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

Abstract—Although most biclustering formulations are NP-hard, in time series expression data analysis, it is reasonable to restrict the problem to the identification of maximal biclusters with contiguous columns, which correspond to coherent expression patterns shared by a group of genes in consecutive time points. This restriction leads to a tractable problem. We propose an algorithm that finds and reports all maximal contiguous column coherent biclusters in time linear in the size of the expression matrix. The linear time complexity of CCC-Biclustering relies on the use of a discretized matrix and efficient string processing techniques based on suffix trees. We also propose a method for ranking biclusters based on their statistical significance and a methodology for filtering highly overlapping and, therefore, redundant biclusters. We report results in synthetic and real data showing the effectiveness of the approach and its relevance in the discovery of regulatory modules. Results obtained using the transcriptomic expression patterns occurring in *Saccharomyces cerevisiae* in response to heat stress 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.

1 Introduction

Recent developments in DNA chips enable the simultaneous measurement of the expression level of a large number of genes (virtually all the genes of an organism) for a given experimental condition [27]. In this context, several nonsupervised machine learning methods have been extensively used in the analysis of gene expression data obtained from these microarray experiments. More recently, biclustering, a technique that aims at finding subgroups of genes that exhibit highly correlated behaviors in a subgroup of conditions has emerged as a way to identify potential regulatory mechanisms. 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 have been extensively studied and documented [6], [21], [28]. We believe that 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], [28]. Most specific versions of this problem have been shown to be NP-hard [31], 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 [37]; limits are imposed on the size of the biclusters that can be found, in order to obtain reasonable runtimes. Furthermore, the inherent difficulty of the biclustering 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 have also led many authors to a formulation based on a discretized matrix [3], [12], [14], [16], [17], [18], [19], [20], [23], [30], [33], [35], [37], [41]. Unfortunately, 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. In this experimental setup, the researcher is particularly interested in biclusters with contiguous columns, corresponding to samples taken in consecutive instants of time, which identify coherent expression patterns shared by a group of genes in consecutive time points.

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

- *S.C. Madeira is with the Departamento de Informática, Universidade da Beira Interior, Rua Marques D'Ávila e Bolama, 6201-001 Covilhã, Portugal. She is also with the Knowledge Discovery and Bioinformatics (KDBIO) Group, INESC-ID, Rua Alves Redol 9, Apartado 13069, 1000-029 Lisbon, Portugal, and with the Instituto Superior Técnico, Lisbon Technical University, Av. Rovisco Pais, P-1049-001 Lisboa, Portugal. E-mail: smadeira@kdbio.inesc-id.pt.*
- *M.C. Teixeira and I. Sá-Correia are with the Biological Sciences Research Group, Departamento de Engenharia Química e Biológica, Instituto Superior Técnico, Lisbon Technical University, Av. Rovisco Pais, P-1049-001 Lisboa, Portugal. E-mail: {mnpct, isacorreia}@ist.utl.pt.*
- *A.L. Oliveira is with the the Knowledge Discovery and Bioinformatics (KDBIO) Group, INESC-ID, Rua Alves Redol 9, Apartado 13069, 1000-029 Lisbon, Portugal, and also with the Instituto Superior Técnico, Lisbon Technical University, Av. Rovisco Pais, P-1049-001 Lisboa, Portugal. E-mail: aml@inesc-id.pt.*

Manuscript received 10 Apr. 2007; revised 10 Oct. 2007; accepted 18 Feb. 2008; published online 21 Mar. 2008.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2007-04-0043. Digital Object Identifier no. 10.1109/TCBB.2008.34.

1545-5963/10/$26.00 © 2010 IEEE Published by the IEEE CS, CI, and EMB Societies & the ACM

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], [42] and their relevance in the identification of regulatory processes. The biological support for this reasoning is the key observation 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. We also propose a statistical test that can be used to score the identified CCC-Biclusters and to sort them by increasing value of the probability that they have appeared by a random coincidence of events and a method to filter and remove highly overlapping and, therefore, redundant, CCC-Biclusters. We show the effectiveness of the proposed approach in recovering planted CCC-Biclusters in synthetic data sets and its ability to find regulatory modules in real data.

This paper is organized as follows: Section 2 surveys the related work. Section 3 provides the problem formulation. Section 4 describes the algorithm. Section 5 proposes a scoring schema for CCC-Biclusters based on statistical significance and similarity measures. Section 6 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 7 shows a comparison with a heuristic approach developed for biclustering time series expression data and an application of CCC-Biclustering to the discovery of regulatory modules in yeast by using 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 coexpressed genes, which are shown to be coregulated by a set of common transcription factors (TFs) and highly functionally enriched in one or more Gene Ontology (GO) terms. Finally, Section 8 presents the conclusions and directions for future work.

2 Related Work

2.1 Biclustering Algorithms for Time Series Expression Data

Although a large number of biclustering algorithms have been proposed to address the general problem of biclustering [21], [28], to date and to our knowledge, only two recent proposals have addressed the problem of biclustering in time series expression data [12], [42].

Zhang et al. [42] proposed the CC-TSB algorithm, which is based on the work by Cheng and Church [6] and 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 in Section 7.1.

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 our method. The exact complexity of their algorithm is hard to estimate from the description, but it is at least $\Omega(|R||C|^2)$, and hence, the CCC-Biclustering algorithm we propose is at least a factor of $\Theta(|C|)$ times faster.[1]

2.2 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], [33], [37], percentage of values [2], [32], equal-width intervals [2], [32], equal frequency [35], linear order between the conditions [3], [4], [16], [17], [18], [19], and statistically significant states [30], [41].

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], [29]. These techniques use either two [8], [15], [29] 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 Costa et al. [7], who claim 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.

3 Problem Definition

3.1 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, which represent distinctive activation levels. In the simpler case, Σ may contain only two symbols, one used for *no-regulation* and the 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.

