Heuristic Approaches for Time-Lagged Biclustering

Joana P. Gonçalves Sara C. Madeira

INESC-ID and Instituto Superior Técnico, Technical University of Lisbon, Portugal
{jpg,smadeira}@kdbio.inesc-id.pt

ABSTRACT

Identifying patterns in temporal data supports complex analyses in several domains, including stock markets (finance) and social interactions (social science). Clinical and biological applications, such as monitoring patient response to treatment or characterizing activity at the molecular level, are also of interest. In particular, researchers seek to gain insight into the dynamics of biological processes, and potential perturbations of these leading to disease, through the discovery of patterns in time series gene expression data. For many years, clustering has remained the standard technique to group genes exhibiting similar response profiles. However, clustering defines similarity across all time points, focusing on global patterns which tend to characterize rather broad and unspecific responses. It is widely believed that local patterns offer additional insight into the underlying intricate events leading to the overall observed behavior. Efficient biclustering algorithms have been devised for the discovery of temporally aligned local patterns in gene expression time series, but the extraction of time-lagged patterns remains a challenge due to the combinatorial explosion of pattern occurrence combinations when delays are considered. We present heuristic approaches enabling polynomial rather than exponential time solutions for the problem.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms, Problem Complexity**]: Nonnumerical problems—*pattern matching*; G.2.1 [**Discrete Mathematics**]: Combinatorics; I.5 [**Pattern Recognition**]; J.3 [**Computer Applications**]: Life sciences

General Terms

Algorithms, Performance

Keywords

biclustering, pattern recognition, time series, gene expression, temporal patterns

1. INTRODUCTION

Gene expression is a dynamic process, reflecting changes orchestrated by the underlying regulatory mechanisms involved in cellular control. Temporal gene expression profiling enables to monitor the responses of a large number of regulatory players over time and is recognized as a key strategy to gain insight into the intricate circuitry of gene regulation. Ultimately, the analysis of time series gene expression data is critical to advance our understanding of complex biological mechanisms involved in processes such as growth and development, disease susceptibility and progression, and response to treatment [2, 4]. When studying gene expression measured over time, local transcriptional patterns assume a major relevance, as genes are expected to behave coherently with different subsets of partners mostly within the time frames of the biological tasks in which they participate together. Biclustering is a suitable solution for the discovery of local patterns, but its general formulation has been shown NP-hard upon reduction to the maximum edge biclique problem [13]. Many general purpose biclustering algorithms have been proposed in the literature [9]. However, most are unsuitable for the analysis of time series data given that they disregard important temporal properties, such as time point dependency and biological process inherent time contiguity. Notably, in the case of temporal data, the assumption that biological processes last for a contiguous period of time motivates the discovery of local patterns spanning consecutive time points. This observation leads to a reasonable restriction in the search space of local patterns, enabling a linear time solution for the temporal biclustering problem [12]. Although the most efficient temporal biclustering algorithm to date and also effective in unraveling biologically meaningful biclusters in real data [10, 12], this approach finds only local patterns that are temporally aligned.

Patterns exhibiting time-lagged relationships among the expression profiles of different genes are an important aspect of gene regulation, as target genes are often activated with a certain time delay rather than simultaneously. Temporal programs of expression in which genes are activated one by one in a predefined order are well-known and can be generated by widespread network topologies, including regulatory cascades [1]. The identification of time-lagged patterns has been addressed before [8, 11], but the approaches put forward are hampered by the potential explosion of pattern combinations which makes exhaustive enumeration unfeasible. In this work, we discuss the exponential complexity of a solution for identifying and exhaustively reporting occurrences of time-lagged local patterns in temporal data [11]. We further propose heuristic approaches achieving a significant reduction of the result space. Finally, we show that the heuristic approaches lead to polynomial solutions which identify meaningful time-lagged patterns in real data.

2. METHODS

In this section, we first introduce some important concepts and describe the original CCC-Biclustering algorithm [12] devised to identify instances of local expression patterns occurring in the same time frame across a subset of temporal profiles. We then describe an extension of CCC-Biclustering to enable the identification of instances of local temporal expression patterns occurring at potentially different starting time points. Finally, we describe interesting heuristic approaches that prune the result space and thus avoid the potentially exponential number of maximal CCC-Biclusters with time-lags that can emerge with exhaustive enumeration.

2.1 Time series gene expression matrix

Let M' be an expression matrix defined by a set of genes (rows), G, and a set of time points (columns), T, where M'_{ij} represents the expression of gene i at time point j. Real values in M' are discretized to a set of symbols, Σ, representing activation levels in a new matrix M. Any discretization is eligible. A popular approach in the analysis of time series data [8, 12] consists in converting matrix M' into M, where $M_{ij} \in \Sigma$ reflects the trend between the expression states of gene i in time points j and $j+1$, respectively. In this case, we use alphabet $\Sigma = \{D, N, U\}$, where D, N, and U denote *down-regulation*, *no-change* and *up-regulation* (Fig. 1).

	$T1$	$T2$	$T3$	$T4$	$T5$
$G1$	0.07	0.73	-0.54	0.45	0.25
$G2$	-0.34	0.46	-0.38	0.76	-0.44
$G3$	0.22	0.17	-0.11	0.44	-0.11
$G4$	0.70	0.71	-0.41	0.33	0.35
$G5$	0.70	0.17	0.70	- 0.33	0.75

(a) Original matrix.

	$T1$	$T2$	$T3$	$T4$	$T5$
$G1$	N	U	D	U	N
$G2$	D	U	D	U	D
$G3$	N	N	N	U	N
$G4$	U	U	D	U	U
$G5$	U	D	U	D	U

(b) Discretized matrix.

Figure 1: Time series expression matrices. Illustrative (a) time series expression matrix, together with its (b) discretized version, using alphabet $\{D, N, U\}$.

2.2 Finding Temporally Aligned Local Patterns

We briefly describe CCC-Biclustering, an efficient algorithm for mining temporally aligned local patterns in a time series gene expression matrix.

2.2.1 (Maximal) CCC-Bicluster

A CCC-Bicluster, M_{IJ}, is defined as a subset of genes $I \subseteq G$ and a subset of contiguous time points $J \subseteq T$ such that $M_{ij} = M_{lj}, \forall i, l \in I$ and $\forall j \in J$. This means that every gene in I shares the same expression pattern spanning the time points in J. A CCC-Bicluster is maximal (Fig. 3) if adding rows to I violates the coherence of the expression pattern (row-maximality) and adding a symbol to the beginning or end of the expression pattern induces changes in I (left-/right-maximality). CCC-Biclusters pertaining a single row are biologically uninteresting and are thus discarded.

