Identification of Regulatory Modules in Time Series Gene Expression Data using a Linear Time Biclustering Algorithm

Sara C. Madeira, Miguel C. Teixeira,

Isabel Sá-Correia and Arlindo L. Oliveira, *Member, IEEE*

Abstract

Several non-supervised machine learning methods have been used in the analysis of gene expression data obtained from microarray experiments. Recently, biclustering, a non-supervised approach that performs simultaneous clustering on the row and column dimensions of the data matrix, has been shown to be remarkably effective in a variety of applications. The goal of biclustering is to find subgroups of genes and subgroups of experimental conditions, where the genes exhibit highly correlated behaviors. These correlated behaviors correspond to coherent expression patterns and can be used to identify potential regulatory modules possibly involved in regulatory mechanisms.

Many specific versions of the biclustering problem have been shown to be NP-complete. However, when we are interested in identifying biclusters in time series expression data, we can restrict the problem by finding only maximal biclusters with contiguous columns. This restriction leads to a tractable problem. Its motivation is the fact that biological processes start and finish in an identifiable contiguous period of time, leading to increased (or decreased) activity of sets of genes forming biclusters with contiguous columns. In this context, we propose an algorithm that finds and reports all maximal contiguous column

Sara C. Madeira is with the Knowledge Discovery and BIOinformatic (KDBIO) team of INESC-ID, Lisbon, Portugal. She is also with University of Beira Interior, Covilhã, Portugal, and Instituto Superior Técnico, Technical University of Lisbon, Portugal. E-mail: smadeira@di.ubi.pt.

Miguel C. Teixeira is with the Biological Sciences Research Group, Centre for Biological and Chemical Engineering/IBB-Institute for Biotechnology and Bioengineering, Instituto Superior Técnico, Technical University of Lisbon, Portugal. E-mail: mnpct@ist.utl.pt.

Isabel Sá-Correia is with the Biological Sciences Research Group, Centre for Biological and Chemical Engineering/IBB-Institute for Biotechnology and Bioengineering, Instituto Superior Técnico, Technical University of Lisbon, Portugal. She is also with Instituto Superior Técnico, Technical University of Lisbon, Portugal. E-mail: isacorreia@ist.utl.pt.

Arlindo L. Oliveira is with the Knowledge Discovery and BIOinformatic (KDBIO) team of INESC-ID, Lisbon, Portugal. He is also with Instituto Superior Técnico, Technical University of Lisbon, Portugal. E-mail: aml@inesc-id.pt.

coherent biclusters (CCC-Biclusters), in time linear in the size of the expression matrix. Each relevant CCC-Bicluster identified corresponds to the discovery of a coherent expression pattern shared by a group of genes in a contiguous subset of time-points and identifies a potentially relevant regulatory module. The linear time complexity of CCC-Biclustering is obtained by manipulating a discretized version of the gene expression matrix and using efficient string processing techniques based on suffix trees.

We report results in synthetic and real data that show the effectiveness of the approach and the relevance of CCC-Biclustering in the discovery of regulatory modules. These results were obtained by applying the algorithm to the transcriptomic expression patterns occurring in *Saccharomyces cerevisiae* in response to heat stress. The results show not only the ability of the proposed methodology to extract relevant information compatible with documented biological knowledge, but also the utility of using this algorithm in the study of other environmental stresses, and of regulatory modules, in general.

Index Terms

Biclustering, time series gene expression data, expression patterns, regulatory modules.

I. Introduction

Recent developments in DNA chips enable the simultaneous measure of the expression level of a large number of genes (virtually all the genes of an organism) for a given experimental condition (sample) [26]. A special type of gene expression data is time series expression data obtained from microarray experiments performed in successive time periods. The analysis of this kind of expression data can be made at a number of levels including experimental design, pattern recognition and network inference [1]. In this context, non-supervised methods, such as clustering and biclustering, represent important techniques. In this paper we study biclustering, a technique that has recently shown to be remarkably effective in a variety of applications in biological data analysis and other data mining tasks. The importance of biclustering in the identification of groups of genes with coherent expression patterns, and its advantages (when compared to clustering) in the discovery of local expression patterns has been extensively studied and documented [6], [21], [27]. The use of these techniques is therefore critical to identify the dynamics of biological systems as well as the different groups of genes involved in each biological process.

Many approaches to biclustering in expression data have been proposed to date [21], [27]. Most formulations of this problem are NP-hard [30], and almost all the approaches presented to date are heuristic and are not guaranteed to find optimal solutions. In a few cases, exhaustive search methods have been used [36], but limits are imposed on the size of the biclusters that can be found, in order to obtain reasonable runtimes. Moreover, the inherent difficulty of this problem when dealing with the original expression matrix and the great interest in finding coherent behaviors regardless of the exact numeric values in the matrix, has led many authors to a formulation based on a discretized version of the expression matrix [3], [12], [14], [16]–[20], [23], [29], [32], [34], [36], [40]. The discretized versions remain, in general, NP-hard.

There exists, however, an important restriction to the biclustering problem that has not been extensively explored and that leads to a tractable problem. This restriction is applicable when the expression data corresponds to snapshots in time of the expression level of the genes. Under this experimental setup, the researcher is particularly interested in biclusters with contiguous columns corresponding to samples taken in consecutive instants of time.

Our motivation to restrict the biclustering problem to the analysis of times series expression data and the identification of contiguous columns biclusters is twofold. First, time series expression experiments are an increasingly popular method for studying a wide range of biological phenomena and can therefore be used to answer a wide range of biological questions [1]. Second, several authors have already pointed out the importance of biclusters with contiguous columns [12], [41], and their importance in the identification of gene regulatory processes. In fact, the activation of a set of genes under specific conditions corresponds, in many cases, to the activation of a particular biological process. The key observation is the fact that biological processes start and finish in a contiguous but unknown period of time, leading to increased (or decreased) activity of sets of genes that can be identified as biclusters with contiguous columns.

In this context, we propose the CCC-Biclustering algorithm, which finds and reports all maximal contiguous column coherent biclusters (CCC-Biclusters) in time linear in the size of the expression matrix by processing a discretized version of the original expression matrix and using efficient string processing techniques based on suffix trees.

This paper is organized as follows: Section II surveys the related work. Section III provides the problem formulation. Section IV describes the algorithm. Section V proposes a scoring schema for CCC-Biclusters based on statistical significance and similarity measures. Section VI presents experimental results performed with synthetic data, which show experimentally the predicted linear time complexity of the algorithm and its ability to recover planted CCC-Biclusters when coupled with the proposed statistical significance and similarity measures. Section VII shows a comparison of CCC-Biclustering with a heuristic approach developed and an application of CCC-Biclustering to the discovery of regulatory modules in yeast by using gene expression data related with the yeast response to heat stress. These results show the ability of the algorithm to discover biologically relevant CCC-Biclusters, corresponding to co-expressed genes, which are shown to be co-regulated by a set of common transcription factors and highly functionally enriched in one or more Gene Ontology terms. Finally, Section VIII presents the conclusions and directions for future work.

II. Related Work

A. Biclustering Algorithms for Time Series Expression Data

Although a large number of biclustering algorithms has been proposed to address the general problem of biclustering [21] [27], to date and to our knowledge, only two recent proposals have addressed the problem of biclustering in time series expression data [12], [41].

Zhang et al. [41] proposed the CC-TSB algorithm based on the work by Cheng and Church [6], that uses directly the values in the expression matrix. Due to its heuristic nature, this approach is not guaranteed to find the optimal set of biclusters. We will compare our method against this work of Zhang et al. in Section VII-A.

A different approach, from Ji and Tan [12], works with a discretized expression matrix. As in the present work, they are also interested in identifying biclusters formed by consecutive columns. Therefore, if appropriately implemented, their idea would generate exactly the same biclusters as the ones generated by out method. The exact complexity of their algorithm is hard to estimate from the description, but it is at least $O(|R||C|^2)$, and hence the CCC-Biclustering algorithm we propose is at least a factor of $\Theta(|C|)$ times faster[1].

B. Discretization Techniques used in Time Series Expression Data Analysis

Most discretization techniques commonly applied to gene expression data use absolute expression values based on the following concepts: average and standard deviation [14], [32], [36], percentage of values [2], [31]; equal width intervals [2], [31]; equal frequency [34]; linear order between the conditions [3], [4], [16]–[19] and statistically significant states [29], [40].

[1] Moreover, the implementation made available by the authors has complexity that is exponential on the number of columns.

