THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Computing 1, Singapore 117590

TRB6/08

Finding Time-lagged 3D Clusters

Xin Xu, Ying Lu, Kian-Lee Tan and Anthony K.H. Tung

June 2008

T e c h n i c a l R e p o r t

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

OOI Beng Chin
Dean of School

Finding Time-lagged 3D Clusters

Xin Xu
Natl. University Hospital
xu.xin@nus.edu.sg

Ying Lu
UIUC
yinglu@uiuc.edu

Kian-Lee Tan, Anthony K. H. Tung
Natl. University of Singapore
{tankl,atung}@comp.nus.edu.sg

ABSTRACT

Existing 3D clustering algorithms on $gene \times sample \times time$ expression data do not consider the *time lags* between correlated gene expression patterns. Besides, they either ignore the correlation on *time subseries*, or disregard the *continuity* of the time series, or only validate pure shifting or pure scaling coherent patterns instead of the general *shifting-and-scaling patterns*. In this paper, we propose a novel 3D cluster model, S^2D^3 Cluster, to address these problems, where S^2 reflects the shifting-and-scaling correlation and D^3 the 3-Dimensional $gene \times sample \times time$ data. Within the S^2D^3 Cluster model, expression levels of genes are shifting-and-scaling coherent in both sample subspace and time subseries with arbitrary time lags. We develop a 3D clustering algorithm, $LagMiner$, for identifying interesting S^2D^3 Clusters that satisfy the constraints of regulation (γ), coherence (ϵ), minimum gene number ($MinG$), minimum sample subspace size ($MinS$) and minimum time periods length ($MinT$). Experimental results on both synthetic and real-life datasets show that $LagMiner$ is effective, scalable and parameter-robust. While we use gene expression data in this paper, our model and algorithm can be applied on any other data where both spatial and temporal coherence are pursued.

1. INTRODUCTION

1.1 Motivation

Recent advances in microarray technologies have made it possible to measure the expression profiles of thousands of genes simultaneously. Clustering gene expression data is one of the most important tasks since similar expression profiles of genes could indicate a related function or the same cellular pathway [9]. To ensure statistical robustness, gene expression profiles of multiple **samples** are typically taken and these samples are usually generated either from the cells of the same tissue under varying conditions or from tissues of different patients with the same disease/illness. Furthermore, to analyze the changes of a cell over time, gene expression of the same sample can also be taken at different time point giving rise to time varying profiles. When the expression profiles of thousands of genes are taken for multiple samples over multiple time points, this give rise to a $gene \times sample \times time$ dataset which is the focus of our study in this paper.

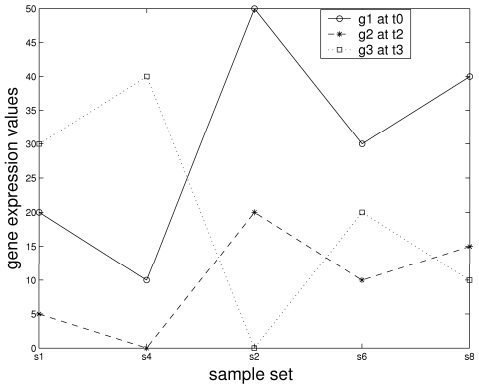


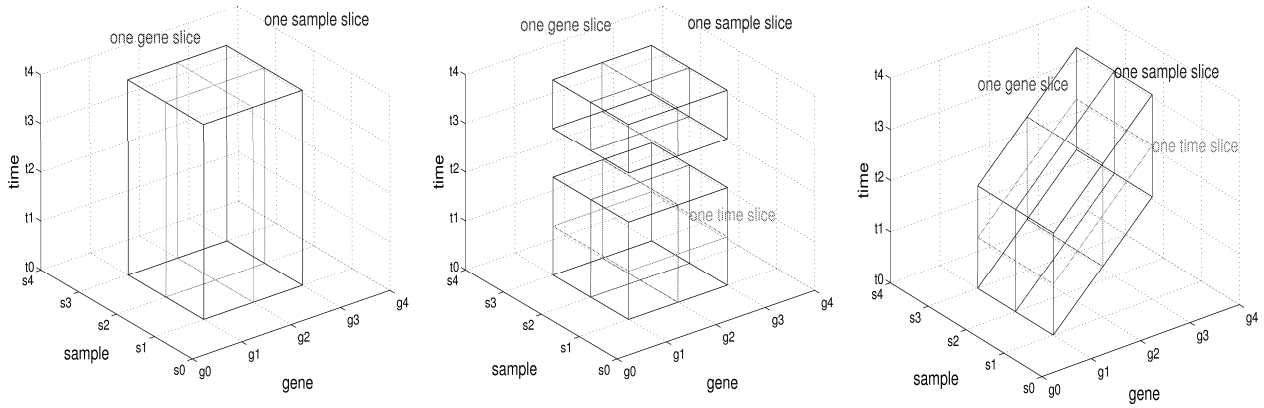
Figure 1: Time-lagged Correlation

Various clustering algorithm for gene expression datasets had been developed [7, 27, 25, 14]. In these algorithms, clusters of gene expression profiles can be found in a subset of samples (gene × sample clustering), during a certain time course (gene × time clustering), or within a combination of the two (gene × sample × time clustering). However, none of the existing clustering algorithms has considered *time lagged correlation* in $gene \times sample \times time$ dataset. For instance, the up-regulation / down-regulation of gene g_1 across five samples at time point t_0 may boost/repress the transcription of gene g_2 at a later time point t_2 and repress/boost the transcription of g_3 at time point t_3 across these samples, as shown in Figure 1.

Existing 3D clustering algorithms are problematic in capturing correlated time periods, time continuity and coherency.

The pioneering $gene \times sample \times time$ clustering algorithm, the Sample-Gene and Gene-Sample search algorithm (SG/GS for short) [14], computes pattern similarity based on Pearson's correlation coefficient using the *entire* time course. In this way, it ignores the correlation on time periods and simply casts the 3D clustering task into a single biclustering one as claimed in [27].

Another 3D clustering algorithm TRICLUSTER [27] identifies 3D clusters. However, it only searches for coherent patterns within the same time points and ignores the *continuity of time series*. Such time discontinuity is not desirable as a time series experiment represents a dependent experimental environment [4]. In addition, TRICLUSTER only detects pure scaling correlations i.e. the expression of two genes in the same clusters are a constant multiplier of each other instead of the more general *shifting-and-scaling*



(a) SG/GS (b) TRICLUSTER (c) LagMiner

Figure 2: Comparison of Lagminer against Previous 3D Clustering Algorithms

patterns proposed in [25] where genes in the same clusters are similar to each other with a constant multiplier and a constant additive factor. As a result, TRICLUSTER cannot detect the three shifting-and-scaling patterns Figure 1 where g_2 and g_1 are positively correlated with a constant multiplier of 0.5 and an additive factor of -5 while g_2 and g_3 are negatively correlated with a constant multiplier of -0.5 and an additive factor of 20 [1]

1.2 Goal

In this paper, we propose a novel 3D cluster model which we call S^2D^3 Cluster, to address these problems (S^2 denoteshifting-and-scaling correlation and D^3 denote the 3-Dimensional $gene \times sample \times time$ data). Within the S^2D^3 cluster model, expression levels of genes are shifting-and-scaling coherent in both sample subspace and time periods with arbitrary time lags. We develop an efficient 3D clustering algorithm, $LagMiner$, to identify interesting S^2D^3 Clusters.

In Figure 2, an example is given to illustrate the different clusters found by SG/GS, TRICLUSTER and LagMiner. Assume we want to find 3D clusters from the $gene \times sample \times time$ expression data consisting of the expression levels of five genes, g_0, g_1, g_2, g_3, and g_4 from five samples, s_0, s_1, s_2, s_3 and s_4, at five continuous time points, t_0, t_1, t_2, t_3 and t_4. The possible sample output clusters of SG/GS, TRICLUSTER and LagMiner are as follows.

SG/GS only outputs clusters across the entire time course such as $\{g_1,g_2,g_3\} \times \{s_1,s_2,s_3\} \times \{t_0,t_1,t_2,t_3,t_4\}$ as Figure 2 (a) shows. From Figure 2 (b), we can see that the time periods of the cluster discovered by TRICLUSTER consists of discontinuous time points, $\{t_0,t_1,t_3,t_4\}$. Neither SG/GS nor TRICLUSTER can detect the time-lagged 3D cluster found by LagMiner in Figure 2 (c) where the time lags among the genes exhibit an arithmetic progression and the correlated time periods can be arbitrarily shifted.

We refer to the projection of 3D cluster on a corresponding gene/sample/time point as a gene/sample/time slice. Note that because of the time-lagged correlation, the time slice of the 3D cluster in Figure 2 (c), $\{g_1t_1,g_2t_2,g_3t_3\} \times \{s_1,s_2,s_3\}$, is no longer perpendicular to the time axis.

[1]To see a concrete example, see that the expression level of gene g_2 in sample s_4 at time t_2 is 0.5 times the expression level of gene g_1 in sample s_4 at time t_0 plus an additive factor of -5. We leave it to reader to see that g_1 and g_3 are related to each other in a similar way

1.3 Contributions

In summary, we make the following contributions in this paper:

(1) We propose a novel 3D cluster model, S^2D^3 Cluster, which successfully addresses the problems of time-lagged correlation, time periods correlation, time continuity and shifting-and-scaling coherence.

(2) Our cluster model is flexible and can be easily generalize to model gene × sample and gene × time clusters. Compared with previous 2D shifting-and-scaling model [25], it is more consistent and systematic.

(3) We develop an efficient algorithm for finding S^2D^3 cluster called LagMiner. Unlike the TRICLUSTER algorithm which evaluates the coherency of each dimensions individually, LagMiner performs coherency pruning in all the three dimensions simultaneously. Consequently, as we shall see in our experimental study, LagMiner is orders of magnitude faster than TRICLUSTER. Experiments on real-life 3D gene expression data also indicate the S^2D^3 clusters that are found by LagMiner are meaningful in biology.