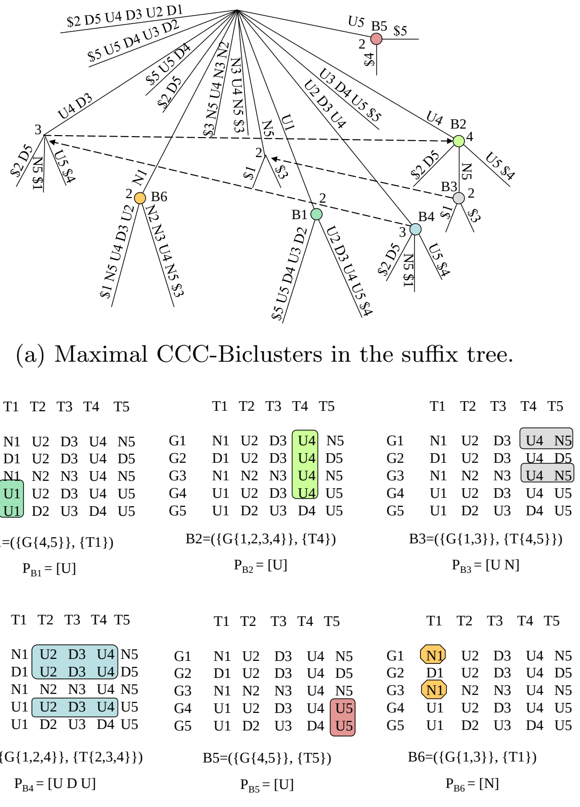
	$T1$	$T2$	$T3$	$T4$	$T5$
$G1$	N1	U2	D3	U4	N5
$G2$	D1	U2	D3	U4	D5
$G3$	N1	N2	N3	U4	N5
$G4$	U1	U2	D3	U4	U5
$G5$	U1	D2	U3	D4	U5

(a) Transformed matrix.

$G1$	N1	U2	D3	U4	N5	\$1
$G2$	D1	U2	D3	U4	D5	\$2
$G3$	N1	N2	N3	U4	N5	\$3
$G4$	U1	U2	D3	U4	U5	\$4
$G5$	U1	D2	U3	D4	U5	\$5

(b) Strings for suffix tree.

Figure 2: Transformed matrix and strings used in suffix tree construction: (a) discretized matrix from Fig. 1(b) after alphabet transformation; (b) strings obtained from matrix (a) and used to build the tree.



(a) Maximal CCC-Biclusters in the suffix tree.

(b) Maximal CCC-Biclusters in the transformed matrix.

Figure 3: Maximal CCC-Biclusters identified in the transformed matrix from Fig. 2(a). These are shown in: (a) the suffix tree built for the strings in Fig. 2(b), and (b) the matrix from Fig. 2(a).

2.2.2 CCC-Biclustering

To find all maximal CCC-Biclusters, CCC-Biclustering first performs a simple alphabet transformation that appends the column number to each symbol in the discretized matrix (Fig. 2). This transformation ensures that patterns match only when both the symbol and time point match, and therefore when the patterns are temporally aligned. Additional adaptations can be introduced, namely to allow support for missing values [10]. Regarding the rows of the transformed matrix as strings, denoting gene expression profiles, a generalized suffix tree $\mathcal{T}$ [7] is then built to match the common local patterns in the profiles and identify the maximal CCC-Biclusters. Such identification relies on the following relationship between maximal CCC-Biclusters and nodes in $\mathcal{T}$: every right and row-maximal CCC-Bicluster with at least two rows corresponds to one internal node in $\mathcal{T}$ and every internal node in $\mathcal{T}$ corresponds to one right and row-maximal CCC-Bicluster with at least two rows. Right- and row-maximality of the CCC-Bicluster identified by an internal node v are guaranteed by generalized suffix tree construction. Left-maximality of a CCC-Bicluster identified by an internal node v is guaranteed when either v has no incoming suffix links [7] or it has incoming suffix links only from nodes for which the number of leaves in their subtree is equal to the number of leaves in the subtree rooted at v. CCC-Biclustering uses efficient string matching techniques to find these nodes and report all maximal CCC-Biclusters in time linear on the size of the expression matrix.

2.3 Finding Time-Lagged Local Patterns

We address the goal of finding occurrences of the same pattern which might not necessarily be temporally aligned. We shall focus on the general case of unbounded time lags, although alternative definitions are also possible. For completeness, in this section we present a sample matrix with some missing values (Fig. 4).

2.3.1 (Maximal) CCC-Biclusters with time lags

We first introduce key concepts and definitions of time lag, starting pattern and CCC-Bicluster with time lags.

Definition 1 (Time Lag). *Absolute difference between the left-most time points of two distinct occurrences of a given pattern.*

Definition 2 (Starting Pattern). *The left-most occurrence amongst all occurrences of a given pattern.*

Definition 3 (CCC-Bicluster with time lags). *A CCC-Bicluster with time lags M_{IJ} is a CCC-Bicluster such that $M_{ij} = M_{lJ_l}$, for all rows $i, l \in I$ and contiguous columns $j_l \in J_l$, where J_l is the set of contiguous columns corresponding to a single occurrence of the pattern in row l and such that $j_l = j + lag_l$ and lag_l is the time lag between the occurrence of the pattern in row l and the starting pattern.*

Definition 4 (Maximal CCC-Bicluster time lags). *A CCC-Bicluster with time lags M_{IJ} is maximal if no rows can be added to I and no contiguous columns can be added to any J_l, for all $l \in I$, while maintaining the coherence property in Definition 3. A CCC-Bicluster is maximal if it is row-maximal, left-maximal and right-maximal.*

In an unbounded time lag setting, the range of possible time lags is artificially bounded by the number of time points in the time series gene expression matrix, thus $[0, |T| - 1]$. We further note that the pattern denoted by a given CCC-Bicluster with unbounded time lags M_{IJ} may occur multiple times in the expression profile of any gene in I. However, only one occurrence per gene is considered in the definition and computation of M_{IJ}.

2.3.2 Maximal CCC-Bicluster node identification

We propose and describe below the three steps of an efficient algorithm enabling the identification of all nodes carrying maximal CCC-Biclusters with unbounded time lags in a time series gene expression matrix.

