Bioinformatics Advance Access published May 6, 2004

Analyzing Time Series Gene Expression Data

Ziv Bar-Joseph

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15217

Abstract

Motivation: Time series expression experiments are an increasingly popular method for studying a wide range of biological systems. However, when analyzing these experiments researchers face many new computational challenges. Algorithms that are specifically designed for time series experiments are required so that we can take advantage of their unique features (such as the ability to infer causality from the temporal response pattern) and address the unique problems they raise (for example, handling the different non uniform sampling rates).

Results: We present a comprehensive review of the current research in time series expression data analysis. We divide the computational challenges into four analysis levels: Experimental design, data analysis, pattern recognition and networks. For each of these levels we discuss computational and biological problems at that level and point out some of the methods that have been proposed to deal with these issues. Many open problems in all of these levels are discussed. This review is intended to serve as both, a point of reference for experimental biologists looking for practical solutions for analyzing their data, and a starting point for computer scientists interested in working on the computational problems related to time series expression analysis.

Contact: zivbj@cs.cmu.edu

1 Introduction

DNA microarray experiments are usually classified based on the type of array that is used in the experiment (cDNA and Oligonucleotide arrays) or according to the organism that is profiled. In this paper we distinguish between static and time series experiments. In *static* expression experiments, a snapshot of the expression of genes in different samples is measured while in *time series* expression experiments a temporal process is measured. Another important difference between these two types of data is that while static data from a sample population (for example, ovarian cancer patients) is assumed to be independent identically distributed (i.i.d.), time series data exhibits a strong autocorrelation between successive points.

Much of the early work on analyzing time series expression experiments used methods developed originally for static data (Spellman *et al.*, 1998; Friedman *et al.*, 2000; Zhu *et al.*, 2000; Troyanskaya *et al.*, 2001). More recently several new algorithms specifically targeting time series expression data were presented. As we discuss in this review, these algorithms both solve problems that are unique to time series expression data, and also allow us to fully utilize this data by taking advantage of its unique features. Gene expression is a temporal process. Different proteins are required (and synthesized) for different functions and under different conditions. Even under stable conditions, due to the degradation of proteins, mRNA is continuously transcribed and new proteins are generated. This process is highly regulated. One of the most important ways in which the cell regulates gene expression is by using a feedback loop. Some of the proteins are transcription factors (TFs). These proteins regulate the expression of other genes (and possibly, their own expression) by either initiating or repressing transcription. When cells are faced with a new condition (such as starvation (Natarajan *et al.*, 2001), infection (Nau *et al.*, 2002) and stress (Gasch *et al.*, 2000)), they react by activating a new expression program. In many cases, the expression program starts by activating a few transcription factors, which in turn activate many other genes that act in response to the new condition. Taking a snapshot of the expression profile following a new condition can reveal some of the genes that are specifically expressed under the new condition. However, in order to determine the complete set of genes that are expressed under these conditions, and to determine the interaction between these genes, it is necessary to measure a time course of expression experiments. This allows us to determine not only the stable state following a new condition, but also the pathway and networks that were activated in order to arrive at this new state.

Examples of time series expression experiments

The main purpose of this section is to demonstrate the wide range of biological questions that time series expression data can be used to answer. Many of these questions involve computational aspects, as we discuss below.

Table 1: Time series expression experiments

Reference	Method of arrest	Duration	Cell cycle length	Sampling rate
WT alpha Spellman *et al.* (1998)	alpha mating factor	0-119m	64m	every 7m
WT cdc15 Spellman *et al.* (1998)	temperature sensitive cdc15 mutant	10-290m	112	ev. 20m for 1 hr, ev. 10m for 3 hr, ev. 20 min for final hr
WT cdc28 Cho *et al.* (1998)	temperature sensitive cdc28 mutant	0-160m	85m	every 10m
fkh1/fkh2 knockout Zhu *et al.* (2000)	alpha mating factor	0-210m	105m	ev. 15m until 165m, then after 45m
yox1/yhp1 knockout Pramilla *et al.* (2002)	alpha mating factor	0-120m	60m	every 10m

Table 1: Summary of five different time series expression experiments. WT- wild type, m-minutes. All of these experiments were performed to study the cell cycle system in yeast. Note that the sampling rates are not always uniform, and vary between the different experiments. In addition, the cell cycle duration (the time it takes the cells to divide) differs depending on the experiment condition.

Biological systems: One of the most extensively studied systems is the cell cycle system. This system plays an important role in development, cancer, and many other biological processes, and has thus been extensively studied over the last four decades (Simon *et al.*, 2001). In table 1 we present 5 different time series expression experiments that were carried out to study various aspects of this system in yeast. At least as many time series expression experiments were carried out to study this system in humans (Whitfield *et al.*, 2002). A number of other systems have also been studied with time series expression experiments, including, among others, the circadian clock in mouse and humans (Storch *et al.*, 2002; Panda *et al.*, 2002).

Genetic interactions and knockouts: While an expression time course of a WT systems is useful to determine the set of genes that function in a system, and the order in which they operate, in order to study the function of individual genes we need to carry out knockout experiments. In a knockout experiment a gene is deleted from the genome, and the resulting strains are profiled using expression experiments. Such experiments allow us to determine the down stream effects of the knockout gene, which in turn can be used to identify target genes and to construct genetic interaction networks. Many knockout expression experiments have been carried out in the static case (Hughes *et al.*, 2000). More recently many knockout time courses are becoming available. These include cell cycle double knockouts (Zhu *et al.*, 2000; Pramilla *et al.*, 2002) and knockouts under stress conditions (Gasch *et al.*, 2000).