The rest of this paper is organized as follows. We review some related work on gene expression data clustering in Section 2. We will present the S^2D^3 cluster model in Section 3. LagMiner will be introduce in Section 4. In Section 5, we present our experimental results and we will conclude in Section 6.

2. RELATED WORK

A wealth of work had been done on clustering gene expression data. According to the space, there are $gene \times sample$ clustering, $gene \times time$ clustering and the latest $gene \times sample \times time$ clustering.

2.1 Gene × Sample Clustering

Some gene × sample clustering algorithms judge profile similarity on full sample space while others evaluate similarity on sample subspace. This include hierarchical clustering [8], self-organizing map [22], and K-means [23]. Hierarchical algorithms merge genes with the most similar expression

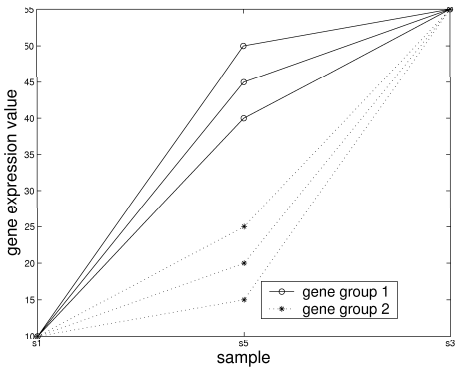


Figure 3: Bias of regcluster

profiles iteratively in a bottom-up manner. Self-organizing map and K-means partition genes into user-specified k optimal clusters.

The subspace clustering algorithms, on the other hand, cluster genes on a subset of samples. Some subspace clustering algorithms [1, 2, 3, 6, 12, 13, 19] are density-based, requiring genes of the same cluster to be dense and close to each other. These algorithms assign each gene to one cluster only. However, a gene may involve in several pathways, thus participating in several clusters. The heuristic biclustering algorithm [7] was the first piece of work that allows overlapping gene members between different clusters. The later pattern-based clustering algorithms [24, 26, 27] identify genes whose expression profiles are shifting or scaling patterns w.r.t. a coherence measure ϵ. These genes are likely to be sparse and far in the correlated subspace. The tendency-based clustering algorithms [5, 16, 17] discover genes whose expression levels change synchronously across samples.

More recently, another pattern-based algorithm, regcluster [25], has been proposed to detect the much more general shifting-and-scaling patterns. regcluster ensures that, within a cluster, the expression level difference of any two genes between any two samples are approximately proportional w.r.t. a coherence threshold ϵ. regcluster also applies a regulation threshold γ to control the regulation significance. regcluster selects two samples with the lowest gene expression levels as the baseline pair for similarity comparison. Such a strategy makes regcluster rather sensitive to the baseline dimension pair. For example, the two diagonal-symmetric gene groups in Figure 3 have exactly the same degree of coherence instinctively. However, regcluster is biased towards gene group 1 whose ratio scores[2] are much more coherent (ranging from 0.125 to 0.5) than those of gene group 2 (ranging from 2 to 8).

2.2 Gene × Time Clustering

Most gene × time clustering algorithms judge profile similarity on the whole time course and are unaware of timing variability of correlated biological processes, such as [15, 20]. GeneXplorer [15] uses Pearson's correlation coefficient to measure the overall similarity of gene expression profiles. An autoregressive model accounting for the dependency of temporal observations is adopted in [20] which fits the expression levels at later time points into a linear function of previous p time points.

[2]ratio of expression difference between samples s_5 and s_3 to that between baseline samples s_1 and s_5, $(P_{g_i,s_3} - P_{g_i,s_5})/(P_{g_i,s_5} - P_{g_i,s_1})$

However, time-lagged correlation has been generally overlooked by previous clustering algorithms, although it is a well-known issue for time series data analysis in bioinformatics [4, 21]. Besides, as stated in [10], gene expression profiles may be correlated at some shorter time periods rather than the whole time course. No gene × time clustering algorithms have considered both time-lagged correlation and correlation with shorter time period as far as we know.

2.3 Gene × Sample × Time Clustering

The SG/GS algorithm [14] evaluates pattern similarity with Pearson's correlation coefficient on the whole time course. It simply casts the 3D clustering problem into a 2D one. A more recent clustering algorithm TRICLUSTER [27] identifies real 3D clusters. However, TRICLUSTER has not considered the time-lagged correlation. It also ignores the continuity of time series data and evaluates profile coherence on disjoined time points. In addition, it validates pure scaling patterns only, but not the general shifting-and-scaling patterns [25].

Notation	Description
G_0, S_0, T_0	set of all genes $\{g_0, g_1, ..., g_{N-1}\}$, set of all samples $\{s_0, s_1, ..., s_{M-1}\}$ and the full time period of the dataset $(t_0, t_1, t_2..., t_{L-1})$ respectively
G,S	a set of genes, $G \subseteq G_0$ and a set of samples, $S \subseteq S_0$ respectively
T_i, $t_{m_i(l)}$	a time periods for gene g_i, $(t_{m_i}, t_{m_i+1}, ..., t_{m_i+l-1})$. m_i, the first time point of time periods T_i is call the time lag coefficient of T_i
t_{m_i}	Abbreviated form of $t_{m_i(1)}$ i.e. length of time period is 1
T	a set of time periods, {T_1, T_2, ..., }
GTP,$\widehat{g_i t_{m_i(l)}}$	the correlated time periods $t_{m_i(l)}$ for a gene g_i
$G.T$	a set of GTP, $\{g_i T_i\}_i = \{\widehat{g_i t_{m_i(l)}}\}_i$
P_{g_i,s_j,t_k}	the expression level of gene g_i on sample s_j at time point t_k
P_{g_j,t_k}	the expression level of gene g_j at time point t_k for a sample slice
P_{g_i,S,t_k}	a set of gene expression value $\{Pg_i, s'_1, t_k, ..., Pg_i, s'_{\|S\|}, t_k\}$ where $S = \{s'_1, ..., s'_{\|S\|}\}$
MinScore	minimum expression level change ratio for a set of genes G across three samples s_u, s_v and s_w, $Min_{g_i \in G} \frac{P_{g_i,\ s_v} - P_{g_i,\ s_u}}{P_{g_i,\ s_w} - P_{g_i,\ s_u}}$
MaxScore	maximum expression level change ratio for a set of genes G across three samples s_u, s_v and s_w, $Max_{g_i \in G} \frac{P_{g_i,\ s_v} - P_{g_i,\ s_u}}{P_{g_i,\ s_w} - P_{g_i,\ s_u}}$
TORD	relative order of time points in the periods
MSGTP	the maximum set of GTPs on a sample set.
$G.T \times S$	a time-lagged 3D cluster of genes from G at time periods in T respectively across samples in S
A	A set of generic entities $\{a_1, ..., a_n\}$
P_{a_j,s_k}	the expression level of entity a_j in sample s_k

Table 1: Notations

3. CLUSTER MODEL

3.1 Notation

For simplicity, we make use of several notations and concepts in our cluster model and algorithm. A short description of these notations and concepts is provided in Table 1. Formal definitions and illustrating examples will be given afterwards.

expression value	g_1					g_2					g_3				
	s_0	s_1	s_2	s_3	s_4	s_0	s_1	s_2	s_3	s_4	s_0	s_1	s_2	s_3	s_4
t_0	*	10	20	5	*	*	*	*	*	*	*	*	*	*	*
t_1	*	15	35	5	*	*	30	50	20	*	*	*	*	*	*
t_2	*	20	10	25	*	*	40	80	20	*	*	20	10	25	*
t_3	*	*	*	*	*	*	50	30	60	*	*	15	−5	25	*
t_4	*	*	*	*	*	*	*	*	*	*	*	10	20	5	*

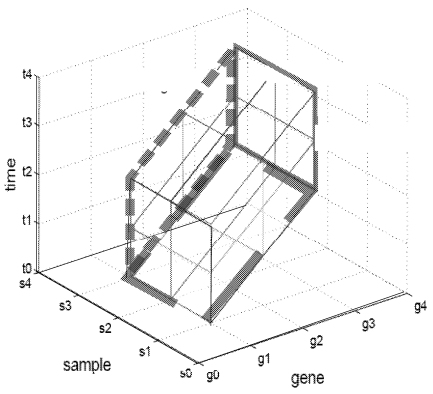
(a) Three Gene Slices

expression value	s_1					s_2					s_3				
	t_0	t_1	t_2	t_3	t_4	t_0	t_1	t_2	t_3	t_4	t_0	t_1	t_2	t_3	t_4
g_0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
g_1	10	15	20	*	*	20	35	10	*	*	5	5	25	*	*
g_2	*	30	40	50	*	*	50	80	30	*	*	20	20	60	*
g_3	*	*	20	15	10	*	*	10	−5	20	*	*	25	25	*
g_4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

(b) Three Sample Slices

expression value	(g_1t_0, g_2t_1, g_3t_2)					(g_1t_1, g_2t_2, g_3t_3)					(g_1t_2, g_2t_3, g_3t_4)				
	s_0	s_1	s_2	s_3	s_4	s_0	s_1	s_2	s_3	s_4	s_0	s_1	s_2	s_3	s_4
g_0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
g_1	*	10	20	5	*	*	15	35	5	*	*	20	10	25	*
g_2	*	30	50	20	*	*	40	80	20	*	*	50	30	60	*
g_3	*	20	10	25	*	*	15	−5	25	*	*	10	20	5	*
g_4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

(c) Three Time Slices



(d) S^2D^3Cluster

Figure 4: An Embedded Time-Lagged 3D Cluster

Let D be the *gene* × *sample* × *time* ($G_0 \times S_0 \times T_0$) expression data, where $G_0 = \{g_0, g_1, ..., g_{N-1}\}$, $S_0 = \{s_0, s_1, ..., s_{M-1}\}$, and T_0 is a continuous time series, $T_0 = (t_0, t_1, t_2, ..., t_{L-1})$. We denote the expression level of gene g_i on sample s_j at time point t_k as P_{g_i,s_j,t_k}, where $0 \leq i < N$, $0 \leq j < M$ and $0 \leq k < L$. To further simplify notation, we will overload P in two ways. The notations P_{g_i,t_k} will be used to represent the gene expression level in a sample slice which is essentially a 2D array where the value in the third dimension is fixed. Also, we will use P_{g_i,S,t_k} to represent a set of gene expression value $\{Pg_i, s'_1, t_k, ..., Pg_i, s'_{|S|}, t_k\}$ where $S = \{s'_1, ..., s'_{|S|}\}$.