	$T1$	$T2$	$T3$	$T4$	$T5$
$G1$	_	U	D	U	N
$G2$	D	U	_	U	_
$G3$	_	N	_	U	N
$G4$	U	_	D	U	U
$G5$	U	_	U	D	U

(a) Discretized matrix.

$G1$	U	D	U	N	\$1
$G2$	D	U	\$2		
	U	\$2			
$G3$	N	\$3			
	U	N	\$3		
$G4$	U	\$4			
	D	U	U	\$4	
$G5$	U	\$5			
	U	D	U	\$5	

(b) Set of strings.

Figure 4: Discretized matrix and strings for use in CCC-Biclustering with unbounded time lags: (a) illustrative discretized matrix with missing values; (b) strings obtained from matrix (a) and used in the suffix tree for CCC-Biclustering with time lags.

Step 1 - Matrix preprocessing and set of strings.

When focusing on local temporal patterns potentially occurring with a time lag, we want any resembling patterns to match regardless of their starting point. In this context, alphabet transformation becomes unnecessary and is therefore not performed (Fig. 4). Consider that each row of the discretized matrix is regarded as a string, corresponding to a temporal gene expression profile. We further deal with missing values as follows. The string is split into multiple substrings, taking the symbol denoting a missing value as the separator character. We append to each substring the terminator characters denoting the row it originated from.

Step 2 - Pattern matching (suffix tree construction).

Similar to the original CCC-Biclustering algorithm, the extended version allowing for unbounded time lags also builds a generalized suffix tree $\mathcal{T}$ to find similar variations/values across consecutive time points in the gene profiles, here termed temporal local patterns, leading to the identification of maximal CCC-Biclusters with unbounded time lags.

Step 3 - Identification of promising (maximal) nodes.

We briefly introduce the relationship between internal nodes in $\mathcal{T}$ and maximal CCC-Biclusters with unbounded time lags in the discretized matrix, based on which the identification algorithm is constructed (Fig. 5). It can be shown that an internal node v in $\mathcal{T}$ identifies at least one maximal CCC-Bicluster with (unbounded) time lags if, in addition to the conditions necessary for a node to identify a maximal CCC-Bicluster (outlined in section 2.2.2), the number of different genes (rows) in the subtree rooted at v is at least 2, as set out in the following definition:

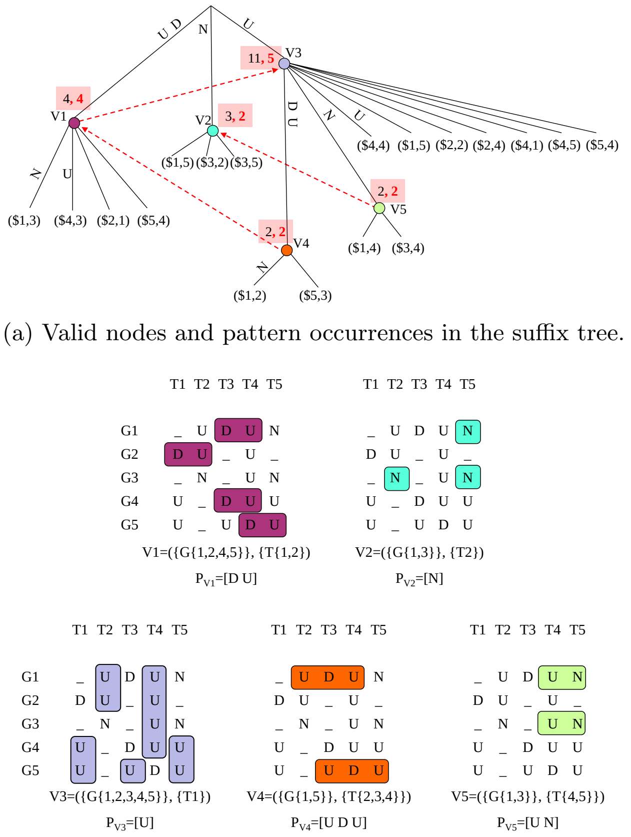
Definition 5 (MaxNode with Time Lags). *An internal node v of $\mathcal{T}$ is a MaxNode with time lags iff it satisfies one of the following conditions:*

- *It does not have incoming suffix links;*
- *It has incoming suffix links only from nodes u_i such that, for each u_i, $L(u_i) < L(v)$;*

together with the following condition:

- *The number of distinct genes in the subtree rooted at v is at least two, $G(v) \geq 2$.*

We motivate the necessity of adding the last condition as follows. Without the alphabet transformation, a given



(a) Valid nodes and pattern occurrences in the suffix tree.

	T1	T2	T3	T4	T5
G1	_	U	D	U	N
G2	D	U	_	U	_
G3	_	N	_	U	N
G4	U	_	D	U	U
G5	U	_	U	D	U

V1=({G{1,2,4,5}}, {T{1,2})

P_{V1}=[D U]

	T1	T2	T3	T4	T5
G1	_	U	D	U	N
G2	D	U	_	U	_
G3	_	N	_	U	N
G4	U	_	D	U	U
G5	U	_	U	D	U

V2=({G{1,3}}, {T2})

P_{V2}=[N]

	T1	T2	T3	T4	T5
G1	_	U	D	U	N
G2	D	U	_	U	_
G3	_	N	_	U	N
G4	U	_	D	U	U
G5	U	_	U	D	U

V3=({G{1,2,3,4,5}}, {T1})

P_{V3}=[U]

	T1	T2	T3	T4	T5
G1	_	U	D	U	N
G2	D	U	_	U	_
G3	_	N	_	U	N
G4	U	_	D	U	U
G5	U	_	U	D	U

V4=({G{1,5}}, {T{2,3,4}})

P_{V4}=[U D U]

	T1	T2	T3	T4	T5
G1	_	U	D	U	N
G2	D	U	_	U	_
G3	_	N	_	U	N
G4	U	_	D	U	U
G5	U	_	U	D	U

V5=({G{1,3}}, {T{4,5}})

P_{V5}=[U N]

(b) Pattern occurrences per valid node in the matrix.