Some discretization techniques have, however, been proposed specifically for time series gene expression data and are based on the transitions in expression states between successive time-points [8], [11], [12], [15], [28]. These discretization techniques use either two [8], [15], [28] or three symbols [11], [12] and are usually preceded by a normalization step which standardizes the gene expression time series to zero mean and unit standard deviation.

When studying the impact of discretization on biclustering we have concluded that the techniques based on transitions between time-points obtain better results than those using absolute values [22]. This fact confirms our intuition and that of Schliep et al. [7], who claims that the methods for time series expression analysis that take explicitly into account the temporal dependencies between the time-points should perform better than those that neglect them.

III. Problem Definition

A. Gene Expression Data and Discretized Expression Matrix

Let A' be an $|R|$ row by $|C|$ column gene expression matrix defined by its set of rows (genes), R, and its set of columns (conditions), C. In this context, A'_{ij} represents the expression level of gene i under condition j. Let A'_{iC} and A'_{Rj} denote row i and column j of matrix A', respectively.

In this work, we address the case where the gene expression levels in matrix A' can be discretized to a set of symbols of interest, Σ, that represent distinctive activation levels. In the simpler case, Σ may contain only two symbols, one used for *no-regulation* and other for *regulation*, $\{N, R\}$, or simply $\{0, 1\}$. Another widely used possibility, is to consider a set of three symbols, $\{D, N, U\}$, meaning *DownRegulated*, *NoChange* and *UpRegulated*, or simply $\{-1, 0, 1\}$. In other applications, the values in matrix A' may be discretized to a larger set of symbols. After the discretization process, matrix A' is transformed into matrix A. $A_{ij} \in \Sigma$ represents the discretized value of the expression level of gene i under condition j.

We use the discretization proposed by Ji and Tan [11], [12]. The discretized matrix A is obtained in two steps. In the first step, A' is transformed into an $A'' = |R| \times (|C| - 1)$ matrix of variations. Once matrix A'' is generated, the final discretized matrix A, also with $|R|$ rows and $|C| - 1$ columns, is obtained in a second step by binning the values of the transformed matrix considering a threshold $t > 0$ (see Equation (1)).

$$A''_{ij} = \begin{cases} \frac{A'_{i(j+1)} - A'_{ij}}{|A'_{ij}|} & \text{if } A'_{ij} \neq 0 \\ D & \text{if } A'_{ij} = 0 \text{ and } A'_{i(j+1)} < 0 \\ U & \text{if } A'_{ij} = 0 \text{ and } A'_{i(j+1)} > 0 \\ N & \text{if } A'_{ij} = 0 \text{ and } A'_{i(j+1)} = 0 \end{cases} \qquad A_{ij} = \begin{cases} D & \text{if } A''_{ij} \leq -t \\ U & \text{if } A''_{ij} \geq t \\ N & \text{otherwise} \end{cases} \tag{1}$$

B. Biclusters in Discretized Expression Data

Consider now the matrix A, corresponding to the discretized version of matrix A'.

***Definition** 1 (Bicluster):* A bicluster is a sub-matrix A_{IJ} defined by $I \subseteq R$, a subset of rows, and $J \subseteq C$, a subset of columns. A bicluster with only one row or one column is called *trivial*.

The goal of biclustering algorithms is to identify a set of biclusters $B_k = (I_k, J_k)$ such that each bicluster satisfies specific characteristics of homogeneity. These characteristics vary in different applications [21]. In this work we will deal with biclusters that exhibit coherent evolutions:

***Definition** 2 (CC-Bicluster):* A *column coherent bicluster* A_{IJ} is a bicluster such that $A_{ij} = A_{lj}$ for all rows $i, l \in I$ and columns $j \in J$.

Finding all maximal biclusters satisfying this coherence property is known to be an NP-hard problem [30].

C. CC-Biclusters in Time-Series Expression Data

Since we are interested in the analysis of time series expression data, we can restrict the attention to potentially overlapping biclusters with arbitrary rows and contiguous columns [12], [41]. This fact leads to an important complexity reduction and transforms this particular version of the biclustering problem into a tractable problem. In this context, we can define the type of biclusters we are interested in this work and the important notion of maximality:

***Definition** 3 (CCC-Bicluster):* A *contiguous column coherent bicluster*, A_{IJ}, is a subset of rows $I = \{i_1, \ldots, i_k\}$ and a contiguous subset of columns $J = \{r, r+1, \ldots, s-1, s\}$ such that $A_{ij} = A_{lj}$, for all rows $i, l \in I$ and columns $j \in J$. Each CCC-Bicluster defines a string S that is common to every row in I for the columns in J.

***Definition** 4 (Row-Maximal CCC-Bicluster):* A CCC-Bicluster A_{IJ} is *row-maximal* if we cannot add more rows to I and maintain the coherence property referred in Definition 3.

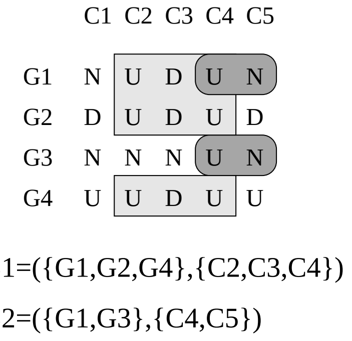
Definition *5 (Left-Maximal and Right-Maximal CCC-Bicluster):* A CCC-Bicluster A_{IJ} is *left-maximal/right-maximal* if we cannot extend its expression pattern S to the left/right by adding a symbol (contiguous column) at its beginning/end without changing its set of rows I.

Definition *6 (Maximal CCC-Bicluster):* A CCC-Bicluster A_{IJ} is maximal if no other CCC-Bicluster exists that properly contains A_{IJ}, that is, if for all other CCC-Biclusters A_{LM}, $I \subseteq L \wedge J \subseteq M \Rightarrow I = L \wedge J = M$.

This definition implies Lemma 1, that we present without proof:

Lemma *1:* Every CCC-Bicluster is right, left and row maximal.

Figure 1 shows an example of a discretized expression matrix together with its two maximal non-trivial CCC-Biclusters.



	C1	C2	C3	C4	C5
G1	N	U	D	U	N
G2	D	U	D	U	D
G3	N	N	N	U	N
G4	U	U	D	U	U

B1=({G1,G2,G4},{C2,C3,C4})

B2=({G1,G3},{C4,C5})

Fig. 1. Example of a discretized matrix with two maximal non-trivial CCC-Biclusters, B1 and B2. The strings UDU and UN correspond to the expression patterns of B1 and B2, respectively.

We can now formulate the problem solved in this work: identify and report all maximal CCC-Biclusters, given a discretized expression matrix. In order to do so we propose a linear time biclustering algorithm that uses efficient string processing techniques based on suffix trees.

IV. Biclustering Time Series Expression Data using Suffix Trees

A. Strings and Suffix Trees

The definitions used are adapted from Gusfield [10], a well know reference on the subject.

Definition *7 (String, Substring and Suffix):* A *string* S is an ordered list of symbols over an alphabet Σ (with $|\Sigma|$ symbols) written contiguously from left to right. For any string S (with $|S|$ symbols), $S[i..j]$ ($1 \leq i, j \leq |S|$) is its (contiguous) *substring* starting at position i and ending at position j. $S[i..|S|]$ is the *suffix* of S that starts at position i.

Definition *8 (Suffix Tree and Generalized Suffix Tree):* A *suffix tree*, T, of a string S is a rooted directed tree with exactly $|S|$ leaves, numbered 1 to $|S|$, such that: 1) each internal node in T, other than the root, has at least two children and each edge is labeled with a nonempty substring of S; 2) no two edges out of a node have edge-labels beginning with the same symbol; 3) for any leaf i, the label of the path from the root to the leaf i exactly spells out the suffix of S that starts at position i. A *generalized suffix tree* is a suffix tree built for a set of strings S_i.

In order to construct a suffix tree obeying this definition, when one suffix of S matches a prefix of another suffix of S, we add a symbol(terminator), that does not appear anywhere else in the string, to its end (usually the symbol $\$$ is used). In the case of generalized suffix trees, we add a unique terminator to the end of each string S_i.

Suffix trees can be built in time linear in the size of the string S [33] [25] [39]. Generalized suffix trees, can be easily obtained by consecutively building the suffix tree for each string S_i. This construction is linear in the sum of the sizes of the set of strings S_i.

Definition *9 (Edge-Length):* Given a node v in T, the *edge-length* of the edge leading to v, $E(v)$, is the number of symbols other than terminators in the path from node u to v, where u is the parent of v.

Definition *10 (String-Depth and String-Label):* The *string-depth* of a node v in T, $P(v)$, is the sum of all edge-lengths in the path from the root to v. This path is the *string-label* of v.

