A Semi-Supervised Learning Method for Coherent Pattern Detection from Gene-Sample-Time Series Datasets

Chun Tang[1*], Murali Ramanathan[2], Daxin Jiang[1] and Aidong Zhang[1]

[1] Department of Computer Science and Engineering,

[2] Department of Pharmaceutical Sciences,

State University of New York at Buffalo, Buffalo, NY 14260, USA

Running Title: Semi-Supervised Coherent Pattern Detection

[*]To whom correspondence should be addressed.

ABSTRACT

Motivation. DNA microarrays provide simultaneous, semi-quantitative measurements of the expression levels of thousands of genes from a single experimental sample. The availability of such data sets can enhance our understanding of gene function and regulation if the patterns underlying gene expression data can be identified. In this paper, we study the problem of coherent pattern detection from gene-sample-time series expression data sets. These data sets result from microarray experiments in which a gene expression time series obtained on multiple samples using microarrays; the gene identities represent the 1^{st} dimension, the sample properties represent the 2^{nd} dimension and time represents the 3^{rd} dimension. Such Gene-Sample-Time Series (GST) data arise naturally in microarray experimental designs, e.g., when the pharmacodynamics of gene expression in responder and non-responder groups is investigated.

Results. A new semi-supervise learning method is proposed to search coherent blocks from gene-sample-time series data set. Each block contains a subset of genes and a subset of samples such that the patterns within the block are coherent along the time series. The coherent blocks may identify the samples corresponding to some phenotypes (e.g., disease states), and suggest the candidate genes correlated to the phenotypes. We empirically evaluate the performance of our approaches on a real microarray data of the pharmacodynamics of gene expression in multiple sclerosis patients after interferon-β treatment.

Availability. Software code written in MATLAB is available on request from the first author.

Contact. chuntang@cse.buffalo.edu.

INTRODUCTION

DNA microarrays, which can measure expression levels of thousands of genes across different experimental conditions (DeRisi *et al.*, 1996), are now being widely used in for biological and biomedical research for mRNA expression profiling. The identification and analysis of the gene expression patterns is a significant but difficult research problem (Iyer *et al.*, 1999). Patterns of gene expression indicate gene level responses and can provide insight into the mechanisms of diseases and therapies.

Microarray data sets and experimental designs can be characterized as either gene-sample data sets or as gene-time series data sets for purposes of computational pattern recognition. Operationally, gene-sample data sets (Golub *et al.*, 1999) contain gene expression measurements across collections of samples whereas gene-time series data sets (Eisen *et al.*, 1998; Iyer *et al.*, 1999) contain gene expression measurements in a sample undergoing a biological process of interest at a series of time points. Both gene-time data and gene-sample data can be represented by 2-dimensional *gene expression matrices*, in which the rows are the genes and the columns are samples in gene-sample data or in the case of gene-time data, ordered time points (Figure 1). Each cell in a gene expression matrix contains real valued data that represents the expression level of a specific gene in a given sample or at a certain time point.

A variety of machine learning, biostatistical and cluster analysis algorithms have been investigated for searching homogeneous sample groups from gene-sample matrices (Eisen *et al.*, 1998; Ben-Dor *et al.*, 1999; Hastie *et al.*, 2001) and for identifying co-expressed genes in gene-time series matrices (Azuaje, 2000; Golub *et al.*, 1999; Slonim *et al.*, 2000). Recently, in the course of defining treatment responders and non-responders from the pharmacodynamics of gene expression in multiple sclerosis patients after drug treatment, we were faced with gene-sample-time series (GST) data sets (Weinstock-Guttman *et al.*, 2003). In general, a GST data set is a 3-dimensional matrix, where in each cell represents the expression level of a certain gene of a certain sample at a certain time point (Figure 2(A)). By convention, the gene identities represent the 1^{st} dimension, the sample properties represent the 2^{nd} dimension and time represents the 3^{rd} dimension. Equivalently, as shown in Figure 2(B), a GST data set can also be regarded as a two dimensional matrix, such that each cell contains a time series pattern of a specific gene of a specific sample. For the GST matrices, the goal is identify the coherent blocks each contains a subset of genes whose expression levels rise and fall coherently under a subset of samples. The sub-matrices formed by such genes and samples become rational targets for additional research.

The existing techniques are incapable or inadequate for coherent block detection from GST data sets because they do not adequate account for the temporal structure within samples. The problem of coherent block detection is NP-hard and simplistic, unsupervised data mining can produce very large number of qualified blocks (Pei *et al.*, 2003; Wang *et al.*, 2002). The size and redundancy of such analysis results can make interpretation difficult and intimidating to users. However, because biomedical research experiments are usually well controlled and hypothesis-driven, it is possible to incorporate domain knowledge to direct pattern searching, and we use a semi-supervised approach for facilitating coherent block detection.

In this paper, we tackle the problem of coherent block detection from GST microarray data sets and make the following contributions:

- We propose a model for incorporating domain knowledge into coherent pattern detection in GST data. Our model is flexible and allows new rules and domain knowledge to be easily incorporated.
- We design a heuristic learning strategy that iteratively improves the quality of coherent blocks.
- We conduct an extensive empirical evaluation on microarray data sets and demonstrate that the proposed method identifies biologically interesting, coherent patterns from real data sets.

The remainder of the paper is organized as follows. In Section 2, we introduce the coherent block detection model and an approach for assessing of coherent patterns. Our method is evaluated for an experimental data set from drug treated multiple sclerosis patients in Section 3 followed by the Discussion in Section 4.

THEORY AND METHODS

PROBLEM DESCRIPTION

Given a gene-sample-time series data matrix, $M = \{w_{i,j,k} | 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq t\}$ where n is the number of genes, m is the number of samples, and t is the number of time points, the goal is to find a set of *coherent GST-blocks*. Each coherent GST-block is a sub-matrix of M denoted by $\{G', S', T'\}$, where the genes exhibit coherent patterns across a *time interval* (a series of consecutive time points) on the sample subset. Figure 2(B) shows two examples of coherent GST-blocks. The gene subset, G', and the sample subset S', can be any subsets of G and S, the original gene and sample sets, respectively. However, the time subset, T', must contain only a series of consecutive time points from the original time set T.

A *pattern* $\vec{P}_{ij}$ of length l is a vector of standardized expression levels for gene i in sample j at l consecutive time points, i.e., $\vec{P}_{ij} = \{w_{i,j,k}, w_{i,j,k+1}, \ldots, w_{i,j,k+l}\}$. Thus, the three-dimensional matrix M can be viewed as a two dimensional matrix with n rows and m columns, each cell is a *pattern*.

Two patterns are coherent if they have the same length and their similarity (according to the similarity measures discussed in the next section) is high, i.e., it exceeds a given threshold.

A sub-matrix of M denoted by $\{G', S', T'\}$, is a coherent GST-block iff:

(1) For any pairs genes g_a and g_b and any pairs of samples s_i and s_j, where $g_a, g_b \in G'$ $s_i, s_j \in S'$, pattern $\vec{P}_{ai}$ are coherent with pattern $\vec{P}_{bj}$ across the time interval T'.

(2) The number of elements, $||G'||$, $||S'||$ and $||T'||$ in the G', S' and T' dimensions meet or exceed their respective user-provided thresholds min_g, min_s and min_t for block size. Mathematically, the constraint: $||G'|| \geq min_g$, $||S'|| \geq min_s$ and $||T'|| \geq min_t$ ensures that a large number of small or trivial blocks are not generated.

Figure 3 summarizes the problem definition with inputs and outputs.

ALGORITHM

Our method first generates a series of patterns. Each pattern is regarded as a *seed* of a coherent GST-block. Then we search the whole data set to find the coherent blocks based on the seeds. The seed generation is a semi-supervised domain knowledge incorporation phase. The coherent block searching phase is an unsupervised learning approach.

