Improved Methods for Inferring Regulatory Networks from Temporal Expression Data

R. Brian Potter
potterrb@comcast.net
Student ID: 5153081

BIOC218 – Spring 2003

Over the past few years, the advent of microarray technology has enabled the simultaneous measurement of the expression levels of thousands of genes. When the expression levels of these genes are measured at multiple time points during an experiment, the result is a temporal expression profile. These expression profiles may be processed to extract the underlying gene regulatory network relationships. This paper broadly reviews the methods available for exploring time-series expression data. It then focuses on some of the inherent problems of using correlation-based methods (in particular), after which two recent methods are described that attempt to overcome these problems. Finally, preliminary results are presented for a new similarity measure and optimizing technique intended to overcome these same problems. The yeast cell cycle data of Spellman *et. al.* [26] is used to evaluate these new methods.

INTRODUCTION

Microarray technology has made it possible to interrogate thousands of genes simultaneously. A series of time-related microarray experiments may be used to monitor a biological process such as the cell cycle [26], cellular differentiation, and environmental responses such as cell starvation [20]. While the simultaneous monitoring of thousands of genes over tens of experimental time points saves days, weeks, and even years of laboratory time, the huge amounts of data generated require analysis techniques especially tailored for extracting information from large datasets.

Gene Regulatory Network Inference

Differential Equations. Many methods have been developed that may be applied for inferring regulatory relationships from time-series gene expression data. The most intuitive approach is differential equations, since the underlying biological networks behave accordingly. Activation, inhibition and self-degradation are all rate-limited processes ideally modeled by differential equations. While this approach may be intuitive, however, modeling gene regulatory networks using differential equations involves the simultaneous determination of potentially thousands of parameters. Currently available time-series datasets are woefully inadequate for this task, having been sampled too infrequently, resulting in an underdetermined problem. This problem of underdetermination has been addressed using the assumption that gene regulatory networks are sparse and therefore limiting the non-zero elements in the dataset to a number that allows a solution [9][11]. Underdetermination has also been addressed by applying multiple regression analysis to the equations [18].

Model-based Methods. Model-based approaches have also been used. The mixed-effects model with B-splines [21] models the time dependency of the gene expression data and associated noise. It uses an expectation maximization (EM) algorithm to cluster the all of the temporal gene expression patterns into a fixed number of clusters, after which a smoothed mean gene expression curve is built from each resulting

cluster. Two problems with this method are how to determine the number of clusters (a parameter that must be specified ahead of time) and how to deal with outlier patterns that are forced by the algorithm into a cluster. The duplication growth model [4] is a method that models biological events such gene duplication, partial duplication, and rewiring. While it is more biologically motivated than the purely statistical techniques discussed below, it is not a fully physico-chemical method like the rated-based methods using differential equations.

Fourier Analysis. An obvious approach to analyzing time-series data is Fourier analysis, especially for identifying cell cycle regulated genes. Spellman *et. al*. [26], whose cell cycle data is used to assess the methods proposed in this paper, used correlation-based clustering to identify similar genes, but used a Fourier algorithm to assess periodicity. An improvement to this approach was subsequently suggested [3] that involves performing a Fast Fourier Transform on the data, reconstructing the original pattern from a small number of high-order Fourier coefficients and then calculating a sum of squares error for the reconstructed pattern. Then the data is randomly permuted and a vector of sum of squares error values produced for each original pattern. Finally, a p-value is calculated for each that allows vectors to be selected as cell cycle regulated below a specified false positive rate.

Statistical Methods. Statistical and probabilistic methods have also been proposed, such as an ANOVA-based permutation test [24], cluster analysis followed by graphical Gaussian modeling (GGM) [29], and a dynamic Bayesian network approach first proposed in [22] and further investigated in [23]. The GGM approach allows analysis of whole genome expression data, but only allows inference of regulatory relationships among clusters of genes, rather than among individual genes. In addition, the method assumes that the data is drawn from a multivariate normal distribution. The dynamic Bayesian network approach does not lend itself well to whole genome datasets, as its learning algorithm is computationally intensive. The approach may, however, be valuable in determining causality in regulatory relationships when run on large datasets such as whole genome (the data for [23] included a mere 169 genes). Clearly, then, improvements need to be made on the performance of the learning algorithm employed to permit exploration of these larger datasets.

Perhaps the most popular statistical approach for analyzing gene expression data is singular value decomposition (SVD) [1][10], also known as principal component analysis (PCA) or Karhunen-Loève expansion. SVD is a linear transformation of the temporal expression data from original the N-genes x M-time points space into a reduced M-eigengenes x M-eigenarrays space, where here it is assumed that $M << N$ (as is the case for existing whole genome datasets). The resulting eigengenes and eigenarrays are orthogonal and therefore may represent independent, biologically meaningful regulatory processes. In [31], the authors recognize that gene networks are sparse; because of this, they use SVD to construct a family of candidate networks, and then use robust regression to choose the sparsest network in the family as the most meaningful biological gene network. Correspondence analysis, a method related to SVD in that it reduces the dimensionality of the data, allows for the comparison within and between two variables simultaneously (e.g. genes with genes, genes with experiments, and experiments with experiments) [13].

Machine-Learning Methods. The final category of methods for inferring gene regulatory networks is machine-learning techniques, which can be further divided into unsupervised and supervised learning. Some of the earliest methods of finding related genes in expression data are clustering methods, and are still popular today. Clustering methods are identified by the clustering algorithm used and by the distance metric used by the clustering algorithm. Examples of algorithms include hierarchical clustering [12], k-means clustering [28], and graph theoretic approaches to clustering [5]. Examples of distance metrics include Euclidean distance, correlation coefficient, Pearson correlation, and many others. (The correlation-based methods that are the focus of this paper would include any clustering algorithms that use the correlation coefficient or some other correlation-based metric as its distance metric.) A related, but more sophisticated method is self-organizing maps (SOM) [27]. Finally, relevance networks [8][7]

are an unsupervised learning technique that uses entropy and mutual information measures to evaluate gene-gene associations and to cluster genes.

