1, y_2, \ldots, y_n] \) be a specific occurrence of the trend \( r \) in the time series \( x \).

**Time interval constraint \( \theta_t \):** although we allow trends to skip over time points, it is often preferable to have few skipped points or none at all. Given a trend \( y = [y_1, y_2, \ldots, y_n] \), this constraint is formalized as follows: \( \forall j, (i_j - i_{j+1}) \leq \theta_t \), where \( 1 \leq j \leq (k - 1) \) and \( \theta_t \) is a positive threshold.

**Variance constraint \( \theta_v \):** as we often prefer trends associated with significant expression changes, we may require \( y \) to have large variance. The variance constraint is formalized as \( \sigma(y) \leq \theta_v \), where \( \theta_v \) is a positive real number. \( \sigma(y) \) and \( y \) are the standard deviation and mean of \( y \), respectively.

**Definition 6 (Anti-monotonic constraint).** Let \( y \) be a sub-vector of \( x \), i.e., \( y \subseteq x \), and let \( \Theta \) be a constraint on \( y \). If \( y \) does not comply with \( \Theta \), then no vector \( z \) satisfying \( y \subseteq z \subseteq x \) will comply with \( \Theta \). Such constraints are called “anti-monotonic constraints.”
1. Discover "frequent associations" for individual genes

Inputs: (1) For the $g$-th gene, the set of time series vectors from the $m$ datasets $\mathbb{D} = \{x_1, x_2, \ldots, x_m\}$. Each vector $x_g$ is the $g$-th row vector of the dataset $D_g$. (2) $\alpha$ is the minimum length of a trend, and $\beta$ is the minimum number of datasets in which a trend must occur to be considered frequent. The time interval constraint $\Theta_T$ and variance constraint $\Theta_V$ are also required.

Output: all couples $(D, T_g)$ recording the occurrence of a qualified trend for the $g$-th gene. $D = \{D_1, \ldots, D_m\}$ is a subset of the $m$ datasets. $T_g = (T(D_1), \ldots, T(D_m))$, where $T(D_i)$ is the collection of time points at which the trend occurs in the $D_i$-th dataset.

II. Discover "frequent temporal associations": groups of genes that exhibit their frequent associations simultaneously across multiple datasets.

Inputs: (1) all couples $(D, T_g)$ and (2) $\xi$, the minimum number of genes in a FTAP.

Output: all triplets $(G, D, T)$ that qualify as FTAPs.

Fig. 5. Decomposition of the FTAP discovery problem.

All constraints are useful for reducing the search space. However, only anti-monotonic constraints can be used to reject all possible super-trends after a given candidate trend has been ruled out. $\Theta_T$ is anti-monotonic constraint, but $\Theta_V$ is not.

### 3.6. Problem formulation and decomposition

Based on the above definitions, we formulate the problem of FTAP discovery as follows.

#### 3.6.1. Problem statement

Given $\alpha$, the minimum length of a trend; $\beta$, the minimum number of datasets in which a frequent trend must occur; and $\xi$, the minimum number of genes, our goal is to identify all FTAPs $(\mathbb{G}, \mathbb{D}, \mathbb{T})$ satisfying Definition 5.

#### 3.6.2. Problem decomposition

As discussed previously, there are two associations in a FTAP: "frequent association" and "temporal association". Thus it is natural to begin by discovering frequent trends in individual genes across the available datasets (denoted Subproblem-I). We will then compare the frequent trends of different genes to look for alignments along the time axis (denoted Subproblem-II).

There are $m$ datasets $\{D_1, D_2, \ldots, D_m\}$, each with $n$ genes and $l$ time points $\{D_i \subseteq \mathbb{R}^{n \times l}\}$. The rows of $D_i$ correspond to genes, and the columns correspond to time points. The inputs and outputs of the two subproblems are summarized in Fig. 5.