Data Normalization and Standardization

Global normalization ensures that the mean and the standard deviation across all genes are zero and unity, respectively, for each sample, j, and each time point, k . It was applied to reduce the measurement bias. The normalized value, $u_{i,j,k}$, for each gene (with non-normalized expression value $w_{i,j,k}$) is:

$$u_{i,j,k} = \frac{w_{i,j,k} - \overline{w}_{j,k}}{\sigma_{j,k}} \ , \tag{1}$$

where

$$\overline{w}_{j,k} = \frac{1}{n}\sum_{i=1}^{n} w_{i,j,k} \ , \ \sigma_{j,k} = \frac{1}{n-1}\sqrt{\sum_{i=1}^{n}(w_{i,j,k} - \overline{w}_{j,k})^2} \ .$$

Standardization, which is analogous to normalization, but reduces scale differences between temporal profiles, was applied to ensure that geometrically similar patterns were placed in the same blocks. The standardized value, $v_{i,j,k}$, is obtained from the normalized expression values of gene i and sample j in a pattern of length l using:

$$v_{i,j,k} = \frac{u_{i,j,k} - \overline{u}_{i,j}}{\tau_{i,j}} \ , \tag{2}$$

where

$$\overline{u}_{i,j} = \frac{1}{l}\sum_{i=1}^{l} u_{i,j,k} \ , \ \tau_{i,j} = \frac{1}{l-1}\sqrt{\sum_{i=1}^{l}(u_{i,j,k} - \overline{u}_{i,j})^2} \ .$$

Coherence Measurement

Prior to defining coherence values, it is necessary to identify a suitable similarity measures for comparing two patterns. A variety of similarity measures are available from the clustering literature; we used the centered Pearson correlation coefficient because it is widely used and is scale and shift independent.

If $\vec{X} = \{x_1, x_2, ...x_l\}$ and $\vec{Y} = \{y_1, y_2, ...y_l\}$ are two pattern vectors of length l containing standardized gene expression values, then Pearson correlation coefficient $\rho(\vec{X}, \vec{Y})$ for the similarity between $\vec{X}$ and $\vec{Y}$ ranges from -1 (perfect anti-correlation) to $+1$ (perfect correlation) and is given by:

$$\rho(\vec{X}, \vec{Y}) = \frac{\sum_{i=1}^{l}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{l}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{l}(y_i - \overline{y})^2}} \ , \tag{3}$$

where

$$\overline{x} = \frac{1}{l}\sum_{i=1}^{l} x_i \ , \ \overline{y} = \frac{1}{l}\sum_{i=1}^{l} y_i \ .$$

The coherence value $C(G', S', T')$ of a block $\{G', S', T'\}$ measures the similarity between the patterns comprising the block relative to the centroid for the block. If a block $\{G', S', T'\}$ contains patterns $\vec{P}_{ij}$ corresponding to genes $i \in G'$ and samples $j \in S'$ contained in the coherence value $C(G', S', T')$ across the interval T' is given by:

$$C(G', S', T') = \frac{\sum_{i \in G'} \sum_{j \in S'} \rho(\vec{P}_{ij}, \vec{P}_{centroid})}{||G'|| \times ||S'||}. \tag{4}$$

The centroid for the block $\vec{P}_{centroid}$ is a pattern which has the same time interval T' as every other pattern in the block. The value at each time point of the centroid is the average value at the corresponding time point of all patterns in the block. Thus $\vec{P}_{centroid} = \{\overline{v}_k, \overline{v}_{k+1}, \ldots, \overline{v}_{k+l}\}$, where $\overline{v}_{k+x} = \frac{\sum_{i \in G'} \sum_{j \in S'} v_{i,j,k+x}}{||G'|| \times ||S'||}$ $(0 \leq x \leq l)$.

Seed Generation

Every coherent block is constructed from a seed such that the patterns within the block share similarity to the seed. The three methods for seed generation in the algorithm are described below.

Method 1: Domain Expert Selected Seeds In this approach, the domain expert identifies specific genes, and their mean expression pattern is used as seed. Typically, biomedical scientists tend to select specific genes that have been implicated in the cellular processes and physiological processes of interest by prior work. The algorithm utilizes the seed to find other genes with similar patterns and these may be novel or poorly characterized genes that could provide a framework for additional experimental investigation. For example, Figure 4 shows the temporal profiles of a set of four genes, $H38522$, $AA704613$, $N92443$ and $N75595$, that are known to be induced by interferon-β, the drug used to treat multiple sclerosis patients in our clinical studies.

Method 2: User Defined Patterns This method is similar to Method 1, except that an synthetic pattern is used instead of the temporal profile of a specific gene. The synthetic patterns may not exist in the data set or the user may be interested in identifying unknown genes that share similarity with the pattern of interest. Figure 5 shows two examples: a pattern that increases during the first several time points and decreases in the last several time points or a pattern decreases first and increases.

Method 3: Seed Searching with Constraints Unlike the first two methods of seed generation, which are supervised and require domain knowledge, Method 3 can be used when the user only has partial or very little knowledge about the pattern of interest.

In this Method, the seeds are identified randomly from the data and excluded from consideration unless they meet one or multiple following seed exclusion rules. These rules are independent from other phases. The users can always add other features or re-define the existing rules.

- It has been previously used or is coherent (to within a user defined threshold) with another previously used seed.
- It does not contain a definite temporal **trend**. The user can define the threshold number of consecutive time points that should be monotonically increasing or monotonically decreasing to constitute a trend. This eliminates patterns with frequent fluctuations, which are more likely represent noise. Figure 6 shows two patterns that differ in the length of the trends. The trend of pattern in Figure 6(A) is 1 while the trend of pattern in Figure 6(B) is 3. We hope to find more patterns having large trend values in the coherence blocks. Thus we use a threshold to qualify seeds. If the trend of the current pattern is less than the threshold, it will not be selected as a seed.
- The highest value or lowest value reaches at a certain **pick point**. The pick point specifies the relative time point that the pattern reaches its highest value or lowest value. Figure 7 shows an example of a pattern. The pattern reaches its highest value at time 1, lowest value at time 4. Thus we have “$highpickpoint'' = 1$ and “$lowpickpoint'' = 4$, respectively.

Importantly, the seed generation phase is independent of the block detection phase to allow sophisticated users to add other seed generation methods for specific applications.

Block Detection

The pseudocode for block detection is shown in Figure 8. The seed pattern is used to initialize each block and the block detection phase employs an iterative process that efficiently inserts or removes the genes and samples into and out of the blocks to improve the coherence value of each block.

The process is initiated by adding a random set of genes and samples to each seed pattern to create block of sufficient size.

In the first loop of the iteration, a randomized sequence of all genes and samples is examined and in the second loop, every block is examined. Genes or samples are either inserted or removed from their current block based on the coherence gain, $\Delta C = C_{after} - C_{before}$, of the action. If the coherence gain is positive, the insert/remove action is conducted, and if negative, the action is conducted with a probability $p = \exp \frac{\Delta C}{C_{before} \times T(i)}$.

The probability function p has two components. The first component, $\frac{\Delta C}{C_{before}}$, is simply the fractional decrease of coherence value and greater fractional decreases reduces the probability that the action will be performed. The second component $T(i)$, is a simulated annealing function that reduces p as the iteration number, i increases. In our implementation, a slow annealing function of the form, $T(0) = 1$ and $T(i) = \frac{1}{1+i}$ is used. Slow annealing is more effective at approaching global optimum than rapid annealing but requires more iteration (Kirkpatrick *et al.*, 1983).

Several additional measures are implemented during iterations of the block detection phase. The length of the gene, sample and time dimensions of a block are checked and maintained during the iteration to ensure that all the blocks meet $||G'|| \geq min_g$ AND $||S'|| \geq min_s$ AND $||T'|| \geq min_t$. The blocks are sorted descending order of coherence values and the blocks that do not have sufficient coherence are discarded. If more seed are generated or provide than necessary, this measure eliminates blocks that are not justified by the data at hand.