Development: Understanding development is key to understanding many genetic diseases. It is natural to use time series expression experiments to study development at the molecular level, and to identify genes that play key role in different stages of development. For example, an 80 time points expression experiment studying the development of the fruit fly *Drosophila* identified many genes that control specific stages in the fly developmental process (Arbeitman *et al.*, 2002). Similar experiments were carried out in other organisms, including the worm *c. elegans* (Kim *et al.*, 2001). More recently, expression experiments have been carried out to study human development. In (Ivanova *et al.*, 2002) human embryonic stem cells have been profiled in order to identify genes that are involved in the specific differentiation of these cells to various tissue types.

Infectious and other disease: Identifying genes that act in response to certain infectious disease is a key issue in developing drugs to fight these disease. Nau *et al.* (2002) studied a time course of human cells that were infected by four different pathogens. Other examples include Huntington disease (Xu *et al.*, 2002) and cancer (Whitfield *et al.*, 2002). As the examples above suggest, expression experiments can be used to answer many biologically important problems. However, as we discuss below, addressing these issues requires us to solve many computational problems as well.

Computational challenges in the analysis of time series expression data

The biological and computational issues that are addressed when analyzing gene expression data in general, and time series expression data in particular can be presented using a hierarchy of four analysis levels: Experimental design, data analysis, pattern recognition and networks.. Each of these levels addresses a specific biological and computational issues, and also serves as a pre-processing step for higher levels in the hierarchy. The rest if this review is devoted to these four levels. For each of these levels we first discuss the computational challenges and biological problems associated with this level, and then summarize some of the methods that have been suggested to solve these problems. For some levels we discuss in greater detail one of the methods that have been suggested.

2 Experimental design

Experimental design is key to the success of any expression experiment. In the past, various aspects of experimental design for general microarray experiments have been studied. For example, Zien *et al.* (2002) studied the number of microarrays required for expression experiments and Ben-Dor *et al.* (2000) studied the combinatorial problem of selecting representative probes for genes sequences in order to minimize cross hybridization. Below we focus on experimental design issues that are unique to time series expression data.

Challenges

An important computational problem for designing time series expression experiments is the determination of sampling rates. If the experiment is under-sampled, the results might not correctly represent the activity of the genes in the duration of the experiments, and key events will be missed. On the other hand, over-sampling is expansive and time consuming. Since many experiments are limited by budget constraints, over-sampling will result in shorter experiment duration, which might lead to missing important genes that participate in the process at a later stage. This problem has also biological consequences, since sampling rates should depend on the transcription and degradation rates of the messenger RNAs. In addition, under sampling can lead to temporal aggregation effects (Bay *et al.*, 2003). These effects may interfere with our ability to infer casual relationships since genes that are conditionally independent may appear as dependent if the sampling rate is too coarse.

Another problem related to some of the time series experiments is the problem of synchronization. When temporal systems are studied, cells need to be arrested so that all cells start at the same phase. Even if the arrest succeeds (which is not always the case (Shedden & Cooper, 2002b)) cells may lose their synchronization after a while (Whitfield *et al.*, 2002). Determining if and when cells go out of synch improves the analysis process, and can help in deciding which of the time points accurately reflect the behavior of the system being studied.

Algorithms for the experimental design level

As can be seen in Table 1, to date, sampling rates depended on biologists intuition, and varied (depending on the lab) even under similar experimental conditions (for example, the three alpha cell cycle experiments (Spellman *et al.*, 1998; Zhu *et al.*, 2000; Pramilla *et al.*, 2002) were sampled every 7, 15 and 10 minutes respectfully). Despite the importance of this problems we are not aware of any work that addressed these issues, perhaps indicating the complexity of this problem. Synchronization has received more attention. Shedden & Cooper (2002a) used a Fourier analysis algorithm to test the synchronizations of different arrest methods. Their method looks at how many genes are best explained by a periodic curve and how many are best explained using a-periodic curve. Using randomization tests, they can determine the actual synchronization achieved by the method, by comparing these sets of genes. They have determined that at least one of the human cell cycle experiments does not achieve considerable synchronization (Shedden & Cooper, 2002b) while most yeast cell cycle experiments did show considerable synchronization.

3 Data analysis level

In this level the focus is on the individual gene, and the issues addressed range from determining the continuous representation for each gene to aligning genes in different experiments and to identifying differentially expressed genes between two or more time series expression experiments.

Challenges

Following a microarray time series experiment, a key challenge is to extract the continuous representation of all genes throughout the course of the experiment. Such a representation enables us to overcome problems related to sampling rate differences and missing values. Unfortunately, the nature of microarray data makes straightforward interpolation difficult. Data are often very noisy and there are few replicates. Thus, simple techniques such as interpolation of individual genes can lead to poor estimates. Additionally, in many cases there are a large number of missing time-points in a series for any given gene, making gene specific interpolation infeasible. A particular problem arises when series are not sampled uniformly such as in (Spellman *et al.*, 1998; Chu *et al.*, 1998; Eisen *et al.*, 1998). Another challenge arises from the variability in the *timing* of biological processes. The rate at which similar underlying processes such as the cell-cycle unfold can be expected to differ across organisms, genetic variants, and environmental conditions. For instance, Spellman *et al.* (1998) analyze time-series data for the yeast cell-cycle in which different methods were used to synchronize the cells. It is clear that the cycle lengths across the different experiments vary considerably, and that the series begin and end at different phases of the cell-cycle (see Table 1). Thus, one needs a method to align such series to make them comparable. Some of the time series experiments are performed to detect periodic genes (Spellman *et al.*, 1998). Identifying such genes is challenging because different genes may have different phase and amplitude, and because of the noise present in all time series expression experiments. Finally, many experiments are carried out to identify genes with altered expression between samples. For instance, one would like to identify genes that have changed significantly after an experimental treatment or that differ between normal and diseased cells. However, comparisons of time series expression data sets are hindered by biological and experimental inconsistencies such as differences in sampling rate, variations in the timing of biological processes, and the lack of full repeats.