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

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, as described as follows:

$$A''_{ij} = \begin{cases} \frac{A'_{i(j+1)} - A'_{ij}}{|A'_{ij}|}, & \text{if } A'_{ij} \neq 0, \\ -1, & \text{if } A'_{ij} = 0 \text{ and } A'_{i(j+1)} < 0, \\ 1, & \text{if } A'_{ij} = 0 \text{ and } A'_{i(j+1)} > 0, \\ 0, & \text{if } A'_{ij} = 0 \text{ and } A'_{i(j+1)} = 0. \end{cases} \quad (1)$$

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 $t > 0$:[2]

$$A_{ij} = \begin{cases} D, & \text{if } A''_{ij} \leq -t, \\ U, & \text{if } A''_{ij} \geq t, \\ N, & \text{otherwise.} \end{cases} \quad (2)$$

3.2 Biclusters in Discretized Expression Data

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

Definition 1 (bicluster and trivial bicluster). *A bicluster $B = (I, J)$ is a submatrix 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* CC-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 [31].

3.3 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], [42]. 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* CCC-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.*

2. This discretization technique should be preceded by a normalization step, which normalizes each gene expression pattern to a given mean and standard deviation, since the binning process uses variations between consecutive time-points and the same threshold t for all the genes in matrix A'. As in the work of Ji and Tan, we standardized A' to zero mean and unit standard deviation by gene and set the threshold t to the standard deviation value ($t = 1$).

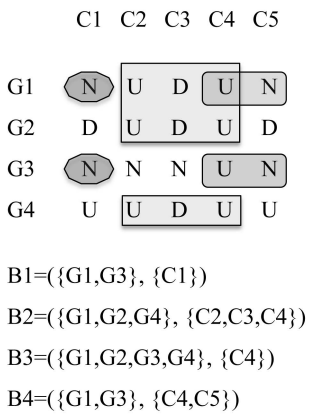

Fig. 1. Example of a discretized matrix with all maximal CCC-Biclusters with at least two rows (B1 to B4). The strings N, UDU, U, and UN correspond to the expression patterns of the maximal CCC-Biclusters B1, B2, B3, and B4, respectively.

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 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 it, 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, which we present without proof.

Lemma 1. *The maximal CCC-Biclusters are exactly the CCC-Biclusters that are right-, left-, and row-maximal.*

Fig. 1 shows an example of a discretized expression matrix together with its maximal CCC-Biclusters. For clarity, we omit the maximal CCC-Biclusters with only one row (gene), which are uninteresting from a biological point of view.

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.

4 Biclustering Time Series Expression Data Using Suffix Trees

4.1 Strings and Suffix Trees

The definitions used in this section are adapted from the work of Gusfield [10], a well-known 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)*

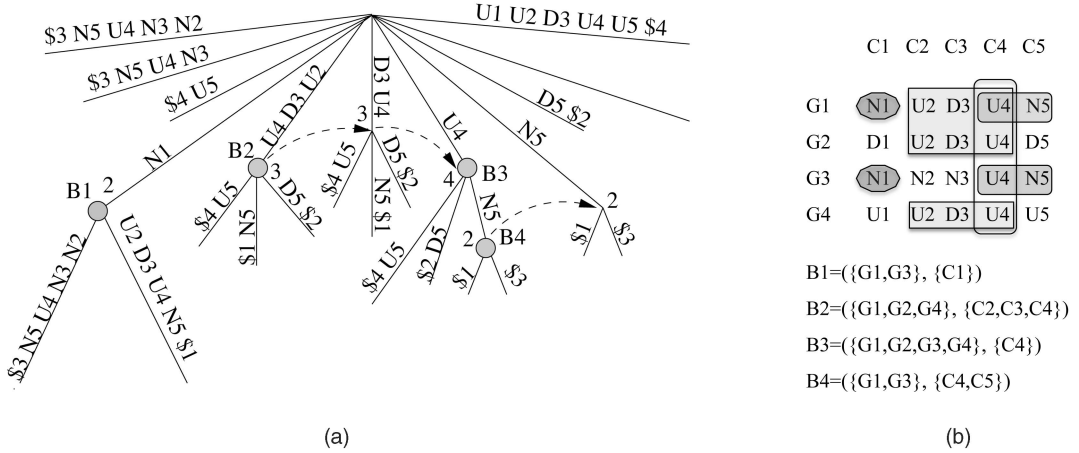

Fig. 2. (a) Generalized suffix tree for the discretized and transformed matrix on the right. The circles identify the maximal CCC-Biclusters with at least two rows (B1 to B4). (b) CCC-Biclusters B1 to B4 shown in the matrix after alphabet transformation as subsets of rows and columns. Remember that the strings N, UDU, U, and UN correspond to the expression patterns of the maximal CCC-Biclusters B1, B2, B3, and B4, respectively.

written contiguously from left to right. For any string S (with $|S|$ symbols), $S[i..j]$ $(i \geq 0, j \leq |S|)$ 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 starting with the same symbol, and 3) for any leaf i, the label of the path from the root to the leaf i exactly spells out the suffix of S starting 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), which 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 using several algorithms [26], [34], [40]. 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 (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 10 (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 u with string label α, then a pointer from v to u is called a suffix link. As a special case, if α is empty, then $x\alpha$ has a suffix link leading to the root.*

4.2 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 $\{S_1, \ldots, S_{|R|}\}$ obtained by applying this alphabet transformation to each row A_{iC} in matrix A. Fig. 2b shows the result of this transformation applied to the discretized matrix shown in Fig. 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 $\{S_1, \ldots, S_{|R|}\}$. 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.

Consider a node v in T together with its string depth $P(v)$. Let $L(v)$ denote the number of leaves in the subtree rooted at v, in case v is an internal node.