The best state for each block, i.e., the highest coherence value achieved among all iterations, is updated at the end of the iteration. The algorithm terminates when no positive action can be obtained during the iteration. Upon termination, the best state of each block is output.

RESULTS

Data Set

The gene-sample-time data set was obtained from our collaborators in the Departments of Pharmaceutical Sciences and Neurology. It consists of array measurements of $4,324$ known genes in 13 multiple sclerosis patients (MS) before and 1, 2, 4, 8, 24, 48, 120, 168 hours and 3 months after intra-muscular interferon-β treatment. The protocols used to obtain the data are described in Weinstock-Guttman *et al.* (2003). MS patients exhibit considerable heterogeneity in their clinical responses to interferon-β: only $30 \sim 40\%$ of patients respond well to treatment and the remainder exhibit varying degrees of partial responsiveness. This pharmacodynamic study was conducted to delineate the genomic effects of interferon-β to delineate the molecular basis of response heterogeneity.

Results on the MS Microarray Data

In the computational experiment, we first used the temporal profiles of 18 genes in Table 1 as seeds for Method 1 of seed selection, denoted as $SeedSet1$. These interferon-β induced genes are involved primarily in the antiviral and immunomodulatory pathways and are biologically interesting; they were examined in detail in Weinstock-Guttman *et al.* (2003). Then, we designed the patterns shown in Figure 9 as the seeds (denoted as $SeedSet2$) by Method 2 of seed selection. The seeds were designed to identify genes whose expression peaked at $2 \sim 4$ hours and $4 \sim 8$ hours. At last, we generated 30 seeds using Method 3, denoted as $SeedSet3$. The threshold of trend constrain was chosen as 2.

We apply our algorithms on the MS microarray data with $min_g = 10$, $min_s = 3$ and $min_t = 4$. In total, 44 coherent GST-blocks are reported. 11 blocks are generated from $SeedSet1$, 10 from $SeedSet2$ and 23 blocks from $SeedSet3$. The patterns of 44 coherent GST-blocks are shown in Figure 10, Figure 11 and Figure 12. These three figures demonstrate that the patterns in each blocks are highly coherent with each other. The highest coherence value of the 44 blocks is 0.9663, the lowest value is 0.7977 while the average coherence value is 0.8961. If we set $min_g = 5$, 57 coherent GST-blocks are reported. 13 blocks are generated from $SeedSet1$, 15 from $SeedSet2$ and 29 blocks from $SeedSet3$. The average coherence value will be 0.8957.

To better understand the coherent block detection results, we feed the genes in each block to *Onto-Express* (http://vortex.cs.wayne.edu/Projects.html) and obtain a hierarchy of functional annotations in terms of *Gene Ontology* for each block. Then, we further investigate the genes and samples in the blocks.

Some interesting observations are obtained as follows.

First, as expected, the majority genes in the blocks are involved in cellular process and physiological process, while genes involved in other biological process (e.g., development, behavior and viral life cycle) are not highly represented. Moreover, among the genes involved in cellular process, those involved in cellular physiological process (cell growth and/or maintenance) are predominant while cell communication is ranked second. Since IFN-β is known to have anti-proliferative activities, the high population of cellular process genes involved in cell growth and/or maintenance is biologically plausible. An example of gene ontology tree for a randomly selected block is shown in Figure 13(A). Figure 13(B) shows the distribution of number of genes in each category of biological process. Among 669 genes in biological process, 407 of them are involved in celluar process, 595 of them are in physiological processes. Figure 13(C) shows the distribution of number of genes in each category of celluar process. For the 407 genes involved in the cellular process, 219 genes are in the cell communication category while 246 of them involved in cell growth and/or maintenance process.

Second, a bunch of genes (e.g., genes $AA664040$, $AA485426$, $AA634166$, $AA157813$, etc.) that are well known for transcriptional signaling and cellular signaling can be identified in the resulted blocks. Those genes, together with the other genes in the same block that are unknown or poorly understood, may serve as switches in the genetic network and hence play an essential role in the biological processes. Thus, studying the time-series of the genes in the coherent GST-blocks may greatly help people understand the regulatory mechanisms behind the response to IFN-β treatment.

Last, coherent GST-blocks also consist of different groupings of samples. For example, in the MS microarray data, the expression data from patients with different responses to IFN-β treatment are collected. Among the 44 reported blocks, 33 blocks consist of 3 samples, 12 blocks consist of 4 samples, while one blocks consist of the 6 samples. The group information of the samples, combined with the gene information from the blocks, provides promising hypothesis for different phenotypes.

DISCUSSION

A variety of supervised and unsupervised methods have been applied to gene-sample or gene-time microarray data sets. In Discussion, we will highlight the similarities and salient differences between our algorithm and other published clustering and subspace clustering methods.

Conventional Methods

Principal components analysis (PCA) has been applied to gene-sample data (Raychaudhuri *et al.*, 2000; Holter *et al.*, 2000; Alter *et al.*, 2000) but the components do not always have useful sample prediction capabilities and often do not capture phenotype structures (Yeung *et al.*, 2001). The poor predictive capabilities of PCA with array data arise because the genes accounting for most of the variance in the data are frequently not the most informative of the class distinction of interest. As a result, a variety of sample data set analysis algorithms (e.g., Ben-Dor *et al.* (20001); Xing and Karp (2001); Tang *et al.* (2004)) have been proposed; these partition the sample dimension to detect their macroscopic phenotypes and identify informative genes that manifest the sample partition.

The gene-time data analysis tools algorithms (e.g. Tavazoie *et al.* (1999); Tamayo *et al.* (1999); Jiang *et al.* (2003)) focus on clustering on the gene dimension to identify groups of co-expressed genes based on their temporal expression patterns. Principal components analysis (PCA) has also been applied to array time series data (Raychaudhuri *et al.*, 2000; Holter *et al.*, 2000; Alter *et al.*, 2000) and a limited number of principal components usually accounts for the essential features of the data set allowing considerably reduced complexity, e.g., the sporulation data was modeled using as few as two principal components (Raychaudhuri *et al.*, 2000). By modeling gene expression as Markov processes, Holter et al. extended the principal components analysis/singular value decomposition to estimate the transition matrix for a subset of the principal components. Because the kinetic data is obtained at a limited number of time points, the general problem of computing the transition matrix for an array containing G genes contains G^2 elements and is ill posed. Transition matrices for clustered data and for interpolated time courses have also been examined (D'Haeseleer *et al.*, 1999). The CAGED software program (Ramoni *et al.*, 2002,?) uses a Bayesian framework for clustering and autoregressive models for representing time series.

Although such clustering algorithms can be applied sequentially to 2-dimensional, gene-sample data or gene-time data to obtain gene or sample partitions, they cannot be applied to 3-dimensional gene-sample-time data.

Subspace Clustering or Bi-clustering

The clustering approaches search exclusive and exhaustive partitions of objects that share the same feature space. However, current thinking in molecular biology holds that only a small subset of genes participate in any cellular process of interest and that a cellular process takes place only in a subset of the samples. This belief calls for the subspace clustering to capture "blocks" formed by a subset of genes across a subset of samples.

A bi-cluster is a subset of genes coherent with a subset of conditions and Cheng and Church introduced a biclustering algorithm (Cheng and Church, 2000); Yang *et al.* (2002) implemented a move-based algorithm to find biclusters more efficiently. However, both biclustering algorithms in adopt heuristic search approaches that do not allow incorporation of domain knowledge (Cheng and Church, 2000; Yang *et al.*, 2002).

Wang *et al.* (2002) gave a new definition of pattern-based cluster on two dimensional microarray data sets and proposed a depth-first algorithm to find all pattern-based clusters from two dimensional microarray data sets. The definition of *pattern-based clusters* have **closure property** that any sub-cluster of a pattern-based cluster is a *pattern-based cluster*. According to the closure property, if a block is not a pattern-based cluster according to the cluster definition, any super set of the block is not possible to be a pattern-based cluster. Thus the pattern-based clusters can be searched in a relatively efficient way by pruning all unnecessary enumeration of all super-sets of a non-cluster block. Pei *et al.* (2003) proposed an efficient algorithm, MaPle to mine the complete set of maximal pattern-based clusters.