We adopt the shifting-and-scaling correlation here because it is the most general correlation criterion to the best of our knowledge. Based on the shifting-and-scaling measurement, we expect the 3D cluster to have coherent time-sample curves at each gene slice (orthonormal to the gene axis), coherent gene-time curves at each sample slice (orthonormal to the sample axis), and also coherent gene-sample curves at each time slice (possibly not orthonormal to the time axis due to time lags).

As an example, refer to Figure 4d, where three gene, sample and time slices of a cluster C' are shown. Both the gene (one example highlighted with continuous line) and sample(one example highlight with short broken line) slices are orthonormal to the gene and sample axis respectively while the time slices(one example highlighted with long broken line) is not orthonormal. As a running example, we also show the actual value of these slices in Figure 4a, Figure 4b and Figure 4c with non-related space marked with '*'. To further show example of the mapping between Figure 4d and Figure 4a, 4b and 4c, we have marked the corresponding yellow, red and blue slices from Figure 4d in Figure 4a, 4b and 4c as well.

Because they are not orthonormal to the time axes, representing the three time slices of Figure 4c in mathematical notation is more complex than representing the sample and gene slices. Generally, this also make representing and defining a time-lagged 3D cluster more complex than previous cluster model which have clusters that are all orthogonal to the axes.

Formally speaking, a time-lagged 3D cluster $C = G \cdot T \times S$ consists of a subset of genes $G \subseteq G_0$, a subset of samples $S \subseteq S_0$, and a set of local time periods $T = \{T_i = (t_p, t_{p+1}, ..., t_{p+l-1}) | g_i \in G, l \leq L|\}$, where T_i is the gene-specific time periods of length l.

To have an intuition on how the mathematical definition of time-lagged 3D cluster come about, we will again use Figure 4 as an example. Note that any three slices from Figure 4a, 4b or 4c will be sufficient to represent C and we pick Figure 4a to derive our representation.

We can see that each gene, g_1, g_2 and g_3 are associate with different time periods: g_1 is associated with time period t_0 to t_2, g_2 is associated to time period t_0 to t_2, and g_3 associated with time period t_0 to t_2. To simplify such a presentation, we denote a time period from t_{m_i} to t_{m_i+l} as $t_{m_i(l)}$ with l being the length of the time period. In the case where $l = 1$, we will simple denote it as t_{m_i}. Thus g_1 is associate with $t_{0(3)}$, g_2 is associate with $t_{1(3)}$ and g_3 is associate with $t_{2(3)}$.

We call each of this association a **gene-time periods (GTP)**, denoted as $\widehat{g_i t_{m_i(l)}}$. This further simplify our representation of the gene-time period association to $\widehat{g_1 t_{0(3)}}$, $\widehat{g_2 t_{1(3)}}$, $\widehat{g_3 t_{2(3)}}$ which represented $\{g_1, g_2, g_3\} \cdot \{t_{0(3)}, t_{1(3)}, t_{2(3)}\}$. Finally, since these gene-time period association are true for sample set $\{s_1, s_2, s_3\}$, our 3D time lagged cluster C is just defined as $\{g_1, g_2, g_3\} \cdot \{t_{0(3)}, t_{1(3)}, t_{2(3)}\} \times \{s_1, s_2, s_3\}$ i.e. $G \cdot T \times S$ where $G = \{g_1, g_2, g_3\}$, $T = \{t_{0(3)}, t_{1(3)}, t_{2(3)}\}$ and $S = \{s_1, s_2, s_3\}$. Alternative, we also represent C as $\{\widehat{g_1 t_0}, \widehat{g_2 t_1}, \widehat{g_3 t_2}\}_3 \times \{s_1, s_2, s_3\}$ in this paper as well.

In the remaining of this section, we will formally present the framework of our cluster model. We will start with 2D shifting-and-scaling cluster model for *gene* × *sample* cluster and *gene*×*time* cluster first. Base on that, we will introduce our 3D shifting-and-scaling cluster model, S^2D^3 cluster, for *gene* × *sample* × *time* cluster.

3.2 Shifting and Scaling Patterns

We will first define a generic notion of shifting and scaling patterns in a general 2D array, $A \times S$ where A represents a general set of entities, $A = \{a_1, ..., a_n\}$ and S is a set of samples, $S = \{s_1, ..., s_m\}$. Later on in this paper,A will be instantiated to different type of entities (eg. genes, time periods) depending on the context of discussion. Here, we assume that each element in the array $A \times S$ is a gene expression value and is denote as P_{a_i,s_j}, $a_i \in A$, $s_u \in S$.

Given any two samples, s_u and s_w, it is always possible to make two elements in A, a_i and a_j perfectly linear correlated across the two samples i.e. we can always find a scaling factor c_1 and a shifting factor c_2 such that $P_{a_i,s_u} = c_1 * P_{a_i,s_w} + c_2$ and $P_{a_j,s_u} = c_1 * P_{a_j,s_w} + c_2$. Thus it does not make sense to define a shifting and scaling patterns for $|S| \leq 2$. [3]

Instead, we will define the coherence of shifting-and-scaling patterns across three and more samples w.r.t. threshold ϵ by proposing a **shifting-and-scaling score (S^2Score)** on every sample triplet. Note that the use of quadruple or more combination of samples can be done but this will not only restrict the clusters found to contain more samples but will also result in more expensive computation which will be discussed later.

Before we explain S^2Score, we will first define the **sample triplet order** on an element a_i as a set of three samples $\{s_u, s_v, s_w\}$ such that $s_u \prec s_v \prec s_w$ if (1) $P_{a_i,s_u} \leq P_{a_i,s_v} \leq P_{a_i,s_w}$ and $u < w$ or (2) $P_{a_i,s_u} \geq P_{a_i,s_v} \geq P_{a_i,s_w}$ and $u < w$. Essentially, the sample triple order ensure that the gene expression of P_{a_i,s_u}, P_{a_i,s_v} and P_{a_i,s_w} are either in increasing or decreasing order.

We measure the coherence of gene expression patterns on a sample triplet with our **shifting-and-scaling score (S^2Score)** which is equal to the range of expression level change ratio, **MaxScore-MinScore**, as defined below.

DEFINITION 1. *($S^2Score(A,\{s_u, s_v, s_w\})$)*
Assume A is a set of entities and $\{s_u, s_v, s_w\}$ is a sample triplet and $s_u \prec s_v \prec s_w$ is a sample triplet order that is the same for all elements in A. Then the shifting-and-scaling score (S^2Score for brevity) of A on $\{s_u, s_v, s_w\}$ is:

$$\begin{aligned} & S^2Score(A, \{s_u, s_v, s_w\}) \\ =& MaxScore - MinScore \\ =& Max_{a_i \in A} \frac{P_{a_i,s_v} - P_{a_i,s_u}}{P_{a_i,s_w} - P_{a_i,s_u}} - Min_{a_i \in A} \frac{P_{a_i,s_v} - P_{a_i,s_u}}{P_{a_i,s_w} - P_{a_i,s_u}} \end{aligned}$$

If entities in A share no common sample triplet order on $\{s_u, s_v, s_w\}$, then we define $S^2Score(A, \{s_u, s_v, s_w\}) = \infty$. □

LEMMA 3.1. *(S^2Score Property)*
Two genes a_i and a_j are perfectly shifting-and-scaling correlated on sample triplet $\{s_u, s_v, s_w\}$ if and only if

$$S^2Score(\{a_i, a_j\}, \{s_u, s_v, s_w\}) = 0.$$

Proof:
(1) Assume $P_{a_i,\{s_u,s_v,s_w\}} = c_1 * P_{a_j,\{s_u,s_v,s_w\}} + c_2$, $c_1 \neq 0$. According to the definition of sample triplet order, a_i and a_j must have the same sample triplet order. Without loss of generality, assume the common order is $s_u \prec s_v \prec s_w$, then we have

$$\frac{P_{a_i,s_v} - P_{a_i,s_u}}{P_{a_i,s_w} - P_{a_i,s_u}} = \frac{c_1 * (P_{a_j,s_v} - P_{a_j,s_u})}{c_1 * (P_{a_j,s_w} - P_{a_j,s_u})} = \frac{P_{a_j,s_v} - P_{a_j,s_u}}{P_{a_j,s_w} - P_{a_j,\, s_u}}.$$

Therefore, $S^2Score(\{a_i, a_j\}, \{s_u, s_v, s_w\}) = 0$.

(2) A zero S^2Score (Def. 1) indicates a_i and a_j have the same triplet order. Without loss of generality, assume $s_u \prec s_v \prec s_w$. Transform the S^2Score equation with this sample triplet order and then we can find a constant

$$c_1 = \frac{P_{a_i,s_v} - P_{a_i,s_u}}{P_{a_j,s_v} - P_{a_j,s_u}} = \frac{P_{a_i,s_w} - P_{a_i,s_u}}{P_{a_j,s_w} - P_{a_j,s_u}}.$$

Similarly, we can find a constant c_2 equal to

$$P_{a_i,s_u} - c_1 * P_{a_j,s_u} = P_{a_i,s_v} - c_1 * P_{a_j,s_v} = P_{a_i,s_w} - c_1 * P_{a_j,s_w}.$$

Thus, a_i and a_j are perfectly shifting-and-scaling correlated on the sample triplet. □

We can also define the S^{2}Score for a set of samples with more than 3 members:

DEFINITION 2. *($S^2Score(A,S)$)*
Assume A is a set of entities and S is a set of samples with more than three members. $S^2Score(A,S) = max\{S^2Score(G, \{s_u, s_v, s_w\})|\forall\{s_u, s_v, s_w\} \subseteq S\}$. □

Based on the above definition, given two entities a_i and a_j which are perfectly shifting-and-scaling correlated with respect to S, S^{2}Score($\{a_i, a_j\}$,S) is always zero. This can be from Lemma 3.1 by observing that a_i and a_j should also be perfectly shifting-and-scaling correlated with respect to all possible triplets from S

3.3 Gene × Sample and Time × Sample Clusters

We will next look at how we will use the concept of shifting and scaling patterns to define what are gene × sample and time × sample clusters by instantiating A either to be a set of genes G or a set of time points T' .