Continuous representation of time series expression data

In (Bar-Joseph *et al.*, 2003c) a method for representing expression profiles by aligned continuous curves is described. Cubic splines are used to represent gene expression curves. Cubic splines are a set of piecewise cubic polynomials, and are frequently used for fitting time-series and other noisy data. For gene expression, the authors use B-splines, a type of spline that represents each point as a linear combination of a set of basis polynomials. By knowing the value of these splines at a set of control points, one can generate the entire set of polynomials using these basis functions. Due to noise and missing values, estimating these splines from expression data for each gene could lead to over-fitting of the data. Instead, spline coefficients of co-expressed genes are constrained to have the same covariance matrix, and thus other genes in the same class are used to estimate the missing values of a specific gene. The goal is to infer, for each gene and each class of genes, the value of their spline control points. The parameters of this model are computed using an EM algorithm. This method provides a superior fit for time series expression data when compared to other methods (which are discussed below), though it is only appropriate for relatively long (> 10 time points) experiments. This method can also be used to continuously align time series expression data (see Figure 1). The algorithm seeks to maximize the similarity between the two sets of expression profiles by adjusting a shift and stretch parameters for one of the profiles, holding the second profile fixed. Finally, this algorithm was used to identify differentially expressed genes in time series expression data (Bar-Joseph *et al.*, 2003b). Using the aligned continuous curves, a global difference measure between these two curves is computed. In order to determine the significance of this global difference, a noise model for individual samples is used to find a curve that best explains this difference. Thus, this algorithm is able to assign significance to temporal expression differences, even when only partial repeats are available (for example, repeats of time point 0).

We conclude this section with a word of caution. While splines are useful for fitting time series expression data, their success is highly dependent on the measured data. In particular, if the data is too noisy or sampled at a very low rate, splines (or any other continuous representation) will not be able to generate an accurate representation of the expression profile.

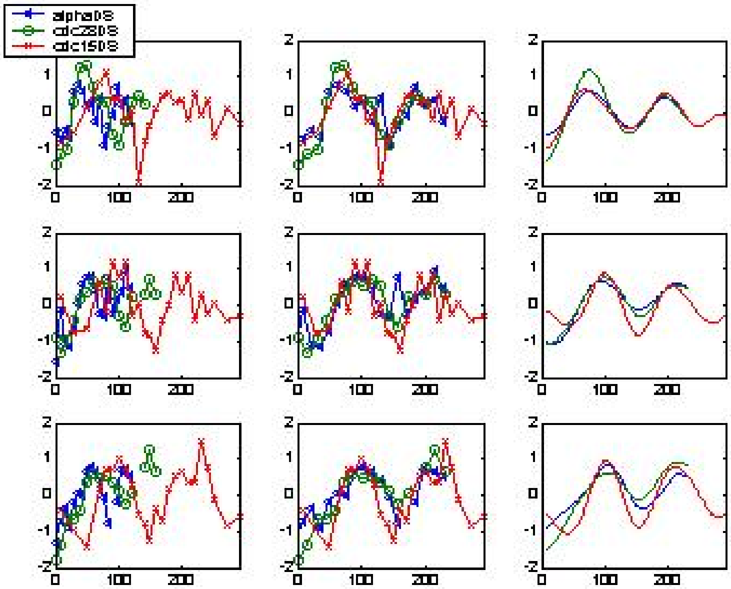

Figure 1: Left panel: Raw values for three cycling genes in three different cell cycle experiments. Middle panel: Alignment of the measured values. Right panel: Aligned continuous curves for the three genes. Using the continuous representation results in reduced noise, correctly making the expression pattern of each of these genes similar across the different synchronization methods.

Other algorithms for the data analysis level

Several papers have used simple interpolation techniques to estimate missing values for gene expression data. Aach & Church (2001) use linear interpolation to estimate gene expression levels for unobserved time-points. D'haeseleer *et al.* (1999) uses spline interpolation on individual genes to interpolate missing time-points. Zhao *et al.* (2001) fitted a statistical model to all genes in order to find those that are cell cycle regulated. This method uses a custom tailored model, relying on the periodicity of the specific dataset analyzed, and is thus less general than the above approaches. As for alignment, Aach & Church (2001) presented a method for aligning gene expression time-series using an algorithm based on Dynamic Time Warping, a discrete method that uses dynamic programming and is conceptually similar to sequence alignment algorithms. Many algorithms have been introduced for identifying genes differentially expressed between two experiments in the static expression case (Golub *et al.*, 1999; Dudiot & et al, in press). However, due to differences in sampling rates, and variations in the timing of biological processes (see Table 1), such methods cannot be directly applied to most time series expression datasets. Relatively few methods were developed for identifying such genes in time series data. These methods include cluster analysis (Zhu *et al.*, 2000; Gasch *et al.*, 2000), in which clusters of genes are compared across the two experiments and Generalized SVD (presented by Alter *et al.* (2003)) which are also used to detect differences between sets of genes, but are less appropriate for comparing individual genes. Other researchers have used custom tailored models (Xu *et al.*, 2002) to identify genes with expression patterns that differ significantly from the assumed model. Identifying periodically expressed genes was also the subject of recent research. In addition to the Shedden and Cooper method discussed above, Wichert *et al.* (2004) used a periodgram, which is also based on Fourier analysis to identify significant peaks in time series expression data.