However, both methods (Wang *et al.*, 2002; Pei *et al.*, 2003) have too many output blocks that are hard to interpret and do further analysis. In (Pei *et al.*, 2003), the authors applied their algorithm on the Yeast microarray data set which contains 2884 genes and 17 conditions. They set $\delta = 0$ that only output blocks in which the genes follow exact same pattern. With the block size varying from 30×9 to 50×7, the number of qualified varies from 5520 to 3.37×10^15. The method in (Wang *et al.*, 2002) outputs every subset of the results in (Pei *et al.*, 2003). Thus the method in (Wang *et al.*, 2002) should give much larger number of blocks. With thousands or even millions of blocks, users can hardly do any further semantic analysis.

Therefore, we have proposed a novel, semi-supervised model to incorporate domain knowledge to detect meaningful coherent blocks from gene-sample-time series microarray data sets. We adopted heuristic searching and simulated annealing techniques to approach global optimum solutions. The problem and its solution have practical bioinformatics applications, e.g., for identifying drug response phenotypes in pharmacodynamic studies employing microarray methods.

ACKNOWLEDGEMENTS

Support from the National Science Foundation (Research Grant 0234895) and the National Institutes of Health (P20-GM 067650) is gratefully acknowledged. The multiple sclerosis work in Dr. Ramanathan's laboratory was supported in part by a grant RG3258A2 from the National Multiple Sclerosis Society.

References

Alter O., Brown P.O. and Bostein D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, Vol. 97(18):10101–10106.

Azuaje, Francisco (2000) Making Genome Expression Data Meaningful: Prediction and Discovery of Classes of Cancer Through a Connectionist Learning Approach. In *Proceedings of IEEE International Symposium on Bioinformatics and Biomedical Engineering*, pages 208–213.

Ben-Dor A., Friedman N. and Yakhini Z. (2001) Class discovery in gene expression data. In *Proc. Fifth Annual Inter. Conf. on Computational Molecular Biology (RECOMB 2001)*, pages 31–38. ACM Press.

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297.

Cheng, Y., Church, G.M. (2000). Biclustering of expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 8:93–103.

DeRisi, J., Penland, L., Brown, PO., Bittner, M.L., Meltzer, P.S., Ray, M., Chen, Y., Su, Y.A., Trent, J.M. (1996). Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nature Genetics*, 14:457–460.

D'Haeseleer P., Wen X., Fuhrman S., Somogyi R. (1999) Linear modeling of mRNA expression levels during CNS development and injury. *Pac Symp Biocomput*, pages 41–52.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868.

Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gassenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,J.R., Caligiuri,M.A. *et al.* (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537.

Hastie, T., Tibshirani, R., Boststein, D., and Brown, P. (2001). Supervised harvesting of expression trees. *Genome Biology*, Vol. 2(1):0003.1–0003.12.

Holter NS., Mitra M., Maritan A., Cieplak M., Banavar JR., Fedoroff NV. (2000) Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proc Natl Acad Sci*, 97(15):8409–8414.

Iyer, V.R., Eisen, M.B., Ross, D.T., Schuler, G., Moore, T., Lee, J.C., Trent, J.M., Staudt, L.M., Hudson, J.Jr., Boguski, M.S. *et al.* (1999) The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87.

Jiang, D., Pei, J., and Zhang, A. (2003) Interactive Exploration of Coherent Patterns in Time-Series Gene Expression Data. In *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, Washington, DC, USA.

Kirkpatrick, S., Gelatt, C.D.Jr., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Pei, J., Zhang, X., Cho, M., Wang, H. and Yu, P.S. (2003) MaPle: A Fast Algorithm for Maximal Pattern-based Clustering. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03).*

Ramoni, M.F., Sebastianin, P., Cohen, P.R. (2002) Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121.

Ramoni, M.F., Sebastiani, P., Kohane, I.S. (2002) Cluster analysis of gene expression dynamics. *Proc Nat Acad Sci USA*, 99(14):9121–9126.

Raychaudhuri, S., Stuart, J.M., Altman, R.B. (2000) Principal components analysis to summarize microarray experiments: application to sporulation time series. *Pac Symp Biocomput*, pages 455–466.

Slonim, D.K., Tamayo, P., Mesirov, J.P., Golub, T.R. and Lander, E.S. (2000) Class Prediction and Discovery Using Gene Expression Data. In *RECOMB 2000: Proceedings of the Fifth Annual International Conference on Computational Biology*. ACM Press.

Tamayo P., Solni D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E.S. and Golub T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, Vol. 96(6):2907–2912.

Tang, C., Zhang, A., and Ramanathan, M. (2004) ESPD: A Pattern Detection Model Underlying Gene Expression Profiles. *Bioinformatics*, 20(6):829–838.

Tavazoie, S., Hughes, D., Campbell, M.J., Cho, R.J. and Church, G.M. (1999). Systematic determination of genetic network architecture. *Nature Genet*, pages 281–285.

Wang, H., Wang, W., Yang, J. and Yu, P.S. (2002) Clustering by Pattern Similarity in Large Data Sets. In *SIGMOD 2002, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 394–405.

Weinstock-Guttman, B., Badgett, D., Patrick, K., Hartrich, L., Hall, D., Baier, M., Feichter, J. and Ramanathan, M. (2003) Genomic effects of interferon-b in multiple sclerosis patients. *Accepted pending revisions Journal of Immunology*.

Xing, E.P. and Karp, R.M. (2001). Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, Vol. 17(1):306–315.

Yang, J., Wang, W., Wang, H., and Yu, P.S. (2002). δ-cluster: Capturing Subspace Correlation in a Large Data Set. In *Proceedings of 18th International Conference on Data Engineering (ICDE 2002)*, pages 517–528.

Yeung, K.Y., Haynor, D.R. and Ruzzo, W.L. (2001) Validating Clustering for Gene Expression Data. *Bioinformatics*, Vol.17(4):309–318.

TABLES

Unigene ID	Gene Name
Hs.833	IFN-stimulated protein, 15kDa
Hs.183487	IFN-stimulated gene (20kDa)
Hs.82030	Tryptophanyl-tRNA synthetase
Hs.241510	IFN-induced protein 41, 30kD
Hs.146360	IFN-induced transmembrane protein 1 (9-27)
Hs.1279	Complement component 1, r subcomponent
Hs.21486	Stat-1, 91kDa
Hs.62661	GBP-1, IFN-inducible, 67kDa
Hs.20315	IFN-induced protein 56
Hs.274382	Protein kinase, IFN-inducible dsRNA dependent
Hs.926	Mx2
Hs.79322	Glutaminyl-tRNA synthetase
Hs.278613	IFN-α-inducible protein 27
Hs.86958	IFN (α, β and ω) receptor 2
Hs.76391	Mx1
Hs.80645	IFN regulatory factor 1
Hs.174195	IFN-induced transmembrane protein 2 (1-8D)
Hs.171862	GBP-2, IFN-inducible
Hs.155530	IFN-γ-inducible protein 16
Hs.77367	monokine induced by IFN-γ
Hs.83795	IFN regulatory factor 2
Hs.277477	MHC, class I, C
Hs.14623	IFN-γ-inducible protein 30
Hs.75415	β_2-Microglobulin

Table 1: The set of IFN-stimulated genes used as seeds.

FIGURE LEGENDS

Figure 1. Figure 1 shows the format of a conventional gene expression data set which is a two dimensional matrix.

Figure 2. The structure of a gene-sample-time series (GST) data set.
(A) The 3-dimensional structure of a GST dataset.
(B) 2-dimensional representation of the GST dataset. The blocks marked “A” and “B” are two examples of coherent GST-blocks.

