Available online at www.sciencedirect.com

ScienceDirect

Journal of Biomedical Informatics 40 (2007) 761–774

Journal of Biomedical Informatics

www.elsevier.com/locate/yjbin

Precedence Temporal Networks to represent temporal relationships in gene expression data

Lucia Sacchi *, Cristiana Larizza, Paolo Magni, Riccardo Bellazzi

Dipartimento di Informatica e Sistemistica, University of Pavia, Via Ferrata n^o 1, 27100 Pavia, Italy

Received 22 September 2006
Available online 10 June 2007

Abstract

The reconstruction of gene regulatory networks from gene expression time series is nowadays an interesting research challenge. A key problem in this kind of analysis is the automated extraction of precedence and synchronization between interesting patterns assumed by genes over time.

The present work introduces Precedence Temporal Networks (PTN), a novel method to extract and visualize temporal relationships between genes. PTNs are a special kind of temporal network where nodes represent temporal patterns while edges identify precedence or synchronization relationships between the nodes.

The method is tested on two case studies: the expression of a subset of genes in the soil amoeba *Dictyostelium discoideum* and of a set of well-studied genes involved in the human cell cycle regulation. The extracted networks reflect the capability of the algorithm to clearly reconstruct the timing of the considered gene sets, highlighting different stages in *Dictyostelium* development and in the cell cycle, respectively.

Keywords: Gene expression; DNA microarrays; Temporal data mining; Temporal Abstraction; Temporal association rules; Temporal networks

1. Introduction

Recent years have seen an increasing interest in the analysis of data coming from DNA microarrays, a measurement technique able to take a genome-wide snapshot of the molecular activity of a cell in terms of its m-RNA abundance. Such data give the unparalleled possibility to directly observe the results of the complex regulation mechanisms underlying protein production in any tissue. The majority of the published studies has been devoted to differential analysis, i.e. the search for a collection of genes which are differentially expressed in a condition of interest (such as tumor vs healthy patients), and to functional studies, i.e. the discovery of gene function. Moreover, a noteworthy research effort dealt with the automated generation of hypotheses on gene regulatory networks through reverse engineering approaches [1,2]. The aim of these studies is to infer regulatory relationships between genes by observing the dynamic behavior of their expression; this is possible, for example, by studying time series of DNA microarray measurements. From a computational viewpoint, a pioneering work was represented by the REVEAL algorithm [3], which extracts networks expressing Boolean relationships between genes through an heuristic search strategy based on mutual information. More recently, several approaches were proposed to derive regulatory networks from DNA microarray data, including methods which model gene expression dynamics through differential equations [4] and Bayesian networks [5]. Given the very nature of microarray data, none of these approaches may however lead to reveal all the biochemical pathways or the physical interactions underlying the observed processes. As a matter of fact, a certain mRNA stream might not always correspond to the same protein, due to potential post-transcriptional or post-translational

[*] Corresponding author.
E-mail address: lucia.sacchi@unipv.it (L. Sacchi).

1532-0464/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2007.06.003

modifications and, even more important, the dynamics of regulatory interactions cannot always be captured by the (low) sampling time available in DNA microarray experiments. For these reasons, it is important to couple genetic network search with descriptive approaches able to derive robust hypotheses on the regulatory mechanisms, to be further verified by wet-lab experiments. For example, it is of interest to describe patterns of synchronized gene expressions, which might be the evidence of a strict relationship between the genes activity. Moreover, it is useful to highlight the temporal relationships between groups of synchronized genes, to understand the temporal sequence of biological sub-processes. To this end the so-called *module networks* were recently introduced. A module is a set of synchronized co-regulated genes which share a common function [6]. Given inputs from both gene expression data and biological knowledge on putative transcription factors for the involved genes, the algorithm searches both for a split of the genes into modules and for a regulation program for each module, which is then used to explain the behavior of genes in the module itself. Such a regulation program is extracted through a probabilistic strategy. Although of interest, the final module network is still sometimes difficult to interpret; the elements in each module may not be completely synchronized and conditional probabilities may express average behaviors, losing complex and "local" relationships, such as, for example, a control action between genes which holds only when a certain gene is overexpressed and not when it is underexpressed.

Traditionally, a qualitative representation of synchronized behaviors is extracted through clustering [7]. Thanks to the recent interest in temporal clustering research [8–10], several computational tools able to extract the main patterns occurring in a set of gene expression temporal profiles are nowadays available. However, temporal clustering is by nature designed to group time series according to their expression profiles and not to extract temporal relationships between them.

To overcome the above mentioned limitations, in this paper we introduce a novel method to express the temporal relationships which occur between gene expression profiles. Differently from the other methods, our approach is centered on the description of precedence and synchronization of gene temporal patterns in a data set.

The method is based on the description of the genes in terms of *patterns*, which are a formalization of the intuitive notion of *interesting behaviors* that the user may want to extract from the available data. The temporal relationships which describe synchronization and precedence between such patterns, and consequently between the involved genes, are extracted through an algorithm that efficiently searches the space of possible relationships that may result between the patterns. Finally, the genes and the corresponding relationships are mapped into a labeled graph, designed to reconstruct the timing of the events occurring during the process under analysis; we call the resulting graph Precedence Temporal Network (PTN).

The paper develops as follows: in Section 2 we first introduce the notion of pattern and its formalization through a qualitative labeling mechanism for the representation of temporal data based on the Temporal Abstraction technique [11–14]. After interesting patterns are retrieved in the data, we show how to derive precedence relationships between such patterns by running a temporal association rule algorithm; the extracted rules are finally mapped into a PTN. Sections 3 and 4 present an evaluation of the method on the reconstruction of the timing of interesting events occurring in a set of genes involved in the developmental program of the soil amoeba *Dictyostelium discoideum* and in a group of genes known to play a key role in human cell cycle regulation.

2. Methods

2.1. Knowledge-based temporal abstraction to describe gene expression patterns

The notion of *pattern* is intuitively related to the representation of a property or a behavior of interest that might be conveniently extracted from data for analysis purposes; such an abstract property is in general of qualitative nature. According to this first definition, a simple pattern may for example be an increasing trend in a variable, while a more complex one might be an up and down behavior repeated several times. When dealing with temporal data, an interval, which represents the period of validity of the selected qualitative property over the measurement time span, is usually associated to the pattern. In this paper we exploit the framework of Temporal Abstractions (TAs) [11–14] to formalize the intuitive definition of pattern as a 'shape' of interest related to an interval of validity.