A gene × sample cluster occur in a time slice (Figure 4c shows three examples) and is essential a set of genes, G, which are shifting-and-scaling correlated with each other across a set of samples S based on the S^{2}Score.

EXAMPLE 1. *Figure 1 shows three perfect shifting-and-scaling patterns, [20, 10, 50], [5, 0, 20] and [30, 40, 0], across $\{s_1, s_4, s_2\}$. The common sample triplet order of the three genes is $s_2 \prec s_1 \prec s_4$. And, $S^2Score(\{g_1, g_2, g_3\}, \{s_1, s_4, s_2\}) = 0$, since $\forall g_i$, $i = 1, 2$ and 3, $\frac{P_{g_i,\, s_1} - P_{g_i,\, s_2}}{P_{g_i,\, s_4} - P_{g_i,\, s_2}} = 0.75$.* □

In real applications however, there are usually no perfect shifting-and-scaling patterns due to noise, experimental errors, etc. Therefore, a coherence threshold ϵ is used to control the degree of coherence.

DEFINITION 3. *(gene × sample cluster w.r.t. ϵ)*
Assume G is a group of genes and S is a set of samples where $|S| \geq 3$, then $C = G \times S$ is a coherent cluster w.r.t. ϵ, if and only if $S^2Score(G, S) \leq \epsilon$. □

Correspondingly, we can also define a time × sample cluster in a gene slice (Figure 4a shows three examples):

[3] Alternatively, it is also possible to limit $|A| \geq 2$ and allow $|S| \geq 2$. We will not elaborate on this symmetric nature of the definition further due to lack of space

DEFINITION 4. *(time × sample cluster w.r.t. ϵ)*
Assume T' is set of time points and S is a set of samples where $|S| \geq 3$, then $C = T' \times S$ is a coherent cluster w.r.t. ϵ, if and only if $S^2Score(T', S) \leq \epsilon$. □

In the case where $\epsilon = 0$ in the two definitions above, we will refer to the corresponding clusters as perfect clusters. In addition to the coherence threshold, ϵ, we also perform a regulation test with threshold γ in LagMiner. This is because biologists are generally interested in gene expression patterns of significant regulations. Cheng and Church [7] state that the utmost important goal of gene expression data analysis is to find a set of genes showing strikingly similar *up-regulation* and *down-regulation* under a set of conditions. We apply γ in our cluster model to ensure a significant expression level difference between any two samples.

3.4 Gene × Time Cluster

Unlike the sample dimension, the time dimension has a fixed order of time points and there may exist some time lags between correlated patterns. Comparatively, the coherence requirement over time series data is more relaxed than that on sample data. One can define the coherence of gene × time cluster as either correlated or anti-correlated in expression levels. Accordingly, we extend gene × sample cluster model w.r.t. ϵ to gene × time cluster model by **relaxing** ϵ to 1.0 .

We further propose the concepts of **time lag coefficient** and **time-lagged order** (**TORD**) to capture expression patterns with arbitrary time lags.

Let t_0 be the starting time point of the time dimension and $T_i = (t_{m_i}, t_{m_i+1}, ..., t_{m_i+l-1})$ be the corresponding time periods for g_i. Then we say the *time lag coefficient* of T_i is m_i i.e. m_i is the starting time slice of T_i. We further define the *time-lagged order (TORD)* of gene g_i on time sequence $T_i = (t_{m_i}, t_{m_i+1}, ..., t_{m_i+l-1})$ as $k_1 \prec k_2 \prec ... \prec k_l$, where $k_i \in \{0, 1, ..., l-1\}$, if either
(1) $P_{g_i,t_{m_i+k1}} < P_{g_i,t_{m_i+k_2}} < ..... < P_{g_i,t_{m_i+k_l}}$ and $k_1 < k_l$, or
(2) $P_{g_i,t_{m_i+k_1}} > P_{g_i,t_{m_i+k_2}} > ..... > P_{g_i,t_{m_i+k_l}}$ and $k_1 < k_l$.

EXAMPLE 2. *As an example of time lag coefficient and TORD. For sample slice s_1 of Figure 4b, the time lag coefficients of g_1, g_2 and g_3 are 0, 1 and 2 respectively. The three shifting-and-scaling patterns of time lags have a common TORD: $0 \prec 1 \prec 2$.* □

We denote gene expression patterns with time lags as $G.T$, where $G = \{g_i\}$ is a group of correlated genes, $T = \{T_i | g_i \in G\}$ is the set of corresponding time periods. We define $G.T$ as a coherent *gene × time* cluster when there is a common TORD among T. The formal definition is given below.

DEFINITION 5. *(gene × time Cluster)*
Assume $G = \{g_i\}$ is a group of genes, $T = \{T_i | g_i \in G\}$ is the set of time periods, where $T_i = (t_{m_i}, t_{m_i+1}, ..., t_{m_i+l-1})$, $|T_i| = l \geq 3$ and m_i is the lag coefficient of g_i. We say $C = G \cdot T$ is a coherent gene × time cluster iff there is a common TORD among T. □

3.5 Gene × Sample × Time Cluster

A time-lagged 3D cluster can be considered as a *combination* of gene × sample and gene × time clusters. Figure 4 shows an ideal one whose time slice/gene slice corresponds to a perfect gene × sample cluster, and sample slice corresponds to a gene × time cluster. The following observation further helps optimize our S^2D^3 Cluster model.

OBSERVATION 3.1. *(gene/time slice combination)*
Assume $G = \{g_i | \forall i\} = \{g_1, g_2, ..., g_n\}$) is a group of genes, S is a set of samples, $T = \{T_i | g_i \in G\}$ is a set of time periods, where $T_i = (t_{m_i}, t_{m_i+1}, ..., t_{m_i+l-1})$ and $l \geq 3$. Let $C = G \cdot T \times S$ be a perfect time-lagged 3D cluster where:

1)each time slice corresponds to a perfect gene × sample

2)each gene slice corresponds to a perfect time × sample cluster

3)each sample slice corresponds to a gene × time cluster

Let $A = \{\widehat{g_i t_{m_i}} | g_i \in G\} = \{g_1 t_{m_1}, g_1 t_{m_1+1}, ..., g_1 t_{m_1+l-1}, g_2 t_{m_2}, g_2 t_{m_2+1}, ..., g_2 t_{m_2+l-1}, ..., g_n t_{m_n}, g_n t_{m_n+1}, ..., g_n t_{m_n+l-1}\}$. We claim that $S^2Score(A, S) = 0$ i.e. $A \times S$ is a perfect cluster.

Proof:
(1) Transitivity of Coherence Property*: If P_{g_i,S,t_u} and P_{g_j,S,t_v} are perfectly shifting-and-scaling correlated, then ($P_{g_i,S,t_u} = c_1 * P_{g_j,S,t_v} + c_2$). Similarly if P_{g_j,S,t_v} and P_{g_k,S,t_w} are also shifting-and-scaling correlated ($P_{g_j,S,t_v} = c_1' * P_{g_k,S,t_w} + c_2'$), then P_{g_i,S,t_u} and P_{g_k,S,t_w} are also perfectly shifting-and-scaling correlated. This is because $P_{g_i,S,t_p} = c_1 * c_1' * P_{g_k,S,t_r} + c_1 * c_2' + c_2$.*

(2) Intersection between Gene and Time Slice*:At each gene slice g_i, $\widehat{g_i t_{m_i}} \times S$ is a perfect gene × sample cluster. That is, $P_{g_i,S,t_{m_i}}$, $P_{g_i,S,t_{m_i+1}}$, ..., and $P_{g_i,S,t_{m_i+l-1}}$ are all perfectly shifting-and-scaling correlated with each other. Likewise, in a time slice $\{g_i t_{m_i} | \forall g_i \in G\}$, $P_{g_1,S,t_{m_1}}$, $P_{g_2,S,t_{m_2}}$, ..., and $P_{g_n,S,t_{m_n}}$ are also perfectly shifting-and-scaling correlated. It can be seen that given any gene slice and time slice, there should be an intersection (refer to Figure 4), $P(g_i, S, g_i t_{m_i})$ which is common in both the slices. Thus any members in the gene or time slice will be perfectly shifting-and-scaling correlated with each other based on transitivity of their relationship with $P(g_i, S, g_i t_{m_i})$*

□

EXAMPLE 3. *The running example of time-lagged 3D cluster in Figure 4 is an example of a perfect, time-lagged 3D cluster: $\{\widehat{g_1 t_0}, \widehat{g_2 t_1}, \widehat{g_3 t_2}\}_3 \times \{s_1, s_2, s_3\}$. Each of its gene slice and time slice corresponds to a perfect 2D shifting-and-scaling cluster. Each of its sample slice, meanwhile, corresponds to a gene × time cluster with a common TORD: $0 \prec 1 \prec 2$ at sample slice s_1 and $1 \prec 0 \prec 2$ at sample slices s_2 and s_3. In the example, $\{\widehat{g_i t_{m_i}} | g_i \in G\}$ correspond to $\{g_1 t_0, g_1 t_1, g_1 t_2, g_2 t_1, g_2 t_2, g_2 t_3, g_3 t_2, g_1 t_0, g_1 t_0\}$ based on Figure 4b. By crossing this set with S in our running example, each of the sample s_1, s_2 and s_3 will be associate with $\{\widehat{g_i t_{m_i}} | g_i \in G\}$ based on their corresponding gene slices in Figure 4b as follow:*

s_1 associated with a_1 = {10, 15, 20, 30, 40, 50, 20, 15, 10}

s_2 associated with a_2 = {20, 35, 10, 50, 80, 30, 10, -5, 20}

s_3 associated with a_3 = {5, 5, 25, 20, 20, 60, 25, 25, 5}

Let $A = \{a_1, a_2, a_3\}$, then we can see that $S^2Score(A,S)=0$. □

As can be seen, Observation 3.1 allows us to linearize each sample slice (which is a 2D array) into a 1D array such that comparing coherency among the 1D arrays is equivalent to comparing coherency among the 2D arrays. Later on in Section 4.1, our enumeration of sample triplets can thus be conducted on a set of 1D arrays making the algorithm a 2D mining algorithm that explore along all the three dimensions in the gene-sample-time space. Compare to the TRICLUSTER [27] algorithm which search in the sample and time slices separately, this simultaneous exploration along all three dimensions (for a small distance) give much more room for pruning.