Figure 3. Figure 3 summarizes the problem definition with inputs and outputs.

Figure 4. Figure 4 shows four seeds identified with domain knowledge.

Figure 5. Figure 5 shows two examples of the synthetic patterns defined by user.

Figure 6. Figure 6 shows two patterns that differ in the length of the trends.

Figure 7. Figure 7 shows the high pick point and low pick point of a pattern.

Figure 8. The pseudo-code of the block detection phase.

Figure 9. Figure 9 shows 6 example patterns of the seeds defined by users.

Figure 10. Figure 10 shows the coherent blocks generated from $SeedSet1$. The title of each block shows the seed ID, the number of patterns in the block and the coherence value of the block. The horizontal axis shows the 9 time points, from before to 3 months. Each polyline is a pattern of a certain gene on a certain sample.

Figure 11. Figure 11 shows the coherent blocks generated from $SeedSet2$. The title of each block shows the number of patterns in the time points of the block. Each polyline is a pattern of a certain gene on a certain sample on the specific time intervals.

Figure 12. Figure 12 shows the coherent blocks generated from $SeedSet3$. The title of each block shows the number of patterns in the block and the coherence value of the block. The horizontal axis shows the 9 time points, from before to 3 months. Each polyline is a pattern of a certain gene on a certain sample.

Figure 13. Figure 13 shows the hierarchy of functional annotations in terms of *Gene Ontology* for an example block.

(A) The gene ontology tree for genes in the block.

(B) The distribution of biological process.

(C) The distribution of cellular process.

FIGURE 1:

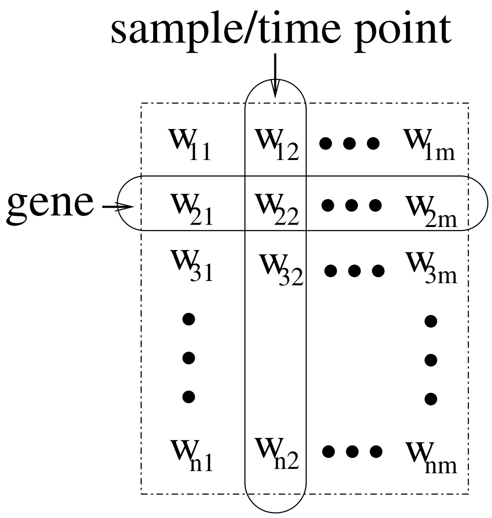

sample/time point
gene
w_{11} w_{12} w_{1m}
w_{21} w_{22} w_{2m}
w_{31} w_{32} w_{3m}
w_{n1} w_{n2} w_{nm}

FIGURE 2:

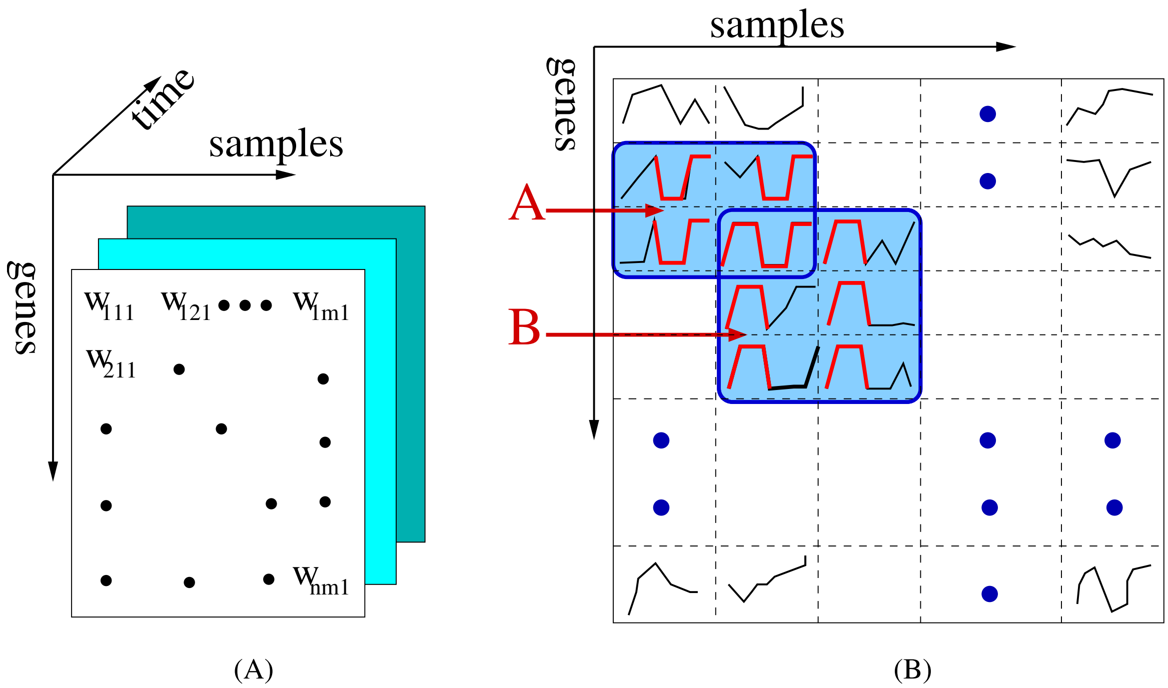

time
samples
genes
W111 W121 W1m1
W211
Wnm1
(A)
samples
genes
A
B
(B)

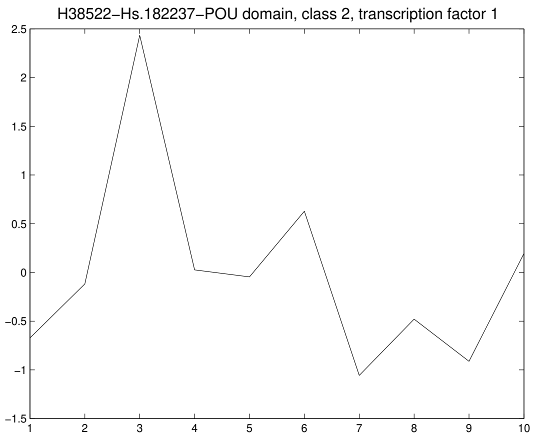
FIGURE 3:

Input:

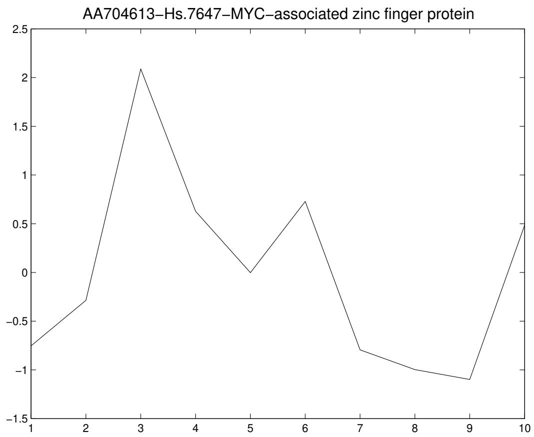
1. m samples, n genes and t time points;
2. the corresponding gene-sample-time series expression matrix M;
3. min_g, min_s and min_t, as the minimum number of genes, samples and length of time interval for block size;
4. approximate number of output blocks K.

Output: K coherent GST-blocks with respect to min_s, min_g and min_t.