The basic feature of the TA technique is the shift from a time-point to an interval-based representation of time series data. Following the data model proposed in [15], raw temporal data are represented as time-stamped entities, called *events*, while their abstract representation is given by TAs as a sequence of intervals, called *episodes*. In the following, we will denote a generic episode as e ≡ (e.start, e.end), where e.start and e.end are, respectively, the starting and the ending point of the interval. A qualitative label, corresponding to a specific behavior of interest, is then used to characterize each episode. The algorithms which are devoted to the generation of episodes from events (or from other episodes) are known as TA *mechanisms*.

Depending on the kind of inputs and outputs of the corresponding mechanisms, Temporal Abstractions can be classified into two main categories:

- *Basic* TAs, solved by mechanisms that abstract time-stamped data into intervals (input data are events and outputs are episodes),
- *Complex* TAs, solved by mechanisms that abstract intervals into other intervals (input and output data are episodes).

Among Basic TAs, we will herein deal with *state* TAs, which are used to detect qualitative patterns corresponding for example to low, high or normal values in numerical or symbolic time series, and *trend* TAs, used to capture increasing, decreasing or stationary courses in numerical time series.

Complex TAs correspond to intervals over which specific temporal relationships between basic or other complex TAs hold; such temporal relationships are usually identified through Allen's temporal operators [16]. Fig. 1 shows an example of how a time series can be conveniently represented through a set of state (Fig. 1a), trend (Fig. 1b), and complex TAs (Fig. 1c).

Thanks to its capability to handle temporal data through qualitative and interval-base features, the TA framework offers a natural way to formalize the intuitive notion of pattern introduced at the beginning of this section. TAs give in fact the possibility of associating a qualitative property of interest, which we will refer to as the *pattern*, to the set of episodes where this behavior is verified in the data. In our analysis context, a pattern may for example be the 'overexpression' of a gene, which will be formalized by a state TA describing the time intervals in which gene expression assumes high values; a complex pattern, such as a 'peak' in gene expression profile, may be represented with a complex TA given by the combination of the two consecutive trend TAs Increasing and Decreasing.

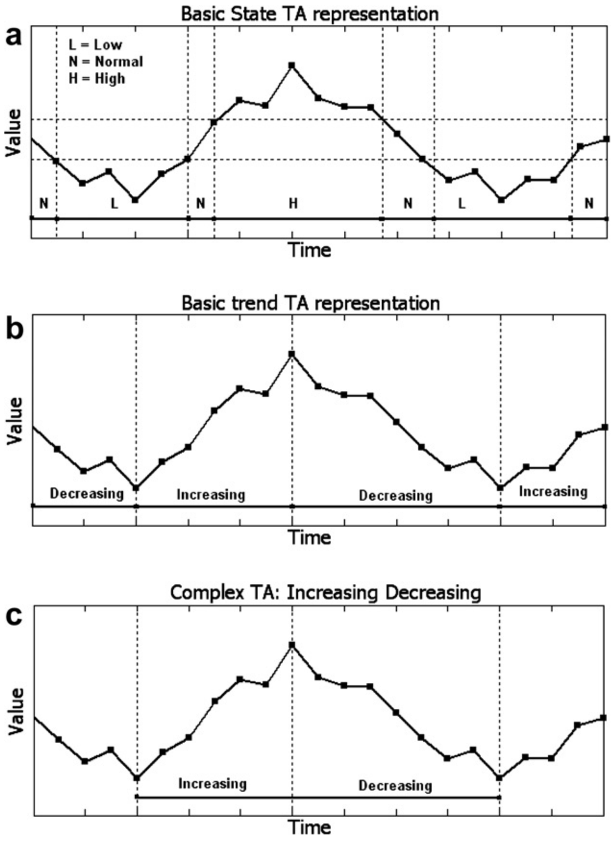

Fig. 1. Representation of time series through Temporal Abstractions. Temporal Abstraction can be applied in several ways to represent temporal data. (a) An example of state TA representation: the time series is described through intervals labeled according to the level of the variable (low, normal, high). (b) The same time series represented through trend TAs: a label which gives information on the trend behavior of the variable is used to describe the intervals detected through suitable mechanisms. (c) An example of complex TA representation: the two basic trend TAs Increasing and Decreasing are associated through Allen's temporal operator MEETS to form the complex abstraction [Increasing Decreasing].

As mentioned above, any TA can be extracted from a set of time series through suitable TA mechanisms. Such mechanisms are typically dependent on a set of parameters specified by the user or by the data analyst according to the application of interest. Examples of these parameters are the minimum slope to trigger the detection of a trend TA, or the threshold values needed to map the quantitative values of a variable (e.g. gene expression) to a set of qualitative state labels (e.g. high or low).

In this work, TAs are exploited to formally define and then detect interesting patterns in a data set of gene expression time series. The retrieval of such patterns represents the first step of our proposed approach.

The procedure to retrieve patterns in a data set develops as follows:

- Define a set of qualitative abstract patterns QAP = $\{p_1, p_2, \ldots, p_n\}$, which are the behaviors one wants to detect in the time series. Each p_i can be defined both as a basic or a complex pattern (e.g. 'Low' or 'Increasing Decreasing'),
- Process the raw time series through suitable TA mechanisms [8,15] in order to represent them through basic TAs (state, trends, or both, according to the set QAP),
- For each $p_i \in$ QAP, identify the intervals in which p_i is verified in the data. If p_i is a complex pattern (e.g. 'Increasing Decreasing'), the retrieval is performed by recursively applying on any variable a complex TA mechanism corresponding to the proper composition of Allen's operators (e.g. find the intervals where one episode of Increase MEETS one episode of Decrease).

2.2. Formalization of the notion of precedence

In order to systematically look for temporal relationships between the patterns, we herein formalize the notion of precedence we want to deal with. In more detail, we consider temporal relationships expressed by the temporal operator *PRECEDES*, defined as follows [17]:

Given two episodes, $E_1 \equiv [e_1.\text{start}, e_1.\text{end}]$ and $E_2 \equiv [e_2.\text{start}, e_2.\text{end}]$, E_1 *PRECEDES* E_2 if $e_1.\text{start} \leqslant e_2.\text{start}$ AND $e_1.\text{end} \leqslant e_2.\text{end}$.

According to this definition, the *PRECEDES* operator synthesizes the following Allen's temporal relationships: *OVERLAPS*, *FINISHED-BY*, *MEETS*, *BEFORE*, *EQUALS* and *STARTS*.