Fig. 2a illustrates the generalized suffix tree obtained from the strings that correspond to the rows of the matrix shown in Fig. 2b. For clarity, this figure does not contain the leaves that represent string terminators that are direct daughters of the root. Each nonterminal 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, right-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 at v. Therefore, each internal 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 edge label is simply a terminator also identify CCC-Biclusters. However, some of these CCC-Biclusters are nonmaximal (internal nodes with string labels $D3U4$ and $N5$ and all leaf nodes whose $P(v)$ is different from the number of columns in A). We will show that an internal node corresponds to a

maximal CCC-Bicluster *iff* there is no incoming suffix link from a node with the same value of $L(v)$. As such, only the internal nodes with string labels $N1$, $U2D3U4$, $U4$, and $U4N5$ identify maximal CCC-Biclusters with at least two rows. These nodes correspond to the maximal CCC-Biclusters $B1=(\{G1,G3\},\{C1\})$, $B2=(\{G1,G2,G4\},\{C2,C3,C4\})$, $B3=(\{G1,G2,G3,G4\},\{C4\})$, and $B4=(\{G1,G3\},\{C4,C5\})$. 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. Moreover, the value of $P(v)$ and the first symbol in the string label of v provide the information needed to identify the set of contiguous columns.

Using the illustrative example in Fig. 2, we have shown that all internal 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 sketches of the proofs, the two lemmas that lead to the theorem that supports our linear time biclustering algorithm. For the sake of clarity, we will consider only CCC-Biclusters with at least two rows. CCC-Biclusters with one row are trivial and uninteresting, and considering them would unnecessarily complicate the proofs.

Lemma 2. *Every right-maximal, row-maximal CCC-Bicluster with at least two rows corresponds to one internal node in T, and every internal node in T corresponds to one right-maximal, row-maximal CCC-Bicluster with at least two rows.*

Proof. Let B be a right-maximal, row-maximal CCC-Bicluster with at least two rows. 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 (possibly the terminator symbol) that differs from the symbol in the other rows in column j, which is the column after the last column in B. Therefore, there is an internal node in T that matches B, and the string label of that node is the string that defines B. For the other direction, each internal node in T defines one string, which is present in two or more rows in matrix A, and at least one of these rows has a symbol different from the others in column j, which is again the column after the last column in B. If this was not the case, there would not exist an internal node in T. □

To distinguish the nodes that correspond to left-maximal CCC-Biclusters, we introduce the following definition.

Definition 11 (MaxNode). *An internal node v of T is called a MaxNode* iff *it satisfies one of the following conditions:*

1. *It does not have incoming suffix links.*
2. *It has incoming suffix links only from nodes u_i such that for every node u_i, $L(u_i) < L(v)$.*

Consider now two nodes in the suffix tree, v_1 and v_2. Notice that if there is a suffix link from node v_1 to node v_2, the CCC-Bicluster defined by v_2 contains one less column than the CCC-Bicluster defined by v_1. This leads us to the last lemma.

Lemma 3. *An internal node in T corresponds to a left-maximal CCC-Bicluster* iff *it satisfies Definition 11.*

Proof. The string label $s = P(v)$ of a node v satisfying the conditions of the lemma defines a CCC-Bicluster B in A with at least two rows. If node v has no incoming suffix links, then it corresponds to either a CCC-Bicluster starting at column 1 in A or is defined by a string s such that xs is present in a single row in A (xs is the string label of a leaf node in T). Therefore, B is left-maximal, since in both cases, it cannot be extended to the left without losing rows. If node v has incoming suffix links from nodes u_i such that $L(u_i) < L(v)$, then B is left-maximal, since the CCC-Biclusters defined by nodes u_i have less rows than B. For the other direction, if an internal node v has one incoming suffix link from a node u such that $L(u) = L(v)$ ($L(u) > L(v)$ can never happen), then the CCC-Bicluster B defined by v can be extended to the left, keeping the same set of rows. Therefore, v does not define a left-maximal CCC-Bicluster. □

We now present our main result.

Theorem 1. *Every maximal CCC-Bicluster with at least two rows corresponds to an internal node in the generalized suffix tree T that satisfies Definition 11, and each of these internal nodes defines a maximal CCC-Bicluster with at least two rows.*

Proof. Let B be a maximal CCC-Bicluster with at least two rows. By Lemma 2, this CCC-Bicluster corresponds to an internal node v in T. Since B is left-maximal (Lemma 1), node v must satisfy Definition 11 (Lemma 2). For the other direction, let v be an internal node in T satisfying Definition 11. By Lemma 2, node v corresponds to a right-maximal, row-maximal bicluster. If v also satisfies Definition 11, then B is also left-maximal, and therefore, B is maximal. □

4.3 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 for the set of strings $\{S_1,\ldots,S_{|R|}\}$, obtained using the alphabet transformation described in Section 4.2, and checks, for each internal node, whether the conditions of Theorem 1 are met. Nodes that do not meet the required conditions are marked as invalid in line 10. All the remaining internal nodes correspond to maximal CCC-Biclusters and are reported.

Algorithm 1. CCC-Biclustering
input: Discretized gene expression matrix A
1 Perform alphabet transformation and obtain $\{S_1,\ldots,S_{|R|}\}$.
2 Build a generalized suffix tree T for $\{S_1,\ldots,S_{|R|}\}$.
3 **for** *each internal node* $v \in T$ **do**
4 Mark v as "Valid."
5 Compute the string depth $P(v)$.
6 **for** *each internal node* $v \in T$ **do**
7 Compute the number of leaves $L(v)$ in the subtree rooted at v.
8 **for** *each internal node* $v \in T$ **do**

9 **if** *there is a suffix link from* v *to a node* u ***and*** $L(u) = L(v)$ **then**
10 Mark node u as "Invalid."
11 **for** *each internal node* $v \in T$ **do**
12 **if** v *is marked as "Valid"* **then**
13 Report the CCC-Bicluster that corresponds to v.