Figure 5: Promising internal nodes and corresponding pattern occurrences in CCC-Biclustering with unbounded time lags: (a) valid internal nodes in the suffix tree built for the set of strings in Fig. 4; (b) occurrences of the pattern denoted by each valid internal node in the discretized matrix (a).

expression profile (row) can match with itself if the same pattern occurs at different time points in the profile. As a result, in the time-lagged setting there can be multiple leaves associated with the same gene (row) under a given internal node v in $\mathcal{T}$. Therefore, we can no longer assume that the branches (guaranteed to be at least two) descending from an internal node belong to different genes and automatically satisfy the quorum, as in the temporally aligned CCC-Biclustering case. Alternatively, we calculate and test the real number of different genes $G(v)$ represented in the leaves in the subtree rooted at v. CCC-Biclustering with time lags is based on the identification of all *MaxNodes with Time Lags*. The relationship between these nodes and the maximal CCC-Bicluster with unbounded time lags they identify is specified by Theorem 1, here presented without proof:

THEOREM 1. *Every maximal CCC-Bicluster with unbounded time lags and at least two genes (rows) can be identified using an internal node in the generalized suffix tree $\mathcal{T}$ that satisfies Definition 5, and each of these internal nodes identifies at least one maximal CCC-Bicluster with unbounded time lags and at least two genes (rows).*

Based on Theorem 1, all internal nodes of interest v in $\mathcal{T}$ can be identified using a limited number of properties which are straightforward to compute, namely the number of leaves and distinct genes in the subtree rooted at v (Fig. 5).

2.3.3 *Complexity of node identification*

Constructing the generalized suffix tree $\mathcal{T}$ for the set of strings can be done in $O(|G||T|)$ time. Since there are $O(|G||T|)$ nodes in $\mathcal{T}$, computing the number of leaves and different genes (rows) under every node v in $\mathcal{T}$ also takes $O(|G||T|)$ time. Likewise, identifying all the internal nodes corresponding to at least one maximal CCC-Bicluster with unbounded time lags and at least two genes (rows) can be performed in $O(|G||T|)$ time. Consequently, the algorithm is able to find all nodes containing maximal CCC-Biclusters with unbounded time lags in $O(|G||T|)$ time.

2.4 Reporting Time-Lagged Local Patterns

In the original version of CCC-Biclustering, dealing with temporally aligned patterns, there could be at most a single occurrence of the pattern per gene. Since the number of maximal CCC-Biclusters is $O(|G||T|)$ and the information to report per CCC-Bicluster is $O(|G|)$, the time necessary for reporting all maximal CCC-Biclusters would be $O(|G|^2|T|)$.

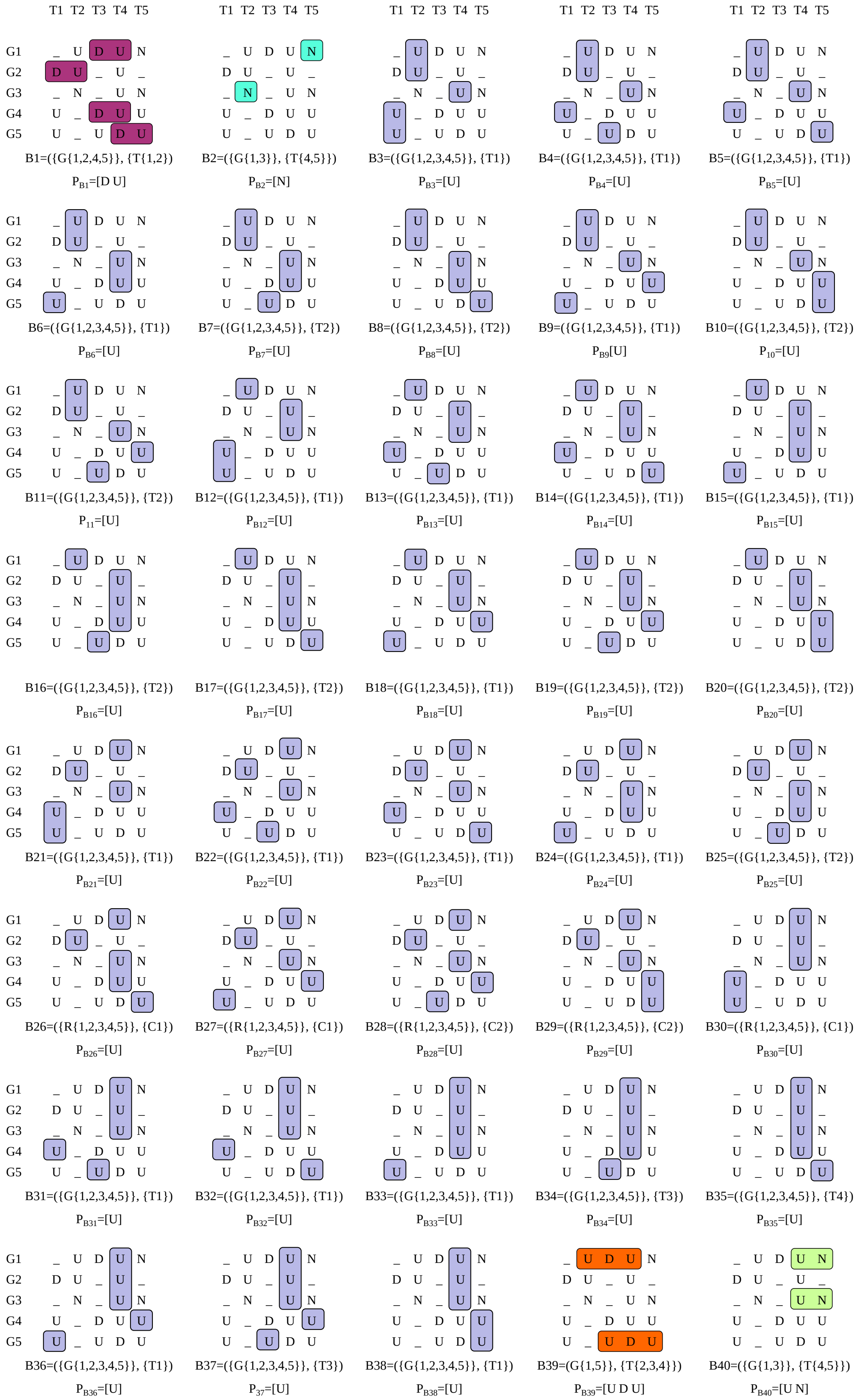
In the time-lagged setting, however, it is possible to find multiple occurrences of a pattern within the profile of a given gene at different starting time points. In such case, the number of maximal CCC-Biclusters with time lags corresponds to the number of all possible combinations of the existing occurrences of the pattern for the different genes. In the worst case, the number of maximal CCC-Biclusters with time lags that can be obtained for a given temporal pattern is therefore $O(|T|^{|G|})$. This means that the number of maximal CCC-Biclusters with time lags can grow exponentially with the number of genes, $|G|$, which makes exhaustive enumeration unfeasible in most cases of interest. We describe the complexity of exhaustively generating all possible CCC-Biclusters with time lags and introduce three alternatives for delivering the information under each valid internal node v in $\mathcal{T}$, such that the reporting step becomes tractable.