The *PRECEDES* relationship may be conveniently constrained in the temporal association rule by three parameters, introduced to allow some restrictions on the mutual position of the involved intervals. These parameters are: the left shift (*LS*), defined as the maximum allowed distance between e_2.start and e_1.start (i.e. the maximum allowed distance between the starting points of the two episodes involved in the relationship), the gap (*G*), defined as the maximum allowed distance between e_2.start and e_1.end and the right shift (*RS*), defined as the maximum allowed distance between e_2.end and e_1.end (i.e. the maximum allowed distance between the ending points of the two episodes involved in the relationship).

The introduction of this parameterization offers a great advantage in terms of efficiency, as it allows to restrict the space of relationships to be investigated and to avoid the retrieval of potentially useless rules, that otherwise would have to be eliminated during the results interpretation step. This is particularly important in gene expression analysis, where thousands of genes are simultaneously studied.

By properly tuning these parameters it is possible to select a subset of relationships that will be evaluated during the analysis; it might not in fact be necessary to always look for all the relationships covered by *PRECEDES*. For example, if one wants to restrict the retrieval only to relationships satisfying the *MEETS* operator, G will have to be set equal to zero; to detect *EQUALS* relationships, both *LS* and *RS* have to be set to zero. Note that, according to the definition of *PRECEDES*, neither *LS* nor *RS* can assume negative values. *G*, on the other hand, can be <0, for example in the case one wants to take into account the *OVERLAPS* operator. To properly direct the search over the rule space, the choice of the parameters values must be performed on the basis of the knowledge available on the problem domain. The user will therefore have to answer to simple questions, like for example: which is the maximum allowed distance between two intervals to consider an extracted rule as meaningful? Is the intersection of the intervals allowed in a potential rule? After how much time do we expect the consequent interval to start?

In the case one wants to carry out an exhaustive search over the rule space, high values for all the parameters have to be set, in order not to violate the corresponding constraints. In this way, the algorithm will extract *all* the precedence relationships present in the data set; this choice is left to the user, considering that this means a considerable increase both in computational time and in the effort of interpreting the results.

2.3. Searching for temporal relationships between patterns

Once a formal definition of the notion of precedence is stated, the algorithm follows a strategy that allows the search for rules of the kind $A \rightarrow_P C$, where a set of contemporaneous patterns, the antecedent A, *PRECEDES* another pattern, the consequent C. A rule where the antecedent is related to the consequent by some kind of temporal relationships is referred to as a *temporal rule* [18]. In our analysis context, an example of such a rule involving basic trend patterns is: "An increasing episode in gene G_1 AND in gene G_2 *PRECEDES* a decreasing episode in gene G_3".

Inspired by the works proposed in the literature for the extraction of temporal association rules from interval-based data [17–20], we recently extended the method proposed in [17] by introducing an algorithm which has the flexibility to handle rules characterized by complex patterns both in their antecedent and consequent [21].

Our algorithm follows two steps: first, it derives simple gene-gene temporal relationships (e.g. G_1 Increasing $\rightarrow_P$ G_3 Decreasing) and then it combines several genes in the antecedent to form more complex rules. This leads to the detection of modules of synchronized and temporally related gene sets. To deal with the specific concerns of this paper, the algorithm was tuned to properly address the particular characteristics of gene expression profiles; we in fact have to manage short time series [10] and to face with the task of analyzing a multivariate problem where each variable represents a single gene. As we will discuss in the following paragraphs, this will in particular influence the choice of the parameters constraints.

Following literature approaches for the mining of temporal association rules, in our extraction system the search is performed through an Apriori-like technique [22], where interesting rules are selected based on thresholds defined on two parameters which are essential for the definition of frequent patterns and for an efficient search over the rule space. These parameters are the *confidence* and the *support*. In this paper we will introduce these two quantities according to the definitions stated in [17] and already adopted in [21], which are summarized in the following.

We introduce:

- *TS*: total duration of the observation period,
- *RTS*: time span corresponding to the union of the episodes in which both the patterns corresponding to the antecedent and the consequent of the rule occur,
- *NAT*: number of times (episodes) the antecedent occurs during *TS*,
- *NARTS*: number of times (episodes) the antecedent occurs during *RTS*.

We thus define:

- $Support(\mathrm{Sup}) = \frac{RTS}{TS}$,
- $Confidence(\mathrm{Conf}) = \frac{NARTS}{NAT}$.

Intuitively, the support offers a measure of the portion of the observation period (i.e. the total duration of the DNA microarray experiment) which is covered by the rule, while the confidence indicates the frequency of occurrence of the rule with respect to the number of episodes of the antecedent. Of course, the setting of these parameters leads to carefully consider the features of the time series under analysis; in the case of gene expression profiles, a proper tuning must take into account the low number of samples which usually characterizes the data. This will lead to constraints for confidence and support which are more restrictive than the ones we would apply in the case of longer time series. Considering short time series, we in fact expect a low number of intervals of validity for a specific pattern within a gene, which corresponds to a low value for *NAT*. Significant rules will indeed be the ones showing a value for *NARTS* very close to the one calculated for *NAT*, i.e. which involve all the episodes of the antecedent as episodes of the rule, resulting in a confidence close to one. In addition, a rule will be judged as significant if it covers a large time span with respect to the entire observation period, i.e. if it shows a value for *RTS* close to the one computed for *TS*. To force this constraint, the threshold on the minimum support must be set to an high value, typically close to one.

The main steps of the algorithm for temporal rule extraction are described in the following pseudocode.

```
// Initialization
Given:
$A_0$ = set of all the TAs that represent the genes. Each $a_i \in A_0$ is the set of
intervals of validity of a pattern over a time series
N = cardinality of $A_0$
min_sup = the threshold for the support
min_conf = the threshold for the confidence
for i=1 to N
   // Set the Consequent as a TA in $A_0$
   set cons = $a_i \in A_0$;
   set k = 1

   // Creation of simple gene - gene interactions:
   for j=1 to N-1, $j \neq i$
         Select a set of intervals $a_j \in A_0 - \{cons\}$
         Apply the PRECEDES operator between $a_j$ and cons.
         if ($a_j \rightarrow_P$ cons and Sup(rule)>=min_sup)
               set $A_1 = A_1 \cup \{a_j\}$
         end
   end
   // Creation of complex rules:
   Repeat:
   set k=k+1;
   Generate the set $A_k$ from $A_{k-1}$ such that each rule in $A_k$:
         -  has cardinality k (conjunction of k patterns in the
            antecedent),
         -  verifies the PRECEDES relationship,
         -  Sup(rule)>=min_sup.
   Until: $A_k$ is empty
   Select from $A_k$ only rules showing confidence Conf>=min_conf
end
```

The algorithm first searches for simple gene-gene relationships by applying the *PRECEDES* temporal operator between the intervals of validity of the patterns of the antecedent and of the consequent of the rule; the method then creates more complex rules by merging synchronized antecedents. Two antecedents are synchronized if the intervals corresponding to the candidate patterns intersect; referring to Allen's temporal operators, two intervals show an intersection if they satisfy any of the relations *EQUALS*, *OVERLAPS*, *FINISHES*, *FINISHED-BY*, *STARTS*, *STARTED-BY*, *DURING*, *CONTAINS*. Considering all these operators for the computation of synchronization allows the involved intervals to be slightly shifted between each other; such shift would not be possible by resorting to the *EQUALS* relation alone.