In additional, Observation 3.1 also allows us to simplify the definition of the S^2D^3 cluster model with respect to a coherence threshold ϵ and time periods length l and given below. Note that the first condition in the definition is essentially a result of Observation 3.1.

DEFINITION 6. *(S^2D^3 Cluster w.r.t. ϵ and l)*
Assume $G = \{g_i\}$ is a group of genes, S is a set of samples, and $T = \{T_i | g_i \in G\}$ is a set of time periods, where $Ti = (t_{m_i}, t_{m_i+1}, ..., t_{m_i+l-1})$ and m_i is the lag coefficient of g_i. Then $C = G \cdot T \times S$ is a time-lagged S^2D^3 cluster iff (1) $S^2Score(\{\widehat{g_i t_{m_i}} | g_i \in G\}, S) \leq \epsilon$; and (2) each sample slice $s_n \in S$ is a gene $\cdot$ time cluster: there exists a common $TORD$ over T_i among G. □

3.6 Model Comparison

Compared with previous shifting-and-scaling model [25], our cluster model is much more consistent and flexible, as summarized below.

Justifiability: As we discussed in Section 2, regcluster model is biased towards cluster whose baseline sample-pair has a large expression difference. Hence, regcluster favors gene group 1 in Figure 3. With our gene × sample cluster model, both gene groups in Figure 3 have the same degree of coherence: $S^2Score(Group_1, \{s_1, s_5, s_3\}) = S^2Score(Group_2, \{s_1, s_5, s_3\}) = 0.222$.

Compactness: Because of the use of baseline sample-pair, the submatrix of a regcluster may not be coherent w.r.t. ϵ any more. However, any submatrix of our 2D or 3D cluster is still coherent w.r.t. ϵ.

Convenience: The coherence threshold in the regcluster model and previous work [24, 27] is difficult to specify in advance, varying from 0 to ∞. Comparatively, our cluster model ensures the S^2Score on any sample triplet is always within $[0, 1]$, 0 indicating perfect shifting-and-scaling correlation.

To filter out the most interesting S^2D^3 Clusters, besides the coherence threshold ϵ, we also apply the regulation threshold γ (similar to that proposed in [25]), the minimum gene number threshold $MinG$, the minimum sample set size $MinS$, and the minimum time periods length $MinT$ (similar to those proposed in [27]) in our algorithm.

4. ALGORITHM LAGMINER

Unlike previous 3D clustering algorithms [14, 27], our S^2D^3 Cluster discovery algorithm, LagMiner, explores on all three dimensions *simultaneously*. Meanwhile, it takes into account all the three issues that are not addressed by the previous algorithms: (1) the *time-lagged* correlation, (2) the *continuity* of time periods data and (3) the general *shifting-and-scaling* correlation.

To implement the simultaneous 3D exploration, we propose a concept of **maximum set of gene-time periods (MSGTP)** over a sample set S. In this way, any maximal 3D clusters with time lags can be recovered from one or more MSGTPs with ease.

DEFINITION 7. *(MSGTP w.r.t. ϵ and γ)*

Input: $D = G_0 \times S_0 \times T_0$: 3D dataset, $MinG$: minimum number of genes, $MinS$: minimum number of samples, $MinT$: minimum time sequence length, γ: regulation threshold and ϵ: coherence threshold.
Output: all validated optimal S^2D^3 Clusters.

1. $MS_0 = MSGTP\_const(D, MinG, MinS, MinT, \epsilon, \gamma)$;
2. **for** each $M = GTPset \times S_m$ in MS_0 ($|S_m| = 3$) **do**
   $SSet\_extend(M)$ and records to MS;
   Prune the non-maximal M in MS;
3. $CS = maxclus\_recover(MS)$;

Output the optimal maximal S^2D^3 Cluster in CS.
Subroutine: $MSGTP\_const()$.

1. **for** each sample triplet $S_m = \{s_u, s_v, s_w\}$ enumerated in depth-first order **do**
   Check all the gene-time periods of length $MinT$;
   Record validated GTP w.r.t. ϵ and γ to $Candi_{S_m}$.
2. Apply $MinS$ compactness property pruning iteratively:
   if candidate GTP conflicts with property 1
   then remove GTP;
   if sample s conflicts with property 2
   then remove GTP for any S_m that contains s;
   if sample pair $s_u - s_v$ conflicts with property 3
   then remove GTP for any S_m that contains $s_u - s_v$;
3. **for** each $Candi_{S_m}$ in depth-first order **do**
   Sort its GTP first by triplet order, second by TORD and last by MinScore;
   Slide window w.r.t. constraints ϵ and $MinG$;
   Record validated MSGTPs into MS_0;
4. return MS_0.

Subroutine: $SSet\_extend(M = GTPset \times S)$.

1. **if** $|S| \geq MinS$ **then** insert M to MS.
2. **for** each extension candidate s' **do**
   Intersect M with one MSGTP on each $\{s_u, s_v, s'\}$ that $s_u, s_v \in S$
   while applying $TORD$, $MinG$ and redundancy pruning;
   if not prunable **then**
   $S' = S \cup \{s'\}$;
   $GTPset'$ = intersected gene-time periods;
   $M_{new} = GTPset' \times S'$;
   if $|S| \geq MinS$ **then**
   Insert M_{new} into MS;
   $SSet\_extend(M_{new})$.

Subroutine: $maxclus\_recover(MS)$.

1. Gene uniqueness processing on MS;
2. Sort $MS = \{M_n = gthset \times S\}$ first by $|S|$ next by $|gthset$ (both descending);
3. **for** each $M_n \in MS$ in sorted order **do**
   Extend the time sequences of M_n in both positive and negative directions;
   Prune subset MSGTP of extended M'_n;
   Record M'_n in CS;
4. return CS;

Figure 5: LagMiner

Assume S is a set of samples, ϵ is the coherence threshold, γ is the regulation threshold ($\gamma > 0$), then the maximum set of gene-time periods (MSGTP) of length l over S that satisfy the constrains of coherence and regulation is a maximal set of gene-time periods $\{\widehat{g_i t_{m_i}}\}_l$ that satisfies the following conditions:

(1)[4] *Coherence at gene/time slices: $\forall$ sample triplet $\{s_u, s_v, s_w\} \subseteq S$, $S^2Score(\{\widehat{g_i t_{m_i}}\}, \{s_u, s_v, s_w\}) \leq \epsilon$;*

(2) Coherence at sample slices: At any sample slice $s_u \in S$, $\forall \widehat{g_i t_{m_i}}$, $TORD(\widehat{g_i t_{m_i}})$ is the same.

(3) Regulation significance: $\forall s_u, s_v \in S$, $\forall g_i t_{m_i+x}$, $0 \leq x < l$, we have either $P_{g_i,s_u,t_{m_i+x}} - P_{g_i,s_v,t_{m_i+x}} \geq P_{g_i,s_v,t_{m_i+x}} \cdot \gamma$ or $P_{g_i,s_v,t_{m_i+x}} - P_{g_i,s_u,t_{m_i+x}} \geq P_{g_i,s_u,t_{m_i+x}} \cdot \gamma$. □

Figure 5 gives an algorithmic description of LagMiner. Details will be explained in the following sections. Generally, LagMiner performs clustering in three basic steps.

(1) *MSGTP Construction* ($MSGTP\_const()$): We construct all possible MSGTPs for each sample triplet $\{s_u, s_v, s_w\} \subseteq S$ w.r.t. ϵ, γ and $MinT$.

(2) *Sample Set Extension* ($SSet\_extend()$): We extend the sample sets of initial MSGTPs to meet $MinS$ constraint.

(3) *Maximal Refinement* ($maxclus\_recover()$): Since we only consider time periods of length $MinT$ at the first two steps, at Step (3), we extend the cluster in the time dimension and then output the **optimal 3D clusters**. This implies that larger weights are assigned to gene dimension and sample dimension. Assume we have two 3D clusters, $C_1 = G_1 \cdot T_1 \times S_1$ and $C_2 = G_2 \cdot T_2 \times S_2$, if $G_1 \subset G_2$, $S_1 \subseteq S_2$, then C_1 is not optimal given C_2.

In many ways, LagMiner adopt the concept of shotgun and assembly approach for human genome sequencing [18, 11] which first break the human DNA into many small fragment and then assemble them into longer sequences by joining fragments which have matching suffix and prefix. In this case, the fragments in LagMiner are the MSGTPs which will be combined to form the final clusters. However, MSGTPs differ from sequence fragments since it is multi-dimensional.

Figure 6 give an overview of the cluster assembly process of LagMiner for a $S^2D^3Cluster$. Initial, basic unit of MSGTP are constructed in the first phase with 3 samples, $MinT$ time units and more than $MinG$ genes. In the second phase, extension are then done along the sample dimension to construct MSGTPs consisting of more than $MinS$ samples. Finally in the last phase, extension are done along the time dimension to construct the final optimal clusters which can have more than $MinT$ time slices.