4 Pattern recognition level

Due to the large number of genes that are profiled in each experiment, clustering is needed to provide a global overview of the experiment results. In addition, clustering was used to determine function for unknown genes (Eisen *et al.*, 1998), to look at expression programs for different systems in the cell (Spellman *et al.*, 1998) and for identifying sets of genes that are specifically involved in a certain type of cancer or other diseases (Alon *et al.*, 1999). Another major challenge in gene expression analysis is effective data organization and visualization. It is thus not surprising that early work on gene expression analysis have focused on this level, and several clustering algorithms have been suggested for gene expression data (Ben-Dor *et al.*, 1999; Tamayo *et al.*, 1999; R. & R., 2000).

Challenges

While clustering is important for all expression experiments (static and time series), there are a number of issues that should be specifically addressed when clustering time series expression data. First, most clustering algorithms (including k-means and self organizing maps (Tamayo *et al.*, 1999)) treat their input as a vector of independent samples, and do not take into account the temporal relationship between the time points, and the duration each time point represents. Thus, these algorithms cannot benefit from the known dependencies among consecutive points. In addition, since many time series are sampled non uniformly, such independence assumption might skew the results. Another important problem is inferring causality from time series expression experiment. Since some genes act as regulators of other genes, by looking at the temporal expression patterns of genes we might be able to infer relationships between regulators and the genes they regulate, and explain how genes are regulated in the cell. When analyzing time series expression datasets we are interested not only in the clusters themselves but also in the relationships between the different clusters. This is especially important when using clustering algorithms for visualization purposes. For time series data, such algorithms should provide an overview of the dynamics of the system as well as the different groups (or clusters) involved. Finally, while tens of thousands of genes are profiled at each experiment, many time series data sets are short ($<$10) and noisy. Even if expression experiments become cheaper, this problem is not likely to disappear, since obtaining large quantities of human samples is an issue. This problem requires the development of novel methods for identifying true patterns in such short datasets.

Hidden Markov models for clustering time series datasets

In (Schliep *et al.*, 2003) the authors present a Hidden Markov Model (HMM) based clustering algorithm for time series expression data. While other clustering algorithm (such as k-means and hierarchical clustering) will produce the same result when the time points are randomly permuted, HMMs take into account the temporal nature of the expression datasets resulting in higher quality clustering. HMMs can be defined by the following parameters: The (hidden) states, S_i, the probability of starting at a given state, π_i, the transition probability from state i to state j $a_{i,j}$, and $b_i(w)$ the emission probability of symbol $w \in \Sigma$ at state S_i. Given gene expression data for n genes denoted by O, our goal is to find a partition of the data into K HMMs $\Lambda_1 \cdots \Lambda_K$ which will maximize the likelihood of the data given the learned HMM model. For gene expression data, the omission probabilities are assumed to be Gaussians with fixed variance (see Figure 2). In order to determine the parameters of this model an iterative algorithm is used. This algorithm performs two steps: In the first steps each gene is assigned the most likely HMM and in the second step the parameters of each HMM are determined using the genes that were assigned to it.

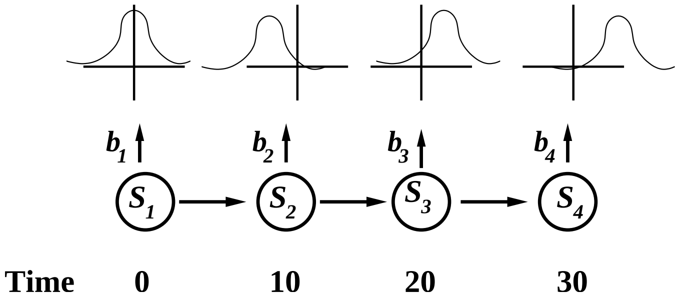

Figure 2: HMMs for clustering time series expression data. Each series is modeled using a HMM which takes into account both, transition from prior state, and omission probability at current state. The omission probabilities (b_is) are assumed to be Gaussians with different mean and fixed variance.

Other algorithms for the pattern recognition level

Alter *et al.* (2000) and Holter *et al.* (2000) used Singular Value Decomposition (SVD) to determine the different phases in the yeast cell cycle, and to order genes in the two dimensional plane based on these phases. They have determined that expression patterns for all cell cycle genes can be reconstructed from a linear combination of two base profiles. As mentioned above, most algorithms for clustering gene expression data were developed for static expression datasets. In order to take into account the actual time that each sampled point represents, the splines framework discussed above can be extended so that it can be used when the class information is not given (Bar-Joseph *et al.*, 2003c). This results in the clustering of the *continuous representation* of each expression profile, which is important for non uniformly sampled data. Other researchers looked at relationships between temporal profiles of different genes. For example, Ramoni *et al.* (2002) studied clustering time series expression data based on their dynamics. Qian *et al.* (2001) used local alignment algorithms to study time-shifted and inverted gene expression profiles and to infer causality from time series profiles. Holter *et al.* (2001) used a time translational matrix to model the temporal relationships between different modes of the Singular Value Decomposition (SVD). In order to improve data visualization and analysis, an algorithm for optimally ordering the leaves of an hierarchical clustering tree was presented in (Bar-Joseph *et al.*, 2003a). By ordering the leaves of this tree, the algorithm allows users to identify not only the clusters but also the relationships between these clusters.