Most clustering techniques are easy to understand, but are often inadequate for inferring gene regulatory networks. Hierarchical clustering assumes that the resulting networks are hierarchical in nature; gene regulatory networks are typically much more complex, with loops, hubs, and multiple input-output relationships (for instance, browse the regulatory pathways stored at KEGG – www.genome.ad.jp/kegg/regulation.html). K-means clustering requires that the number of clusters (k) be specified prior to clustering; unfortunately, the final clusters are heavily dependent on the choice of k. The SOM approach is further complicated in that an initial geometry must be specified in addition to a value for k.

The most widely used supervised learning technique for gene expression data is support vector machines (SVM) [6]. SVMs are supervised in that they use a set of labeled data to "train" the SVM to discriminate between data that belong to different clusters by finding a set of hyperplanes that linearly separate the clusters. The difficulty is often that a set of hyperplanes cannot be found that separate the data in the input space, so the SVM maps the training set to a higher-dimensional feature space in which a set of hyperplanes may be found that do separate the data. This mapping would normally mean increased complexity and computation time (counterintuitive to the objectives of SVD, where a lower-dimensional feature space is sought), but the SVM compensates for this with kernel functions that allow the algorithm to compute the hyperplanes without explicitly representing the higher-dimensional feature space. The biggest danger in this technique is the possibility of overfitting the data and producing trivial solutions.

Challenges

A number of issues must be taken into account when working with gene expression data. Noisy signals can often obscure gene-gene relationships, and should be addressed. Most often with other techniques this is accomplished through the use of replicates; however, microarrays are still relatively expensive, and replicates are not yet widely used. Some methods model a noise component directly [21]. Noise is discussed from a statistical viewpoint in [2], which proposes a model including a noise component for the complex biological processes underlying gene expression data [14]. The Spellman dataset [26] does not include replicates, so the noise contribution to a single gene is not addressed by the approach described in this paper. However, the "noise" within the entire dataset due to time-invariant genes can be filtered in data pre-processing (see the 'FUTURE WORK' section below).

One issue of particular importance is how to deal with missing values. Often a particular spot on a microarray slide has been contaminated or for some other reason is excluded from analysis. Obviously, the preferred solution to this problem is to repeat the experiment for that spot. However, this is not done in practice due to expense. The most common methods for dealing with missing values are replacing them with zeros or with a 'row average' value (i.e., the average expression level of all timepoints in the gene). Statistical methods may also be used, such as least squares estimates and ANOVA. However, these methods are model based, an assumption which can lead to less than satisfactory results. SVD-based and k-nearest neighbor-based missing value imputation methods have also been proposed [30]. Both of these methods remove the model assumption of the statistical methods and perform well. The issue of missing values is addressed in this work in data pre-processing.

Additional issues arise when attempting to elucidate gene networks. For instance, the assumption that biologically related genes and gene products produce similar expression patterns is not always true [31]. As already discussed, many clustering algorithms only formulate hierarchical relationships, which is too simplistic for gene networks [31]. Finally, most techniques are not capable of dealing with time delays in relationships and therefore cannot distinguish causality among genes [2], although some methods have been written that address this issue [23][19].

PROBLEM STATEMENT

One of the simplest approaches for finding relationships between genes is correlation analysis. Correlation-based methods have been popular for analyzing gene expression data because they are intuitive, simple to implement, and produce easy to understand results (typically a hierarchical set of clusters). However, correlation-based methods for clustering of genes and inference of gene networks suffer from a number of problems:

1. A high correlation coefficient between two genes does not necessarily indicate similarly shaped curves [25]. Figure 1(a) shows two curves whose overall shapes are arguably more similar than those in Figure 1(b). However, their corresponding correlation coefficients are 0.58 and 0.87, respectively.

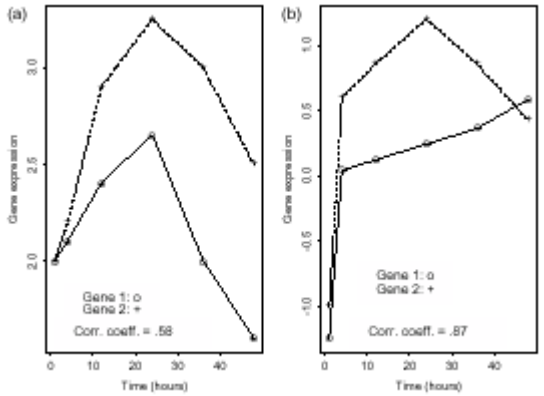


Figure 1. High correlation does not mean similar shapes. (graphic from [25]).

2. It is common biologically that the response of a regulatory target to a promoter or inhibitor lags the expression of that regulator. Typical correlation-based methods cannot identify similarity between expression curves involving time delays, nor can they infer causality.
3. Correlation methods are only good at detecting global similarities between expression curves [19]. Other methods are only good at detecting local similarities [14]. What is desired is an approach that accomplishes both.
4. As described above, most correlation methods cluster data hierarchically. That is, a gene can belong to only one cluster, implying a single biological function. In reality, regulatory relationships are much more complicated.

The window optimization technique described in this paper is intended to identify relationships between genes involving a time delay, and considers both local and global similarity. Directional similarity measure (DSM) is an alternative measure to the correlation coefficient that finds gene relationships based on the shape of expression patterns, overcoming the first issue. Graphical representation of high correlation or high DSM genes exhibits complex behavior in which a gene can participate in many different relationships with other genes. Therefore, a combination of window optimization and DSM can be used to overcome all of the problems described above, and also may indicate causal relationships.

Two very recently proposed methods also attempt to address these issues: *order restricted inference* [25] and the *event method* [19].

Order Restricted Inference

Order restricted inference is a method where a gene expression pattern is first reduced to the relationships between neighboring timepoints. A number of *profiles* are then defined in terms of *linkages* and *nodes*,

and then an algorithm is proposed to evaluate each expression pattern and determine which profile it most closely resembles. Two timepoints within a profile are defined as *linked* if the inequality between them is specified a priori. A timepoint is then said to be a *node* if it is linked with every other parameter in the profile. For instance, if the patterns in Figure 1(a) were considered as profiles, all of the timepoints would be linked since the curve is monotonically increasing up to timepoint four and monotonically decreasing thereafter. That is, tp1 < tp2 < tp3 < tp4 > tp5 > tp6 > … and tp4 is a node, where tp*n* is timepoint n.