4.4 Complexity Analysis of CCC-Biclustering and Implementation Issues

With appropriate data structures at the nodes and using Ukkonen's algorithm [40], the suffix tree construction time is linear on the size of the input matrix, $O(|R||C|)$. The remaining steps of the CCC-Biclustering 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 that becomes important when the alphabet is large [10]. 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 4.2 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 Fig. 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 string 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 nonlinear 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, we guarantee that the branching at the root is performed in constant time, and the total complexity of CCC-Biclustering is $O(|R||C|)$.

5 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 that 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, from both 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-Bicluster. CCC-Biclusters are sorted by increasing order of the computed p-value, and if several of them are very similar, only the most significant ones are kept.

5.1 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 $|R|$ genes and $|C|$ time points. The value of this probability 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: $P = \sum_{j=k}^{n} p^j (1-p)^{n-j}$.

The statistical significance of a CCC-Bicluster B is the p-value(B), which is computed by obtaining the probability of a random occurrence under H_0 of the expression pattern $p_B, k = |I| - 1$ times in $n = |R| - 1$ independent trials, where I is the number of genes in B, and $|R|$ 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, $P(p_B)$, 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 $B = (\{G1, G2, G4\}, \{C2, C3, C4\})$, corresponding to CCC-Bicluster B2 in Fig. 2b, with expression pattern $p_B = U2D3U4$, then

$$P(p_B) = P(U2D3U4) = P(U2)P(D3|U2)P(U4|D3),$$

where

$$P(U2) = \frac{|U2|}{|R|},$$
$$P(D3|U2) = \frac{P(U2D3)}{P(U2)} = \frac{|U2D3|}{|U2|},$$

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 Fig. 2b. 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 $U4$.

To speed up the computation, the value of $P(p_B)$ for each pattern p_B can be computed and stored in the internal nodes

while traversing the suffix tree and at the same time that the maximal CCC-Biclusters are identified. Note that we do not need to compute values of $P(p_B)$ for the leaf nodes, since these CCC-Biclusters with only one row are not reported.

5.2 Similarity Measure

In order to compute the similarity measure 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 two CCC-Biclusters in terms of both genes and conditions and is defined as follows:

$$J(B_1, B_2) = \frac{|B_1 \bigcap B_2|}{|B_1 \bigcup B_2|} = \frac{|B_{11}|}{|B_{01}| + |B_{10}| + |B_{11}|}, \qquad (3)$$

where

$$B_{11} = \Big\{(i,j) : (i,j) \in B_1 \bigwedge (i,j) \in B_2\Big\},$$
$$B_{10} = \Big\{(i,j) : (i,j) \in B_1 \bigwedge (i,j) \notin B_2\Big\},$$

and

$$B_{01} = \Big\{(i,j) : (i,j) \notin B_1 \bigwedge (i,j) \in B_2\Big\},$$

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 measure as defined in (3) 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

$$|B_1 \bigcup B_2| = |B_1| + |B_2| - |B_1 \bigcap B_2|.$$

To speed up the process, only the CCC-Biclusters that have been identified as statistically significant are processed during this last phase of the filtering process.

6 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.

We have first validated the predicted linear time complexity by generating synthetic matrices with random values, in which 10 CCC-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 × columns) to 1,000 × 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 CCC-Biclusters overlapping with the planted biclusters and a large number of artifacts that resulted from random coincidences in the data matrix.

TABLE 1
CCC-Biclusters Planted in the Synthetic 1,000 × 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 1,000 rows by 50 columns example. For this example, two experiments were carried out. In the first experiment, no planted CCC-Biclusters existed. In this case, a total of 42,030 maximal nontrivial CCC-Biclusters were identified. *None* of these CCC-Biclusters passed the statistical significance test described in Section 5.1 at a 1 percent level (after Bonferroni correction). In the second experiment, 10 CCC-Biclusters with the aforementioned dimensions were planted. See Table 1 for details.

CCC-Biclustering identified 40,461 maximal nontrivial CCC-Biclusters in a few seconds. From these, 165 passed the statistical significance test at the 1 percent level (after Bonferroni correction). Most of these are variations of the planted CCC-Biclusters that still pass the statistical significance test because they include a large part of the planted pattern. Table 2 shows that after sorting the CCC-Biclusters using the statistical significance p-value described in Section 5.1 and filtering CCC-Biclusters whose similarity measure, as defined in Section 5.2, is above 25 percent, the proposed approach is able to identify the planted CCC-Biclusters as the top 10 CCC-Biclusters.[3] After filtering the CCC-Biclusters with similarities above 25 percent, only 37 CCC-Biclusters had a (Bonferroni-corrected) p-value below 0.01.[4]

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

3. CCC-Biclusters with IDs 2,438, 15,158, and 14,145 have lost some genes, relative to the original planted CCC-Biclusters. This was 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.

4. 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 measure described in Section 5.2.

TABLE 2
Top 10 CCC-Biclusters Discovered after Filtering CCC-Biclusters with Overlapping above 25 Percent (after Sorting the Discovered CCC-Biclusters Using the Statistical Significance p-Value)

ID	*Expression Pattern*	*#Time-Points (first-last)*	*#Genes*	*p-Value*	*Closest Planted CCC-Bicluster*
24,475	DDNUDDNDDNDD	12 (35-46)	18	4.13E-56	MATCH 7
5,790	UDNNDUNUDNDD	12 (26-37)	19	1.37E-55	MATCH 8
17,868	NUNUUNDDNDNU	12 (21-32)	19	3.72E-55	MATCH 1
25,020	DNDNDDNNNNDD	12 (33-44)	18	6.10E-54	MATCH 10
2,438	UDDUNUDDU	9 (37-45)	23	8.43E-40	MATCH 6 LOST 1 GENE[3]
15,158	NUNDNNDDUNN	11 (30-40)	16	4.08E-37	MATCH 3 LOST 3 GENES[3]
34531	UNUDUDNDUU	10 (2-11)	16	7.17E-34	MATCH 5
16,797	NDUNDNUUU	9 (25-33)	20	8.20E-33	MATCH 9
38,344	NNUDUNNNU	9 (1-9)	15	5.29E-23	MATCH 2
14,145	NDUUDDDD	8 (34-41)	14	1.48E-17	MATCH 4 LOST 1 GENE[3]