5 Networks

The final analysis level is the networks level in which we focus on the interactions between genes, and attempt to build descriptive and predictive models for different systems in the cell. For regulatory networks, the components of such models are the genes (or their protein products) that are involved in a specific system, and the transcription factors that regulate the system. Such models provide a description of the process under investigation, and the interactions that take place during the activation of the system (how are the different genes involved activated ? which genes are turned on first ? which next ? etc.). Predictive models should also be able to address question about different perturbations of the system (what happens when we knockout a specific gene ? what will happen when we add a specific drug to the environment ?). Models are useful for many applications. For example, in drug discovery researchers are interested in identifying proteins that are at the root of a certain disease. Using these models we can determine which genes are the causes, and target them to prevent the spread of the disease. Another important application is to identify side effects of a certain treatment. Targeting a protein can cause a number of side effects which might be toxic to the cell. Using genetic interaction models we can determine likely side effects in advance, and target only those proteins for which these side effects are minimal.

Challenges

A key issue when trying to model a system is data integration or information fusion. Unlike the lower analysis levels, expression data is not enough to accurately reconstruct these networks. Due to the large number of genes, many different hypothesis can be generated to explain a specific expression pattern. In order to constrain the number of possible hypothesis, we need to incorporate additional biological data, and to ’fuse’ such data with gene expression data. For example, genetic regulatory networks are key to understanding expression programs in the cell (Lee *et al.*, 2002). In order to accurately construct such networks, we need to combine disparate biological data sources, including protein-DNA binding data, protein interaction data and expression data.

Another challenge at this level is obtaining perturbation data. For regulatory networks, such data is likely to consist of gene knockouts under different experimental conditions. For temporal systems, we will need time series knockout experiments. However, large quantities of such knockout data will not be available in the near future. Time series expression experiments are expensive, and since the number of potential knockouts is larger than the number of genes (since, in many cases a single knockout does not alter expression, and only a double knockout can hint at the relationships between the knocked out factors and the genes they interact with (Zhu *et al.*, 2000)). Note that many of the additional datasets mentioned above (for example protein-DNA binding data) are also static. Thus, they too require models that can combine time series and static data.

As mentioned above, sampling rates and temporal aggregations can have a negative impact on our ability to correctly reconstruct temporal networks (Bay *et al.*, 2003). Thus, solving problems at lower analysis levels is an important step toward reconstructing temporal interaction networks.

Finally, we need to select an appropriate computational modeling framework for such systems. A generative model for various systems will be the ultimate goal, however, due to the large number of genes involved, the current amount of data cannot support such models on a large scale. One possible intermediate solution is to construct networks from gene modules - sets of genes that are assumed to share a common function or be involved in the same pathway. Algorithms for identifying such modules, and assembling them to temporal networks are an important first step toward modeling such systems.

A dynamic model for the cell cycle system in yeast

In (Lee *et al.*, 2002) a sub-network discovery algorithm for the cell cycle system was presented (see Figure 3). This algorithm works by combining gene expression and protein-DNA binding data as follows. First, the algorithm uses the full set of transcription factors and a large set of expression data to identify gene module; sets of genes that are co-expressed and co-regulated. Next, genes in the generated modules are flagged if they are determined to be involved in the cell cycle system based on their cyclic behavior in a time series gene expression experiment. A statistical test based on the hypergeometric distribution is then used to determine which modules contain a significant number of flagged genes and the transcription factors that regulate these modules are identified. Finally in the third step, the module discovery algorithm is run using relevant expression data, the flagged genes, and the list of identified regulators, producing a set of modules with genes and factors directly involved in the process of interest. In order to present a dynamic model for this system the expression profiles of the genes in the discovered modules are interpolated. Next, one module is selected as an anchor (or time point 0), and the rest of the modules are aligned to this module using the continuous alignment algorithm discussed above. However, in this case only a shift between two

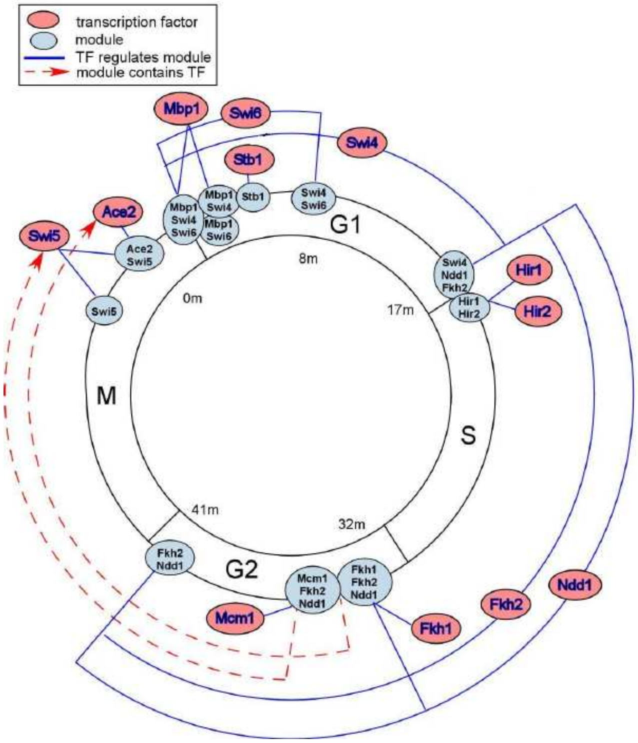

Figure 3: Computational discovery of the cell cycle subnetwork using expression and binding data. This automatically recovered network is extremely similar to the one described in (Simon *et al.*, 2001), which required considerable prior biological knowledge to construct.