2.4. Precedence Temporal Networks

In order to map the obtained set of rules into a PTN, some preliminary points have to be discussed. First of all, the temporal rules extraction algorithm described in Section 2.3 potentially allows the detection of all the possible relationships between the patterns in the set QAP. As an example, let's consider the set QAP defined as QAP = {[Increasing Decreasing], [Increasing]}, whose first element formally describes an intuitive 'up-and-down' activation/deactivation pattern and the second formalizes an 'up' or activation pattern. Denoting Increasing with I and Decreasing with D, there are four possible relationships that might be investigated: $\{ID\} \rightarrow_P ID$, $\{I\} \rightarrow_P I$, $\{ID\} \rightarrow_P I$ and $\{I\} \rightarrow_P ID$. Genes might indeed be related to each other in four different ways, leading to a set of potential interactions where the same gene may play different 'roles', depending on the patterns it satisfies. From an interpretative viewpoint, this doesn't allow a clear mapping between the gene patterns and a graphical representation through a network of precedence relationships. As a consequence, it is in practice advisable to set a preliminary pattern space for the antecedents and for the consequents and then tune the algorithm in order to limit the search over the specified sets. Going back to the previous example, we could for instance choose to map only the rules of the kind $\{ID\} \rightarrow_P I$; in the algorithm we will therefore define two sets, one for the consequents, made up of all the intervals of validity of the I pattern, and the other for the antecedents, made up of the set of genes verifying the complex abstraction ID. In such a way, besides further reducing the computational cost of the procedure, we also allow a more simple and clear way to map the rules into an interaction network.

Starting from the constraints on the set of temporal rules pointed out so far, the mapping into a Precedence Temporal Network is then straightforward. We define a PTN as a kind of temporal network [23] where each node represents a pattern for a specific gene; in the advisable case the space sets for the antecedents and the consequents had been carefully selected, we will get a one-to-one correspondence between nodes and genes. Edges describe the temporal relationships between the nodes (genes) involved in the temporal rules. To include in the representation both the relationships occurring between the elements in the antecedents and the one occurring between the antecedents and the consequents, we distinguish between two types of edges. The first, called *co-occurrence* edges, link elements characterized by the simultaneity of their temporal events (i.e. the members of the antecedent), while the second, called *precedence* edges, map the *PRECEDES* temporal relationship. Moreover, within the precedence edges, we specify strong edges, which link two nodes that verify the precedence operator in all the rules in which they are involved, and weak edges, which instead associate two nodes related by a precedence relationship in some cases, while evaluated as contemporaneous in others.

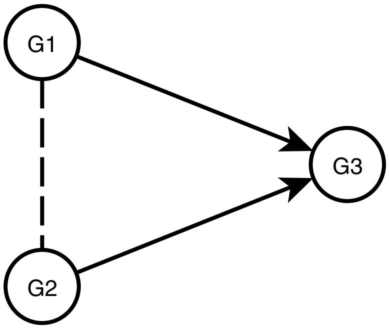

Fig. 2. Example of a simple PTN. Mapping of the rule "An increasing episode in gene G_1 AND in gene G_2 PRECEDES a decreasing episode in gene G_3" through the corresponding PTN. The dashed edge corresponds to the co-occurrence connection between G_1 and G_2 (elements of the antecedent), while the black arrows represent the precedence connections stated by the rule.

Fig. 2 shows the graph obtained by mapping the simple rule "An increasing episode in gene G_1 AND in gene G_2 *PRECEDES* a decreasing episode in gene G_3", defined over the set QAP = {[Increasing], [Decreasing]} and considering {Increasing} $\rightarrow_P$ Decreasing as the target rule. In the resulting graph each node represents a gene, the dashed edge represents the co-occurrence connection between G_1 and G_2 (elements of the antecedent), while black arrows represent the precedence connections defined by the rule.

3. Results

In this section, we present the results on two different case studies on DNA microarray experiments. First, we analyze the reconstruction of developmental events taking place in *D. discoideum* cells and, second, we apply our method to derive temporal relationships between a set of genes involved in the human cell cycle.

3.1. Reconstructing the timing of developmental events in D. discoideum

In this first experiment, we run our algorithm to derive a PTN on a dataset made up of time series collected during the development of the soil amoeba *D. discoideum*. Upon starvation, single *Dictyostelium* cells stop growing as unicellular entities and aggregate to form a multicellular organism through a process which lasts about 24 h. Under these conditions, the developmental program of the amoeba is characterized by a well-studied and highly coordinate set of cellular, physiological and morphological changes [24]. Up to now, several functional studies have been able to relate the different stages of the development to changes in gene expression and a lot of genes have been already found to be developmentally related [24]. The task of reconstructing the timing of developmental events in *Dictyostelium* cells is a good benchmark to validate our method, since much is known about gene expression during the development of such amoeba. To test our algorithm we chose to use a dataset of whole-genome transcriptional profiles of wild type cells [25], made up of time series of 13 time points that record measurements taken every 2 h over an observation period of 24 h, equal to the duration of *D. discoideum* developmental course. From the original dataset we selected 24 genes with known activation time during development [26]; these genes are listed in Table 1.

Fig. 3 shows the PTN reconstructed by our algorithm when searching for temporal relationships between different intervals of activation of the genes. To represent the activation of a gene, we chose to detect intervals where the expression value of the gene is greater than zero; for this reason we defined the set QAP as QAP = {'Activated'}, which is a pattern expressing information on the level of expression of the genes in the dataset. A state TA mechanism was then run on the time series to detect the intervals of validity of the pattern over the data. Given an expression profile $x = (x_1, x_2, \ldots, x_{13})$, this mechanism first creates a qualitative profile $y = (y_1, y_2, \ldots, y_{13})$ such that:

$$\begin{cases} y_i = \text{`}On\text{'} & if\ x_i > 0 \\ y_i = \text{`}Off\text{'} & if\ x_i \leqslant 0 \end{cases}$$

Activation intervals are detected by merging consecutive points labeled as 'On'. The detection of an activation interval is triggered when a minimum number of consecutive 'On' time points is found; this number was herein set to 3.