TABLE 3
Comparison of the Results Obtained by the CCC-Biclustering and CC-TSB Algorithm [42]

CC-TSB Algorithm (sorted by MSR)			CCC-Biclustering (sorted by MSR)			CCC-Biclustering (sorted by p-Value, overlap $\leq$ 25 Percent)			
#Time-Points (first-last)	*#Genes*	MSR	*#Time-Points (first-last)*	*#Genes*	MSR	*#Time-Points (first-last)*	*#Genes*	*MSR*	*p-Value*
17 (1-17)	1,447	411.7	3 (11-13)	49	26.3	5 (9-13)	88	165.2	2.40E-24
16 (2-16)	1,016	7,039.1	5 (11-15)	24	45.0	9 (5-13)	21	104.6	5.30E-24
17 (1-17)	1,730	42,830.3	2 (14-15)	558	47.7	7 (11-17)	31	107.6	1.02E-20
17 (1-17)	1,366	47,338.4	4 (3-6)	24	48.2	10 (2-11)	16	116.5	4.72E-20
17 (1-17)	1,671	47,887.9	6 (11-16)	20	48.4	6 (8-13)	189	241.8	1.74E-18

7 Experimental Results with Real Data Sets

7.1 Comparison with Heuristic Algorithms

The CC-TSB algorithm [42] described in Section 2 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 data set and using the same parameters used by the authors.

In this context, we used the yeast cell-cycle data set publicly available [5], described by Tavazoie et al. [38] and processed by Cheng and Church [6]. We used 2,884 genes selected by Cheng and Church [6] and removed the ORFS with missing values and the ones that no longer exist in the Saccharomyces Genome Database (SGD). As in [42], 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 matrix using the technique based on transitions between time points proposed by Ji and Tan [11], [12] and described in Section 3.1.[5]

In Table 3, we report the sizes and the mean squared residue (MSR) for the top five biclusters (evaluated by the MSR, which is the merit function minimized in the 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 a 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 five CCC-Biclusters discovered by sorting the results using the p-value described in Section 5.1 and filtering CCC-Biclusters with similarities above 25 percent. 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 of 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 data set. This means that this algorithm is in fact looking for gene clusters and not biclusters, which makes it useless for the purposes of identifying local patterns.

7.2 Application to the Identification of Regulatory Modules

To assess the biological relevance of CCC-Biclusters in real data, we used a data set from Gasch et al. [9], concerning the yeast response to heat shock. This data set comprises five different time-points along the first hour of exposure to 37 °C (0′, 5′, 15′, 30′, and 60′). The first time-point is an

5. Before the discretization, we normalize the data to zero mean and unit standard deviation, as described in Section 3.1. However and in order to compare our results with those of the CC-TSB algorithm (which computes the MSR values using the original values in A'), we computed the MSR values for the CCC-Biclusters presented in Table 3 using the original (not normalized) expression values in A'. We also set the threshold t to the standard deviation value ($t = 1$).

TABLE 4
Summary of the CCC-Biclusters Passing the Statistical Test at the 1 Percent Level after Bonferroni Correction (after Filtering CCC-Biclusters with Similarity above 25 Percent)

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	0	1.31E-04 (3)	87.91
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	0	4	2.41E-05 (4)	11.49
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

average of three replicates of time zero. The data set was preprocessed as in Section 7.1.

Since we were interested in CCC-Biclusters with high statistical significance, the set of 167 maximal CCC-Biclusters discovered (using 1.7 seconds of a 2.2-GHz Intel Core 2 Duo) was then sorted in ascending order according to the statistical p-value described in Section 5.1. From these, only 25 were considered as highly significant at the 1 percent level after applying the Bonferroni correction for multiple testing. In order to avoid the analysis of highly overlapping CCC-Biclusters, we then computed the similarities between the sorted CCC-Biclusters using the Jaccard similarity score, as described in Section 5.2, and filtered CCC-Biclusters with a similarity greater than 25 percent. This filtering process removed 9 of the 25 CCC-Biclusters originally selected.

Table 4 shows a summary of the remaining 16 CCC-Biclusters analyzed using the GO annotations obtained using the GoToolBox [25]. 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 a 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].[6]

From these 16 CCC-Biclusters, six (Tables 5 and 6) were analyzed in more detail, corresponding to chronological expression patterns selected as described below (Figs. 3 and 4). For these CCC-Biclusters selected for describing either transcriptional up-regulation or down-regulation patterns, we analyzed in detail the GO annotations together with information about transcriptional regulation available in the YEASTRACT database [39].[7]

7.2.1 CCC-Biclusters Describing Transcriptional Up-Regulation Patterns

The first three CCC-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 Fig. 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 5). 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 of 1.65E-04) and "regulation of transcription from RNA polymerase II promoter" (p-value of 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 TFs associated with 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 5, reveals the occurrence of highly significant terms, including "carbohydrate metabolism" (p-values of 7.33E-08

6. Note that 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.

7. In Tables 5 and 6, presented in the next sections, column 1 identifies the CCC-Bicluster, column 2 lists relevant TFs coregulating the set of genes in the CCC-Biclusters, column 3 lists the percentage of genes in the CCC-Biclusters that are coregulated 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-values not passing the Bonferroni test at the 1 percent level are marked with *.