The algorithm for order restricted inference then is as follows:

1. Pre-specify a collection of candidate profiles (e.g., monotonically increasing, monotonically decreasing, up-down, down-up, cyclical, etc.).
2. Obtain estimates of each timepoint in an expression pattern under each of the candidate profiles (the technique is described in [25]).
3. Compute a distance measure from each profile to the corresponding expression pattern estimate. The authors here chose the l_∞ norm, the maximum difference between the estimates of two linked parameters.
4. BOOTSTRAPPING STEP. Combine the M actual observations from all the timepoints into a vector of length MT (where T is the number of timepoints) and draw T simple random samples with replacement, each of size M. Repeat steps 2 and 3 for each bootstrap sample to build a distribution of l_∞ norms.
5. Assign the pattern to a profile if its l_∞ norm is above the α^{th} percentile, where α is chosen ahead of time. If the pattern is below the α^{th} percentile, or if two profiles tie, then do not assign the pattern to any profile.
6. Repeat steps 2-5 for each gene.

Order restricted inference naturally clusters patterns into classes with other similarly shaped patterns by making explicit use of ordering information among timepoints, satisfy certain optimality properties as discussed in [25] and its references, and also allows the user to specify an acceptable Type I error rate based on the bootstrapping distribution. Specifically, this method overcomes the first problem above by being able to group patterns specifically by shape. It does not address the other two problems, however, and further suffers in that the target profiles must be selected and defined a priori. In addition, the algorithm would need to be adapted to accommodate more subtle patterns; for instance, it cannot currently distinguish between up-down profiles that rise quickly and peak and those that rise more slowly.

Event Method

Correlation-based methods are capable of finding global similarities between gene expression patterns. Other methods, such as the edge-detection method described in [14], are adept at identifying strong local similarities. The *event method* strives to do both.

The event method algorithm is as follows:

1. Convert the segments between timepoints in a gene expression pattern into events. Each event is either a rising (R), constant (C), or falling (F) event such that the pattern of *n* timepoints is converted to a string of *n*-1 events:
    - The data is smoothed using a sliding window technique
    - The slope is determined between each pair of timepoints
    - The slopes are converted to events. This is intuitive with the exception of the C event. A slope is considered to be constant based on a threshold parameter that specifies the percentage of data points for the gene that are to be classified as constant. Any segments above the upper threshold boundary are classified as R, and any below the lower threshold boundary are classified as F.

2. The event strings are aligned using a modified Needleman-Wunsch algorithm. This alignment takes into account time delays between aligned events, but assumes only positive time delays (i.e., gene A activates gene B is assumed; the assumption that gene B activates gene A would be a separate execution of the algorithm).
3. Finally, a scoring function is used to score the alignment. This scoring function imposes a linear time delay penalty for gaps in the event alignment.

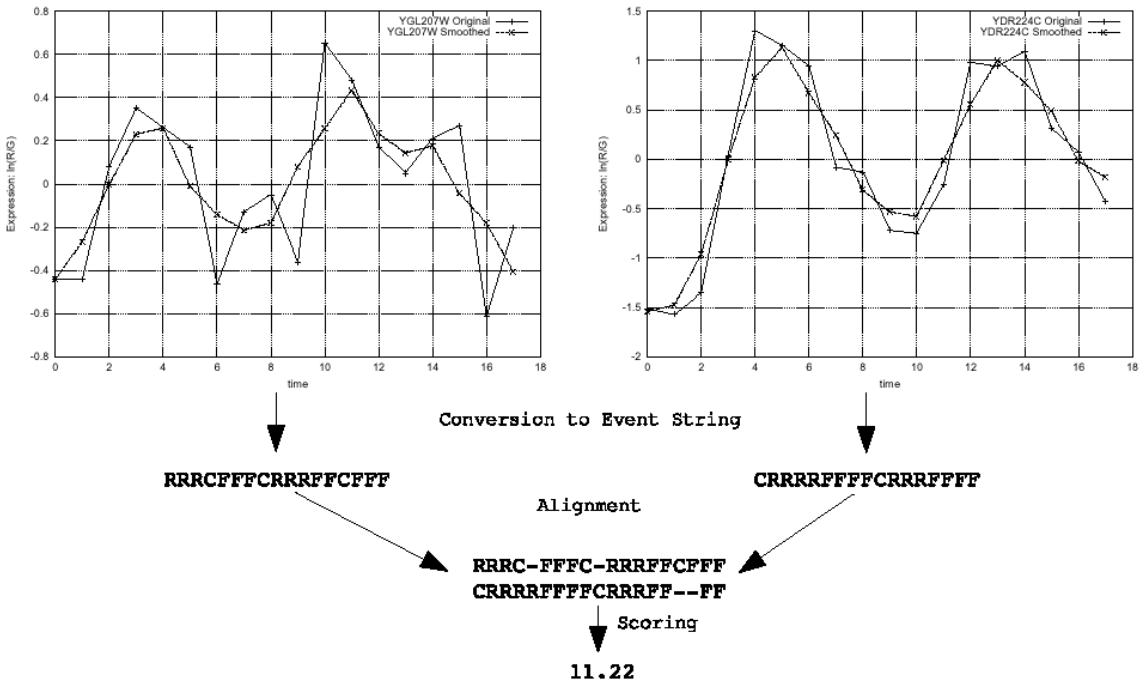


Figure 2. The event method as applied to the genes YGL207W and YDR224C. (graphic from [19]).

An example application of the event method is shown in Figure 2.

The event method is computationally efficient, and seemingly addresses all of the problems listed above. It is able to cluster patterns based on the similarity of their shapes, is able to account for delays, and can detect both global and local similarities. The event method is a data-driven machine-learning algorithm, not requiring that profiles be defined a priori, as in the order restricted inference technique. It was found to perform as well as, but not necessarily better than correlation-based techniques, when using the Spellman data [26] for testing.

PROPOSED SOLUTION

The event method seems to address all of the problems of correlation-based methods for inferring gene regulatory networks from temporal expression data. It does not, however, claim to outperform correlation-based methods. The modifications to correlation-based methods identified below are an attempt not only to address the issues of shape conformance, dealing with time delays, and finding local and global similarities, but also hopes to improve upon the performance (i.e., ability to detect true positives) of correlation-based techniques.