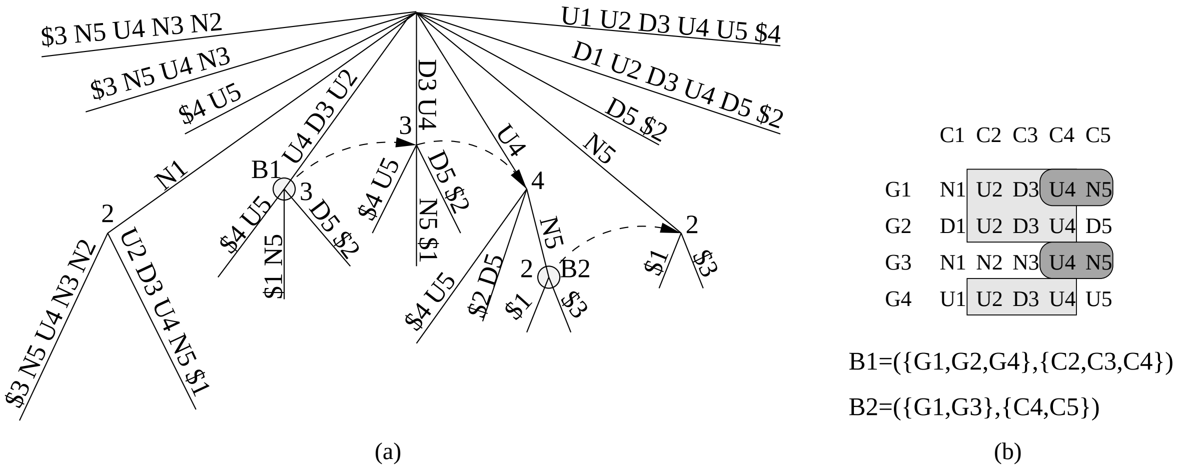
Definition *11 (Suffix Link):* Let $x\alpha$ denote an arbitrary string, where x denotes a single symbol and α denotes a (possibly empty) substring. For any internal node v with string-label $x\alpha$, if there is another node $s(v)$ with string-label α, then a pointer from v to $s(v)$ is called a suffix link. The pair $(v, s(v))$ will denote the suffix link from v to $s(v)$. As a special case, if α is empty $x\alpha$ has a suffix link leading to the root, $(v, root)$.

B. CCC-Biclusters and Suffix Trees

We now develop our linear-time biclustering algorithm. Before presenting the central idea of this work, which relates CCC-Biclusters and nodes in a suffix tree, we introduce a simple alphabet transformation (performed as a preprocessing step in the algorithm) that appends the column number to each symbol in the matrix. For that, we consider a new alphabet $\Sigma' = \Sigma \times \{1, \ldots, |C|\}$, where each element Σ' is obtained by concatenating one symbol in Σ and one number in the range $\{1, \ldots, |C|\}$. Consider now the set of strings $S_i = \{S_1, \ldots, S_{|R|}\}$ obtained by applying this alphabet transformation to each row A_{iC} in matrix A. Figure 2(b) shows the result of this transformation applied to the discretized matrix shown in Figure 1.

We will now show that the maximal CCC-Biclusters in the original matrix A correspond exactly to nodes in the generalized suffix tree T built from the set of strings[2].

Consider a node v in T together with its string-depth, $P(v)$, and its edge-length, $E(v)$. Let $L(v)$ denote the number of leaves in the sub-tree rooted at v, in case v is an internal node.



	C1	C2	C3	C4	C5
G1	N1	U2	D3	U4	N5
G2	D1	U2	D3	U4	D5
G3	N1	N2	N3	U4	N5
G4	U1	U2	D3	U4	U5

B1=({G1,G2,G4},{C2,C3,C4})

B2=({G1,G3},{C4,C5})

(a) (b)

Fig. 2. (a) Generalized suffix tree for the discretized matrix on the right. The circles identify the Maximal Non-Trivial CCC-Biclusters (B1 and B2). (b) CCC-Biclusters B1 and B2 showed in the matrix alphabet transformation as subsets of rows and columns. Remember that the strings UDU and UN correspond to the expression patterns of B1 and B2, respectively.

Figure 2(a) illustrates the generalized suffix tree obtained from the strings that correspond to the rows of the matrix in Figure 2(b). For clarity, this figure does not contain the leaves that represent string terminators that are direct daughters of the root. Each non-terminal node, other than the root, is labeled with the value of $L(v)$, the number of leaves in its subtree. We show the suffix links between nodes although (for clarity) we omit the suffix links pointing to the root.

By analyzing this illustrative example, it is easy to verify that every internal node in T corresponds to one row-maximal CCC-Bicluster in matrix A. This is so because an internal node v in T corresponds to a given substring that is common to every row that has a leaf rooted in v. Therefore, node v defines a CCC-Bicluster that has $P(v)$ columns and a number of rows equal to $L(v)$. It is also true that all leaves except those whose string-label is simply a terminator also

[2] We will show later that the increase in the alphabet size resulting from the alphabet transformation will not affect the linear-time construction and manipulation of the suffix tree.

identify CCC-Biclusters. However, some of the CCC-Biclusters are trivial, since they represent CCC-Biclusters with only one column (nodes with edge-labels $N1$, $U4$ and $N5$), or are represented by leaves (trivial CCC-Biclusters with only one row). Others are non-maximal (nodes with edge-labels $D3U4$ and $N5$), since they have an incoming suffix link from a node with the same number of leaves. We will show that an internal node is maximal *iff* there is no incoming suffix link from a node with the same value of $L(v)$. As such, only the internal nodes with edge-labels $U2D3U4$ and $U4N5$ identify maximal, non-trivial CCC-Biclusters. These nodes correspond to the maximal CCC-Biclusters $(\{G1, G2, G4\}, \{C2, C3, C4\})$ and $(\{G1, G3\}, \{C4, C5\})$.

Note that, the rows in each CCC-Bicluster, identified by a given node v, are obtained from the terminators of the leaves in its subtree. Furthermore, the value of $P(v)$ and the first symbol of the string-label of v provide the information needed to identify the set of columns.

Using the illustrative example in Figure 2 we have shown that all nodes in the generalized suffix tree T correspond to CCC-Biclusters in matrix A, and that some of these CCC-Biclusters may not be maximal. We will now present, with only sketches of proofs, the two lemmas that lead to the theorem that supports our linear time biclustering algorithm.

***Lemma** 2:* Every right-maximal CCC-Bicluster corresponds to one node in T.

Proof: Let B be a CCC-Bicluster that cannot be extended to the right by adding a column at the right, that is, a right-maximal CCC-Bicluster. Since B is a CCC-Bicluster, every row in B shares the substring that defines B. Since B is right maximal, at least one of the rows in B must have a symbol that differs from the symbol in the other rows, in the first column to the right that is not in B. Therefore, there is a node in T that matches B and the string-label of that node is the string that defines B. ■

***Lemma** 3:* Let node v_1 correspond to a CCC-Bicluster B_1 and node v_2 correspond to a CCC-Bicluster B_2. Then, if there is a suffix link from node v_1 to node v_2, (v_1, v_2), CCC-Bicluster B_2 contains one less column than CCC-Bicluster B_1.

Proof: Follows directly from the definition of suffix links. ■

***Theorem** 1:* Let v be a node in the generalized suffix tree T. If v is an internal node, then v corresponds to a maximal CCC-Bicluster *iff* $L(v) > L(u)$ for every node u such that there is a suffix link from u to v, (u, v). If v is a leaf node, then v corresponds to a maximal CCC-Bicluster *iff* the string-depth of v is equal to $|C|$ ($P(v) = |C|$), and the edge-label of v has symbols other than terminators ($E(v) > 0$)). Furthermore, every maximal CCC-Bicluster in the matrix corresponds to a node v satisfying one of these conditions.

Proof: Let B be a maximal CCC-Bicluster and S the string that defines B. Now, S must lead to a node v (by Lemma 2), that is either an internal node or a leaf node.

If node v is an internal node and does not have an incoming suffix link, the conditions of the theorem are met. In fact, we can conclude that v corresponds to a left-maximal CCC-Bicluster (see Definition 5 and Lemma 3). Moreover, v is also row-maximal and right-maximal (see Definitions 4 and 5 and Lemma 2). Consider now that v has an incoming suffix link. Since B is also left-maximal, every node u that defines a CCC-Bicluster B' with a set of columns properly containing the set of columns of B and one more column than B must have $L(u) < L(v)$ (by Lemma 3). This happens because B' cannot contain all the rows in B (otherwise, B would not be left-maximal). Therefore, it is sufficient to check that every internal node u that has a suffix link directed at v has $L(u) < L(v)$ to ensure that node v corresponds to a maximal CCC-Bicluster. On the other hand, if $L(u) = L(v)$ (it can never happen that $L(u) \geq L(v)$), then B' would have one more column than B, and the same set of rows. Therefore, B would not be not maximal.