Fig. 6 illustrates this search process. The complete collection of datasets, genes, and time series is illustrated in Fig. 6(a). Subproblem-I discovers which trends occur frequently among the datasets, for one gene at a time. We employed standard techniques of frequent sequential pattern mining to design the search algorithm for Subproblem-I. Since the procedure is considers each gene independently, it can be easily implemented in a parallel computing framework. Its output is a table of all frequent trends and their occurrences in the dataset, as shown in Fig. 6(b). Subproblem-II looks for alignments between the frequent trends of different genes along the time axis. The final result is a set of frequent patterns with identical temporal associations. We model Subproblem-II as a frequent itemset mining problem, and adopted an existing algorithm for this purpose. This problem can also be parallelized by dividing the search space (i.e., restricting the collection to trends of length $k$ see Section 4.2).
4. The algorithm

4.1. Subproblem-I: discovering frequent trends for each gene

Given a set of time-series vectors corresponding to the same gene, identifying all the frequent trends is not a trivial task. There are \( k \) candidate trends for a time series of length \( k \). As presented in Property 3 of Section 3.3, the support (frequency) of a trend never increases with length (i.e., Apriori property). We can exploit this property to efficiently discover all frequent trends, starting with those of length \( x \) and working our way up to longer trends. A formal description of this process is given in Algorithm 1.

Algorithm 1 consists of three steps. (1) Initialization: set \( k \) to \( x \) and enumerate all candidate trends of length \( k \). (2) Frequency calculation: identify all frequent trends among the length-\( k \) candidates. (3) Extension: extend all the frequent length-\( k \) trends to length \( k + 1 \), and repeat steps (1)–(3). By approaching longer trends in this manner, a great many candidates are pruned in each iteration thanks to the Apriori property. To further increase efficiency, anti-monotonic constraints are also evaluated during the search for frequent trends. If the constraints invalidate a trend, all of its extensions will also fail. According to our experience, imposing even one anti-monotonic constraint such as \( \Theta_r \), greatly speeds up Algorithm 1. All other constraints (such as \( \Theta_v \)) are checked at the end (Line 20) of Algorithm 1.

The routine \( \text{EnumOccurrences} \) (Line 6 of Algorithm 1) discovers every sub-vector \( y \) of \( x \) whose rank vector is equal to \( r \), i.e., \( r_y = r \). We developed an efficient, depth-first search algorithm that recursively matches each element of \( r \) with every possible element of \( y \). The algorithm progresses through the elements of \( r \) one by one, until the entire rank vector has been checked. The elements of \( r \) and \( y \) are always matched from highest to lowest.

Algorithm 1 FrequentTrends

\[
\begin{align*}
\text{Input:} & \quad II = \{x_1, \ldots, x_m\}, m \text{ times-series vectors} \\
& \quad \text{(2) minimum trend length} \ z, \text{ minimum dataset support} \ \beta \\
& \quad \text{(3) } \Theta_v, \text{ the thresholds of anti-monotonic constraints such as } \Theta_r \\
& \quad \text{(4) } \Theta_0, \text{ the threshold of other constraints such as } \Theta_v \\
\text{Output:} & \quad \text{a list of couples } (S, T) \text{ identifying all the frequent trends in } II. \\
1: & \quad k \leftarrow x, \text{ then generate all length-} k \text{ trends and store in } \mathcal{R}_x; \" \text{ all trends with length less than } x \text{ are trivial.}\" \\
2: & \quad \text{repeat} \\
3: & \quad \forall r \text{ in the set of candidate length-} k \text{ trends } \mathcal{R}_x \text{ do} \\
4: & \quad \forall r \text{ in } II \text{ do} \\
5: & \quad \text{EnumOccurrences}(r, x, \Theta_v); \" \text{Find all occurrences of } r \text{ in } x \text{ complying with the anti-monotonic constraints defined in } \Theta_v.\" \\
6: & \quad \text{end for} \\
7: & \quad \text{end for} \\
8: & \quad \text{if } \text{supp}(r) > \beta \text{ then} \\
9: & \quad \mathcal{R}_x \leftarrow \mathcal{R}_x \cup r; \\
10: & \quad \text{if } k > x \text{ then} \\
11: & \quad \text{Record all its occurrence information to the couple } (S, T); \" \text{This trend qualifies as frequent.}\" \\
12: & \quad \text{end if} \\
13: & \quad \text{end if} \\
14: & \quad \text{end for} \\
15: & \quad \text{if } \mathcal{R}_x = \emptyset \text{ then} \\
16: & \quad \mathcal{R}_x \leftarrow 0, \text{ extend all qualified length-} k \text{ trends in } \mathcal{R}_x \text{ to generate a list of candidate length-} (k + 1) \text{ trends, and store them in } \mathcal{R}_z; \\
17: & \quad k \leftarrow k + 1; \\
18: & \quad \text{end if} \\
19: & \quad \text{until } \mathcal{R}_x \text{ is empty} \\
20: & \quad \text{Remove all occurrences } (S, T) \text{ which do not comply with any constraints in } \Theta_0.
\end{align*}
\]