Window Optimization

The first proposed improvement is to use a window optimization approach to maximize the correlation coefficient for a given gene (thus finding not only global, but also local similarities) and to account for delays. ***Window optimization guarantees an optimal solution, in that it finds the longest subsection of***

an expression pattern that produces the highest correlation. Window optimization may also be applied to measures other than correlation, such as DSM.

The optimization algorithm calculates the correlation between two patterns A and B as follows:

1. Choose a minimum window size, w_{min}. This must be at least three timepoints in order to minimally discriminate between curve shapes, since two timepoints can only describe a single line segment. The maximum window size is the number of timepoints in the entire pattern (T).
2. Choose a maximum delay, d_{max}. A requirement is that $w_{min} + d_{max} \leq T$. Note that here, as with the event method, negative delays are avoided by considering the reverse relationship separately (e.g., gene B regulates gene A).
3. Now calculate the correlation c_{wd} for each combination of starting position $0 \leq p \leq T\text{-}w$ on pattern A, window size $w_{min} \leq w \leq T$ and delay $0 \leq d \leq \min(d_{max}, T\text{-}w)$. Here, the min() function ensures that $w + d \leq T$ in all cases.
4. Choose the max(abs(c_{wd})) as the correlation c between the two patterns. Store the values of p, w, and d for the winning c_{wd}; if there is a tie between two combinations of p, w, and d, select the combination whose score is also a maximum (see below – for now, simply note that the score also considers window size and number of missing values).

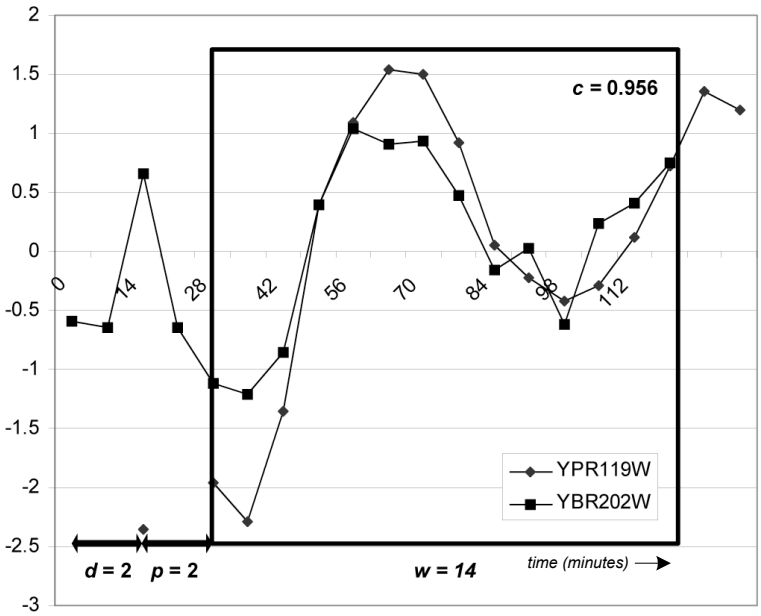


Figure 3. Windowed optimized correlation of the gene expression patterns of genes YPR119W and YBR202W (data from alpha factor-based cell cycle synchronization [26]).

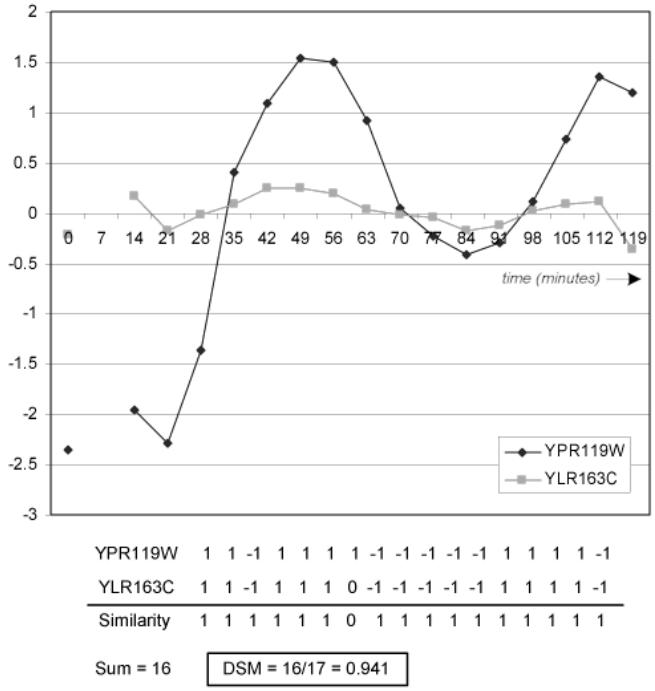
The window optimization algorithm finds the most highly correlated subset of timepoints between two patterns. Values for p, w, and d are not specified in advance, and can be different for each pattern pair evaluated; as such, it is able to identify both local and global similarities in a completely flexible manner. In addition, it is able to detect relationships involving delays. The correlation is chosen based on its magnitude only (i.e., no preference is given to correlation vs. anti-correlation), but the sign of the selected correlation can be combined with delay information to infer causality.

Figure 3 shows windowed correlation as applied to *S. cerevisiae* genes YPR119W and YBR202W. For this pair, the optimal window was found to be 14 timepoints, with a delay of two timepoints and a starting position at timepoint two (numbering of timepoints starts at zero). Notice that timepoint one for YPR119W is a missing value, and is skipped by the algorithm. This will be discussed further in the sections below on scoring and data pre-processing.

It is difficult to assess the computational complexity of the window optimization algorithm, since it relies on user-specified runtime parameter settings. Full window ($w=T$, $p=d=0$) operation is equivalent to non-windowed correlation, and is of complexity $O(T)$, where T is the number of timepoints in each gene expression pattern. For $w_{min}<w<T$, $0<p<T-w$, $0<d<T-w-p$ (here the worst case is considered for all parameters, so $d_{max} = T-w-p$, considering all available timepoints), the number of cases considered is $w(T-w)(T-w-p)$ summed over all windows $w_{min}<w<T$, or a worst case complexity $O(T^3)$. When run in target mode (i.e., scan a list of target genes against the entire genome), worst-case execution time is $O(nT^3)$, where n is the number of genes (ORFs) in the genome. Whole genome scanning (i.e., consider all possible relationships between all genes) would then take $O(n^2T^3)$ execution time.