Considering both the small number of time samples characterizing the profiles and the fact that the considered experiment is aimed at monitoring the entire developmental course in *Dictyostelium* cells, we chose to search for precedence relationships over the whole observation time span. As it was mentioned in Section 2.2, this is possible by letting the rule extraction parameters (*LS*, *G* and *RS*) assume values higher than the length of the examined time series. For completeness, a list of the algorithm design parameters and their corresponding settings is detailed in Table 2.

For the sake of clarity, the network in Fig. 3 is visualized by mapping genes according to the background knowledge about their activation time during development; all the

Table 1
Dictyostelium discoideum gene set

Early development	*carA*, *nagA*, *cprD*, *acaA*, *dscA*, *pdsA*, *cadA*, *manA*, *regA*, *gbfA*, *rasD*, *carB*
Late development	*pspA*,*ecmB*, *tagB*, *tagC*, *cotA*, *cotB*, *cotC*, *cotD*, *pspB*, *spiA*, *yelA*, *culA*

List of the 24 genes considered for the analysis of *D. discoideum* gene expression time series. Genes are divided into activated in an early developmental stage (until aggregation is complete) and activated late into the developmental program.

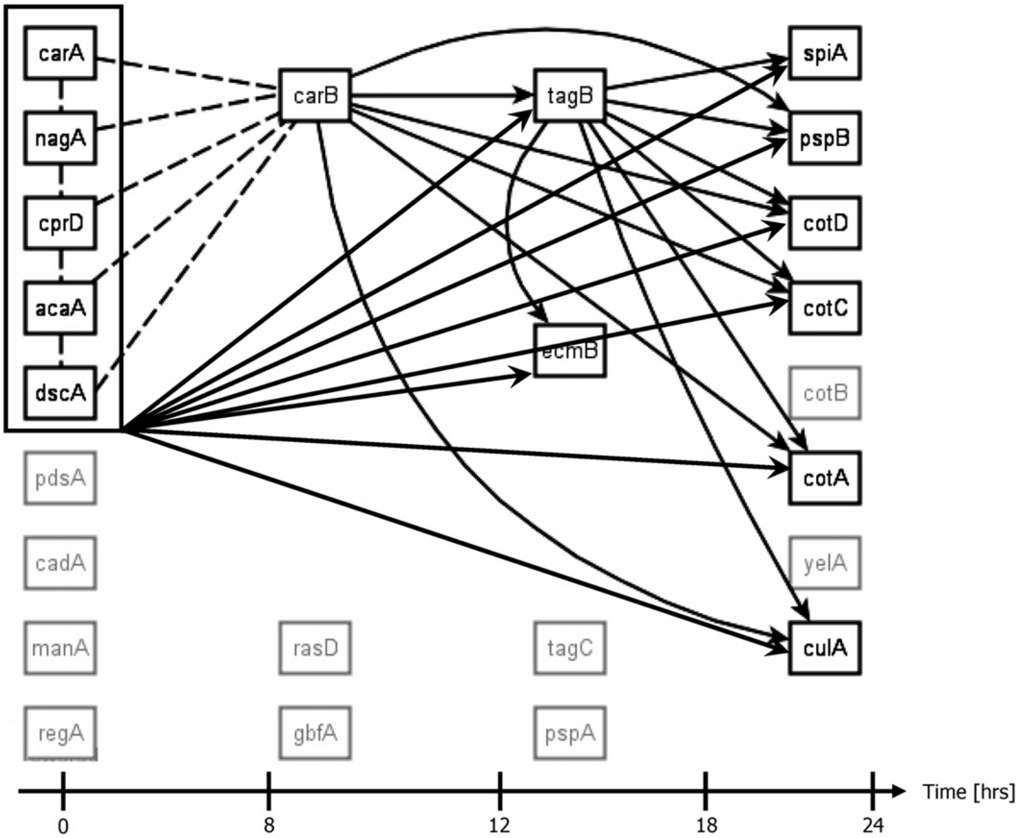

Fig. 3. PTN for *Dictyostelium discoideum* gene expression time series. PTN extracted by running our algorithm on 24 developmental time series of *D. discoideum* [25]. Relationships between intervals of activation of the genes are explored. The network is mapped according to the background knowledge about the developmental stage of activation [26], with early genes on the left. Dashed edges represent co-occurrence connections while black arrows describe strong precedence links. Grey nodes correspond to genes for which no interesting interactions were detected by the rule extraction algorithm.

Table 2
Timing of activation events in *Dictyostelium* cells: parameters setting

Parameter	Value
min_conf	0.8
min_sup	1
LS	30
G	60
RS	30

List of the parameters involved in the rule extraction algorithm and details on their setting in the *Dictyostelium* case study. High values for *LS*, *G* and *RS* allow the user to extract precedence relationships occurring during the whole observation interval. Thresholds for confidence and support close to one (parameters *min_conf* and *min_sup*) are used to extract significant rules even in the presence of short gene expression profiles.

'early' genes are therefore drawn as contemporaneous on the left of the graph, while the others are positioned according to the time axis depicted at the bottom of the picture. Grey nodes in the network represent genes for which no significant relationships were derived.

3.2. Finding temporal relationships between marker genes in the human cell cycle

In this second case study we test the performances of our algorithm in reconstructing the timing of events taking place between genes involved in the human cell cycle. In this example, we show results on a set of gene expression time series coming from one of the five genome-wide experiments on a human cancer cell line (HeLa) published in [27]. In this experiment the profiles are made up of 47 time points, with measurements taken every hour over an observation period of 46 h. Given an estimate cell cycle duration of 15 h, the experiment monitors an average of three cell cycles. From the original set of genes we extracted 20 target elements which we then considered for further analyses; such genes make up a set of well-studied genes showing an expression peak in a specific cell cycle phase [27]. This feature makes them suitable for a validation of our method, since the patterns to be searched and the temporal sequence of the peaks are known. The genes are listed in Table 3, together with the cell cycle phase in which their expression profile is known to show a peak.

To formalize the concept of *peak* expression, which corresponds to a complex shape where a Decreasing interval follows an Increasing one, the set QAP was defined as QAP = {[Increasing Decreasing]}, which is a complex pattern obtained as the composition of two basic trends through the *MEETS* operator. Trend detection is herein performed through a traditional sliding window algorithm for the piecewise linear segmentation of time series [28,21], and the qualitative labels are assigned to the intervals

Table 3
Human cell cycle related gene set

Peak phase	Genes
G_1/S boundary	*CCNE1*, *E2F1*, *CDC6*, *PCNA*
S	*RRM2*, *RAD51*, *RFC4*, *DHFR*
G_2	*CCNF*, *CCNA2*, *TOP2A*, *CDC2*
M	*STK15*, *BUB1*, *PLK1*, *CCNB1*
M/G_1 transition	*CDKN3*, *VEGFC*, *RAD21*, *PTTG1*

The set of 20 genes analyzed to derive the PTN from the experiment on the human cell cycle [27]; genes are grouped according to the cell cycle phase in which they show a peak in the expression profile.