If node v is a leaf node the conditions of the theorem are met. In fact, if B is maximal the string-depth of v must be equal to $|C|$. Otherwise, B would not be left-maximal since it could be extended to the left by adding all the symbols at its left in S. Furthermore, the string-label of v cannot be only a terminator symbol ($E(v) = 0$), otherwise B would also not be maximal. This is so based on the following: (1) if the parent of v is the root, then B would not be left maximal since it could be extended to the left by adding to it at least the column corresponding to the last symbol in S; (2) if the parent of v is an internal node other than the root, then B would not be maximal either since it could be extended by adding to it the rows that correspond to the remaining leaves in the subtree rooted at its parent. ■

C. CCC-Biclustering: A Linear Time Biclustering Algorithm for Finding and Reporting all Maximal CCC-Biclusters

Theorem 1 directly implies that there is an algorithm that finds and reports all maximal CCC-Biclusters in a discretized and transformed gene expression matrix A in time linear in the size of the matrix. Algorithm 1 builds a suffix tree over the set of strings $S_i = \{S_1, \ldots, S_{|R|}\}$, obtained using the alphabet transformation described in Section IV-B and checks, for each node, whether the conditions of Theorem 1 are met. Nodes that do not meet the required conditions are marked as invalid in lines 11 and 14. All the remaining nodes correspond to maximal CCC-Biclusters and are reported.

Algorithm 1: CCC-Biclustering

```
   input: Discretized gene expression matrix $A$
1  Perform alphabet transformation and build a generalized suffix tree $T$ for the resulting set
   of strings $S_i = \{S_1, \ldots, S_{|R|}\}$.
2  for each node $v \in T$ do
3      Mark $v$ as "Valid".
4      Compute the string-depth $P(v)$.
5      if $v$ is a leaf node then
6          Compute the edge-length $E(v)$.
7  for each internal node $v \in T$ do
8      Compute the number of leaves $L(v)$ in the subtree rooted at $v$.
9  for each node $v \in T$ do
10     if $v$ is an internal node and there is a suffix link $(v, s(v))$ and $L(s(v)) = L(v)$ then
11         Mark node $s(v)$ as "Invalid".
12     else
13         if $v$ is a leaf node and ($P(v) \neq |C|$ or $E(v) = 0$) then
14             Mark node $v$ as "Invalid".
15 for each node $v \in T$ do
16     if $v$ is marked as "Valid" then
17         Report the CCC-Bicluster that corresponds to $v$.
```

D. Complexity Analysis of CCC-Biclustering and Implementation Issues

With appropriate data structures at the nodes and using Ukkonen's algorithm [39], the suffix tree construction time is linear on the size of the input matrix, $O(|R||C|)$. The remaining steps of our algorithm are also linear since they are performed using depth first searches (*dfs*) on the suffix tree. Since any tree has fewer internal nodes than leaves, the linear time complexity of Algorithm 1 is an immediate result.

One issue, however, deserves a special reference. It is a well known fact that the complexity of suffix tree construction has a dependence on the alphabet size [10] that becomes important when the alphabet is large. Therefore, one has to ensure that the increase in the alphabet size from $|\Sigma|$ to $|C||\Sigma|$ due to the alphabet transformation described in Section IV-B does not affect the linear time complexity of our algorithm. In fact, only one internal node, the root, has a number of children that depends on the number of columns. As can be observed in the suffix tree for the example in Figure 2, all internal nodes other than the root have a number of children that is not affected by the number of columns. This is so because, after the alphabet transformation, the edge label of an internal node corresponds to an expression pattern common to a set of genes, between a contiguous set of time-points, which always starts at a *specific* time-point. This leads to a maximum number of children that is $O(|\Sigma|)$ and not $O(|C||\Sigma|)$.

Internal nodes that have as children only leaf nodes with edges labeled by terminator symbols, may have a number of children that grows with the number of rows in the matrix but this number does not depend on the number of columns. The dependence on the number of rows is not a problem since standard implementations of generalized suffix trees avoid non-linear dependencies on the number of terminators by using the appropriate data structures. In this context, and since the alphabet transformation only influences the outdegree of the root, by using an array at the root to store the pointers to the children (instead of the linked lists used in the other nodes), one guarantees that the branching at the root is performed in constant time, and the total complexity of CCC-Biclustering algorithm is $O(|R||C|)$.

V. Scoring CCC-Biclusters using Statistical Significance and Similarity Measures

Since applying biclustering to real gene expression matrices can produce hundreds or even thousands of biclusters, an objective evaluation of the quality of the biclusters discovered is crucial. In fact, the inspection of biclustering results can be prohibitive without an efficient scoring approach which enables sorting and filtering the results according to a statistical scoring criterion. The statistical significance of the results can then be combined with measures of biological significance in order to produce a set of interesting and potentially useful biclusters, both from the statistical and biological point of view. For CCC-Biclusters, we propose the use of a scoring criterion, which combines two criteria: (1) statistical significance of expression pattern, and (2) similarity with another overlapping CCC-Biclusters.

A. Statistical Significance

We propose to measure the statistical significance of a CCC-Bicluster B of size $|I|\times|J|$, where I is the set of genes and J is the set of contiguous time-points, and expression pattern p_B, against the null hypothesis, H_0, that assumes that the expression values of genes evolve independently.

Under the null hypothesis, it is possible to compute, using reasonable simplifying assumptions, the probability of a CCC-Bicluster of the considered size and expression pattern occurring by chance in an expression matrix with $|G|$ genes and $|C|$ time-points. This value is obtained by computing the *tail of the binomial distribution*, P, which gives the probability of an event with probability p occurring k or more times in n independent trials, $P = \sum_{j=k}^{n} p^j (1-p)^{n-j}$.

In this context, the statistical significance of a CCC-Bicluster B, is the *p*-value(B), computed by obtaining the probability of a random occurrence under H_0 of the expression pattern p_B $k = |I| - 1$ times in $n = |G| - 1$ independent trials, where I is the number of genes in B and $|G|$ is the total number of genes in the gene expression matrix.

We use the simplifying assumption that the probability of occurrence of a specific expression pattern p_B is adequately modeled by a first order Markov Chain, with state transition probabilities obtained from the values in the corresponding columns in the matrix. For example, if $B = (\{G1, G2, G4\}, \{C2, C3, C4\})$ in Figure 2(b), $p_B = P(U2D3U4) = P(U2)P(D3|U2)P(U4|D3)$, where $P(U2) = \frac{|U2|}{|G|}$, $P(D3|U2) = \frac{P(U2D3)}{P(U2)} = \frac{|U2D3|}{|U2|}$ and $P(U4|D3) = \frac{P(D3U4)}{P(D3)} = \frac{|D3U4|}{|D3|}$. These probabilities are, in this case, computed using the gene expression matrix after alphabet transformation in Figure 2(b). The values $|U2|$, $|U2D3|$, $|D3|$ and $|D3U4|$ correspond, respectively, to the number of occurrences of symbol $U2$, the number of transitions from $U2$ to $D3$, the number of occurrences of symbol $D3$, and the number of transitions from $D3$ to $U4$.

B. Similarity Measure

In order to compute the similarity score between two CCC-Biclusters, $B_1 = (I_1, J_1)$ and $B_2 = (I_2, J_2)$, we use the Jaccard Index. In this work, this score is used to measure the overlap between CCC-Biclusters both in terms of genes and conditions and is defined as follows:

$$J(B_1, B_2) = J((I_1, J_1), (I_2, J_2)) = \frac{|B_1 \bigcap B_2|}{|B_1 \bigcup B_2|} = \frac{|B_{11}|}{|B_{01}| + |B_{10}| - |B_{11}|}, \tag{2}$$

where $B_{11} = \{(i,j) : (i,j) \in B_1 \bigwedge (i,j) \in B_2\}$, $B_{10} = \{(i,j) : (i,j) \in B_1 \bigwedge (i,j) \notin B_2\}$ and $B_{01} = \{(i,j) : (i,j) \notin B_1 \bigwedge (i,j) \in B_2\}$, for the genes $i \in I_1 \bigcup I_2$ and the conditions $j \in J_1 \bigcup J_2$.