sets is allowed, and thus the shift parameter can be used to determine the actual starting time (with respect to the selected time point 0 module) of the aligned module. This process is repeated for all modules, resulting in a temporal ordering of the discovered modules. Note that the actual activation time of the factors can be determined even if their expression profiles does not change under the experimental condition by using the time of the modules they regulate. This results in the correct assignment of factors to different stages in the cell cycle system, without directly observing their protein levels.

Other algorithms for the network analysis level

Recently, a number of papers discuss the use of dynamic Bayesian networks (DBNs) for modeling time series expression data. DBNs are an extension of Bayesian networks (BNs), which have been successfully applied to model static expression data (Pe'er *et al.*, 2001). The main advantage of DBNs for gene expression data is that unlike BNs, which are acyclic, DBNs allow for cycles, which are common in many biological systems. In addition, DBNs can also improve our ability to learn causal relationships by relying on the temporal nature of the data. Below we mention some of these papers and their conclusions.

Ong *et al.* (2002) generated DBNs to model response to physiological changes in E. coli. Their method used prior biological knowledge for grouping together genes in the same operon that are transcribed together and are thus co-regulated. This allowed them to reduce the parameters of their model and increase its significance. They tested their method on 169 genes in 9 operons from E. coli and concluded that their DBN network correctly identified correlations with related genes for 4 of the 9 operons.

Perrin *et al.* (2003) presented a DBN model containing hidden variables (that is, nodes for which we do not have direct observation) to overcome both biological and measurement noise. Their model uses an extension of the linear regression model with normally distributed noise. They applied their method to model the DNA repair network in E. coli, focusing on the 8 main genes in that system. In general they have found that their method was able to extract the main regulatory circuits for this system. As for prediction, they observed a very high correlation between the prediction of the generated network for the next time step and the actual values observed (0.97) and a somewhat lower correlation for similar prediction of multiple steps (0.65).

Kim *et al.* (2003) use DBNs to model a 45 genes subnetwork of the cell cycle system in yeast. By comparing the resulting network with a previously determined network from the KEGG database they have concluded that many of the edges can be correctly identified using DBNs.

In order to test the application of DBNs to gene expression data, and to determine their accuracy, Husmeier (2003) performed a simulation based analysis. Unlike with real biological data, in a simulation based study we know what the correct network is, and so it is possible to compare the resulting network and the true (underlying) network. This was done by selecting a significance threshold for each edge, and determining the true positive (how many correct edges were recovered) and false positive (how many recovered edges do not exist in the true network) rates. Husmeier concludes that while the global network recovered by DBNs is not useful, local structures can be recovered to a certain extent.

As can be seen from the above papers, DBNs seem like a promising direction for modeling temporal system. However, currently most work is limited to the analysis of a small set of genes or to simulation studies. As we discuss below, more data and improved computational tools are required in order to obtain better large scale models for these systems.

6 Discussion

As can be seen from the challenges sections above, time series expression data raises many new computational problems. Some of these problems have been addressed (though there is still a lot of room for improvement) while others will require new computational tools. The way I see it, the interesting computational problems (leading to interesting computational algorithms) remain on two levels: The experimental design level and the networks level. While the other levels still present a number of unsolved problems (for example - dealing with short time series expression experiments), these two levels have received relatively little attention to date, and are ripe for further research. Below I outline some promising research directions for these two levels.

One of the key issues in the experimental design level is determining sampling rates for time series expression experiments. As mentioned above, these rates currently depend on the intuition of the experimentalists. One possible direction for determining these rates is to use an online algorithm. This algorithm can start by sampling at an initial (low) frequency. Next, the algorithm will seek to determine whether the initial frequency is appropriate or not by computing confidence intervals for a reconstructed curve based on the current available data. If not, the algorithm should choose a new point to sample. This process should be repeated until the required confidence level is reached.

Synchronization is another issue which deserves more attention. Recently, two methods have been presented to deconvolve cell cycle expression data in order to overcome synchronization loss. Both methods assume that cell cycle rates for yeast cell population follow a normal distribution. Using this assumption, Lu *et al.* (2004) presented a method for resynchronizing time series expression data by assuming that expression profiles follow a specific pattern (sinusoids). Bar-Joseph *et al.* (2004) used a different method which relies on external information (FACS or budding index) to determine a model for synchronization loss, and then uses this model to deconvolve expression data. Both methods seem to work well for yeast, but the problem of deconvolving human cell cycle data, which is not synchronized for one complete cycle, is still open. Extension of the models presented in these papers is a promising direction for solving this problem.