Directional Similarity Measure (DSM)

The second proposed improvement is called the *directional similarity measure (DSM)*, and is a technique similar to the event method in that it is interested in finding patterns with similar shapes. DSM is also a data-driven machine-learning algorithm in that it does not require candidate profiles to be defined *a priori*, and like the event method it also converts n timepoints into n-1 events or *segments*.



YPR119W	1	1	-1	1	1	1	1	-1	-1	-1	-1	-1	1	1	1	1	-1
YLR163C	1	1	-1	1	1	1	0	-1	-1	-1	-1	-1	1	1	1	1	-1
Similarity	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1

Sum = 16 DSM = 16/17 = 0.941

Figure 4. Calculation of DSM between genes YPR119W and YLR163C shows that high DSM <u>*does*</u> mean similar shapes (data from alpha factor-based cell cycle synchronization [26]).

The DSM is calculated between two patterns A and B as follows:

1. Convert the n timepoints of pattern A into n-1 segments.
   - If a segment's slope is positive, assign a +1 for the segment.
   - If the slope is negative, assign a –1 for the segment.
   - Otherwise, the slope is constant; assign a 0 for the segment.
2. Handle missing values as follows:
   - If the missing values are at the beginning of the pattern, consider all segments up to the first timepoint that contains a value as constant.
   - If the missing values are at the end of the pattern, consider all segments past the last timepoint that contains a value as constant.

- For any stretch of one or more missing values between two timepoints t_a and t_b that contain values, consider the entire span of segments between t_a and t_b as a single segment and evaluate according to step 1. Assign the resultant value to all individual segments between t_a and t_b.

3. Repeat steps 1 and 2 for pattern B.
4. Multiply the values produced for corresponding timepoints on patterns A and B together, and then sum up the resulting values. Divide by the number of values summed to get a value in the range [-1, 1].

Figure 4 shows this process graphically. In this case, only the 7th segment disagrees between the two patterns, where YPR119W is increasing and YLR163C is constant. Note that the algorithm very simplistically handles constant segments – the enclosing timepoints must be exactly equal (to whatever precision the timepoints themselves were specified), and corresponding segments on patterns A and B that are both constant are rewarded a similarity value of 0. A future enhancement to this scheme would be to include a threshold for identifying constant segments similar to that used in the event method, and to assign to matching constant segments between two patterns a +1 or –1, depending on whether the DSM is positive or negative without the addition of this similarity (i.e., to ensure that matching constant segments are always rewarded).

The computational complexity of the DSM algorithm is O(T), where T is the number of timepoints in each gene expression pattern. When run in target mode, execution time is O(nT), where n is the number of genes (ORFs) in the genome. Whole genome scanning would then take O(n^2T) execution time.

Scoring

The resultant correlation from the window optimization technique and DSM values from directional similarity measurements are also scored as follows:

```
Score = abs(corr)*abs(DSM)*tpFraction*confidence*100%;
```

The absolute values for correlation and DSM are used to ensure that both highly correlated and anti-correlated (or similarly shaped and mirror-image shaped) patterns are identified as equally important. In instances where only correlation or DSM is used, the unused measure is set equal to 1.

The tpFraction component measures the fraction of the total pattern that is encompassed in the window selected by window optimization:

```
tpFraction = w/T
```

The confidence value assesses the quality of the data within the window:

```
confidence = (# missing values in w)/w
```

The tpFraction component therefore rewards longer matches, while the confidence component attenuates matches containing missing values.

RESULTS AND DISCUSSION

To assess these methods, the classic yeast (*S. cerevisiae*) cell cycle data of Spellman *et. al*. was used [26]. Four sets of experiments were performed to synchronize the yeast cell cycle: alpha factor-based, cdc15-based, cdc28-based, and elutriation-based synchronization. Each of these experiments produced an array of results ranging from 14 to 24 timepoints across the entire genome (6178 genes or ORFs). The website for this data (cellcycle-www.stanford.edu) provides both the raw data and processed data. The processed data (normalized log ratio values) was chosen for this study to avoid extensive data processing re-work.

Futcher presents a simplified model of yeast cell cycle regulation (see Figure 5), which in its simplest form is a negative feedback oscillator between the transcription factors SBF/MBF and Clb2 [15]. SBF is a transcription factor composed of the DNA-binding protein Swi4 and regulatory component Swi6, while MBF is composed of the DNA-binding protein Mbp1 and Swi6. Clb2 is a mitotic cyclin that binds and activates the Cdc28 kinase. The boxes in Figure 5 represent the clusters identified in [26], which are regulated as shown by MBF, SBF or Clb2.

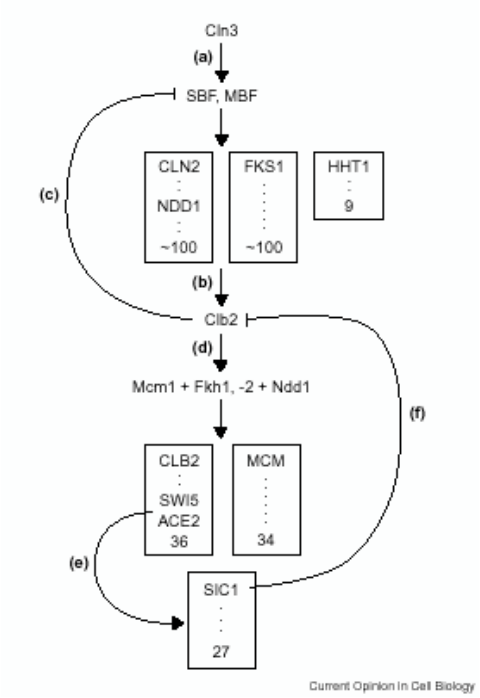


Figure 5. Yeast cell cycle transcriptional network (graphic from [15]).

Figure 5 indicates that detection of the genes in these clusters may be used to test window optimization and DSM. For instance, the gene for Clb2 (YPR119W) can be used to search the yeast genome for high scoring hits. These hits can then be compared to the members of the CLB2, MCM, and SIC1 clusters to assess the ability of the algorithms to recover the clusters from [26]. The same could be done using the Mbp1, Swi4, and Swi6 genes to recover the CLN2, FKS1, and HHT1 clusters. Unfortunately, this author was unable to confirm the member genes for clusters CLN2 and FKS1 from the website listed above, so the preliminary results presented are based solely on assessment of Clb2.