Compare to TRICLUSTER [27] which search in the sample and time space separately, LagMiner first explore along all three dimensions (for a small distance) in order to construct the basic unit of MSGTP. This allow us to first prunes the search space along all three dimensions. Searching in sample and time space separately will inevitably results in an extremely large number of candidate 2D clusters. As we will show later, LagerMiner is much more efficiently than TRICLUSTER because of this.

[4]This condition is one reason why the smallest unit of triplet is used instead of quadruple or larger set of samples. Testing all combination of samples can be expensive for larger sample set.

4.1 MSGTP Construction

In subroutine $MSGTP\_const()$, we firstly discover all candidate gene-time subseris (GTP) of length $MinT$ for a MSGTP over a sample triplet $\{s_i, s_j, s_k\}$. Then we perform $MinS$ compactness checking and 1D window sliding to obtain the initial MSGTPs on sample triplets.

GTP Discovery: for each sample triplet $S_m = \{s_u, s_v, s_w\}$ and $\forall \widehat{g_i t_{m_i}} \in G \cdot T$ of time sequence length $MinT$, we carry out

(1) regulation test: We test whether $\forall g_i t_{m_i+x}$, $0 \leq x < MinT$, the expression level difference across any two samples in S_m is significant w.r.t. γ, and

(2) coherence test: We test whether $S^2Score(\widehat{g_i t_{m_i}}, S_m) \leq \epsilon$.

If $\widehat{g_i t_{m_i}}$ satisfies these two conditions, then it is a candidate GTP for S_m and will be stored in the candidate list, $Candi_{S_m}$, together with its associated S^2Score range [$MinScore$, $MaxScore$] (Definition 1).

Compactness Pruning: Recall that in a valid 3D cluster, there should be at least $MinS$ correlated samples. Owing to the compact property of our S^2D^3 Cluster model (Section 3.6), each candidate GTP must satisfy the $MinS$ **compactness property** below, assuming $MinS > 3$:

(1) Each candidate GTP should cover at least C^3_{MinS} different sample triplets.

(2) Among the sample triplets that GTP covers, each sample must occur at least C^2_{MinS-1} times.

(3) Among the sample triplets that GTP covers, each sample-pair must occur at least C^1_{MinS-2} times.

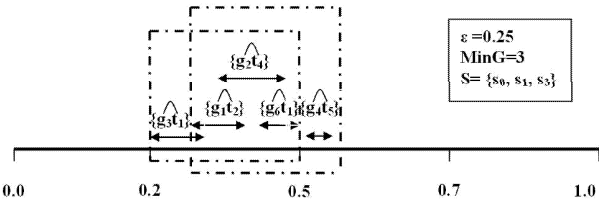
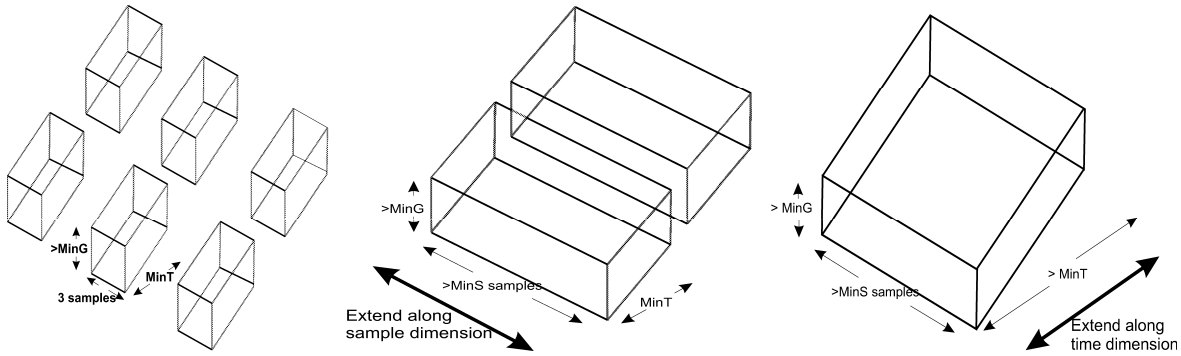


Figure 7: 1D Window Sliding

1D Window Sliding: For each sample triplet S_m, we sort candidate GTP in $Candi_{S_m}$ first by sample triplet order, next by TORD and last by $MinScore$. We slide a window of length ϵ over the sorted GTPs of the same sample triplet order and the same TORD. If the number of genes within the window is above $MinG$, then an initial MSGTP w.r.t. $MinG$, $MinS$, $MinT$, ϵ and γ is found. This process is termed 1D window sliding since each candidate GTP occupies a space of [MinScore, MaxScore], as opposed to a single value in previous studies.

EXAMPLE 4. *For example, with respect to our running example in Figure 4, Figure 7 shows two MSGTPs on sample*



(a) After MSGTP Construction (b) After Sample Extension (c) Optimal 3D Cluster Discovery

Figure 6: Assembling MSGTPs to form S^2D^3 Cluster

triplet $\{s_0, s_1, s_3\}$ *when* $\epsilon = 0.25$ *and* $MinG = 3$*:* $\{\widehat{g_3t_1}, \widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\} \times \{s_0, s_1, s_3\}$ *and* $\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}, \widehat{g_4t_5}\} \times \{s_0, s_1, s_3\}$. □

4.2 Sample Set Extension

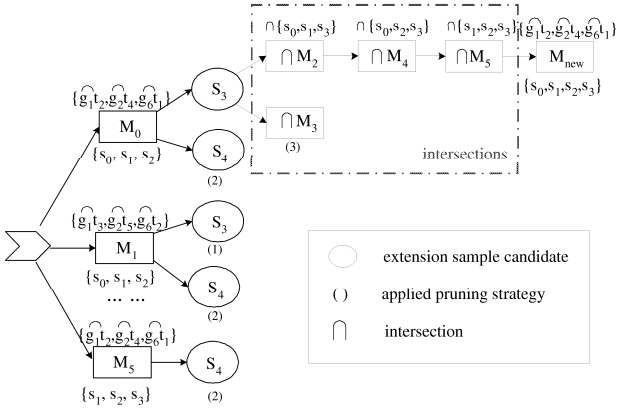


Figure 8: Sample Set Extension

We note that the sample sets of the initial MSGTPs obtained in Step 1 only consist of three samples. In Subroutine *SSet_extend*(), we extend the sample sets of initial MSGTPs to meet the $MinS$ constraint. After extension, the maximal MSGTPs are stored in MS.

To avoid redundancy, we impose an ORD order on the samples: $s_0 \prec s_1 \prec ... \prec s_n$. We extend the sample set in depth-first order of ORD. Assume the sample set of the current MSGTP, M, is S, then the extension candidates of M are those samples of a lower ORD rank than samples in S: $\{s' | \forall s_u \in S, s_u \prec s'\}$. To extend M's sample set from S to $S \cup \{s'\}$, we recursively intersect M with one MSGTP on each newly involved sample triplet, $\{s_u, s_v, s' | s_u, s_v \in S, s_u \prec s' \text{ and } s_v \prec s'\}$.

The following pruning strategies are applied in *SSet_extend*():

(1) TORD Pruning: if the gene-time periods of the two MSGTPs for intersection have different TORD on any common sample, the subsequent search stops (Please refer to Definition 6).

(2) $MinG$ Pruning: if the number of genes after intersection falls below $MinG$, the subsequent search stops.

(3) Redundancy Pruning: if the new MSGTP after intersection has been explored already, the subsequent search stops.

Example 5. *Let's look at an example of sample set extension. Table 2 shows the initial MSGTPs constructed on sample triplets over dataset D. There are five samples in D:* s_0, s_1, ..., *and* s_4. *Assume* $MinS = 3$*, the sample set extension procedure proceeds as Figure 8 shows. For* M_0 *on sample triplet* $\{s_0, s_1, s_2\}$*, the extension candidates are* s_3 *and* s_4*. Staring from* s_3*,* M_0 *intersects with one MSGTP of three sample triplets, each consists of* s_3 *and two samples from* $\{s_0, s_1, s_2\}$*:* $\{s_0, s_1, s_3\}$*,* $\{s_0, s_2, s_3\}$ *and* $\{s_1, s_2, s_3\}$*. Specifically,* M_0 *intersects with* M_2*,* M_4 *and* M_5*. We obtain* $M_{new} = \{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\} \times \{s_0, s_1, s_2, s_3\}$*. Since no MSGTP exists on sample triplet containing* s_4*, we can no longer extend* M_{new}*. The SSet_extend() subroutine then goes back to the intersection of* M_0 *with* M_3*. Since the intersection is redundant with* M_{new}*, subsequent search is pruned.*

The next MSGTP to extend is M_1 *on sample triplet* $\{s_0, s_1, s_2\}$*.* M_1*'s extension candidates are* s_3 *and* s_4*. When extending with* s_3*, no MSGTP on sample triplet* $\{s_0, s_1, s_3\}$ *has the same TORD as* M_1 *on samples* s_0 *and* s_1*, thus the extension with* s_3 *stops (TORD pruning strategy).* M_1 *is also not extendable with* s_4*, since* s_4 *is not available in any sample triplets.*

The only extension candidate for M_2*,* M_3*,* M_4 *and* M_5 *is* s_4 *which is not available in any sample triplets, therefore the subroutine stops.*

Given M_{new}*,* M_0*,* M_4 *and* M_5 *are no longer maximal and thus are pruned. The maximal MSGTPs in MS are* M_{new}*,* M_1*,* M_2 *and* M_3. □

4.3 Optimal 3D Cluster Recovery

In Subroutine *maxclus_recover*(), we search and output optimal S^2D^3 Cluster in the following steps.

First, we check whether each MSGTP, M_n, contains gene-time periods of the same gene. We split M_n if so. After that, each M_n is a candidate S^2D^3 Cluster that satisfies the $MinG$, $MinS$, ϵ, γ and $MinT$ constraints.

Second, we sort the processed MSGTPs first in descending order of sample set size, next in descending order of gene set size, and last in enumeration order. In this way, an optimal 3D clusters can always be discovered before the non-optimal ones.