Similarly, the gene similarity and condition similarity can be computed, respectively, as follows: $J(I_1, I_2) = \frac{|I_1 \bigcap I_2|}{|I_1 \bigcup I_2|}$ and $J(J_1, J_2) = \frac{|J_1 \bigcap J_2|}{|J_1 \bigcup J_2|}$. Note that, in practice, and since $|B_1| = |I_1| \times |J_1|$ and $|B_2| = |I_2| \times |J_2|$, the similarity score as defined in Equation 2 can be computed easily using the fact that $|B_1 \bigcap B_2| = |I_1 \bigcap I_2| \times |J_1 \bigcap J_2|$ and $|B_1 \bigcup B_2| = |B_1| + |B_2| - |B_1 \bigcap B_2|$.

VI. Experimental Results with Synthetic Data

In order to validate experimentally the proposed approach in terms of the predicted linear time complexity and the ability to recover relevant CCC-Biclusters, we performed experiments on synthetic data, using a prototype implementation of the algorithm coded in Java, and a 3 GHz Pentium-4 machine running Linux with 1GB of memory. We have first validated the predicted linear time complexity by generating synthetic matrices with random values, in which 10 biclusters, with dimensions ranging from 15 to 25 rows and 8 to 12 columns, were hidden. The size of the matrices varied from 250×50 (rows $\times$ columns) up to 1000×250. We used a three symbol alphabet, $\Sigma = \{U, D, N\}$. These experiments have also shown a clear linear relationship between the variation of the CPU time and the size of the input data matrix over several orders of magnitude [23]. In all cases, we recovered the *planted* CCC-Biclusters, together with a number of biclusters overlapping with the planted biclusters and a large number of artifacts that result from random coincidences in the data matrix.

TABLE I

CCC-Biclusters planted in the synthetic 1000×50 matrix.

ID	Expression Pattern	#Time-Points(first-last)	#Genes
1	NUNUUNDDNDNU	12(21-32)	19
2	NNUDUNNNU	9(1-9)	15
3	NUNDNNDDUNN	11(30-40)	19
4	NDUUDDDD	8(34-41)	16
5	UNUDUDNDUU	10(2-11)	16
6	UDDUNUDDU	9(37-45)	24
8	UDNNDUNUDNDD	12(26-37)	19
9	NDUNDNUUU	9(25-33)	20
10	DNDNDDNNNNDD	12(33-44)	18

For illustrative purposes, we describe here the results obtained in the 1000 rows by 50 columns example. For this example, two experiments were carried out. In the first experiment, no planted biclusters existed. In this case, a total of 41097 maximal non-trivial CCC-Biclusters were identified. *None* of these biclusters passed the statistical significance test described in Section V-A at a 1% level (after Bonferroni correction). In the second experiment, 10 CCC-Biclusters with the aforementioned dimensions were planted (see Table I). The algorithm identified 40461 maximal non-trivial CCC-Biclusters in a few seconds. From these, 165 passed the statistical significance test at the 1% level (after Bonferroni correction). Most of these are variations on the

TABLE II

Top 10 CCC-Biclusters discovered after filtering CCC-Biclusters with overlapping above 25% (after sorting the discovered CCC-Biclusters using the statistical significance p-value)

ID	*Expression Pattern*	*#Time-Points*	*#Genes*	*p-Value*	*Closest Planted CCC-Bicluster*
24475	DDNUDDNDDNDD	12(35-46)	18	4.13E-56	MATCH 7
5790	UDNNDUNUDNDD	12(26-37)	19	1.37E-55	MATCH 8
17868	NUNUUNDDNDNU	12(21-32)	19	3.72E-55	MATCH 1
25020	DNDNDDNNNNDD	12(33-44)	18	6.10E-54	MATCH 10
2438	UDDUNUDDU	9(37-45)	23	8.43E-40	MATCH 6 LOST 1 GENE[1]
15158	NUNDNNDDUNN	11(30-40)	16	4.08E-37	MATCH 3 LOST 3 GENES[1]
34531	UNUDUDNDUU	10(2-11)	16	7.17E-34	MATCH 5
16797	NDUNDNUUU	9(25-33)	20	8.20E-33	MATCH 9
38344	NNUDUNNNU	9(1-9)	15	5.29E-23	MATCH 2
14145	NDUUDDDD	8(34-41)	14	1.48E-17	MATCH 4 LOST 1 GENE[1]

planted CCC-Biclusters. Table II shows that after sorting the CCC-bicluster using the statistical significance p-value described in Section V-A and filtering CCC-Biclusters whose similarity measure as defined in Section V-B is above 25%, the proposed approach is able to identify the planted CCC-Biclusters as the top 10 CCC-Biclusters[1]. After filtering the biclusters with similarities above 25% only 37 biclusters had a (Bonferroni corrected) p-value below 0.01.

These results confirm that CCC-Biclustering, when coupled with the proposed scoring schema based on statistical significance and similarity measures[2], can be effectively used to identify even relatively small CCC-Biclusters that are statistically significant.

VII. Experimental Results with Real Datasets

A. Comparison with Heuristic Algorithms

The CC-TSB algorithm [41] described in Section II aims at finding groups of genes that exhibit coherent evolution on a subset of contiguous columns. Since this heuristic biclustering algorithm uses the gene expression values directly without relying on a discretization step we decided to compare its results with those of CCC-Biclustering in the same dataset and using

[1] Lost genes are caused by the (artificial) way in which CCC-Biclusters were planted. When two or more CCC-Biclusters are overlapping, the expression patterns in the overlapping submatrices are those of the last planted CCC-Bicluster. For this reason, the genes in overlapping zones are lost for the previously planted CCC-Biclusters.

[2] When coupled only with the statistical significance test, CCC-Biclustering is already able to identify the planted CCC-Biclusters. However, there is a number of highly overlapping CCC-Biclusters, which prevent the discovery of the 10 CCC-Biclusters in the top 10 and can thus be filtered efficiently using the similarity score described in Section V-B.

the same parameters used by the authors. As such, we used the yeast cell-cycle dataset publicly available [5], described by Tavazoie et al. [37] and processed by Cheng and Church [6]. We used 2884 genes selected by Cheng and Church [6] and removed the ORFS with missing values and the ones that no longer exists in SGD (Saccharomyces Genome Database). As in [41] we set the parameters α and β to 300 and 1.2, respectively, and used their algorithm in the matrix with the remaining genes to find 100 biclusters. In order to apply CCC-Biclustering we first discretized this preprocessed data using the technique based on transitions between time-points proposed by Ji and Tan [11], [12] and described in Section III-A. We have also used the three symbol alphabet $\Sigma = \{D, N, U\}$ and $t = 1$.

TABLE III

COMPARISON OF THE RESULTS OBTAINED BY THE CCC-BICLUSTERING CC-TSB ALGORITHM [41].

CC-TSB-Biclustering (sorted by MSR)			CCC-Biclustering (sorted by MSR)			CCC-Biclustering (sorted by *p*-value, overlap $\leq$ 25%)			
#Conds (first-last)	#Genes	MSR	#Conds (first-last)	#Genes	MSR	#Conditions (first-last)	#Genes	MSR	*p*-value
17(1-17)	1447	411.7	3(11-13)	49	26.3	5(9-13)	88	165.2	2.40E-24
16(2-16)	1016	7039.1	5(11-15)	24	45.0	9(5-13)	21	104.6	5.30E-24
17(1-17)	1730	42830.3	2(14-15)	558	47.7	7(11-17)	31	107.6	1.02E-20
17(1-17)	1366	47338.4	4(3-6)	24	48.2	10(2-11)	16	116.5	4.72E-20
17(1-17)	1671	47887.9	6(11-16)	20	48.4	6(8-13)	189	241.8	1.74E-18

In Table III we report the sizes and the mean squared residue (MSR) for the top 5 biclusters (evaluated by the MSR, which is the merit function minimized in CC-TSB algorithm) obtained by each method. In the case of CCC-Biclustering, when sorting by MSR was used, and in order to avoid the discovery of CCC-Biclusters with small number of genes corresponding to small matrices with a small MSR, we filtered those with less than 20 genes. We report also the top 5 CCC-Biclusters discovered by sorting the results using the *p*-value described in Section V-A and filtering CCC-Biclusters with similarities above 25%. These results show that the statistical significance test used for CCC-Biclusters is able to find highly significant expression patterns shared by a relatively large number of genes with a small MSR.

The results obtained using the CC-TSB algorithm show that the heuristic proposed by Zhang et al. is not effective. In fact, the restriction imposed on the columns that can be removed makes the algorithm converge rapidly to a local minimum, from which it does not escape. The obtained values for the MSR show clearly the weakness of the method. Moreover, the method converges to biclusters with a high number of columns, which are, in most cases, all the columns in the dataset. This means that the CC-TSB algorithm is in fact looking for gene clusters, and not biclusters, which makes it useless for the purposes of identifying local patterns.