2.4.1 *Exhaustive lagged-CCC-Bicluster enumeration*

Exhaustive enumeration takes $O(|G|^2|T|^{1+|G|})$ time in the worst case, considering that: (i) there can be $O(|T|^{|G|})$ maximal CCC-Biclusters with time lags to report per each of the $O(|G||T|)$ potentially valid internal nodes in $\mathcal{T}$; (ii) for each maximal CCC-Bicluster with time lags, we have to report $O(|G|)$ genes together with the starting point of the pattern occurrences for each gene. This combinatorial explosion is practically challenging and possibly unfeasible. Take the rather small (5×5) illustrative matrix in this section as an example (Fig. 4 and 5), for which we obtain 40 maximal time-lagged CCC-Biclusters with two genes (rows) (Fig. 6).

2.4.2 *Lagged pattern occurrences under valid nodes*

If we disregard the definition of maximal CCC-Bicluster with time lags, we may report only the information contained in the leaves of the subtree rooted at each internal node marked as valid. We consider two options. First, we can report the gene identifier for each of the $O(|G|)$ genes in the subtree of each valid internal node in $\mathcal{T}$, without specifying the starting time points of all the $O(|T|)$ occurrences of the pattern. This type of reporting takes $O(|G|^2|T|)$.



B1=({G{1,2,4,5}}, {T{1,2}}) P_{B1}=[D U]

B2=({G{1,3}}, {T{4,5}}) P_{B2}=[N]

B3=({G{1,2,3,4,5}}, {T1}) P_{B3}=[U]

B4=({G{1,2,3,4,5}}, {T1}) P_{B4}=[U]

B5=({G{1,2,3,4,5}}, {T1}) P_{B5}=[U]

B6=({G{1,2,3,4,5}}, {T1}) P_{B6}=[U]

B7=({G{1,2,3,4,5}}, {T2}) P_{B7}=[U]

B8=({G{1,2,3,4,5}}, {T2}) P_{B8}=[U]

B9=({G{1,2,3,4,5}}, {T1}) P_{B9}[U]

B10=({G{1,2,3,4,5}}, {T2}) P_{10}=[U]

B11=({G{1,2,3,4,5}}, {T2}) P_{11}=[U]

B12=({G{1,2,3,4,5}}, {T1}) P_{B12}=[U]

B13=({G{1,2,3,4,5}}, {T1}) P_{B13}=[U]

B14=({G{1,2,3,4,5}}, {T1}) P_{B14}=[U]

B15=({G{1,2,3,4,5}}, {T1}) P_{B15}=[U]

B16=({G{1,2,3,4,5}}, {T2}) P_{B16}=[U]

B17=({G{1,2,3,4,5}}, {T2}) P_{B17}=[U]

B18=({G{1,2,3,4,5}}, {T1}) P_{B18}=[U]

B19=({G{1,2,3,4,5}}, {T2}) P_{B19}=[U]

B20=({G{1,2,3,4,5}}, {T2}) P_{B20}=[U]

B21=({G{1,2,3,4,5}}, {T1}) P_{B21}=[U]

B22=({G{1,2,3,4,5}}, {T1}) P_{B22}=[U]

B23=({G{1,2,3,4,5}}, {T1}) P_{B23}=[U]

B24=({G{1,2,3,4,5}}, {T1}) P_{B24}=[U]

B25=({G{1,2,3,4,5}}, {T2}) P_{B25}=[U]

B26=({R{1,2,3,4,5}}, {C1}) P_{B26}=[U]

B27=({R{1,2,3,4,5}}, {C1}) P_{B27}=[U]

B28=({R{1,2,3,4,5}}, {C2}) P_{B28}=[U]

B29=({R{1,2,3,4,5}}, {C2}) P_{B29}=[U]

B30=({R{1,2,3,4,5}}, {C1}) P_{B30}=[U]

B31=({G{1,2,3,4,5}}, {T1}) P_{B31}=[U]

B32=({G{1,2,3,4,5}}, {T1}) P_{B32}=[U]

B33=({G{1,2,3,4,5}}, {T1}) P_{B33}=[U]

B34=({G{1,2,3,4,5}}, {T3}) P_{B34}=[U]

B35=({G{1,2,3,4,5}}, {T4}) P_{B35}=[U]

B36=({G{1,2,3,4,5}}, {T1}) P_{B36}=[U]

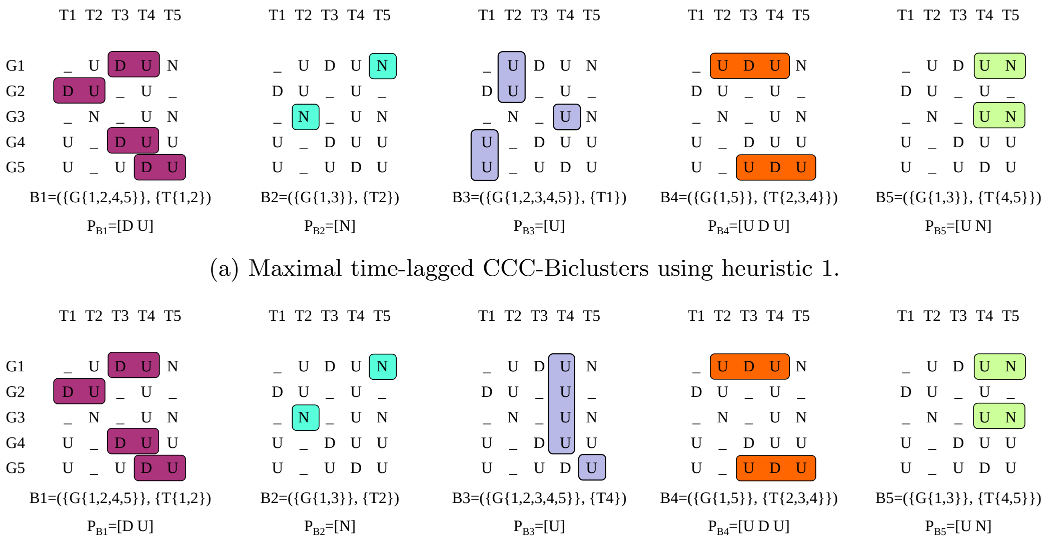
B37=({G{1,2,3,4,5}}, {T3}) P_{37}=[U]

B38=({G{1,2,3,4,5}}, {T1}) P_{B38}=[U]

B39=(G{1,5}}, {T{2,3,4}}) P_{B39}=[U D U]

B40=({G{1,3}}, {T{4,5}}) P_{B40}=[U N]

Figure 6: Exhaustive enumeration of maximal CCC-Biclusters with unbounded time lags.