TABLE 5
CCC-Biclusters Describing Transcriptional Up-Regulated Patterns

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

TABLE 6
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	Swi4p	16.50	cell organization and biogenesis	6.95E-08
		Mbp1p	12.37	cell cycle	1.30E-07
		Arr1p	11.34	nucleobase, nucleoside, nucleotide and nucleic acid metabolism	7.04E-07
		Rpn4p	9.79	biopolymer metabolism	8.41E-07
		Ino4p	8.76	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

and 4.87E-21) or "energy derivation by oxidation of organic compounds" (p-values of 3.60E-06 and 3.92E-20), related to energy generation, and "response to stimulus" (p-values of 3.28E-08 and 1.51E-16) or "response to stress" (p-values of 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 [39], each of the three referred CCC-Biclusters was grouped based on the sharing of specific transcriptional regulators mediating the coregulation 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 percent, 23 percent, and 19 percent of the genes in CCC-Biclusters 39, 27, and 14, respectively. Moreover, in agreement with previous knowledge, Msn2p

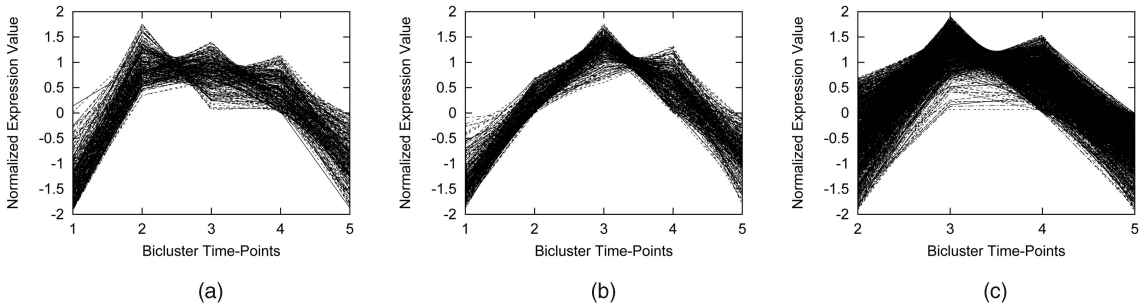

Fig. 3. Expression patterns of the CCC-Biclusters describing transcriptional up-regulation. (a) CCC-Bicluster 39. (b) CCC-Bicluster 27. (c) CCC-Bicluster 14.

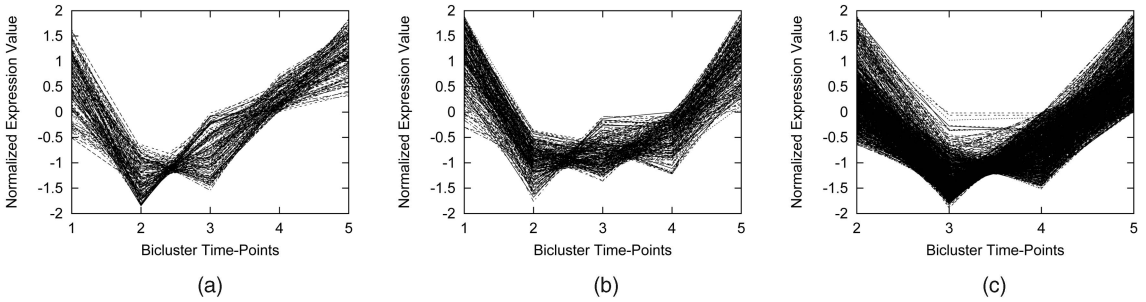

Fig. 4. Expression patterns of the CCC-Biclusters describing transcriptional down-regulation. (a) CCC-Bicluster 147. (b) CCC-Bicluster 151. (c) CCC-Bicluster 124.

and Msn4p, regulators of the general stress response in yeast, appear as major contributors to the heat-induced transcriptional activation. Msn2p regulates 14 percent, 21 percent, and 19 percent of the genes in each of the biclusters, respectively [13]. A third TF also presumably implicated in the regulation of the three biclusters is Rpn4p, regulating 14 percent, 18 percent, and 18 percent of the genes in each of the biclusters, respectively. This TF stimulates the expression of the proteasome genes, involved in the degradation of denatured or unnecessary proteins in stressed yeast cells [9].

Although the TFs regulating the three temporal stages of heat-shock-induced coactivated transcription are apparently the same, the majority of the genes in each of the three CCC-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 TF may act on different target genes in different temporal states.

7.2.2 CCC-Biclusters Describing Transcriptional Down-Regulation Patterns

The remaining three CCC-Biclusters analyzed include genes whose expression was down-regulated abruptly during the first 5 minutes of exposure (CCC-Bicluster 147, comprising 144 genes, and CCC-Bicluster 151, comprising 232 genes) and with a short delay, between 5 and 15 minutes of exposure (CCC-Bicluster 124, comprising 904 genes). See Fig. 4 for details.

The GO-based analysis of CCC-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 6). However, it is interesting to observe that the most significant terms associated with CCC-Bicluster 147 include "protein amino acid glycosylation," (p-value of 1.03E-03), "glycoprotein biosynthesis," (p-value of1.48E-03), and "steroid biosynthesis" (p-value of 3.78E-03). Indeed, steroid/sterol biosynthesis and glycoprotein biosynthesis are linked to the plasma membrane and cell wall reconfiguration, which are important aspects of the heat shock response [36] and appear in this profile to be among the first steps of yeast adaptation to heat shock.