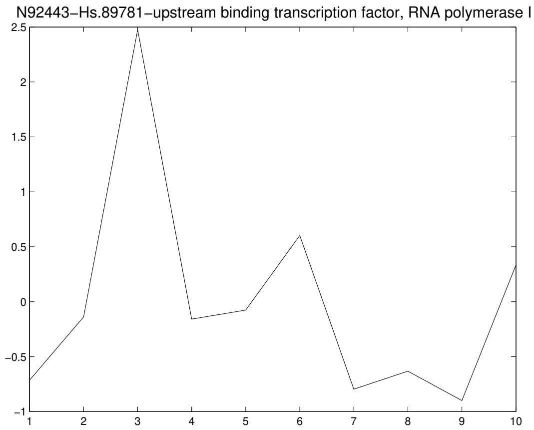
FIGURE 4:



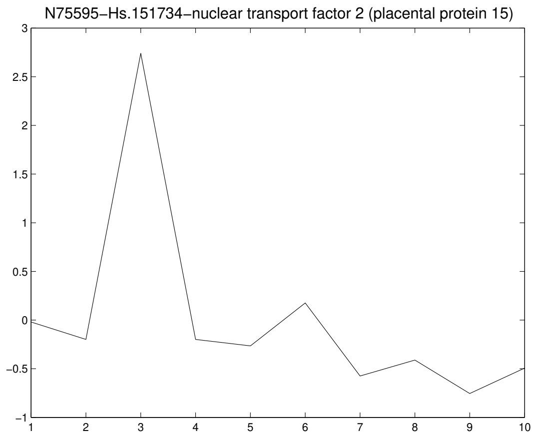
A. H38522



B. AA704613



C. N92443



D. N75595

FIGURE 5:

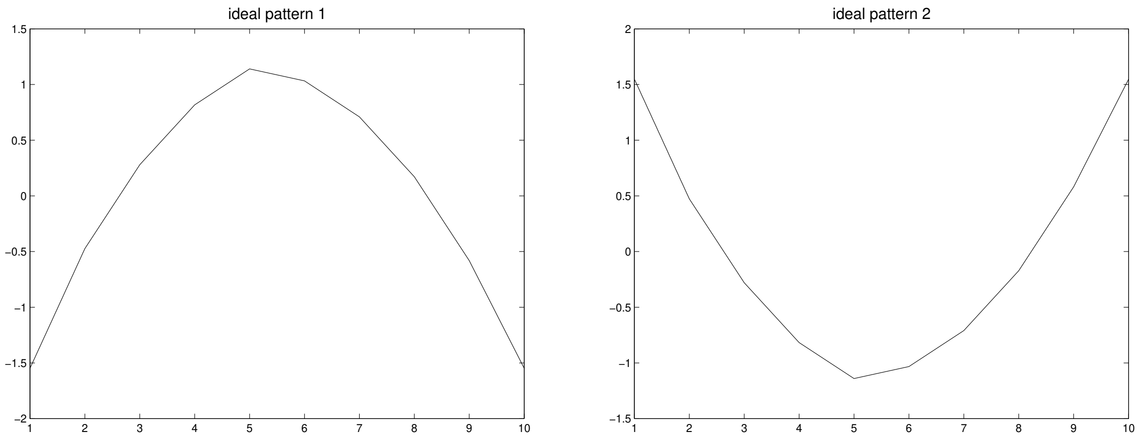
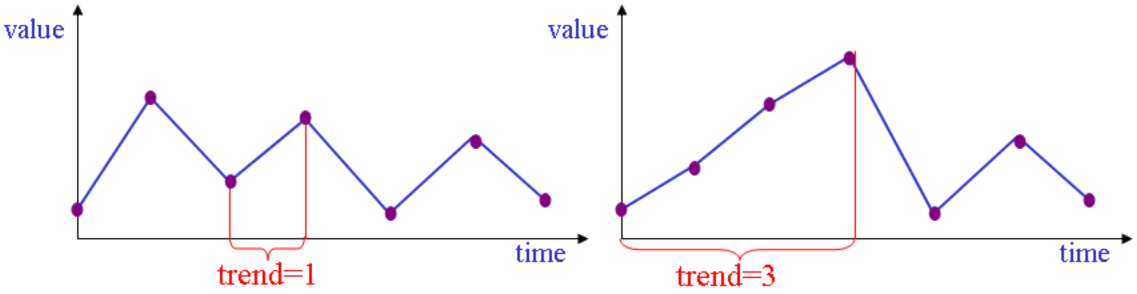


FIGURE 6:



(A) (B)

FIGURE 7:

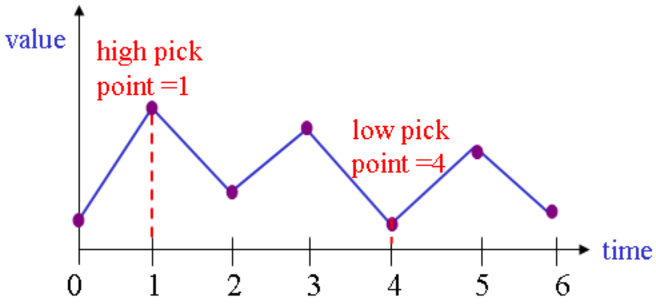


FIGURE 8:

Block Detection Phase:

1) Expand each seed to a block.

2) Repeat:

List an sequence of genes and samples randomly;

For each gene or sample along the sequence, do:

For each candidate block

compute ΔCoh for the possible insert/remove;

if $\Delta Coh > 0$, then conduct the action;

else if $\Delta Coh(G', S', T') < 0$,

then conduct the action with

probability $p = \exp(\frac{\Delta Coh}{Coh \times T(i)})$.

3) Until no positive action can be conducted in one iteration.

Output the best state of each block.

FIGURE 9:

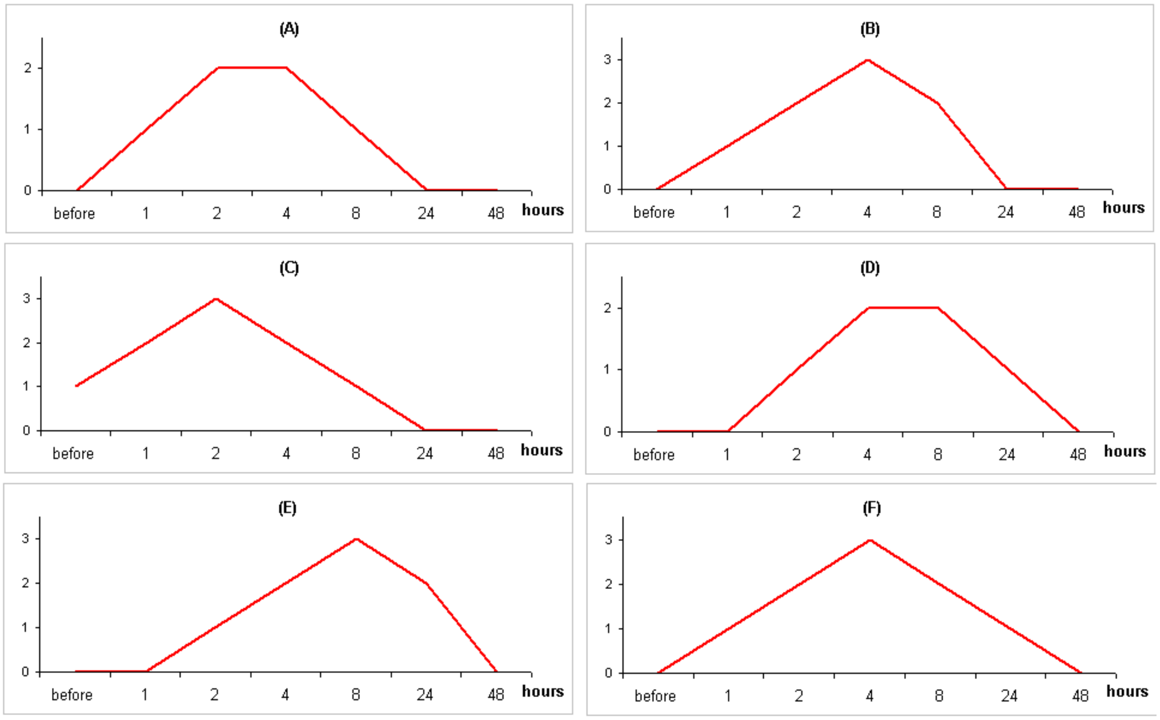


FIGURE 10:

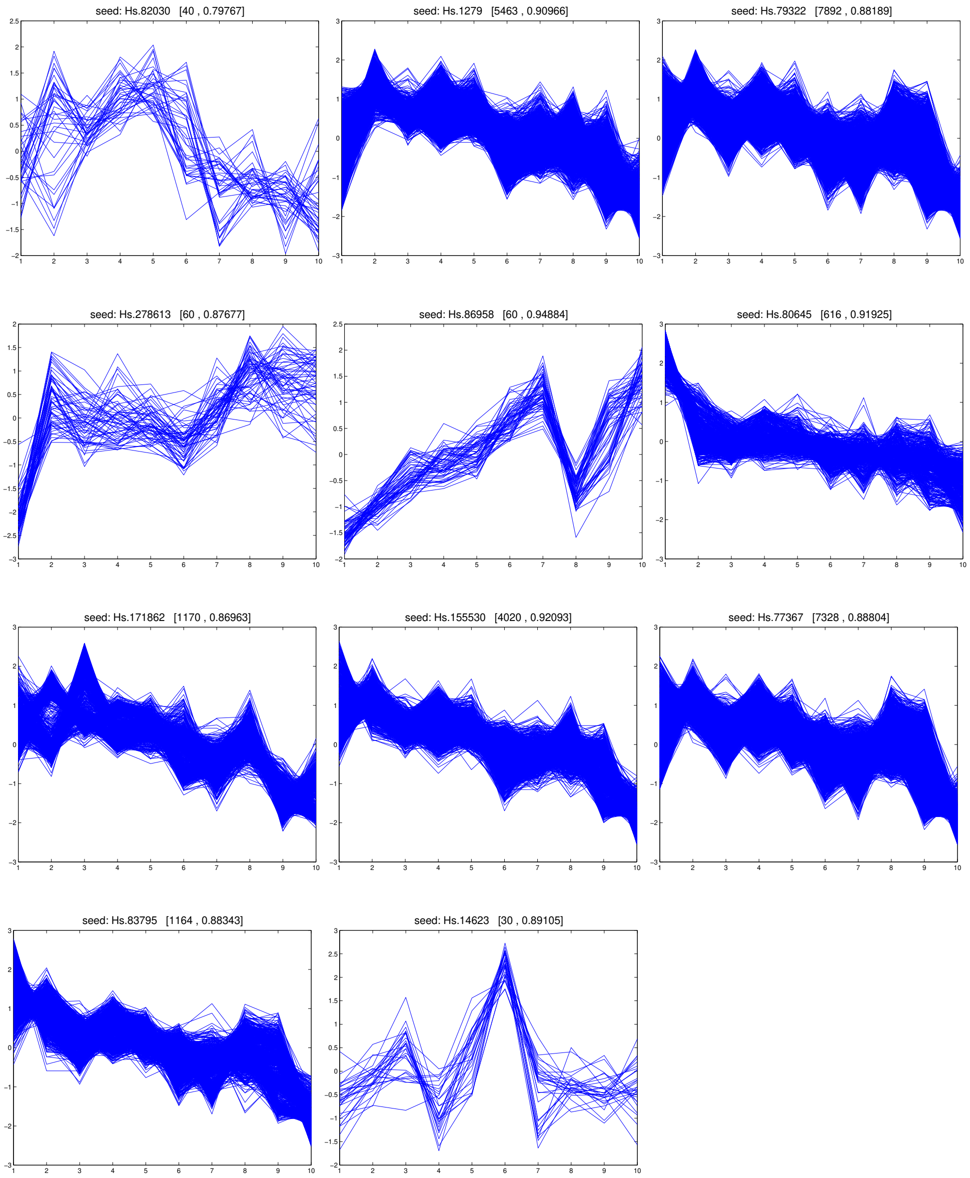
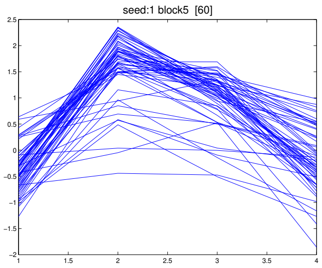
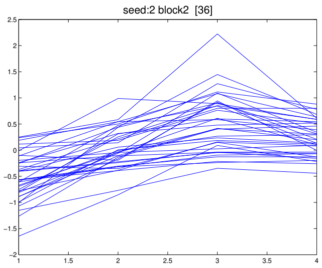

seed: Hs.82030 [40 , 0.79767]
seed: Hs.1279 [5463 , 0.90966]
seed: Hs.79322 [7892 , 0.88189]
seed: Hs.278613 [60 , 0.87677]
seed: Hs.86958 [60 , 0.94884]
seed: Hs.80645 [616 , 0.91925]
seed: Hs.171862 [1170 , 0.86963]
seed: Hs.155530 [4020 , 0.92093]
seed: Hs.77367 [7328 , 0.88804]
seed: Hs.83795 [1164 , 0.88343]
seed: Hs.14623 [30 , 0.89105]

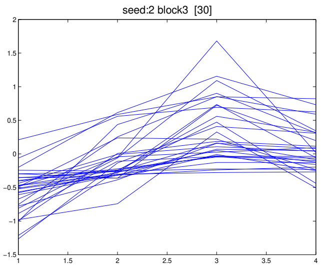
FIGURE 11:



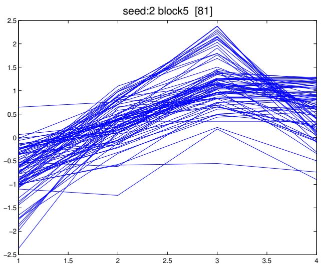
1,2,4,8 hour



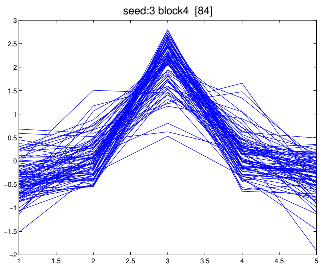
1,2,4,8 hour



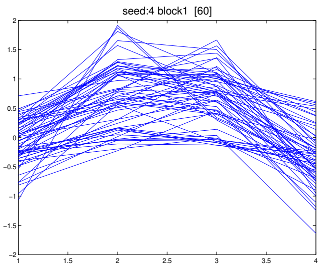
1,2,4,8 hour



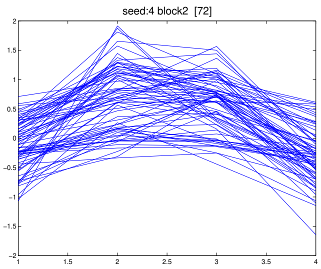
1,2,4,8 hour



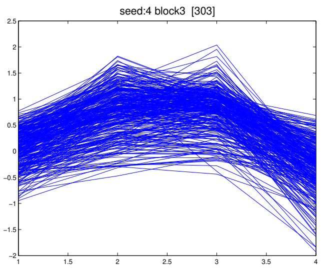
before,1,2,4,8 hour



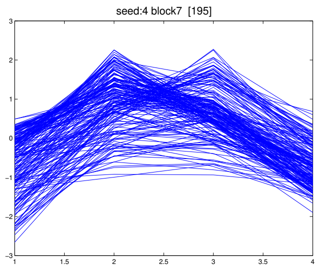
2,4,8,24 hour



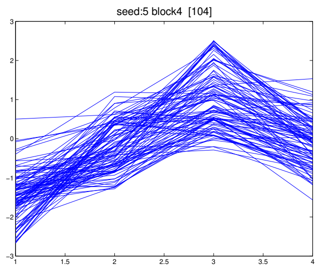
2,4,8,24 hour



2,4,8,24 hour

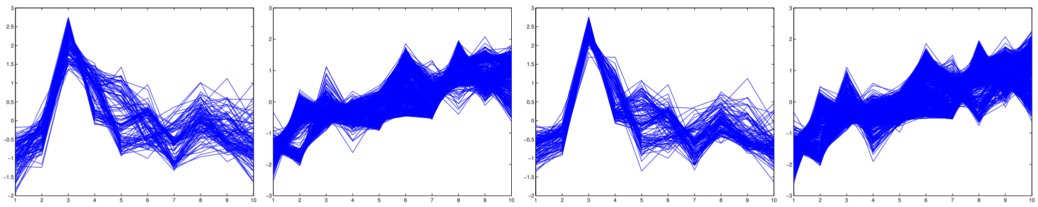


2,4,8,24 hour

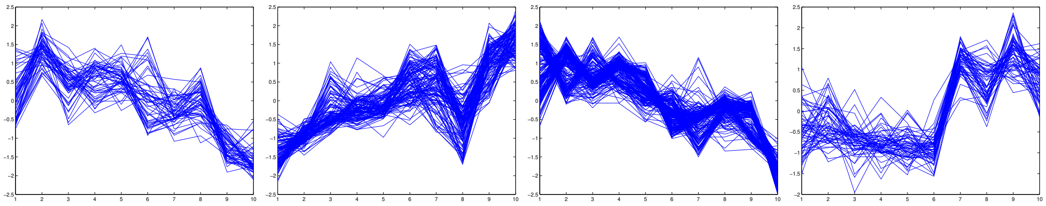


2,4,8,24 hour

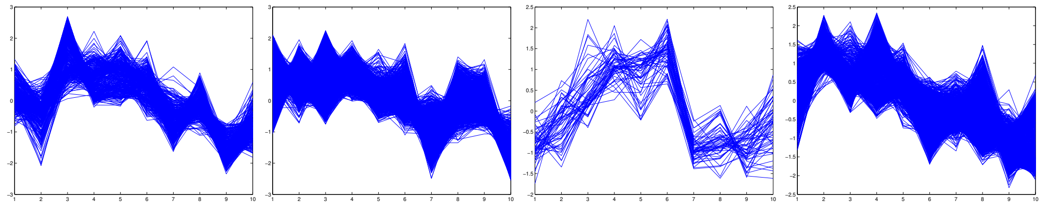
FIGURE 12:



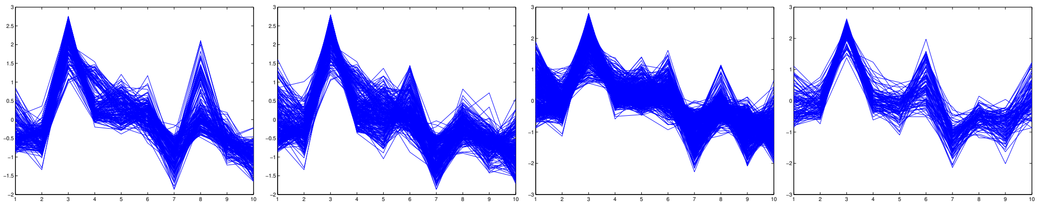
[105 0.8877] [704 0.8879] [114 0.9277] [813 0.8553]



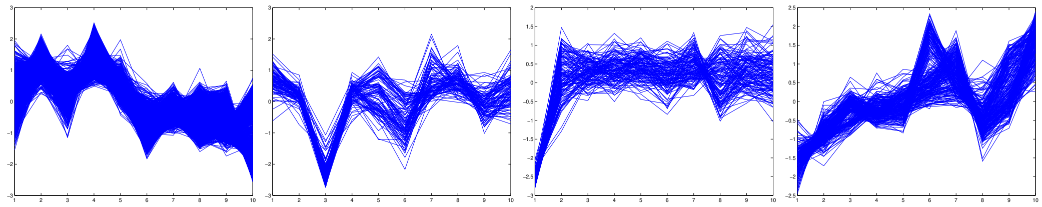
[54 0.8860] [123 0.8991] [222 0.9040] [60 0.9051]



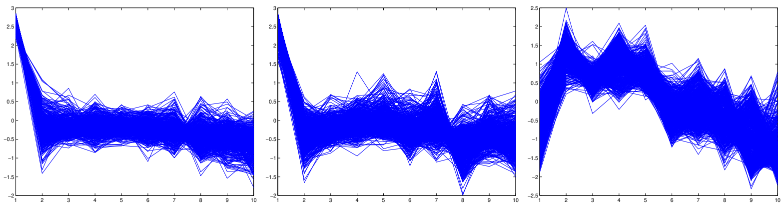
[372 0.8811] [2622 0.8725] [57 0.8941] [1914 0.8687]



[180 0.9160] [288 0.9080] 522 0.8882] [102 0.9122]

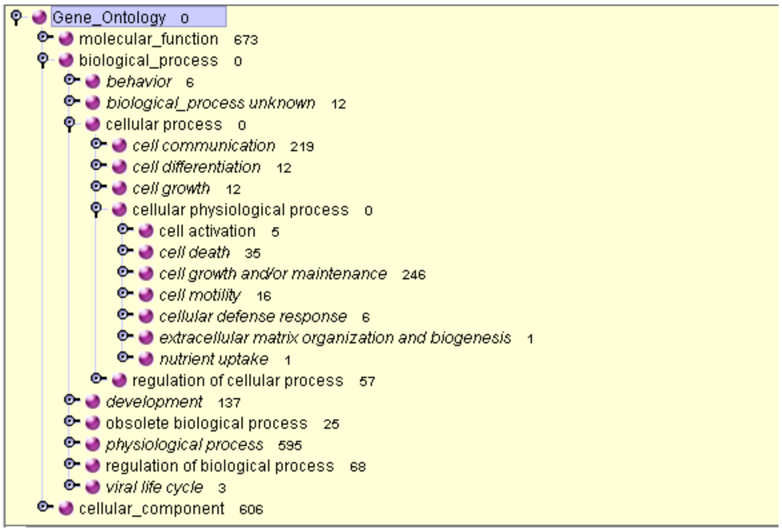


[4755 0.8775] [108 0.8987] [136 0.9073] [258 0.8952]

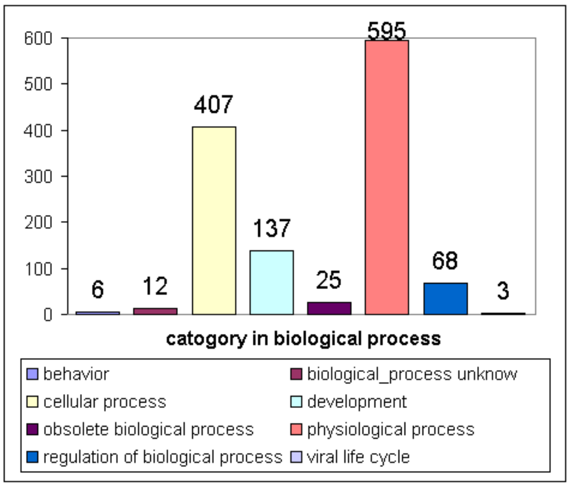


[702 0.9663] [906 0.9519] [855 0.8908]

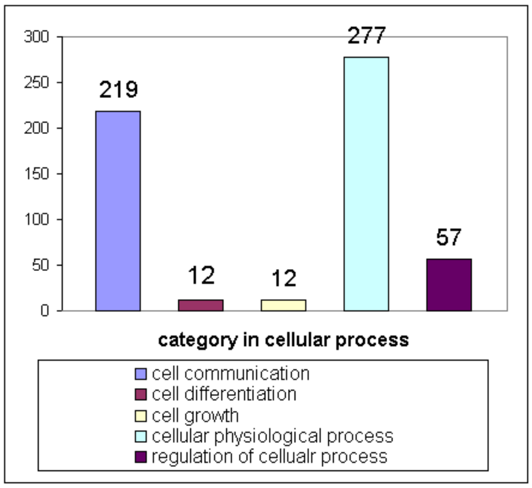
FIGURE 13:



(A)



(B)



(C)