(a) Maximal time-lagged CCC-Biclusters using heuristic 1.

(b) Maximal time-lagged CCC-Biclusters using heuristic 2.

Figure 7: Maximal CCC-Biclusters with unbounded time lags obtained using: (a) heuristic approach 1 - choosing the earliest (left-most) pattern occurrence per gene; and (b) heuristic approach 2 - choosing the earliest starting time point per gene larger than the most frequent starting time point among all occurrences of the pattern (if non existant, choose the closest starting point smaller than the most frequent one).

Second, we can report the gene identifier for each of the $O(|G|)$ genes in the subtree together with the starting time points of all occurrences of the pattern in each gene expression profile. Reporting this information takes $O(|G|^2|T|^2)$ and would allow us to draw a representation such as that in Fig. 5(b).

2.4.3 Single lagged CCC-Bicluster per valid node

A third alternative consists in reporting a single CCC-Bicluster per valid internal node in $\mathcal{T}$. In this case, the reporting step takes $O(|G|^2|T|)$ time to generate a maximal CCC-Bicluster with time lags containing $O(|G|)$ genes, together with a single time point per gene, for each of the $O(|G||T|)$ potentially valid internal nodes in $\mathcal{T}$. We present two distinct heuristic approaches for choosing a single maximal CCC-Bicluster with unbounded time lags per valid node in $\mathcal{T}$. Fig. 7 shows the output expected when applying each of these heuristic approaches to the matrix of Fig. 4.

Heuristic 1 - Left-most occurrence per gene.

Consider the left-most time point among the starting positions of all the occurrences of the pattern in the subtree rooted at a valid node v in $\mathcal{T}$ as the starting point of a cascade of delayed activations/inhibitions. In this context, it is reasonable to store for each gene only the starting time point of the left-most occurrence of the pattern.

Heuristic 2 - Most frequent starting time point.

Consider the most frequent time point among the starting positions of all the occurrences of the pattern in the subtree rooted at a valid internal node v in $\mathcal{T}$, denoted as p, as the starting time point of a cascade of delayed activations/inhibitions. In this context, we propose the following heuristic approach: for each gene, we always store the starting point of the first occurrence starting at or after p; whenever such occurrence does not exist, we store the starting point of the closest occurrence before p.

2.5 Ensuring Pattern Occurrence Maximality

Using only the information in the subtree rooted at a valid internal node v in $\mathcal{T}$ to compute the CCC-Biclusters with unbounded time lags may lead to combinations of genes (rows) and time points (columns) defining non-left maximal CCC-Biclusters with unbounded time lags. Take as an example the CCC-Bicluster with unbounded time lags in Fig. 5, defined by the occurrences of the pattern N at time point $T5$ in the expression profiles of genes $G1$ and $G3$ found under node $V2$. This CCC-Bicluster with unbounded time lags is non left-maximal, given that node $V5$ defines another CCC-Bicluster with unbounded time lags with a larger pattern UN occurring in the profiles of the same genes, $G1$ and $G3$, and spanning time points $T4$ and $T5$.We must therefore guarantee that the generated sets of starting points can effectively be used to compute maximal CCC-Biclusters with unbounded time lags, in all the cases where we need to report not only the genes but also the starting points of the occurrences of the pattern.

This is achieved by storing a bit array per internal node, $colors(v)$, such that each position of the array is set to 1 if the corresponding occurrence in the subtree rooted at node v can be used to compute row and column combinations leading to maximal CCC-Biclusters with unbounded time lags. Since the size of a colors array is $O(|G||T|)$ and there are $O(|G||T|)$ nodes in the suffix tree $\mathcal{T}$, computing all colors arrays takes $O(|G|^2|T|^2)$ time using a depth-first traversal of $\mathcal{T}$ and computing a bitwise OR of the bit arrays. An update procedure, also $O(|G|^2|T|^2)$ is further applied to $colors(u)$, whenever u has an outgoing suffix link pointing to internal node v, such that $L(u) < L(v)$ and $L(v) > G(v)$.

Table 1: CCC-Biclusters with unbounded time lags, obtained using heuristic 1 (choosing the left-most occurrence per gene under each valid internal node, section 2.4.3). We show only the 10 CCC-Biclusters with unbounded time lags, obtained using heuristic 1 and yielding the largest number of highly significant Gene Ontology terms. The CCC-Biclusters are sorted in decreasing order of the number of highly significant terms.

Rank	Bicluster ID	Pattern	#Genes	#Time points	Starting points	#Highly Sig. Terms	Best p-value
1	169	NNU	1088	4	$\{5', 10', 15', 20', 30'\}$	35	4.29×10^{-23}
2	184	$NNUU$	385	5	$\{5', 10', 15', 20'\}$	35	2.67×10^{-25}
3	224	$NNUUN$	292	6	$\{5', 10', 15'\}$	35	2.73×10^{-26}
4	441	$DNNUU$	156	6	$\{5', 10', 15'\}$	32	4.60×10^{-10}
5	166	NUU	633	4	$\{5', 10', 15', 20', 30'\}$	23	7.90×10^{-14}
6	176	$NUUN$	443	5	$\{5', 10', 15', 20'\}$	22	4.57×10^{-17}
7	383	DNN	1269	4	$\{5', 10', 15', 20', 30'\}$	22	8.32×10^{-27}
8	444	$DNNNU$	383	6	$\{5', 10', 15'\}$	22	1.87×10^{-15}
9	308	$NNNUUN$	161	7	$\{5', 10'\}$	21	7.25×10^{-14}
10	399	$DNNU$	360	5	$\{5', 10', 15', 20'\}$	20	8.78×10^{-9}

3. RESULTS

In order to show the usefulness of the proposed heuristic approaches, we applied CCC-Biclustering with time lags to a real expression time series dataset. We used data from Gasch et al. [5], concerning *Saccharomyces cerevisiae*'s response to heat shock. This data set comprises expression levels of 6142 genes measured at eight distinct time points (5', 10', 15', 20', 25', 30', 40', 60' and 80') for over an hour of exposure to 37°C. Similar to previous analyses of this kind [12], we first filtered all genes with missing values and normalized the expression levels per gene to zero mean and unit standard deviation. We also discretized the preprocessed matrix using a technique based on transitions between time points, proposed by Ji and Tan [8], as previously done in the application of CCC-Biclustering to real datasets [12].