Also shown in Table 6 is the fact that the genes in CCC-Bicluster 151 are associated by the GOToolBox, with high significance, to GO terms such "cell organization and biogenesis," (p-value of 6.95E-08), "cell cycle" (p-value of 1.30E-07), and "mitotic cell cycle" (p-value of 3.32E-06), suggesting cell cycle repression, which is in agreement with growth arrest upon sudden exposure to 37 °C. This is consistent with the fact that 16.5 percent and 12.4 percent of the down-regulated genes in this CCC-Bicluster are documented targets of the TFs Swi4p and Mbp1p, respectively, both forming complexes with Swi6p to control cell cycle G1-S transition. Finally, CCC-Bicluster 124 comprises a number of genes involved in RNA and protein synthesis (see Table 6 for details). GO terms such as "RNA processing" (p-value of 6.65E-38) or "ribosome biogenesis" (p-value of 8.23E-63) appear among the most significant GO terms associated to this CCC-Bicluster. Indeed, the inhibition of ribosome biosynthesis and the

repression of rRNA synthesis, associated with the general stress response program, are also features of the heat shock response [9]. In agreement with this observation, the TFs Sfp1p and Rap1, associated with ribosome biogenesis and rRNA synthesis, appear as the main regulators of this CCC-Bicluster.

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.

8 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 allowing up to a given number of errors per gene relative to the string that defines the CCC-Bicluster [24].

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

Acknowledgments

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 are original. The software described in this paper, as well as the data sets and examples used, is available at http://www.kdbio.inesc-id.pt/software/ccc-biclustering. This web page will be updated in order to provide documentation related to the use of the software and a user-friendly interface to CCC-Biclustering, enabling an intuitive use of the algorithm in real data sets. This work was partially supported by Projects POSI/SRI/47778/2002, BioGrid, POSI/EIA/57398/2004, DBYeast, POSI/BIO/56838/2004, and ARN, PTDC/EIA/67726/2006 financed by FCT, Fundação para a Ciencia e Tecnologia, and the POSI program.

References

[1] Z. Bar-Joseph, "Analyzing Time Series Gene Expression Data," *Bioinformatics*, vol. 20, no. 16, pp. 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*, vol. 3, no. 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," *Proc. Sixth Int'l Conf. Computational Biology (RECOMB '02)*, pp. 49-57, 2002.

[4] S. Bleuler and E. Zitzler, "Order Preserving Clustering over Multiple Time Course Experiments," *Proc. Third European Workshop Evolutionary Computation and Bioinformatics*, pp. 33-43, 2005.

[5] Y. Cheng and G.M. Church, "Biclustering of Expression Data—Supplementary Information," http://arep.med.harvard.edu/biclustering/, Sept. 2006.

[6] Y. Cheng and G.M. Church, "Biclustering of Expression Data," *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB '00)*, pp. 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*, vol. 21, no. 10, pp. 2544-2545, 2004.

[8] S. Erdal, O. Ozturk, D. Armbruster, H. Ferhatosmanoglu, and W.C. Ray, "A Time Series Analysis of Microarray Data," *Proc. Fourth IEEE Symp. Bioinformatics and Bioeng. (BIBE '04)*, pp. 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*, vol. 11, pp. 4241-4257, 2000.

[10] D. Gusfield, "Algorithms on Strings, Trees, and Sequences," *Computer Science and Computational Biology Series*. Cambridge Univ. Press, 1997.

[11] L. Ji and K. Tan, "Mining Gene Expression Data for Positive and Negative Co-Regulated Gene Clusters," *Bioinformatics*, vol. 20, no. 16, pp. 2711-2718, 2004.

[12] L. Ji and K. Tan, "Identifying Time-Lagged Gene Clusters Using Gene Expression Data," *Bioinformatics*, vol. 21, no. 4, pp. 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 and Cellular Biology*, vol. 13, pp. 248-256, 1993.

[14] M. Koyuturk, W. Szpankowski, and A. Grama, "Biclustering Gene-Feature Matrices for Statistically Significant Dense Patterns," *Proc. Eighth Int'l Conf. Research in Computational Molecular Biology (RECOMB '04)*, pp. 480-484, 2004.

[15] A. Kwon, H. Hoos, and R. Ng, "Inference of Transcriptional Regulation Relationships from Gene Expression Data," *Bioinformatics*, vol. 19, no. 8, pp. 905-912, 2003.

[16] J. Liu, W. Wang, and J. Yang, "Biclustering in Gene Expression Data by Tendency," *Proc. Third Int'l IEEE CS Computational Systems Bioinformatics Conf. (CSB '04)*, pp. 182-193, 2004.

[17] J. Liu, W. Wang, and J. Yang, "A Framework for Ontology-Driven Subspace Clustering," *Proc. ACM SIGKDD '04*, pp. 623-628, 2004.

[18] J. Liu, W. Wang, and J. Yang, "Gene Ontology Friendly Biclustering of Expression Profiles," *Proc. Third IEEE CS Computational Systems Bioinformatics Conf. (CSB '04)*, pp. 436-447, 2004.

[19] J. Liu, W. Wang, and J. Yang, "Mining Sequential Patterns from Large Data Sets," *Advances in Database Systems*, vol. 18, Kluwer Academic Publishers, 2005.

[20] S. Lonardi, W. Szpankowski, and Q. Yang, "Finding Biclusters by Random Projections," *Proc. 15th Ann. Symp. Combinatorial Pattern Matching (CPM '04)*, pp. 102-116, 2004.