B. Application to the Identification of Regulatory Modules

To assess the biological relevance of the CCC-biclusters in real data, we used a dataset from Gasch et al. [9], concerning the yeast response to heat shock. This dataset comprises 5 different time-points along the first hour of exposure to 37°C (0', 5', 15', 30' and 60'). The first time-point is an average of three replicates of time zero. The dataset was preprocessed as in Section VII-A.

Since we were interested in CCC-Biclusters with high statistical significance the set of 167 maximal non-trivial CCC-Biclusters discovered was then sorted in ascending order according to the statistical *p*-value described in Section V-A. From these only 25 were considered as highly significant at the 1% level after applying the Bonferroni correction for multiple testing. In order to avoid the analysis of highly overlapping CCC-Biclusters, we have then computed the similarities between the sorted biclusters using the Jaccard similarity score, as described in Section V-B, and filtered CCC-Biclusters with similarity greater than 25%. This filtering process removed 9 of the 25 CCC-Biclusters originally selected.

Table IV shows a summary of the remaining 16 CCC-Biclusters analyzed using the Gene Ontology (GO) annotations obtained using the GoToolBox [24]. To perform the analysis for functional enrichment we used the p-values obtained using the hypergeometric distribution to access the over-representation of a specific GO term. In order to consider a CCC-Bicluster to be *highly significant*, we require its genes to show highly significant enrichment in one or more of the "biological process" ontology terms by having a Bonferroni corrected p-value below 0.01. A CCC-Bicluster is considered as *significant* if at least one of the GO terms analyzed is significantly enriched by having a (Bonferroni corrected) *p*-value in the interval $[0.01, 0.05[$.[2]

From these 16 CCC-Biclusters, six (Tables V and VI)[3] were analyzed in more detail, cor-

[2] Although we only consider as functionally enriched the terms with Bonferroni corrected p-values below 0.01 (for high statistical significance), or below 0.05 (for statistical significance), the p-values presented in the text are without correction.

[3] In Tables V and VI, column 1 identifies the CCC-Bicluster, column 2 lists relevant transcription factors(TFs) co-regulating the set of genes in the CCC-Biclusters, column 3 lists the percentage of genes in the CCC-Biclusters that are co-regulated by the TF in column 2. Finally columns 4 and 5 list relevant GO terms in the transcriptomic response of *Saccharomyces cerevisiae* to heat stress, together with the hypergeometric geometric *p*-values. The p-palues not passing the Bonferroni test at the 1% level are marked with *.

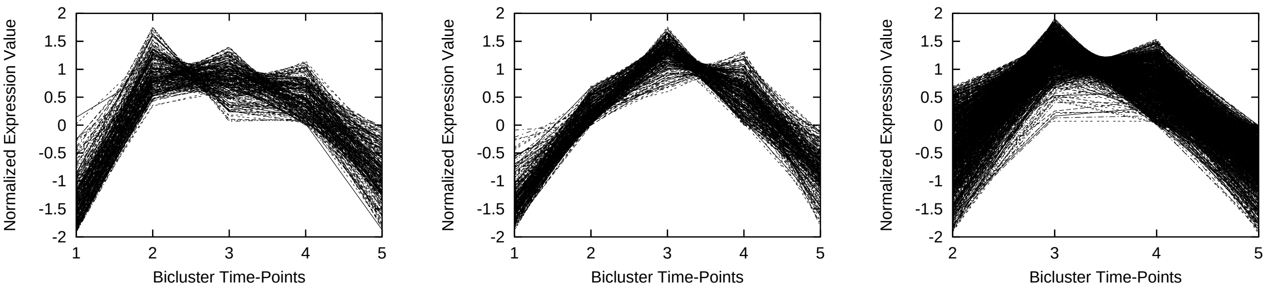
responding to chronological expression patterns selected as described below (Figures 3 and 4). For these CCC-Biclusters, selected for describing either transcriptional up-regulation or down-regulation patterns, we analyzed in detail the Gene Ontology annotations together with information about transcriptional regulation available in the YEASTRACT database [38].

TABLE IV
SUMMARY OF THE CCC-BICLUSTERS PASSING THE STATISTICAL TEST AT THE 1% LEVEL AFTER BONFERRONI CORRECTION (AFTER FILTERING CCC-BICLUSTERS WITH SIMILARITY ABOVE 25%)

ID	*Variation Pattern*	*#Time-Points (first-last)*	*#Genes*	*Sorting P-value*	*#p-values <0.01*	*# p-values 0.01≤<0.05*	*Best p-value (Level>2)*	*Dataset Frequency*
124	DNU	4(2-5)	904	2.56E-84	40	8	8.23E-63(7)	18.52
14	UND	4(2-5)	1091	1.64E-58	62	12	2,79E-24(5)	10.78
27	UUND	5(1-5)	290	3.69E-44	7	6	3.28E-08(3)	21.59
39	UNND	5(1-5)	258	8.65E-42	0	0	1.65E-04(3)	8.18
151	DNNU	5(1-5)	232	3.99E-31	12	2	3.19E-14(3)	93.26
48	UDUD	5(1-5)	182	1.35E-26	0	1	6.98E-05(3)	90.11
142	DUDU	5(1-5)	248	2.84E-24	8	19	4.37E-09(4)	41.27
43	UNDD	5(1-5)	109	6.56E-24	0	0	1.97E-04(11)	4.62
147	DNUU	5(1-5)	144	6.03E-21	0	3	4.50E-05(3)	87.07
83	NUNN	5(1-5)	224	1.90E-16	2	4	1.41E-05(6)	10.13
42	UNDN	5(1-5)	131	3.30E-11	2	1	6.85E-06(9)	4.44
148	DNUN	5(1-5)	192	6.00E-11	4	4	7.68E-07(3)	88.08
159	DDUU	5(1-5)	56	1.37E-07	0	0	1.14E-03(6)	13.64
79	NUUN	5(1-5)	97	4.41E-07	2	3	2.46E-06(3)	20.00
92	NNUN	5(1-5)	52	3.88E-05	2	0	1.64E-06(4)	27.27
99	NNDN	5(1-5)	39	4.79E-05	1	0	2.13E-05(6)	13.79

1) CCC-Biclusters describing Transcriptional Up-Regulation Patterns: The first three biclusters analyzed include genes whose expression was up-regulated (1) abruptly during the first 5 minutes of exposure (CCC-Bicluster 39, with 258 genes), (2) slowly during the first 15 minutes of exposure (CCC-Bicluster 27, with 291 genes) and (3) with a short delay, between 5 and 15 minutes of exposure (CCC-Bicluster 14, with 1091 genes) (See Figure 3, for details).

The analysis of the first bicluster (CCC-Bicluster 39) using the GOToolBox revealed that there are no GO terms with a (Bonferroni corrected) p-value below or equal to 0.01 associated to this specific gene list (Table V). This may occur as a consequence of an unspecific wide initial response to stress, in which the transcription of a number of genes, belonging to a large number of different biological functions, is up-regulated. It is, nonetheless, noteworthy that the most significant terms associated to this bicluster are "signal transduction" (p-value



(a) CCC-Bicluster 39

(b) CCC-Bicluster 27

(c) CCC-Bicluster 14

Fig. 3. Expression patterns of the CCC-Biclusters describing transcriptional up-regulation.

1.65E-04) and "regulation of transcription from RNA polymerase II promoter" (p-value 2.80E-02). This conclusion is consistent with the activation during the first 5 minutes following yeast exposure to heat shock of signaling cascades and transcription factors associated the transcriptional machinery, which will mediate stress specific responses in the subsequent time-points. A similar GO-based analysis of the second and third biclusters (CCC-Biclusters 27 and 14), also presented in Table V, reveals the occurrence of highly significant terms, including "carbohydrate metabolism" (p-values 7.33E-08 and 4.87E-21) or "energy derivation by oxidation of organic compounds" (p-values 3.60E-06 and 3.92E-20), related to energy generation, and "response to stimulus" (p-values 3.28E-08 and 1.51E-16) or "response to stress" (p-values 4.88E-06 and 1.02E-15), related to the cellular response to heat shock. These terms are consistent with the induction of protein folding chaperones aiming at protecting against, and recovering from, protein unfolding with associated energetic expenses. The transcriptional induction of genes involved in alternative carbon source metabolism and respiration, in the presence of glucose, is considered a consequence of a sudden decrease in cellular ATP concentration, caused by ATP-consuming stress defense mechanisms [9].