Table 4
Timing of events during the cell cycle—the extracted rules

Operator: *PRECEDES*
QAP = {[Increasing Decreasing]}
Parameters: *min_conf* = 1, *min_sup* = 2/3, *LS* = 5, *G* = 8, *RS* = 5

Antecedent	Consequent	Confidence	Support
CCNE E2F1 CDC6 PCNA	RRM2	1	0.681
CCNE E2F1 CDC6 PCNA	RAD51	1	0.681
E2F1 PCNA	CDC2	1	0.702
RRM2 RAD51	TOP2A	1	0.702
RRM2 RAD51	CCNF	1	0.702
RRM2 RAD51	CCNA2	1	0.723
RRM2 RAD51	STK15	1	0.702
RRM2 RAD51 CDC2	BUB1	1	0.681
RRM2 RAD51 CDC2 TOP2A	PLK1	1	0.681
CDC2 TOP2A CCNA2	RAD21	1	0.957
RAD51 CDC2	VEGFC	1	0.681
RRM2 RAD51	CDKN3	1	0.702

The set of rules derived by the algorithm applied to the human cell cycle data. Rules of the kind {[Increasing Decreasing]} $\rightarrow_{\mathrm{P}}$ [Increasing Decreasing] are extracted. The values for the parameters are reported in the first line of the table.

according to the slopes of the segments that make up the approximating curve. The complex abstraction is then detected by applying the *MEETS* operator to the intervals of validity of the two trends. This definition of QAP results in a one-to-one correspondence between genes and nodes in the network.

The rules extracted by the algorithm are listed in Table 4, together with their confidence and support. The parameters in the rule extraction algorithm were set in order to extract temporal rules holding within a single cell cycle period; considering an average cell cycle duration of 15 h, with measurements taken every hour, we set: $LS = 5$, $RS = 5$, $G = 8$. As regards confidence and support, we herein considered as significant only rules with a confidence equal to 1 and a support greater than 2/3 (parameters *min_conf* and *min_sup*). As it was already pointed out in Section 2.3, these values had to be properly tuned in order to deal with the special case of short gene expression time series; to this aim, an high threshold for the confidence constrains all the intervals verifying the pattern in the antecedent to be also episodes of the rule. Moreover, the constraint on the support forces the rule time span to be at least equal to the 2/3 of the entire observation interval.

The PTN corresponding to the rules in Table 4 is shown in Fig. 4. In this example the network is visualized according to two strategies: in Fig. 4a a visualization based on background knowledge is depicted. Genes are indeed drawn in a position corresponding to the cell cycle phase where they show a peak in gene expression. Dashed edges represent co-occurrence connections, while black arrows describe strong precedence links. The picture shows also two weak precedence connections, represented by grey arrows, which correspond to the two rules: {*RRM2*,*RAD51*} $\rightarrow_{\mathrm{P}}$ *TOP2A* and {*RRM2*,*RAD51*,*CDC2*,*TOP2A*} $\rightarrow_{\mathrm{P}}$ *PLK1*. Fig. 4b shows how the network can be unrolled on the basis of the intervals of validity of the patterns involved in the rules. Genes are located on the time axis in a position which is consistent with the interval over which the peak had been detected. This is a completely data-driven picture of the results, which is automatically obtained after running the algorithm.

To better highlight the features which distinguish PTNs from traditional methods for gene networks reconstruction, the analysis on the 20 cell cycle genes was also performed through an algorithm for dynamic Bayesian networks (DBNs) reconstruction. Such method is implemented in the publicly available software tool Banjo 1.0.5 (http://www.cs.duke.edu/~amink/software/banjo/), which allows both static and dynamic networks extraction [29,30]. The extracted DBN is depicted in Fig. 5: in Fig. 5a genes are positioned following the scheme proposed in Fig. 4a, while Fig. 5b offers a more standard visualization for the DBN. Besides a visual comparison of the two networks, which gives a first description of the similarities and differences between the two structures, a quantitative evaluation of the results in terms of precision and recall was also performed. The two algorithms were compared in terms of their capability of reconstructing precedence