As for the networks level, as mentioned above it is unlikely that we will have a large number time series knockout datasets in the near future. Unlike time series, there are already large amounts of static knockout data (Hughes *et al.*, 2000) available. Thus, one way to overcome this limitation is to construct models and frameworks which will efficiently combine a small amount of time series wild type and knockouts datasets with larger amounts of static knockout datasets. Such methods will use time series data to construct temporal models and static data to determine parameters and interactions for these models.

Another promising direction is information fusion. The better our prior assumptions the more accurate the resulting networks will be. Such priors on the network structure can be obtained from other high throughput data sources including protein-protein and protein-DNA interactions. This has been already done for static expression data and we anticipate similar success with time series data. Protein-protein interaction and protein-DNA binding data can also be used to overcome two other issues related to modeling biological systems: The huge search space for network structures and the fact that gene expression data does not always correlate with protein levels (while network nodes usually correspond to proteins). Specifically, many transcription factors are expressed at low levels and are activated by post translational modification. By relying on external sources for determining network structure we can still identify regulation events even though they do not appear in the expression data. Another way to address the latter problem is to use perturbation (knockout) data, as mentioned above.

Many recent attempts to model static systems in the cell rely on gene modules. Different papers define gene modules in different ways, but informally such modules can be thought of as a small collection of genes that share similar function in a specific system. The main advantage of using modules is the gain in statistical confidence obtained from a collection of genes versus binary (gene gene) interactions. This should be especially useful for time series analysis because of the limited data available. As mentioned above, some of the current work on modeling temporal systems have already used modules ((Lee *et al.*, 2002; Ong *et al.*, 2002)) and I anticipate that they will be further used in the future.

Finally, while some success has been achieved in modeling temporal systems, I believe that we are just at the early stages of this work. In particular, building a complete generative model for a large system in the cell remains one of the holy grails of computational biology.

References

Aach, J. & Church, G.M. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics*, **17**, 495–508.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. & Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, **96**, 6745–6750.

Alter, O., Brown, P. & Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, **97**, 10101–6.

Alter, O., Brown, P. & Botstein, D. (2003). Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *PNAS*, **100(6)**, 3351–6.

Arbeitman, M., Furlong, E., Imam, F., Johnson, E., Null, B., Baker, B., Krasnow, M., Scott, M., Davis, R. & White, K. (2002). Gene expression during the life cycle of drosophila melanogaster. *Science*, **298**, 2270–75.

Bar-Joseph, Z., Demaine, E., Gifford, D., Hamel, A., Srebro, N. & Jaakkola, T. (2003a). k-ary clustering with optimal leaf ordering for gene expression data. *Bioinformatics*, **19**, 1070–78.

Bar-Joseph, Z., Gerber, G., Jaakkola, T., Gifford, D. & Simon, I. (2003b). Comparing the continuous representation of time series expression profiles to identify differentially expressed genes. *Proc. Natl. Acad. Sci. USA (PNAS)*, **100(18)**, 10146–10151.

Bar-Joseph, Z., Gerber, G., Jaakkola, T., Gifford, D. & Simon, I. (2003c). Continuous representations of time series gene expression data. *Journal of Computational Biology*, **3-4**, 341–356.

Bar-Joseph, Z., Farkash, S., Gifford, D., Simon, I. & Rosenfeld, R. (2004). Deconvolving cell cycle expression data with complementary information. *Bioinformatics (Proceedings of ISMB)*, to appear.

Bay, S.D., Chrisman, L., Pohorille, A. & Shrager, J. (2003). Temporal aggregation bias and inference of causal regulatory networks. In *Proceedings of the IJCAI Workshop on Learning Graphical Models for Computational Genomics*.

Ben-Dor, A., R., S. & Z., Y. (1999). Clustering gene expression patterns. *Journal of Computational Biology*, **6**, 281–297.

Ben-Dor, A., Karp, R., Schwikowski, B. & Yakhini, Z. (2000). Universal dna tag systems: A combinatorial design scheme. In *Proceedings of The Sixth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, 65–75.

Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D. & Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell.*, **2(1)**, 65–73.

Chu, S., DeRisi, J. & et al (1998). The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

D'haeseleer, P., Wen, X., Fuhrman, S. & Somogyi, R. (1999). Linear modeling of mrna expression levels during cns development and injury. In *Pac. Symp. on Biocomputing*, 41–52.

Dudiot, S. & et al (in press). Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. *Statistica sinica*.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *PNAS*, **95**, 14863–14868.

Friedman, N., Linial, M., Nachman, I. & Pe'er, D. (2000). Using bayesian network to analyze expression data. *Journal of Computational Biology*, **7**, 601–620.

Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D. & Brown, P. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, **11(12)**, 4241–4257.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gassenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J. & et al (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Holter, N., Mitra, M., Maritan, A., Cieplak, M., Banavar, J. & Fedoroff, N. (2000). Fundamental patterns underlying gene expression profiles: simplicity from complexity. *PNAS*, **97**, 8409–14.

Holter, N.S., Maritan, A., Cieplak, M., Fedoroff, N. & Banavar, J. (2001). Dynamic modeling of gene expression data. *PNAS*, **98**, 1693–1698.

Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H. & et al (2000). Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, **19**, 2271–2282.

Ivanova, N., Dimos, J., Schaniel, C., Hackney, J., Moore, K. & Lemischka, I. (2002). A stem cell molecular signature. *Science*, **298**, 601–604.