Using the computational tools from the YEASTRACT database [38], each of the three referred biclusters was grouped based on the sharing of specific transcriptional regulators mediating the co-regulation of clustered genes. As expected, based on the literature, the heat-shock factor Hsf1p comes out as one of the major regulators of these three biclusters, regulating 16%, 23% and 19% of the genes in biclusters 39, 27 and 14, respectively. Moreover, in agreement with previous knowledge, Msn2p and Msn4p, regulators of the general stress response in yeast, appear as major contributors to the heat-induced transcriptional activation, regulating 14%, 21% and 19% of the

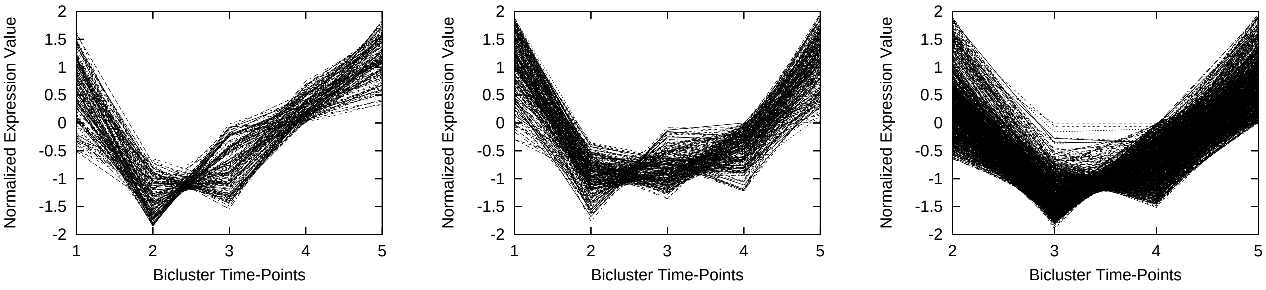
TABLE V

CCC-BICLUSTERS DESCRIBING TRANSCRIPTIONAL UP-REGULATED PATTERNS.

ID	*Pattern*	*TFs*	*%*	*Relevant GO Terms Enriched*	*p-value*
39	Early drastic up-regulation	Sok2p Arr1p Hsf1p Msn2p Rpn4p	23.89 16.37 15.93 14.16 14.16	signal transduction regulation of transcription from RNA polymerase II promoter	1.65E-04* 2.80E-02*
27	Early slow up-regulation	Hsf1p Sok2p Msn2p Rpn4p Msn4p	23.62 22.14 20.66 18.45 17.71	response to stimulus carbohydrate metabolism regulation of carbohydrate metabolism generation of precursor metabolites and energy energy derivation by oxidation of organic compounds response to stress carbohydrate biosynthesis	3.28E-08 7.33E-08 3.51E-07 1.86E-06 3.60E-06 4.88E-06 5.63E-06
14	Middle up-regulation	Hsf1p Sok2p Msn2p Rpn4p Msn4p	23.62 22.14 20.66 18.45 17.71	generation of precursor metabolites and energy carbohydrate metabolism energy derivation by oxidation of organic compounds cellular carbohydrate metabolism response to stimulus response to stress	2.79E-24 4.87E-21 3.92E-20 1.24E-16 1.51E-16 1.02E-15

genes in each of the biclusters, respectively [13]. A third transcription factor also presumably implicated in the regulation of the three biclusters is Rpn4p, regulating 14%, 18% and 18% of the genes in each of the biclusters, respectively. This transcription factor stimulates the expression of the proteasome genes, involved in the degradation of denatured or unnecessary proteins in stressed yeast cells [9]. Although the transcription factors regulating the three temporal stages of heat-shock induced co-activated transcription are apparently the same, the majority of the genes in each of the three biclusters do not overlap. As an example, the Hsf1p-target heat shock genes seem to be activated at different time-points: as a more drastic response *HSP10* and *HSP42* are up-regulated within the first 5 minutes of heat shock, while *HSP104*, *HSP26*, *HSP78*, *SSA4* and *SSE2* transcript levels are only maximal after 15 minutes of heat shock and the expression of *HSC82*, *HSP82*, *SSA1*, *SSA2*, *SSA3*, *SSC1*, *SSE1*, *CPR6* and *STI1* only increases between 5 and 15 minutes of heat shock exposure. This may suggest that these sets of chaperones play their roles at different times of the adaptive process. It also suggests that each transcription factor may act on different target genes in different temporal states.

2) CCC-Biclusters describing Transcriptional Down-Regulation Patterns: The remaining three analyzed biclusters include genes whose expression was down-regulated (1) and (2) abruptly during the first 5 minutes of exposure (CCC-Biclusters 147, comprising 144 genes, and 151, comprising 232 genes), and (3) with a short delay, between 5 and 15 minutes of exposure (CCC-Bicluster 124, comprising 904 genes). See Figure 4 for details.



(a) CCC-Bicluster 147 (b) CCC-Bicluster 151 (c) CCC-Bicluster 124

Fig. 4. Expression patterns of the CCC-Biclusters describing transcriptional down-regulation.

The GO-based analysis of bicluster 147 indicates that there are no GO terms with a (Bonferroni corrected) *p*-value below 0.01 associated to this specific gene list (See Table VI). However, it is interesting to observe that the most significant terms associated with bicluster 147 include “protein amino acid glycosylation” (*p*-value 1.03E-03), “glycoprotein biosynthesis” (*p*-value 1.48E-03) and “steroid biosynthesis” (*p*-value 3.78E-03). Indeed, steroid/sterol biosynthesis and glycoprotein biosynthesis are linked to plasma membrane and cell wall reconfiguration, which are important aspects of the heat shock response [35] and appear in this profile to be among the first steps of yeast adaptation to heat shock. Also shown in Table VI is the fact that the genes in bicluster 151 are associated by GOToolBox, with high significance, to GO terms such “cell organization and biogenesis” (*p*-values 6.95E-08), “cell cycle” (*p*-value 1.30E-07) and “mitotic cell cycle” (*p*-value 3.32E-06), suggesting cell cycle repression, which is in agreement with growth arrest upon sudden exposure to 37^{o}C. This is consistent with the fact that 16.5% and 12.4% of the down-regulated genes in this bicluster are documented targets of the transcription factors Swi4p and Mbp1p, respectively, both forming complexes with Swi6p to control cell cycle G1-S transition. Finally, bicluster 162 comprises a number of genes involved in RNA and protein synthesis (see Table VI for details). GO terms such as “RNA processing” (p-value 6.65E-38) or “ribosome biogenesis” (p-value 8.23E-63) appear among the most significant GO terms associated to this bicluster. Indeed, the inhibition of ribosome biosynthesis and the repression of rRNA synthesis, associated with the general stress response program, is also a feature of the

TABLE VI

CCC-Biclusters describing transcriptional down-regulated patterns

ID	*Pattern*	*TFs*	*%*	*Relevant GO Terms Enriched*	*p-value*
147	Early drastic down-regulation, followed by rapid up-regulation	Ste12p	16.67	regulation of progression through mitotic cell cycle	4.46E-05*
		Rap1p	15.83	steroid biosynthesis	3.78E-04*
		Swi4p	15.00	biopolymer glycosylation	1.03E-03*
		Rpn4p	13.33	protein amino acid glycosylation	1.03E-03*
		Ino4p	11.67	steroid metabolism	1.05E-03*
				protein targeting to ER	1.33E-03*
				glycoprotein biosynthesis	1.48E-03*
				sterol biosynthesis	1.52E-03*
				glycoprotein metabolism	1.58E-03*
151	Early drastic down-regulation, followed by late up-regulation	Mbp1p	12.37	cell organization and biogenesis	6.95E-08
		Arr1p	11.34	cell cycle	1.30E-07
		Rpn4p	9.79	nucleobase, nucleoside, nucleotide and nucleic acid metabolism	7.04E-07
		Ino4p	8.76	biopolymer metabolism	8.41E-07
				mitotic cell cycle	3.32E-06
				regulation of biological process	7.17E-06
				regulation of physiological process	1.08E-05
124	Delayed down-regulation	Sfp1p	33.00	ribosome biogenesis	8.23E-63
		Rap1p	20.89	ribosome biogenesis and assembly	8.38E-62
		Rpn4p	18.91	cytoplasm organization and biogenesis	8.38E-62
		Arr1p	16.19	rRNA processing	1.42E-48
		Fhl1p	12.36	RNA metabolism	3.36E-40
				RNA processing	6.65E-38
				rRNA metabolism	1.58E-35
				organelle organization and biogenesis	6.49E-30
				nucleobase, nucleoside, nucleotide and nucleic acid metabolism	1.20E-29
				cell organization and biogenesis	1.79E-23

heat shock response [9]. In agreement with this observation, the transcription factors Sfp1p and Rap1, associated with ribosome biogenesis and rRNA synthesis, appear as the main regulators of these two biclusters.