[21] S.C. Madeira and A.L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24-45, Jan./Mar. 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, Dec. 2005.
[23] S.C. Madeira and A.L. Oliveira, "A Linear Time Algorithm for Biclustering Time Series Expression Data," *Proc. Fifth Workshop Algorithms in Bioinformatics (WABI '05),* pp. 39-52, 2005.
[24] S.C. Madeira and A.L. Oliveira, "An Efficient Biclustering Algorithm for Finding Genes with Similar Patterns in Time-Series Expression Data," *Proc. Fifth Asia-Pacific Bioinformatics Conf. (APBC '07),* pp. 67-80, 2007.
[25] 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,* vol. 5, no. 12, p. R101, 2004.
[26] E. McCreight, "A Space Economical Suffix Tree Construction Algorithm," *J. ACM,* vol. 23, pp. 262-272, 1976.
[27] G.J. McLachlan, K. Do, and C. Ambroise, "Analysing Microarray Gene Expression Data," *Wiley Series in Probability and Statistics,* 2004.
[28] I. Van Mechelen, H.H. Bock, and P. De Boeck, "Two-Mode Clustering Methods: A Structured Overview," *Statistical Methods in Medical Research,* vol. 13, no. 5, pp. 979-981, 2004.
[29] C. Mollër-Levet, S. Cho, and O. Wolkenhauer, "DNA Microarray Data Clustering Based on Temporal Variation: FCV and TSD Preclustering," *Applied Bioinformatics,* vol. 2, no. 1, pp. 35-45, 2003.
[30] T.M. Murali and S. Kasif, "Extracting Conserved Gene Expression Motifs from Gene Expression Data," *Proc. Eighth Pacific Symp. Biocomputing (PSB '03),* vol. 8, pp. 77-88, 2003.
[31] R. Peeters, "The Maximum Edge Biclique Problem Is NP-Complete," *Discrete Applied Math.,* vol. 131, no. 3, pp. 651-654, 2003.
[32] R.G. Pensa, C. Leschi, J. Besson, and J. Boulicaut, "Assessment of Discretization Techniques for Relevant Pattern Discovery from Gene Expression Data," *Proc. Fourth Workshop Data Mining in Bioinformatics (BIOKDD),* 2004.
[33] 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,* vol. 22, no. 10, pp. 1282-1283, 2006.
[34] P. Weiner, "Linear Pattern Matching Algorithms," *Proc. 14th IEEE Symp. Switching and Automata Theory (SWAT '73),* pp. 1-11, 1973.
[35] Q. Sheng, Y. Moreau, and B. De Moor, "Biclustering Microarray Data by Gibbs Sampling," *Bioinformatics,* vol. 19, no. 2, pp. 196-205, 2003.
[36] T.M. Swan and K. Watson, "Stress Tolerance in a Yeast Sterol Auxotroph: Role of Ergosterol, Heat Shock Proteins and Trehalose," *FEMS Microbiology Letters,* vol. 7, pp. 169-191, 1998.
[37] A. Tanay, R. Sharan, and R. Shamir, "Discovering Statistically Significant Biclusters in Gene Expression Data," *Bioinformatics,* vol. 18, no. 1, pp. 136-144, 2002.
[38] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church, "Systematic Determination of Genetic Network Architecture," *Nature Genetics,* vol. 22, pp. 281-285, 1999.
[39] 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,* vol. 34, pp. D446-D451, Jan. 2006.
[40] E. Ukkonen, "On-Line Construction of Suffix Trees," *Algorithmica,* vol. 14, pp. 249-260, 1995.
[41] C. Wu, Y. Fu, T.M. Murali, and S. Kasif, "Gene Expression Module Discovery Using Gibbs Sampling," *Genome Informatics,* vol. 15, no. 1, pp. 239-248, 2004.
[42] Y. Zhang, H. Zha, and C.H. Chu, "A Time-Series Biclustering Algorithm for Revealing Co-Regulated Genes," *Proc. Fifth IEEE Int'l Conf. Information Technology: Coding and Computing (ITCC '05),* pp. 32-37, 2005.

Sara C. Madeira received the BSc degree in computer science from the University of Beira Interior in 2000 and the MSc degree in information systems and computer engineering from Lisbon Technical University, Lisbon, Portugal, in 2002. She is currently a student in the Instituto Superior Técnico, Lisbon Technical University, where she is finishing her PhD degree in the area of biclustering algorithms for time series gene expression data analysis. She is also a researcher in the Knowledge Discovery and Bioinformatics (KDBIO) Group, INESC-ID, Lisbon, Portugal, and a lecturer in the Departamento de Informática, Universidade da Beira Interior, Covilhã, Portugal. Her research interests include bioinformatics, systems biology, data mining, and machine learning.

Miguel C. Teixeira received the BSc and PhD degrees in chemical engineering and biotechnology from Lisbon Technical University, Lisbon, Portugal, in 2000 and 2004, respectively. He is currently an assistant professor in the Instituto Superior Técnico (IST), Lisbon Technical University. He is also a researcher at the Institute for Biotechnology and Bioengineering (IBB)/CEBQ, IST. His research interests include yeast molecular and cellular biology; response to stress and multidrug resistance (MDR); functional genomics, proteomics, and bioinformatics; and toxicogenomics. Since 2002, he has published 15 peer-reviewed publications in international scientific journals and contributed to the public database YEASTRACT.

Isabel Sá-Correia received the BSc and PhD degrees in chemical engineering and biotechnology from Lisbon Technical University, Lisbon, Portugal, in 1975 and 1984, respectively. She was a visiting assistant professor at the University of Illinois, Chicago, from 1985 to 1986. She is currently a professor responsible for the area of biological sciences in the Instituto Superior Técnico (IST), Lisbon Technical University, and the Institute for Biotechnology and Bioengineering (IBB)/CEBQ, IST. Her research interests include molecular and cellular microbiology; yeast biology; response to stress; bacterial molecular epidemiology; functional genomics, proteomics, and bioinformatics; and microbial toxicogenomics and pathogenomics. She has published more than 150 peer-reviewed publications in international scientific journals and 30 gene sequences in GenBank and contributed to the public database YEASTRACT.

Arlindo L. Oliveira received the BSc and MSc degrees in electrical and computer engineering from Lisbon Technical University, Lisbon, Portugal, in 1986 and 1989, respectively, and the PhD degree in electrical engineering and computer science from the University of California, Berkeley, in 1994. He is currently a professor in the Instituto Superior Técnico, Lisbon Technical University. He is also a senior researcher at the Knowledge Discovery and Bioinformatics (KDBIO) Group, INESC-ID, Lisbon, Portugal. His research interests include bioinformatics, systems biology, string processing, algorithm design, combinatorial optimization, machine learning, logic synthesis, and automata theory. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.