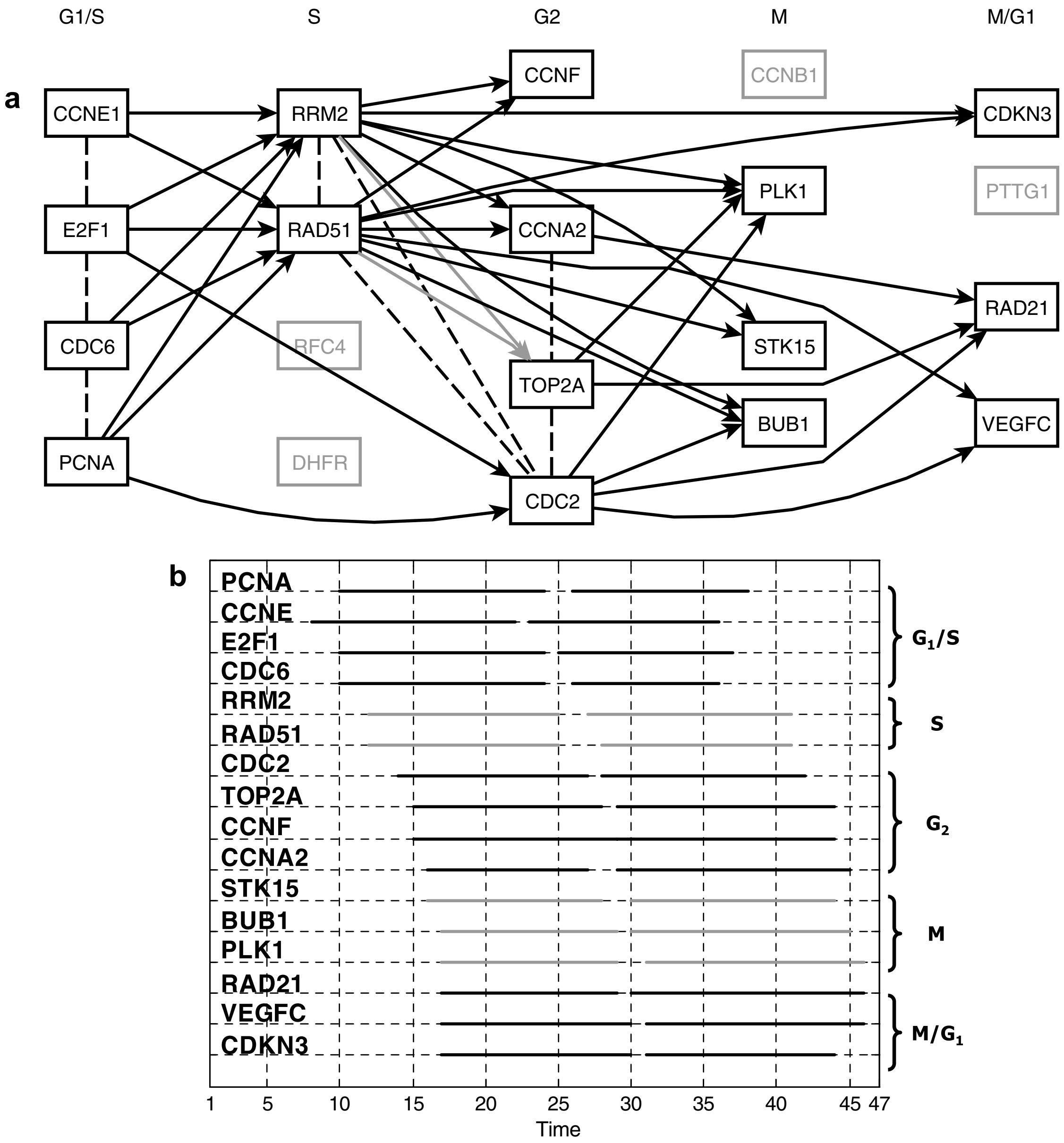

Fig. 4. PTN for genes involved in the human cell cycle. (a) PTN extracted by running our algorithm on HeLa cells time series [27]. Relationships between peaks in gene expression profiles were derived. Genes are drawn according to the phase of the cell cycle where they are known to show a peak in gene expression. Dashed edges represent co-occurrence connections, black arrows describe strong precedence links and grey arrows represent two weak precedence connections that correspond to the two rules: $\{RRM2, RAD51\} \rightarrow_P TOP2A$ and $\{RRM2, RAD51, CDC2, TOP2A\} \rightarrow_P PLK1$. Grey nodes correspond to genes for which no interesting interactions were detected by the rule extraction algorithm. (b) Time intervals of validity of the pattern [Increasing Decreasing] for the genes represented into the network. The different cell cycle phases of peak expression are highlighted on the right-hand side of the picture.

relationships and biological interactions. Tables 5 and 6 report the evaluation results; in particular, the values for true positives (*TP*), false positives (*FP*), false negatives (*FN*), precision and recall are shown.

4. Discussion

We will herein discuss the results obtained on the two applications introduced in Section 3. For simplicity, the present section is divided into two sub-sections, each one related to one of the case studies.

4.1. *Dictyostelium discoideum developmental time series*

Analyzing the network represented in Fig. 3 we can observe the following: first, a module of highly synchronized genes was identified; it includes a set of genes expressed in a very early developmental stage (*carA*, *nagA*, *cprD*, *acaA* and *dscA*) and the gene *carB*, which encodes for the cAMP receptor CAR2. This high synchronization reflects the real timing of the genes, and especially of the ones belonging to the first group: these genes are in fact expressed when cell density is getting high, i.e. between 0