Kim, S., Lund, J., Kiraly, M., Duke, K., Jiang, M., Stuart, J., Eizinger, A., Wylie, B. & Davidson, G. (2001). A gene expression map for c. elegans. *Science*, **293**, 2087–92.

Kim, S., Imoto, S. & Miyano, S. (2003). Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics*, **4**, 228–235.

Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., Hannett, N., Harbison, C., Thompson, C., I., S. & et al (2002). Transcriptional regulatory networks in $saccharomyces\ cerevisiae$. *Science*, **798**, 799–804.

Lu, X., Zhang, W., Qin, Z., Kwast, K., & Liu, J. (2004). Statistical resynchronization and bayesian detection of periodically expressed genes. *Nucl. Acids. Res.*, **32**, 447–455.

Natarajan, K., Meyer, M., Jackson, B., Slade, D., Roberts, C., Hinnebusch, A. & Marton, M. (2001). Transcriptional profiling shows that gcn4p is a master regulator of gene expression during amino acid starvation in yeast. *Mol. Cell Biol.*, **21(13)**, 4347–68.

Nau, G., Richmond, J., Schlesinger, A., Jennings, E., Lander, E., & Young, R. (2002). Human macrophage activation programs induced by bacterial pathogens. *PNAS*, **99**, 1503–1508.

Ong, I., Glasner, J. & Page, D. (2002). Modelling regulatory pathways in e. coli from time series expression profiles. *Bioinformatics*, **18 Supp 1**, 241–248.

Panda, S., Antoch, M., Miller, B., Su, A., Schook, A., Straume, M., Schultz, P., Kay, S., Takahashi, J. & Hogenesch, J. (2002). Coordinated transcription of key pathways in the mouse by the circadian clock. *Cell*, **109(3)**, 307–20.

Pe’er, D., Regev, A., Elidan, G. & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17(S1)**, S215–24.

Perrin, B.E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J. & D’Alche-Buc, F. (2003). Gene networks inference using dynamic bayesian networks. *Bioinformatics*, **19**, II138–II148.

Pramilla, T., Miles, S., GuhaThakurta, D., Jemiolo, D. & Breeden, L. (2002). Conserved homeodomain proteins interact with mads box protein mcm1 to restrict ecb-dependent transcription to the m/g1 phase of the cell cycle. *Genes Dev.*, **16(32)**, 3034–3045.

Qian, J., Dolled-Filhart, M., Lin, J., Yu, H. & Gerstein, M. (2001). Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. *J Mol Biol*, **314(5)**, 1053–66.

R., S. & R., S. (2000). Click: A clustering algorithm for gene expression analysis. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 307–316.

Ramoni, M., Sebastiani, P. & Kohane, I. (2002). Cluster analysis of gene expression dynamics. *PNAS*, **99(14)**, 9121–6.

Schliep, A., Schonhuth, A. & Steinhoff, C. (2003). Using hidden markov models to analyze gene expression time course data. *Bioinformatics*, **19**, i264–i272.

Shedden, K. & Cooper, S. (2002a). Analysis of cell-cycle gene expression in saccharomyces cerevisiae using microarrays and multiple synchronization methods. *Nucleic Acids Res*, **30(13)**, 2920–29.

Shedden, K. & Cooper, S. (2002b). Analysis of cell-cycle-specific gene expression in human cells as determined by microarrays and double-thymidine block synchronization. *PNAS*, **99(7)**, 4379–84.

Simon, I., Barnett, J., Hannett, N., Harbison, C., Rinaldi, N., Volkert, T., Wyrick, J., Zeitlinger, J., Gifford, D., Jaakkola, T. & Young, R. (2001). Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, **106**, 697–708.

Spellman, P.T., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast $saccharomyces$ $cerevisia$ by microarray hybridization. *Mol. Biol. of the Cell*, **9**, 3273–3297.

Storch, K., Lipan, O., Leykin, I., Viswanathan, N., Davis, F., Wong, W. & Weitz, C. (2002). Extensive and divergent circadian gene expression in liver and heart. *Nature*, **418**, 78–83.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. & Golub, T. (1999). Interpreting patterns of gene expression with self organizing maps: Methods and applications to hematopoietic differentiation. *PNAS*, **96**, 2907–2912.

Troyanskaya, O., Cantor, M. & et al (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, **17**, 520–525.

Whitfield, M., Sherlock, G., Saldanha, A., Murray, J., Ball, C., Alexander, K., Matese, J., Perou, C., Hurt, M., Brown, P. & Botstein, D. (2002). Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Mol. Biol. Cell*, **13(6)**, 1977–2000.

Wichert, S., Fokianos, K. & Strimmer, K. (2004). Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, **20**, 5–20.

Xu, X., Olson, J. & Zhao, L. (2002). A regression-based method to identify differentially expressed genes in time course studies and its application to inducible huntington's disease. *Hum. Mol. Genet.*, **11**, 1977–1985.

Zhao, L.P., Prentice, R. & Breeden, L. (2001). Statistical modeling of large microarray data sets to identify stimulus-response profiles. *PNAS*, **98**, 5631–5636.

Zhu, G., T., S.P., Volpe, T., Brown, P., Botstein, D., Davis, T. & Futcher, B. (2000). Two yeast forkhead genes regulate cell cycle and pseudohyphal growth. *Nature*, **406**, 90–94.

Zien, A., Fluck, J., Zimmer, R. & Lengauer, T. (2002). Microarrays: How many do you need? In *Proceedings of The Sixth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, 321–330.