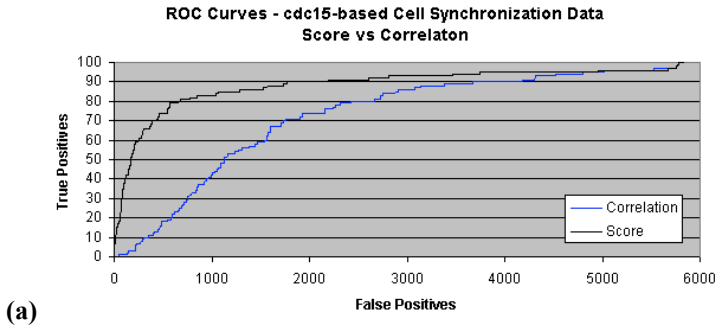
Missing Values

The approach taken to missing values was to leave them as missing values and build the algorithms to handle missing values. Window optimization simply skips any pairing of timepoints in which one or the other of the values is missing. This obviously degrades the resultant correlation, which is reflected in a lower calculated confidence value and therefore a lower score. DSM's method for handling missing values has already been described; again, the confidence value will be lower and this will be reflected in the resultant score.

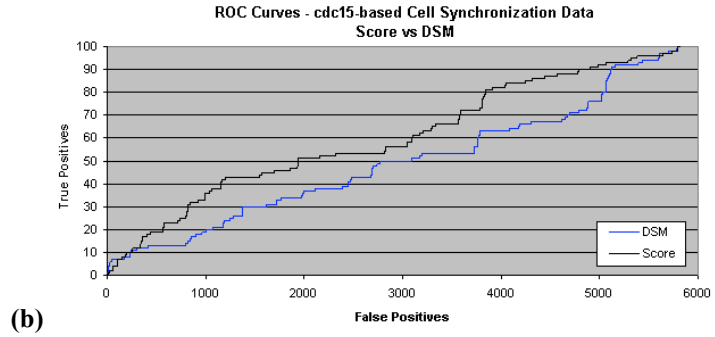
Scoring Strategy

As stated before, the Spellman yeast study includes four separate datasets, labeled alpha, cdc15, cdc28, and elu based on the method used to synchronize the cell cycle. Because of this, four sets of scores are generated for each test run. A rather important task is then to determine how to combine these scores to produce a single value that may be used to evaluate each of the methods.

First, windowed optimized correlation and window optimized DSM was performed on all four datasets. For each set, a ROC curve was generated from the 6178 genes to determine whether the score described above was more effective in recovering the 102 members of the CLB2, MCM and SIC1 clusters. The results, seen in Figure 6, show that for cdc15 the score outperforms the unaltered windowed correlation or windowed DSM in recovering the cluster members. In fact, this is true for all of the datasets. This indicates that the score's inclusion of a factor for data quality and window size does indeed contribute to the scheme's ability to discriminate between related and unrelated genes. Because of this, unaltered windowed correlation and windowed DSM were removed from consideration for evaluating performance.



(a)



(b)

Figure 6. Score outperforms raw correlation/DSM (window optimization employed in all cases).

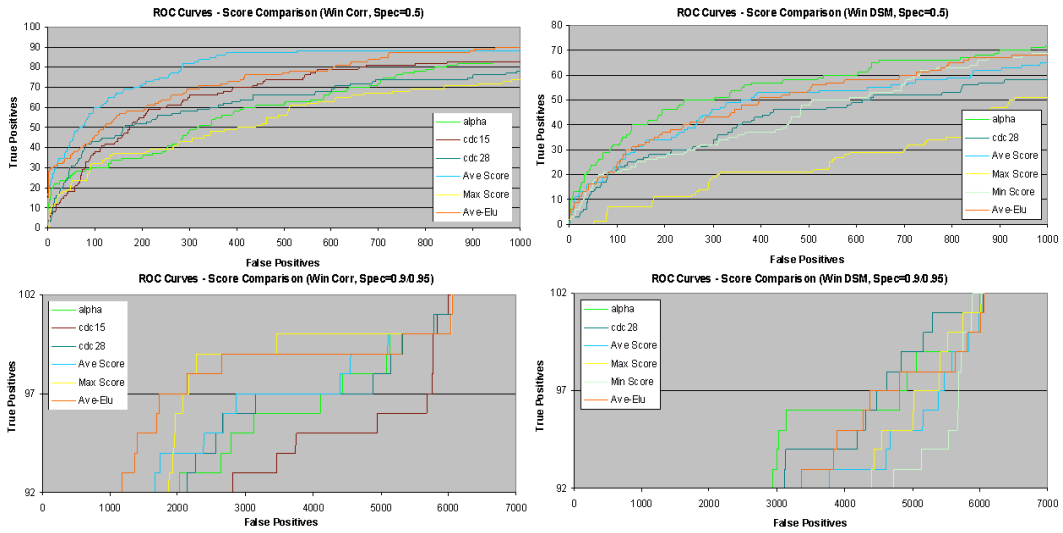


Figure 7. Ave–Elu (Average Score of alpha, cdc15, and cdc28 only, in orange) is best overall performer. Correlation is on the left, DSM on the right. Spec = 0.5 is on top, Spec = 0.9-0.95 is on the bottom.[1]

[1] Spec = TP/(TP+FN). Specificity is defined here as the number of cluster members identified as a fraction of the total number of cluster members. This can be expressed as the number of false positives encountered to reach a certain specificity. For instance, Ave Score reaches Spec=0.5 at approximately FP=100 in the top left chart of Figure 7.

A number of schemes for combining the scores from each of the dataset were next compared to each other and to the individual dataset scores (Figure 7). The schemes investigated were 1) average score, 2) maximum score, and 3) minimum score. Schemes 1 and 2 are intuitive; scheme 3 is based on the "weakest link" theory – the most representative score is that of the lowest scoring dataset. It was also noted that the elu dataset consistently scored poorly; as such, a fourth scheme was added that was the average score of all of the datasets except elu (called Ave-Elu in Figure 7).