4.2. Subproblem-II: aligning trend occurrences across genes

In Subproblem-I, we obtained the frequent trends for each gene and recorded all occurrences of those trends. In Subproblem-II, we will identify gene groupings with several simultaneous frequent trends, and where this pattern is repeated across multiple datasets. As Subproblem-I generates the complete list of length-\( k \) frequent trends, we can state the following important property:

Property 4. All FTAPs of length \( k \) can be identified by mining the list of length-\( k \) frequent trends obtained in Subproblem-I.

This property allows us to divide the search space of Subproblem-II by trend length, greatly simplifying the computation. The individual searches are reduced in size, and can be farmed out in a parallel processing framework.

Let \( (S, T)_g \) (corresponding to a row in Fig. 8) denote a single length-\( k \) frequent trend of gene \( g \). We collect the entire list of length-\( k \) trends of all genes, and mine the list of length-\( k \) FTAPs. Fig. 8(a) illustrates the process for \( k = 3 \). The frequent trend matrix is composed of length-3 vectors. The search for FTAPs can be modeled as a Frequent Itemset Mining (FIM) problem [28]. To explain our algorithm, we briefly digress to a discussion of FIM and related concepts.
FIM is rooted in basket analysis, where “basket” refers to a group of simultaneously purchased items or a bundled set of financial transactions. Given a historical dataset of baskets, market analysts are interested in knowing which items are frequently purchased together. To put it another way, they wish to identify “frequent itemsets”. The frequency of an itemset is defined as the number of baskets in which all the items occur together. This knowledge helps store managers with business decisions such as designing coupons and shelving merchandise to maximize profit.

Let \( I = i_1, i_2, \ldots, i_b \) be a set of items, and let \( D \) be a database of \( m \) transactions. Each transaction \( T \) is a set of items such that \( T \subseteq I \). A transaction \( T \) is said to contain \( X \) if and only if \( X \subseteq T \). \( X \) is called a frequent itemset if at least \( s \) transactions in the database contain \( X \). The output of an FIM algorithm is a list of all itemsets \( X \) that occur in at least \( s \) transactions.

In our case, a single “item” is the doublet \( (\mathcal{S}, \mathcal{T}) \), basically, a subset of time points in a single dataset, e.g., \( D_{16,77,75} \). Each trend (row) of the table can be viewed as a transaction containing these “items”. When we redraw the table of Fig. 8(a) in the form of Fig. 8(b), it is obvious that an FTAP corresponds to a frequent itemset of \( (\mathcal{S}, \mathcal{T}) \) over multiple trends and genes. Many efficient algorithms have already been developed for the frequent itemset mining problem. We employ the winner [30] of the “best implementation” award in the Workshop on Frequent Itemset Mining Implementations. As mentioned earlier, since the mining is done independently for each trend length \( k \), this algorithm is suitable for parallel computing.

5. Experimental study

5.1. Data description

Yeast DNA microarray time-series datasets were collected from the Gene Expression Omnibus (GEO) database. We obtained 18 datasets, each containing more than 10 time points (detailed description of the datasets are provided on the supplementary website: http://zhoulab.usc.edu/FTAP). These time-series datasets contain 5869 yeast genes and are measured under various experimental conditions: cell cycle, hydrogen peroxide response, filamentous-form growth, etc. The log transform was applied to data generated with affymetrix platforms. Expression values of different probes measuring the same gene were averaged. Datasets from replicated experiments were also combined and averaged.