This brief overview of the biological significance of the CCC-biclusters, generated with real data, shows that this method is able to point out the major aspects of a given transcriptional response. In this particular case, the previously identified transcriptional regulons and biological processes underlying the yeast heat shock response emerged from this biclustering analysis. This analysis further emphasizes the importance of obtaining time-course expression profiles to fully understand the several steps that constitute a given stress response and of using suitable computational methods, such as the one described herein. In this analysis we were able to differentiate a number of different expression profiles, contributing to scrutinize, step by step, the yeast cell response to heat shock. Being a thoroughly studied theme, the conclusions from this analysis were not surprising but support the idea that CCC-Biclustering is a powerful tool for the analysis of time-course global expression data.

VIII. Conclusions and future work

This work opened several promising directions for future research. The most immediate direction for development is related with the discovery of imperfect CCC-Biclusters, that is, CCC-Biclusters that allow a given number of errors. Extending the algorithm to handle time-lagged CCC-Biclusters is also a possibility that will be analyzed if the question of time-lagging activation is deemed relevant to the identification of regulatory networks.

The most promising direction for medium and long term research, however, is related with the development of methods for the identification of regulatory networks that use the information about co-regulated genes obtained using biclustering algorithms. This will require the integration of information from different sources, that include gene expression, sequence data and information from the scientific literature. We believe this problem is one of the most important and challenging problems that will be addressed in this area in the coming decade.

Notes

Parts of this work have appeared previously in [23]. However, the statistical methods for ranking CCC-Biclusters, the filtering method for removing highly overlapping biclusters, and the experimental validation with real data is original. This work was partially supported by projects POSI/SRI/47778/2002, BioGrid, POSI/EIA/57398/2004, DBYeast, and POSI/BIO/56838/ 2004 financed by FCT, Fundação para a Ciência e Tecnologia, and the POSI program. The software described in this paper, as well as the datasets and examples used is available at *http://www.inesc-id.pt/kdbio/software/ccc-biclustering*. This web page will be updated in order to provide documentation related to the use of the software, and an user-friendly interface to CCC-Biclustering enabling an intuitive use of the algorithm in real datasets.

References

[1] Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–2503, 2004.

[2] C. Becquet, S. Blachon, B. Jeudy, J-F. Boulicaut, and O. Gandrillon. Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data. *Genome Biology*, 3(12), 2002.

[3] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: The order–preserving submatrix problem. In *Proc. of the 6th International Conference on Computacional Biology*, pages 49–57, 2002.

[4] S. Bleuler and E. Zitzler. Order preserving clustering over multiple time course experiments. In *Proc. of the 3rd European workshop on evolutionary computation and bioinformatics*, pages 33–43, 2005.

[5] Y. Cheng and G. M. Church. Biclustering of expression data - supplementary information. http://arep.med.harvard.edu/biclustering/, [September 20, 2006].

[6] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proc. of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[7] I. G. Costa, A. Schönhuth, and A. Schliep. The Graphical Query Language: a tool for analysis of gene expression time-courses. *Bioinformatics*, 21(10):2544–2545, 2004.

[8] S. Erdal, O. Ozturk, D. Armbruster, H. Ferhatosmanoglu, and W. C. Ray. A time series analysis of microarray data. In *Proc. of the 4rd IEEE Symposium on Bioinformatics and Bioengineering*, pages 366–374, 2004.

[9] 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*, 11:4241–4257, 2000.

[10] D. Gusfield. *Algorithms on strings, trees, and sequences*. Computer Science and Computational Biology Series. Cambridge University Press, 1997.

[11] L. Ji and K. Tan. Mining gene expression data for positive and negative co-regulated gene clusters. *Bioinformatics*, 20(16):2711–2718, 2004.

[12] L. Ji and K. Tan. Identifying time-lagged gene clusters using gene expression data. *Bioinformatics*, 21(4):509–516, 2005.

[13] N. Kobayashi and K. McEntee. Identification of cis and trans components of a novel heat shock stress regulatory pathway in saccharomyces cerevisiae. *Molecular Cell Biology*, 13:248–256, 1993.

[14] M. Koyuturk, W. Szpankowski, and A. Grama. Biclustering gene-feature matrices for statistically significant dense patterns. In *Proc. of the 8th International Conference on Research in Computational Molecular Biology*, pages 480–484, 2004.

[15] A. Kwon, H. Hoos, and R. Ng. Inference of transcriptional regulation relationships from gene expression data. *Bioinformatics*, 19(8):905–912, 2003.

[16] J. Liu, W. Wang, and J. Yang. Biclustering in gene expression data by tendency. In *Proc. of the 3rd International IEEE Computer Society Computational Systems Bioinformatics Conference*, pages 182–193, 2004.

[17] J. Liu, W. Wang, and J. Yang. A framework for ontology-driven subspace clustering. In *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 623–628, 2004.

[18] J. Liu, W. Wang, and J. Yang. Gene ontology friendly biclustering of expression profiles. In *Proc. of the 3rd IEEE Computational Systems Bioinformatics Conference*, pages 436–447, 2004.

[19] J. Liu, W. Wang, and J. Yang. *Mining Sequential Patterns from Large Data Sets*, volume 18 of *Series of Advances in Database Systems*. Kluwer, 2005.

[20] S. Lonardi, W. Szpankowski, and Q. Yang. Finding biclusters by random projections. In *Proc. of the 15th Annual Symposium on Combinatorial Pattern Matching*, pages 102–116, 2004.

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

[22] S. C. Madeira and A. L. Oliveira. An evaluation of discretization methods for non-supervised analysis of time-series gene expression data. Technical Report 42, INESC-ID, December 2005.

[23] S. C. Madeira and A. L. Oliveira. A linear time algorithm for biclustering time series expression data. In *Proc. of 5th Workshop on Algorithms in Bioinformatics*, pages 39–52. Springer Verlag, LNCS/LNBI 3692, 2005.

[24] D. Martin, C. Brun, E. Remy, P. Mouren, D. Thieffry, and B. Jacq. GOToolBox: functional investigation of gene datasets based on gene ontology. *Genome Biology*, 5(12):R101, 2004.

[25] E. McCreight. A space economical suffix tree construction algorithm. *Journal of the ACM*, 23:262–272, 1976.

[26] G. J. McLachlan, K. Do, and C. Ambroise. *Analysing microarray gene expression data*. Wiley Series in Probability and Statistics, 2004.

[27] I. Van Mechelen, H. H. Bock, and P. De Boeck. Two-mode clustering methods: a structured overview. *Statistical Methods in Medical Research*, 13(5):979–981, 2004.

[28] C. Mollër-Levet, S. Cho, and O. Wolkenhauer. DNA microarray data clustering based on temporal variation: FCV and TSD preclustering. *Applied Bioinformatics*, 2(1):35–45, 2003.

[29] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proc. of the Pacific Symposium on Biocomputing*, volume 8, pages 77–88, 2003.

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

[31] R. G. Pensa, C. Leschi, J. Besson, and J. Boulicaut. Assessment of discretization techniques for relevant pattern discovery from gene expression data. In *4th Workshop on Data Mining in Bioinformatics*, 2004.

[32] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(10):1282–1283, 2006.

[33] P.Weiner. Linear pattern matching algorithms. In *Proc. of the 14th IEEE Symposium on Switching and Automata Theory*, pages 1–11, 1973.

[34] Q. Sheng, Y. Moreau, and B. De Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19(2):196–205, 2003.

[35] T. M. Swan and K. Watson. Stress tolerance in a yeast sterol auxotroph: role of ergosterol, heat shock proteins and trehalose. *FEMS Microbiology Letters*, 7:169–191, 1998.

[36] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(1):136–144, 2002.

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

[38] M. C. Teixeira, P. Monteiro, P. Jain, S. Tenreiro, A. R. Fernandes, N. P. Mira, M. Alenquer, A. T. Freitas, A. L. Oliveira, and I. Sá-Correia. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in saccharomyces cerevisiae. *Nucleic Acids Research*, 34:D446–D451, January 2006.

[39] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14:249–260, 1995.

[40] C. Wu, Y. Fu, T. M. Murali, and S. Kasif. Gene expression module discovery using Gibbs sampling. *Genome Informatics*, 15(1):239–248, 2004.

[41] Y. Zhang, H. Zha, and C. H. Chu. A time-series biclustering algorithm for revealing co-regulated genes. In *Proc. of the 5th IEEE International Conference on Information Technology: Coding and Computing*, pages 32–37, 2005.