	Rank@Spec=0.5	Rank@Spec=0.9	Rank@Spec=0.95
Windowed Correlation	2	1	1
Windowed DSM	2(3)	2(3)	1
Windowed Corr + DSM	1	1	1(3)

Table 1. Relative performance of Ave–Elu scoring scheme. A specificity of 0.5 is the false positive rate at which 51 of the 102 cluster patterns have been identified. Rankings are within the four combined scoring schemes (ranks in parentheses are overall, including against individual dataset scores).

These combined scoring schemes were compared to the individual dataset scores under three conditions: 1) windowed correlation, 2) windowed DSM, and 3) windowed correlation with DSM (not shown in Figure 7). The best scoring scheme of the four for windowed correlation was the average score, the best for windowed DSM and windowed correlation with DSM was the average score less the elu dataset. Interestingly, the best performing score for windowed DSM was the alpha dataset score alone, represented in bright green on the right-hand side of Figure 7. Both the alpha and cdc28 dataset individual scores performed well for windowed correlation with DSM. Because a single best scoring scheme was desired in order to compare all of the methods, however, average score less the elu dataset (Ave–Elu) was selected. Table 1 summarizes Ave–Elu's performance in all experiments.

Preliminary Results

Figure 8 presents the results of applying window optimization and DSM to the Spellman data. All of the experiments were evaluated based on their ability to recover the Spellman clusters, either together or individually. The Full Corr, Full DSM and Full CorrDSM experiments fixed the window size at $w=T$, and so $p=d=0$. These experiments use the entire dataset without considering delays, and as such are intended to duplicate standard, non-windowed techniques.

Excluding the CLB2 cluster, which will be discussed separately, these results lead to the following observations:

1. ***Window optimization consistently outperforms non-optimized techniques.*** All of the non-optimized experiments produced shallower ROC curves than all of the window optimized experiments.
2. ***DSM outperforms correlation (non-optimized).*** This is true for the combined cluster set above approximately Spec=0.4; the two measures compete closely for the MCM cluster, with DSM slightly outperforming correlation; DSM outperforms correlation for SIC1 over the full range of specificities.
3. ***Windowed correlation outperforms windowed DSM.***
4. ***DSM can be used to enhance correlation measures (non-optimized).*** This can be seen in that Full CorrDSM outperforms FullCorr.

The CLB2 cluster results are not in line with these observations. Non-optimized correlation outperforms all other measures, followed by window optimized correlation and then non-optimized correlation with DSM. These inconsistent results for the CLB2 cluster may be explained in that these clusters were originally built using a correlation-based method [26], and Clb2 itself belongs to this cluster. As such, the cluster members were assembled based on correlation with Clb2, which may heavily bias these results towards correlation without biological justification. Obviously a close examination of the high scoring genes found by DSM would need to be compared to Clb2 by biological means (sequence similarity,

structural similarity, upstream binding sites, etc.) to determine whether or not the windowed DSM results are biologically more meaningful than the correlation results.

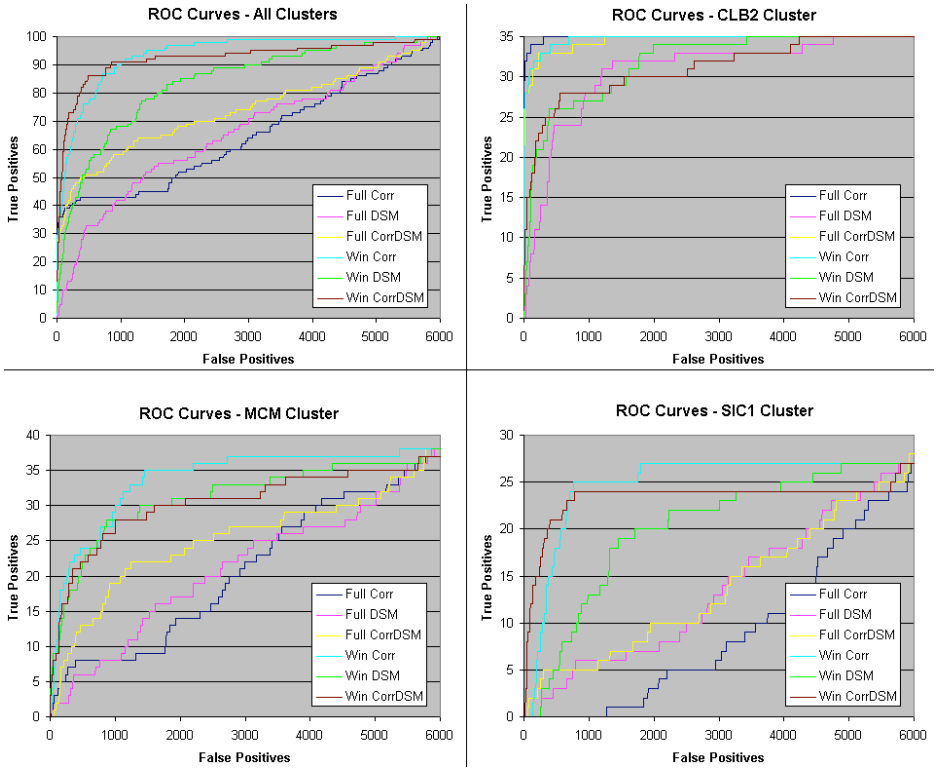


Figure 8. Performance of Window Optimization and DSM on full set of clusters (CLB2, MCM, SIC1) as well as individual clusters.

CONCLUSIONS

Window optimization is guaranteed to produce an optimal score (correlation, DSM, etc.) for a gene expression pattern, identifying both local and global similarity. The preliminary results presented here indicate that window optimization significantly enhances the ability to identify regulatory relationship from time-series gene expression data. The distance similarity measure (DSM) exhibited improved performance over standard full-pattern correlation, but the improvements were not nearly as pronounced as for window optimization. This conclusion must be qualified in that the clusters used to evaluate these methods were initially generated using a correlation-based technique. Biological analysis of early false positives and of cluster members who are identified late by DSM and windowed techniques must be conducted to fully determine whether the standard correlation method or the current methods produce more biologically significant results.