3.1 Exhaustive vs heuristic approach

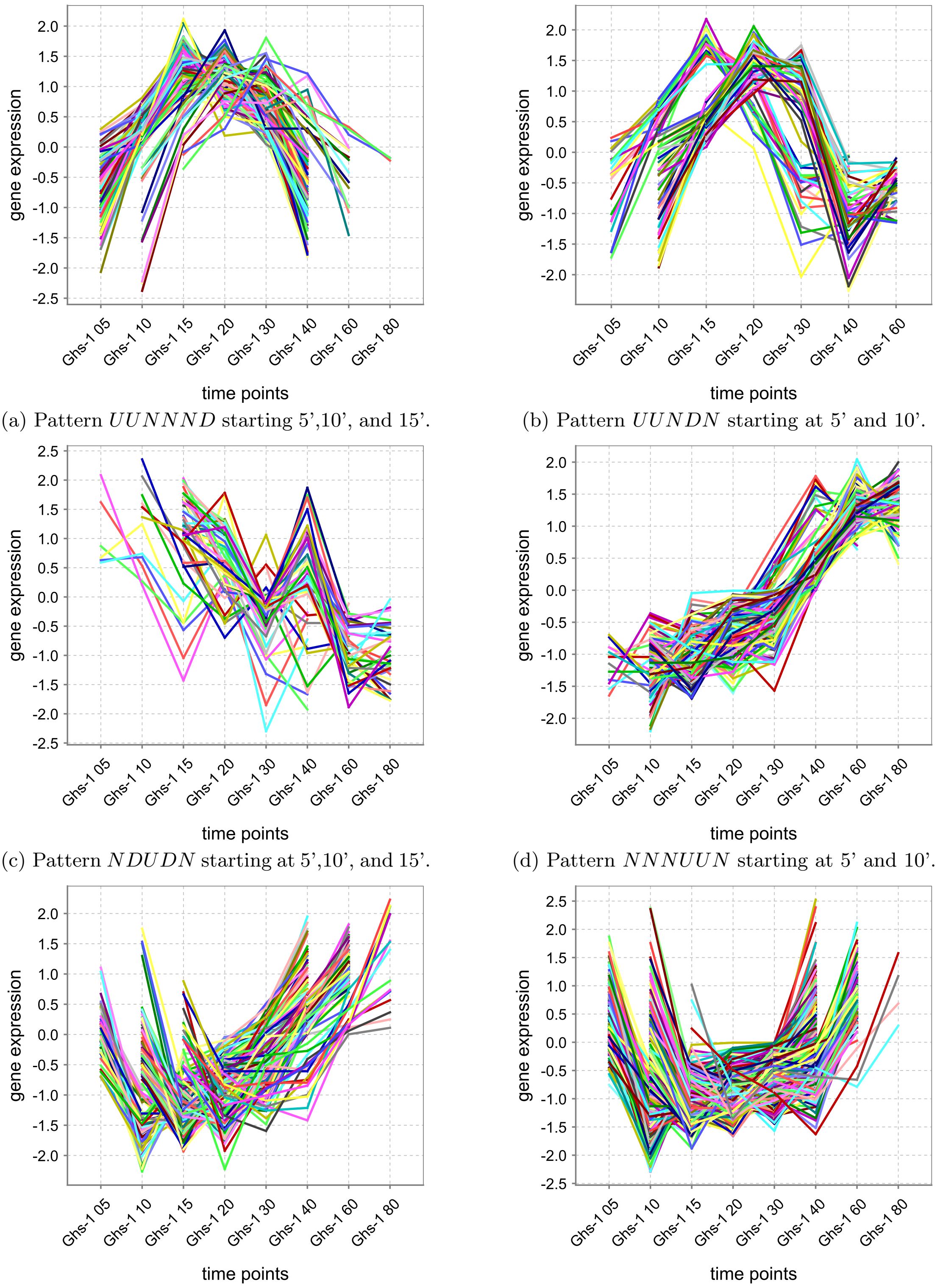
We applied CCC-Biclustering with unbounded time lags using both exhaustive enumeration (section 2.4.1), and the heuristic approach choosing the left-most starting point among all the pattern occurrences per gene (corresponding to heuristic 1, as described in section 2.4.3). Our tests were performed in a machine equipped with an Intel® i7-3632QM CPU and 8GB of RAM running Windows 8 64-bit, which can be considered a reasonably accessible user setting in a modern biology lab. To provide estimates in a real-world context, we integrated and tested the algorithms in BiGGEsTS, our freely available software providing methods for biclustering analysis of time series gene expression data [6]. The exhaustive version rapidly reached the amount of memory available in the system and did not finish computing due to an out of memory error (maximum heap size limit exceeded). For the same input, the heuristic version completed its computation in less than 5 minutes (including time for additional operations, such as bicluster counting and function analysis), with the software staying always under 1GB of memory usage.

3.2 Heuristic results

We provide a brief overview of the results obtained using CCC-Biclustering with unbounded time lags. In this case, we chose to report only one CCC-Bicluster with unbounded time lags per valid internal node according to heuristic 1 (section 2.4.3). This means reporting only the left-most occurrence of the temporal expression pattern per gene under each valid internal node ($MaxNodewithTimeLags$). We restricted the search to CCC-Biclusters with at least 10 genes and 4 time points, in order to focus on reasonably sized biclusters which are more interesting from the biological perspective. The algorithm delivered 569 CCC-Biclusters with unbounded time lags and at least 10 genes and 4 time points.