MSGTP	TORD	$\{s_0, s_1, s_2\}$	MSGTP	TORD	$\{s_0, s_1, s_3\}$
M_0	012,012,102	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}$	M_2	012,012,021	$\{\widehat{g_3t_1}, \widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}$
M_1	102,102,012	$\{\widehat{g_1t_3}, \widehat{g_2t_5}, \widehat{g_6t_2}\}$	M_3	012,012,021	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}, \widehat{g_4t_5}\}$
MSGTP	TORD	$\{s_0, s_2, s_3\}$	MSGTP	TORD	$\{s_1, s_2, s_3\}$
M_4	012,102,021	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}$	M_5	012,102,021	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}$

Table 2: Constructed MSGTPs in MS_0 when $MinS = 3$ and $MinT = 3$

Order	MSGTP	$GTPSet \times SampleSet$
0	M_{new}	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}_3 \times \{s_0, s_1, s_2, s_3\}$
1	M_2	$\{\widehat{g_3t_1}, \widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}_3 \times \{s_0, s_1, s_3\}$
2	M_3	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}, \widehat{g_4t_5}\}_3 \times \{s_0, s_1, s_3\}$
3	M_1	$\{\widehat{g_1t_3}, \widehat{g_2t_5}, \widehat{g_6t_2}\}_3 \times \{s_0, s_1, s_2\}$

Table 3: Sorted MSGTPs

Order	MSGTP	$GTPSet \times SampleSet$
0	M_{new}	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}_3 \times \{s_0, s_1, s_2, s_3\}$
1	M_2	$\{\widehat{g_3t_1}, \widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}_3 \times \{s_0, s_1, s_3\}$
2	M_3	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}, \widehat{g_4t_5}\}_3 \times \{s_0, s_1, s_3\}$
3	M_1'	$\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\}_4 \times \{s_0, s_1, s_2\}$

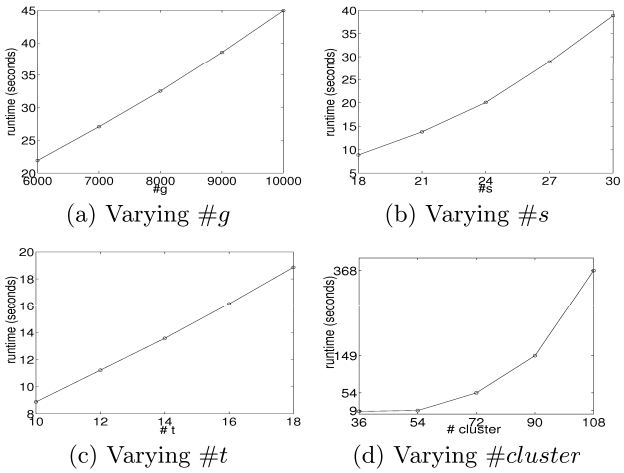
Table 4: Final Optimal S^2D^3 Cluster

Finally, we recover the optimal S^2D^3 Cluster in sorted order. Recall that the MSGTPs found so far consist of gene time periods of length $MinT$. For each MSGTP in sorted order, $M_n = \{\widehat{g_it_{m_i}}\}_{MinT} \times S \in MS$, where the subscript before '$\times$' indicates time periods length, we check whether the time periods of M_n can be extended in the negative or positive direction synchronously. For each $\widehat{g_it_{m_i}}$, the negative direction refers to the time points before t_{m_i}: t_{m_i-1}, t_{m_i-2}, etc. Similarly, the positive direction refers to the time points after $t_{m_i+MinT-1}$: t_{m_i+MinT}, $t_{m_i+MinT+1}$, etc. For M_n, if there are p negative extensions and q positive extensions, then the new MSGTP is $M_n' = \{\widehat{g_it_{m_i-p}}\}_{MinT+p+q} \times S$. During time periods extension, the TORD of new time periods are examined. We mark all the non-optimal MSGTP given M_n', record M_n' in CS, come to the next promising MSGTP and repeat the time periods extension procedure recursively.

EXAMPLE 6. *For example, assuming that maxclus_recover() subroutine sorts the MSGTPs after sample set extension as shown in Table 3. Obviously, the time periods in M_{new}, M_2 and M_3 are not extensible in either direction, and the three MSGTPs cover no subset MSGTPs. However, when it comes to M_1, one negative extension of time periods is made, since $\{\widehat{g_1t_2}, \widehat{g_2t_4}, \widehat{g_6t_1}\} \times \{s_0, s_1, s_2\}$ is subsumed by M_{new}. The final optimal S^2D^3 Cluster output are those listed in Table 4.* □

4.4 Time Complexity

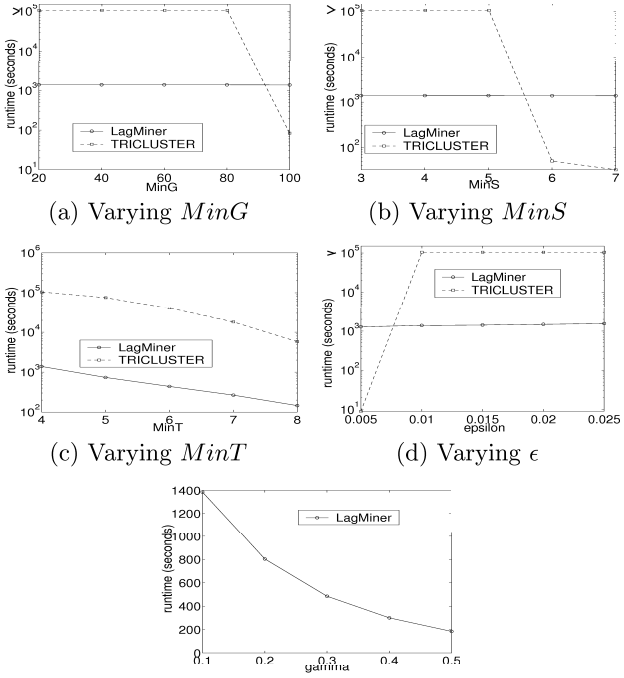
In terms of time complexity, the bottleneck of LagMiner is the MSGTP construction subroutine and sample set extension subroutine. For the 3D dataset $D = G_0 \times S_0 \times T_0$, where $|G_0| = N$, $|S_0| = M$ and $|T_0| = L$, assume there are k validated MSGTPs discovered in MSGTP construction. Then, there are $N*(L-MinT+1)$ gene-time periods to be checked for each sample triplet. There are $M*(M-1)*(M-2)$ sample triplets. Therefore the overall time complexity of MSGTP construction is $O(N \cdot L \cdot M^3)$. Without applying any pruning strategies, the time complexity of sample set extension is exponential with k. By applying $MinG$, $MinS$, $MinT$, ϵ and γ constraints, the efficiency of LagMiner is improved significantly.



(a) Varying $\#g$ (b) Varying $\#s$

(c) Varying $\#t$ (d) Varying $\#cluster$

Figure 9: Scalability Evaluation



(a) Varying $MinG$ (b) Varying $MinS$

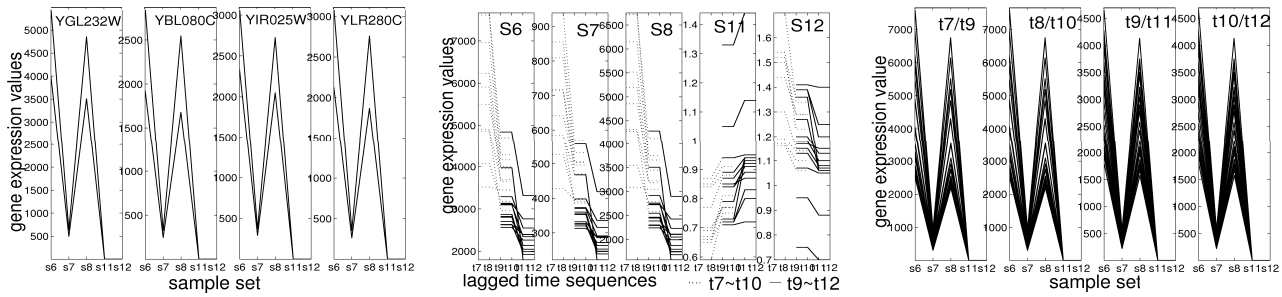
(c) Varying $MinT$ (d) Varying ϵ

(e) Varying γ

Figure 10: Sensitivity Evaluation

5. EXPERIMENTAL EVALUATION

We conduct a series of experiments on both synthetic data and real-life data to compare the performance of LagMiner against TRICLUSTER (discovering pure scaling pattern in 3D space). We perform the scalability test on synthetic dataset by varying the number of genes, the number of samples, the time series length and the number of embedded clusters. We test the sensitivity of LagMiner and



(a) Four Gene Slices (b) Five Sample Slices (c) Four Time Slices

Figure 11: A S^2D^3 Cluster

TRICLUSTER on the real-life yeast 3D dataset [5] that was previously used to evaluate [27]. This 3D dataset contains the expression values of 7679 genes of yeast from 13 samples across 14 consecutive time points. Finally, we evaluate the biological significance of the S^2D^3 clusters discovered in the 3D yeast dataset with the yeast gene ontology term finder.

By setting the time dimension size to one, LagMiner is adapted for 2D *gene* × *sample* clustering. When compared with regcluster (discovering shifting-and-scaling pattern in 2D space) using both synthetic dataset and the real-life yeast 2D dataset in [25], LagMiner turned out to find clusters with more coherency. For space limitation, we will not report the details here.

All experiments are done on a 1.0-GHz Dell PC with 3GB memory running Redhat Linux.

5.1 Scalability

We evaluate the scalability of LagMiner on synthetic datasets obtained with a data generator. The data generator has four input parameters: the number of genes ($\#g$), the number of samples ($\#s$), the number of involved time points ($\#t$), and the number of embedded perfect clusters ($\#cluster$). The default parameters of the data generator are $\#g = 3000$, $\#s = 18$, $\#t = 10$ and $\#cluster = 54$. Only one parameter is varied at each time. The synthetic dataset is initialized with random values ranging from 0 to 10. Then arbitrary time-lagged shifting-and-scaling clusters are embedded into the data. For the embedded perfect clusters, the average sample set size is 6, the time periods length is 5, and the average number of genes is equal to $0.015 * \#g$.