5.2. Experimental setting and processing

We set the minimum trend length \( x = 5 \), as trends over 5 time points are more likely to represent specific patterns. We require a FTAP to occur in at least three datasets and three genes \((\beta = 3 \text{ and } \zeta = 3)\). Additionally, we do not allow trends to skip more than one time point \((\theta_t = 2)\) and require a large variance \((\theta_v = 0.5)\). We ran the algorithm on a Linux parallel computing cluster. The computation time of the algorithm in this setting is taken to be the greatest time required by any one node, even if many nodes were active for much shorter time. The maximum time is about 10 h for both Subproblem-I and Subproblem-II.
Our data-mining procedure resulted in 184,116 patterns, each of which corresponds to a distinct gene group. The patterns range in size from 3 to 156 genes, and involve a total of 4847 genes. In the next section, we analyze the results.

5.3. Analysis of frequent temporal association patterns

To assess whether the patterns obtained are biologically meaningful, we first determined the percentage of gene groups that are functionally homogeneous. Using the Gene Ontology (GO) biological process annotations, we consider a gene group to be functionally homogeneous if the functional homogeneity modeled by the hypergeometric distribution [31] to be significant at the α = 0.005 level. In addition, functional categories must be associated with Gene Ontology nodes at least four levels below the root. By these criteria, 56.28% of the gene groups are functionally homogeneous while randomization test (based on randomly assigned gene groups) only yields 30.38%. Furthermore, among groups containing at least four genes, 72.82% are functionally homogeneous (versus 35.98% by randomization test); and among groups of at least six genes, 75.36% are functionally homogeneous (versus 41.23% by randomization test). The trend does not occur in the corresponding dataset. (b) The second stage, discovery of all length-

It is widely known that while co-expressed genes tend to be functionally related, not all functionally related genes are co-expressed [32]. Identifying functional related genes beyond co-expression has been a major challenge in microarray analysis. Our method identified a significant number of such functionally related gene sets: 38.4% of the groups have average pairwise correlations less than 0.6. The power of our method attributes to its integrative analysis of multiple time-series datasets, since subtle patterns may emerge that could hardly be identified in any individual dataset. In a single dataset, it is difficult to separate complex patterns from the background noise. Our method is able to detect important trends by allowing the patterns to skip time points. This flexibility greatly enhances the sensitivity of our method, even when mining noisy data. In principle, allowing the algorithm to skip multiple time points would also allow us to look for associations between time series with different time units. For example, time series with 10-min and 30-min intervals could be aligned by skipping two time points.

It is difficult to separate complex patterns from the background noise. Our method is able to detect important trends by allowing the patterns to skip time points. This flexibility greatly enhances the sensitivity of our method, even when mining noisy data. In principle, allowing the algorithm to skip multiple time points would also allow us to look for associations between time series with different time units. For example, time series with 10-min and 30-min intervals could be aligned by skipping two time points.

Frequent Itemset Mining

Each trend is a transaction and contains items, each of which corresponds to a set of time points in a dataset, e.g. D1

A frequent itemset corresponds to a FTAP, for example

FTAP :

For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.
them because when a pattern occurs in multiple datasets, it is more easily differentiated from the background.

Fig. 9(a) shows one example of a functionally related gene group without co-expression. The genes DIN7, EAF6 and YLL066C, repeat the same coupled pattern on simultaneous time points in three different datasets (GDS1752, GDS608, GDS2318), representing three different growth conditions. Both EAF6 and DIN7 are known to be involved in DNA repair. The function of gene YLL066C is unknown, but evidences point to its role in DNA repair as well [33]. More specifically, DIN7 is a DNA damage-inducible mitochondrial nuclease that modulates the stability of mitochondrial DNA (mtDNA) in Saccharomyces cerevisiae [34]. EAF6, a subunit of the acetyltransferase complex NuA4, is recruited to the chromatin surrounding the damage and, through a series of chromatin modifications, it too contributes to DNA repair [35]. YLL066C, as a predicted Y element helicase, may play an important role in maintaining and repairing telomere [33]. Despite the genes’ possible involvement in the same biological process, their average pairwise correlations in the three datasets are only 0.24, 0.19, and 0.21, respectively. This association would not be captured by clustering algorithms.