3.2.1 Statistical significance of functional annotations

We further assessed the agreement of the functional annotations of the genes in each CCC-Bicluster by computing the statistical overrepresentation of Gene Ontology (GO) terms [3]. Using the Ontologizer package [14], we calculated a p-value expressing the significance of the ratio of genes annotated with the term in the CCC-Bicluster against the ratio of genes annotated with the term in the population, based on the hypergeometric distribution. To the resulting p-value we applied a Bonferroni correction for multiple testing. For this purpose, we used the most recent ontology and annotation files downloaded from the Gene Ontology repository on May 20, 2013. For approximately 1/6 of the CCC-Biclusters, there was at least one highly significant function (corrected p-value < 0.01). Table 1 presents the 10 CCC-Biclusters with unbounded time lags yielding the largest number of highly statistically significant terms, sorted in decreasing order of this number. This can promote biclusters with shorter and therefore less specific patterns, aggregating genes annotated with a large number of GO terms, to the top. We note, however, that biclusters can alternatively be sorted according to a wide range of criteria, such as pattern length and pattern p-value, among others [6]. Each row corresponds to a given CCC-Bicluster and the different columns show the following information: 'Rank' denotes the CCC-Bicluster rank in the sorted list, 'Bicluster ID' denotes the sequential CCC-Bicluster ID attributed by the algorithm, '#Genes' denotes the number of genes in the CCC-Bicluster, '#Time points' denotes number of time points in the CCC-Bicluster, 'Starting points' contains the time points where occurrences of the CCC-Bicluster pattern start (without specifying in the profiles of which genes), and '#Highly Sig. Terms' denotes the number of highly significant terms. In the last column, 'Best p-value', the table shows the best corrected p-value calculated for a term annotated with the genes in each CCC-Bicluster. Note that, due to the discretization based on variations between time



(a) Pattern $UUNNND$ starting 5',10', and 15'.

(b) Pattern $UUNDN$ starting at 5' and 10'.

(c) Pattern $NDUDN$ starting at 5',10', and 15'.

(d) Pattern $NNNUUN$ starting at 5' and 10'.

(e) Pattern $DNNUU$ starting at 5',10', and 15'.

(f) Pattern $DNNNU$ starting at 5',10', and 15'.

Figure 8: Expression profiles of the genes in several CCC-Biclusters with unbounded time lags.

points, each symbol in the pattern denotes a variation between two consecutive time points and therefore the number of time points is always one unit larger than the number of symbols in the pattern.

3.2.2 Temporal expression cascades

We isolated the expression profiles of the genes in the CCC-Biclusters with unbounded time lags obtained using heuristic 1 (section 2.4.3). Within each CCC-Bicluster with time lags, we further restricted the expression of each group to the time points where the temporal pattern of the CCC-Bicluster with time lags has been identified by the algorithm (Fig. 8). In Fig. 8, we can clearly observe the coherence of expression profiles among the genes within a given CCC-Bicluster with time lags. The expression charts further expose the existence of temporal delays between the patterns of different sets of genes within a bicluster, confirming that the proposed heuristic approach effectively captures this well known phenomenon of temporal cascades in gene regulation. A more thorough biological analysis is out of the scope of this paper, but is envisioned as future work.

4. CONCLUSIONS

In this work, we discussed the complexity of identifying and reporting all maximal CCC-Biclusters with unbounded time lags in a time series gene expression matrix. Essentially, using exhaustive enumeration, the mentioned biclusters are obtained by identifying and computing all valid combinations of instances of local temporal patterns potentially occurring with a time lag in the expression profiles of multiple genes. Due to the combinatorial explosion of the result space, this computation is unfeasible in most cases of interest. For the same reason, available algorithms exhibit exponential time complexity [8, 11]. We addressed this issue by proposing heuristic approaches for time-lagged biclustering, which are biologically reasonable and enable a significant reduction in the result space. Any of the new strategies proposed in this document to report occurrences of time-lagged local patterns found in gene expression time series guarantees that the algorithm runs in polynomial, rather than exponential, time on the size of the input. Using real data, we further showed that the heuristic version was able to: (i) compute successfully in a regular desktop machine for an input that the exhaustive version could not handle without quickly exceeding the total amount of memory available in the system; (ii) retrieve interesting cascades of time-lagged patterns, whose genes were found to be functionally related. As future work, we aim to: fully integrate these heuristic approaches into our software BiGGEsTS [6] (only exhaustive enumeration has been made available to date); and test the different heuristics more extensively and on additional datasets, potentially relating to distinct domains of knowledge.

5. ACKNOWLEDGMENTS

Work supported by Fundação para a Ciência e a Tecnologia (FCT) through projects: PEst-OE/EEI/LA0021/2013 and PTDC/EIA-EIA/111239/2009 (Neuroclinomics).

6. REFERENCES

[1] U. Alon. Network motifs: theory and experimental approaches. *Nature reviews. Genetics*, 8(6):450–61, 2007.

[2] I. P. Androulakis, E. Yang, and R. R. Almon. Analysis of time-series gene expression data: methods, challenges, and opportunities. *Annual Review of Biomedical Engineering*, 9(February):205–28, 2007.

[3] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, and Others. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[4] Z. Bar-Joseph, A. Gitter, and I. Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564, July 2012.

[5] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11(12):4241–57, Dec. 2000.

[6] J. P. Gonçalves, S. C. Madeira, and A. L. Oliveira. BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. *BMC Research Notes*, 2:124, 2009.

[7] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[8] L. Ji and K.-L. Tan. Identifying time-lagged gene clusters using gene expression data. *Bioinformatics*, 21(4):509–16, Feb. 2005.

[9] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004.

[10] S. C. Madeira and A. L. Oliveira. A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. *Algorithms for Molecular Biology*, 4:8, 2009.

[11] S. C. Madeira and A. L. Oliveira. Efficient Biclustering Algorithms for Time Series Gene Expression Data Analysis. In *10th International Work-Conference on Artificial Neural Networks, IWANN 2009 Workshops*, pages 1013–1019, Salamanca, Spain, 2009. Springer.

[12] S. C. Madeira, M. C. Teixeira, I. Sa-Correia, and A. L. Oliveira. Identification of Regulatory Modules in Time Series Gene Expression Data Using a Linear Time Biclustering Algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1):153–165, 2010.

[13] R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651–654, Sept. 2003.

[14] P. N. Robinson, A. Wollstein, U. Böhme, and B. Beattie. Ontologizing gene-expression microarray data: characterizing clusters with Gene Ontology. *Bioinformatics*, 20(6):979–81, Apr. 2004.