Figure 9 shows the runtime of LagMiner when varying $\#g$, $\#s$, $\#t$ and $\#cluster$ with $MinG = 0.015*\#g$, $MinS = 6$, $MinT = 5$, $\epsilon = 0.01$ and $\gamma = 0.0001$. Each time, LagMiner successfully identifies all the perfect time-lagged 3D clusters. The runtime of LagMiner is approximately linear w.r.t. $\#g$ and $\#t$, since the number of candidate gene-time periods are proportional to the product of $\#g$ and $\#t$. As the number of sample triplets increases sharply when increasing $\#s$ and $\#cluster$, runtime of *LagMiner* shows worse scalability with $\#s$ and $\#cluster$. The experimental results are consistent with our time complexity analysis in Section 4.4.

[5]Ideally, we will like to test the algorithms on more real life datasets. Unfortunately, due to the large number of gene expression experiments that must be done to generate a 3D dataset and relatively expensive price associated with each experiment, the yeast dataset is the single, most comprehensive one we can find. This is however expected to change in the next few years as the price of gene chip decrease.

5.2 Sensitivity

We test the sensitivity of *LagMiner* w.r.t. $MinG$, $MinS$, $MinT$, ϵ and γ, as opposed to TRICLUSTER, on the real-life 3D dataset in [27]. The default parameter settings of LagMiner are $MinG = 80$, $MinS = 5$, $MinT = 4$, $\gamma = 0.1$ and $\epsilon = 0.01$. One parameter is varied at each time.

Figure 10 illustrates the runtime of LagMiner and TRICLUSTER when varying parameters $MinG$, $MinS$, $MinT$, ϵ and γ respectively. As can be seen, LagMiner is in general much more efficient than TRICLUSTER, though TRICLUSTER only finds pure-scaling patterns *without* time lags, a specific case of the much more general shifting-and-scaling patterns with arbitrary time lags that LagMiner discovers.

Figures 10 (a), (b) and (d) show that LagMiner is quite insensitive to $MinG$, $MinS$ and ϵ, compared with TRICLUSTER. The runtime of LagMiner slightly decreases from 1399 seconds to 1376 seconds when $MinG$ varies from 20 to 100, smoothly drops from 1387 seconds to 1376 seconds when $MinS$ increases from 3 to 7, and gently increases from 1286 seconds to 1550 seconds when raising ϵ. On the contrary, TRICLUSTER is so sensitive to $MinG$, $MinS$ and ϵ that it cannot finish running within 24 hours when $MinG \leq 80$ or $MinS \leq 5$ or $\epsilon \geq 0.01$. Its runtime suddenly drops below 100 seconds when $MinG = 100$ or $MinS = 6$ or $\epsilon = 0.005$.

The extremely high sensitivity of TRICLUSTER to $MinG$, $MinS$ and ϵ is due to the costly enumeration of excessive 2-dimensional *gene* × *sample* clusters at each time slice. A large proportion of enumerated *gene* × *sample* candidate patterns actually will be pruned with the later $MinT$ constraint, especially when $MinG$, $MinS$, or ϵ constraints are loose. Comparatively, LagMiner avoids unnecessary exploration by performing $MinG$, $MinS$ and $MinT$ pruning simultaneous.

Figure 10 (c) shows that the runtime of TRICLUSTER is approximately two orders of magnitude of that of LagMiner when varying $MinT$ from 4 to 8. Figure 10 (e) shows that the runtime of LagMiner is anti-exponential to γ.

5.3 Effectiveness

LagMiner is able to identify time-lagged correlation. For instance, LagMiner identified 29 S^2D^3 Clusters from the real 3D dataset within 20 minutes when $MinG = 20$, $MinS = 5$, $MinT = 4$, $\gamma = 0.1$ and $\epsilon = 0.02$. We adopt the yeast genome ontology term finder (`http://db.yeastgenome.org/cgi-bin/GO/goTermFinder`) to evaluate the biological significance of the 29 S^2D^3 Clusters. Out of the 29 clusters, 21 clusters turn out to be highly significant w.r.t. one or more GO terms in associated biological processes, cellular

components or gene functions at significant level p=0.005.

One S^2D^3 Cluster of time lag 2 among the 29 3D clusters, C'', is illustrated in Figure 11. C'' has a p-value of 0.00284 in association with mitochondrion inheritance. It consists of 24 gene members, 5 samples and local time periods of length 4. The correlated sample set is $\{s_6, s_7, s_8, s_{11}, s_{12}\}$. In C'', there is a time lag of two between its 11 gene members occupying time periods (t_7, t_8, t_9, t_{10}) and its 13 gene members occupying time periods (t_9, t_{10}, t_{11}, t_{12}). Under the same parameter setting, TRICLUSTER cannot finish running in three days, thus we do not evaluate its findings here [6].

Figure 11 demonstrates the coherence patterns of C'' at gene/sample/time slices. The four gene slices in Figure 11 (a) show the coherent $time - sample$ patterns projected on the 6th, 12th, 18th and 24th gene respectively. For instance, the leftmost gene slice in Figure 11 (a) consists of shifting-and-scaling time curves, t_7-t_{10}, of gene YGL232W across samples s_6, s_7, s_8, s_{11} and s_{12}. The coherent gene curves across time sequences on the five sample slices are shown in Figure 11 (b), in which a time lag of 2 can be observed. Figure 11 (c) demonstrates the coherent gene curves across samples in the 1st, 2nd, 3rd and 4th corresponding time point of correlated time periods. Note that the starting time point is either t_7 or t_9.

6. CONCLUSION

In this paper, we have addressed three issues not addressed by previous 3D clustering algorithms, (1) the *time-lagged* correlation, (2) the *continuity* of correlated time periods, and (3) the general *shifting-and-scaling* correlation, by proposing a novel shifting-and-scaling cluster model, S^2D^3 Cluster. Our cluster model can be generalized to 2-dimensional clustering as well and is much more consistent than the previous 2D shifting-and-scaling model [25].

We have designed algorithm LagMiner for efficient discovery of S^2D^3 Clusters. Unlike TRICLUSTER, which explores the $gene \times sample$ space and $time$ space separately, LagMiner gains much more efficiency by exploring the three dimensions simultaneously. Furthermore, our experimental results on both synthetic and real-life datasets show that: LagMiner is rather scalable for large datasets and robust to parameter settings. The yeast genome ontology term finder also indicates that the S^2D^3 clusters discovered by LagMiner is meaningful in biology.

7. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *ACM SIGMOD*, volume 29, 2000.

[2] C. C. Agrawal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *ACM SIGMOD*, 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD*, 1998.

[4] Z. Bar-Joseph, G. Gerber, D. K. Gifford, and T. S. Jaakkola. A new approach to analyzing gene expression time series data. In *RECOMB*, 2002.

[6]Even if TRICLUSTER is allowed to complete, it will not find the cluster that we shown here since as shown in Figure 2, TRICLUSTER does not capture the concept of time-lagged. Similarly, since the cluster we shown here exists only for 4 out of the 13 time slices, the SG/GS algorithm which make comparison of the genes/samples based on the full time series will not be able to capture it's semantics.

[5] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the ordering-preserving submatrix problem. In *Recomb*, 2002.

[6] C. Bohm, K. Kailing, P. Kroger, and A. Zimek. Computing clusters of correlation connected objects. In *ACM SIGMOD*, 2003.

[7] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology*, 2000.

[8] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analisis and display of genome-wide expression patterns. In *PNAS*, 1998.

[9] T. R. H. et.al. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[10] V. Filkov, S. Skiena, and J. Zhi. Analysis techniques for microarray time-series data. In *RECOMB*, 2001.

[11] A. M. Frieze, F. P. Preparata, and E. Upfal. Optimal reconstruction of a sequence from its probes. *Journal of Computational Biology*, 6(3/4), 1999.

[12] A. Hinneburg and D. A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB*, 1999.

[13] C. hung Cheng, A. W. chee Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *ACM SIGKDD*, 1996.

[14] D. Jiang, J. Pei, M. Ramanathan, C. Tang, and A. Zhang. Mining coherent gene clusters from gene-sample-time microarray data. In *ACM SIGKDD*, 2004.

[15] D. Jiang, J. Pei, and A. Zhang. Interactive exploration of coherent patterns in time-series gene expression data. In *ACM SIGKDD*, 2003.

[16] J. Liu and W. Wang. Op-cluster: Clustering by tendency in high dimensional space. In *ICDM*, 2003.

[17] J. Liu, W. Wang, and J. Yang. Gene ontology friendly biclustering of expression profiles. In *Computational Systems Bioinformatics*, 2004.

[18] F. P. Preparata, A. M. Frieze, and E. Upfal. On the power of universal bases in sequencing by hybridization. In *Proceedings of the Third Annual International Conference on Research in Computational Molecular Biology*, pages 295–301, 1999.

[19] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A monte carlo algorithm for fast projective clustering. In *ACM SIGMOD*, 2002.

[20] M. F. Ramoni, P. Sebastiani, and I. S. Kohane. Cluster analysis of gene expression dynamics. In *Proc. of the National Academy of Sciences*, 2002.

[21] W. A. Schmitt, R. M. Raab, and G. Stephanopoulos. Elucidation of gene interaction networks through time-lagged correlation analysis. In *Genome Research*, 2004.

[22] P. Tamayo, D. Slnim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. In *PNAS*, 1999.

[23] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. In *Nature Genetics*, volume 22, pages 281–285, 1999.

[24] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *ACM SIGMOD*, 2002.

[25] X. Xu, Y. Lu, A. K. Tung, and W. Wang. Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In *ICDE*, 2006.

[26] J. Yang, W. Wang, H. Wang, and P. Yu. δ-clusters: Capturing subspace correlation in a large data set. In *ICDE*, 2002.

[27] L. Zhao and M. J. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In *ACM SIGMOD*, 2005.