To take another example, Fig. 9(b) shows a recurrent temporal coordination of the genes SEC2, SVP26, and YPT31. All three participate in vesicle-mediated transport, where SEC2 is essential for post-Golgi vesicle transport, SVP26 is a component of the early Golgi apparatus and is involved in COP II vesicle transport, and YPT31 mediates intra-Golgi traffic or the budding of post-Golgi vesicles from the trans-Golgi. Their average pairwise correlations in the three datasets are 0.31, 0.11, and 0.63.

The next interesting problem is to elucidate the mechanism underlying the subtle temporal activity coupling. For example, in Fig. 9(b), why does the expression of SEC2 increase initially while the expressions of SVP26 and YPT31 decrease? And why does the trend then rapidly reverse? These are not simple questions, and their answers lie beyond the scope of this paper. However, the advantage of our method is that we need no prior knowledge of the genes are similar. This is important for FTAP to seize groups of functionally related genes whose time series in the single dataset are not similar whatever globally or locally. To the best of our knowledge, we are the first to propose looking for this type of association pattern. Our experimental results demonstrate that the effectiveness of such patterns, and the real-world examples in Fig. 9 intuitively illustrate how three uncorrelated genes are contained in an FTAP.

To discover such complex patterns, we designed an efficient data-mining approach that exploits the anti-monotonic property of frequent trends. We applied the method to 18 yeast microarray time-series datasets, and discovered a large number of FTAPs. A significant fraction of these include functionally related genes beyond co-expression, which could not have been captured by clustering algorithms. Our method is efficient, scalable, and easily computed in parallel. With the rapid accumulation of time-series microarray data, this method could soon provide an atlas of temporal association patterns to facilitate more advanced and detailed studies.

We conclude with suggestions for future work. As our algorithm discovered a great patterns, it is useful to devise a means of statistically evaluating their importance. Secondly, it would be useful to post-process the patterns, arriving at a reduced set of gene groups with fewer overlaps. Finally, our method is equally applicable to time series from other domains such as stock market analysis, geophysics, meteorology, and social sciences. We expect it to discover many novel and complex patterns which have been overlooked by conventional methods.

6. Conclusion

In this paper, we have addressed the challenge of identifying complex temporal couplings in cellular systems. We proposed the novel concept of a “frequent temporal association pattern” (FTAP): multiple genes simultaneously exhibiting different expression trends, where the same pattern can be observed in several microarray datasets. By their very nature, such signals cannot be recognized within a single time-series dataset, but the availability of many such sets makes it possible to separate frequent associations from the background of noisy data and competing processes.

The FTAP has several advantages over traditional methods. First, it can identify the subtle signals which exist only in multiple time-series datasets and thus cannot be captured by co-expression analysis or time-series analysis within the single time-series dataset. Second, unlike existing time-series analysis methods, the FTAP definition does not require that the trends (or curves) of different genes are similar. This is important for FTAP to seize groups of functionally related genes whose time series in the single dataset are not similar whatever globally or locally. To the best of our knowledge, we are the first to propose looking for this type of association pattern. Our experimental results demonstrate that the effectiveness of such patterns, and the real-world examples in Fig. 9 intuitively illustrate how three uncorrelated genes are contained in an FTAP.

To discover such complex patterns, we designed an efficient data-mining approach that exploits the anti-monotonic property of frequent trends. We applied the method to 18 yeast microarray time-series datasets, and discovered a large number of FTAPs. A significant fraction of these include functionally related genes beyond co-expression, which could not have been captured by clustering algorithms. Our method is efficient, scalable, and easily computed in parallel. With the rapid accumulation of time-series microarray data, this method could soon provide an atlas of temporal association patterns to facilitate more advanced and detailed studies.

We conclude with suggestions for future work. As our algorithm discovered a great patterns, it is useful to devise a means of statistically evaluating their importance. Secondly, it would be useful to post-process the patterns, arriving at a reduced set of gene groups with fewer overlaps. Finally, our method is equally applicable to time series from other domains such as stock market analysis, geophysics, meteorology, and social sciences. We expect it to discover many novel and complex patterns which have been overlooked by conventional methods.

References