FUTURE WORK

Many opportunities exist to further exploration. In this paper, window optimization and DSM were used in a targeted way; a potential regulatory gene was first identified, and the genome explored in the context of that gene. Whole genome exploration is also possible, but the range of window sizes and delay values used would need to be limited to obtain reasonable execution times (it takes 1-2 minutes to search the yeast genome for regulatory relationships against one target gene – a full genome scan of the 6178 genes in yeast would then take 100-200 hours). Statistics obtained from the experiments above indicate that a much narrowed range of window values and delays can be used than were employed here.

It would be valuable to perform a direct comparison of DSM and the Event Method [19], as they both assess the shape of patterns. It would also be worthwhile to assess DSM and window optimized DSM

against known clusters of genes whose regulatory relationship was established by means other than correlation, preferably by biological rather than algorithmic assessment. For instance, [17] presents groupings of genes that have been found to have upstream binding sites for SBF, MBF, or both, indicating a potential regulatory relationship.

Improvements may also be made to the techniques proposed in this work. One pre-processing step that was explored and abandoned was filtering the datasets to eliminate patterns that exhibit a low level of variance relative to the rest of the dataset. This filtering was abandoned because it was then difficult to produce a combined score, as the subset of genes filtered from one dataset will not be the same as from another. In addition, some of the methods for handling missing values from [30] could be employed and assessed. Other metrics, such as jack-knife correlation [16] and k-means, could be used with windowing and with DSM. Finally, the concept of distance similarity measure could be expanded by assigning to each segment the percent of the pattern's range that was traversed over the segment (with a positive or negative sign depending on whether the segment is rising or falling) rather than simply a +1/–1.

REFERENCES

[1] Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl. Acad. Sci. 97(18):10101-6, 2000.

[2] Arkin A, Ross J. *Statistical Construction of Chemical Reaction Mechanisms From Measured Time-Series.* J. Phys. Chem. 99:970-979, 1995.

[3] Bair E. *An improved method for identifying cell cycle regulated genes in yeast.* Technical paper, BIOC218, Stanford University, December 6, 2001.

[4] Bhan A, Galas DJ, Dewey TG. *A duplication growth model of gene expression networks*. Bioinformatics 18(11):1486-93, 2002.

[5] Ben-Dor A, Shamir R, Yakhini Z. *Clustering gene expression patterns*. J. Comp. Biol. 6(3-4):281-97, 1999.

[6] Brown et al. *Knowledge-based analysis of microarray gene expression data by using support vector machines*. Proc. Natl. Acad. Sci. 97(1):262-7, 2000.

[7] Butte et al. *Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks.* Proc Natl. Acad. Sci. 97(22):12182-6, 2000.

[8] Butte AJ, Kohane IS. *Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements.* Pac. Symp. Biocomput. 5:418-429, 2000.

[9] Chen T, He HL, Church GM. *Modeling gene expression with differential equations.* Pac. Symp. Biocomput. 4:29-40, 1999.

[10] Dewey TG, Galas DJ. *Dynamic models of gene expression and classification.* Funct. Integr. Genomics 1:269-278, 2001.

[11] de Hoon et. al. *Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtilis using differential equations.* Pac. Symp. Biocomput. 8:17-28, 2003.

[12] Eisen et al. *Cluster analysis and display of genome-wide expression patterns.* Proc. Natl. Acad. Sci. 95:14863-8, 1998.

[13] Fellenberg et al. *Correspondence analysis applied to microarray data.* Proc. Natl. Acad. Sci. 98(19):10781-6, 2001.

[14] Filkov V, Skiena S, Zhi J. *Analysis techniques for microarray time-series data*. J. Comp. Biol. 9(2):317-30, 2002.

[15] Futcher B. *Transcriptional regulatory networks and the yeast cell cycle.* Curr. Opin. Cell Biol. (6):676-83, 2002.

[16] Heyer, LJ, Kruglyak S, Yooseph S. *Exploring expression data: identification and analysis of coexpressed genes*. Gen. Res. 9:11-6-1115, 1999.

[17] Iyer et. al. *Genomic binding sites of yeast cell-cycle transcription factors SBF and MBF.* Nature 409:533-538, 2001.

[18] Kato M, Tsunoda T, Takagi T. *Inferring genetic networks from DNA microarray data by multiple regression analysis*. Genome Inform. Ser. Workshop 11:118-28, 2000.

[19] Kwon AT, Hoos HH, Ng R. *Inference of transcriptional regulation relationships from gene expression data.* Bioinformatics 19: 905-912, 2003.

[20] Loomis WF. *Genetic networks that regulate development in Dictyostelium cells.* Microbiol. Rev., 60(1):135-50, 1996.

[21] Luan Y, Li H. *Clustering of time-course gene expression data using a mixed-effects model with B-splines.* Bioinformatics 19(4):474-82, 2003.

[22] Murphy K, Mian S. *Modelling gene expression data using dynamic Bayesian networks.* Technical Report, Computer Science Division, University of California, Berkeley, CA.

[23] Ong IM, Glasner JD, Page D. *Modelling regulatory pathways in E. coli from time series expression profiles.* Bioinformatics 18 Suppl 1:S241-8, 2002.

[24] Park et al. *Statistical tests for identifying differentially expressed genes in time-course microarray experiments.* Bioinformatics, 12;19(6):694-703, 2003.

[25] Peddada et al. *Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference*. Bioinformatics 19(7):834-841, 2003.

[26] Spellman et al. *Comprehensive identification of cell cycle regulated genes of the yeast Sacccharomyces cerevisiae by microarray hybridization*. Mol. Biol. of the Cell 9:3273-3297, 1998.

[27] Tamayo et al. *Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation*. Proc. Natl. Acad. Sci. 96(6):2907-12, 1999.

[28] Tavazoie et al. *Systematic determination of genetic network architecture.* Nat. Genet. Jul;22(3):281-5, 1999.

[29] Toh H, Horimoto K. *Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling.* Bioinformatics 18(2):287-97, 2002.

[30] Troyanskaya et al. *Missing value estimation methods for DNA microarrays*. Bioinformatics 17(6):520-5, 2001.

[31] Yeung MK, Tegner J, Collins JJ. *Reverse engineering gene networks using singular value decomposition and robust regression.* Proc. Natl. Acad. Sci. 99(9):6163-8, 2002.