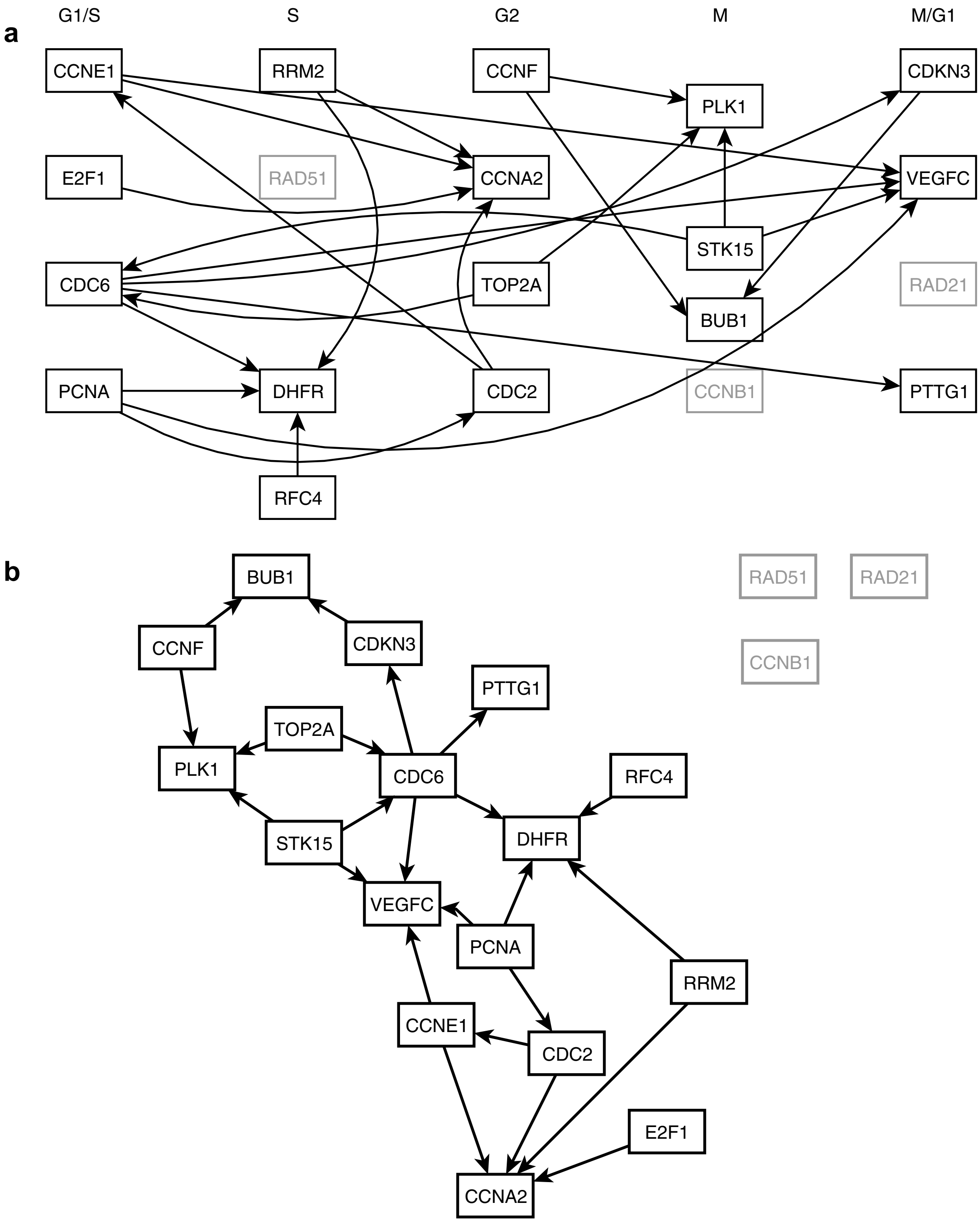

Fig. 5. DBN for genes involved in human cell cycle. (a) Dynamic Bayesian network extracted by running the software Banjo 1.0.5 (http://www.cs.duke.edu/amink/software/banjo/) on HeLa cells time series [27]. Genes are drawn according to the phase of the cell cycle where they are known to show a peak in gene expression. Several edges connecting 'late' genes to 'early' genes can be observed, together with edges connecting genes peaking in the same cell cycle phase. This reveals the different nature of the connections with respect to the ones found in a PTN. (b) A more standard visualization for the DBN.

and 8 h. *carB* is instead expressed when cells are aggregated into tight mounds (6–8 h). These genes are indeed all belonging to the first stage of the development, which lasts until aggregation is complete (10–12 h).

The analysis of the precedence connections extracted by the algorithm (black arrows in the picture) reveals that the sequence of activation of the genes was correctly reconstructed: almost all the genes activated in a specific phase of development are in fact connected with genes activated in the next one. For example, the edge connecting gene *carB* to gene *tagB* states that *tagB* is activated later in development with respect to *carB*. This consideration highlights one of the most important features of PTN representation: the visualized relationships describe the temporal

Table 5
Comparative evaluation of PTNs and DBNs on the extraction of documented biological interactions

Method	*TP*	*FP*	*FN*	Precision	Recall
PTN	3	39	11	0.071	0.214
DBN	4	19	10	0.174	0.286

Evaluation in terms of precision and recall of the performance of PTNs and DBNs in reconstructing known biological interactions reported in the BioGRID database (http://www.thebiogrid.org/). Herein *TP* are the true positives, *FP* the false positives and *FN* the false negatives.

Table 6
Comparative evaluation of PTNs and DBNs on the extraction of precedence relationships

Method	*TP*	*FP*	*FN*	Precision	Recall
PTN	28	3	84	0.9	0.25
DBN	10	12	102	0.45	0.09

Evaluation in terms of precision and recall of the capability of PTNs and DBNs of reconstructing precedence relationships between cell cycle related human genes. Herein *TP* stands for true positives, *FP* for false positives and *FN* for false negatives.

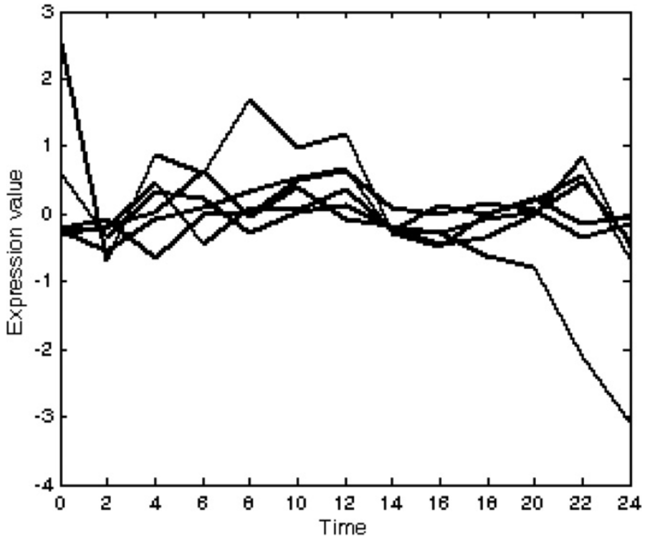

Fig. 6. *Dictyostelium discoideum* gene expression profiles for a set of genes excluded from the PTN. Gene expression profiles for a set of genes for which no interesting interactions were detected by the rule extraction algorithm. The time series show a great variability due to noise effects which influence the state TA detection algorithm; this results in poor performances when deriving activation intervals and, as a consequence, precedence temporal rules.

sequence of particular events occurring during a specific biological process (in this case *activation* events), while they are not aimed at giving information about the possible functional or causal correlation between the involved genes. The connection between *carB* and *tagB*, for example, is not accompanied by any documented regulatory relationship between the two genes. Hypotheses on the possible regulation relationships may however be suggested by the analysis of the specific links in a PTN; these will have to be further investigated through wet-lab experiments for biological validation.

It is important to point out that some of the genes were not extracted by the algorithm as belonging to any significant rule. No interesting activation pattern was indeed found in the corresponding data. A proper explanation for this behavior comes from the analysis of the expression profiles of the involved genes (see Fig. 6, where some of these genes are depicted). As it appears from the picture, the time series show a great variability, probably due to noise effects which influence the state TA detection algorithm; this results in poor performances when deriving activation intervals and, as a consequence, precedence temporal rules.

In addition, we can notice that the reconstructed PTN also shows a ‘false positive’ edge, i.e. a precedence connection between two genes which are instead known to be activated during the same developmental stage. These genes are *ecmB* and *tagB*. Also in this case a deeper analysis of the raw data can help to understand the reason why this link was created. Fig. 7 shows the gene expression profiles of the two genes; in addition, two segments corresponding to the intervals which triggered the detection of the rule $\{tagB\} \rightarrow_{\mathrm{P}} ecmB$ are depicted (black line for *tagB* and grey line for *ecmB*). As it can be noticed, the algorithm extracted a meaningful rule according to the input time series and the selected parameters: an interval of activation for *tagB* is in fact found at the beginning of the observation time span and it is followed by an interval of overexpression of *ecmB* at the end of the observation period. The two intervals last at least three time points and are therefore appropriate to extract a rule with significant support. Therefore, also the detection of false positives may be due to the presence of noise in the data; as it is shown in this example, the effects of noise can be only mitigated by properly choosing the algorithm parameters, but not completely eliminated.

For a more formal evaluation of the performance of the method in terms of precision and recall, we refer the reader to Section 4.2, where a detailed comparison with a standard technique for gene networks reconstruction, the Dynamic Bayesian Network (DBN), is carried out.

4.2. Analysis of cell cycle-related human genes

From the analysis of the network depicted in Fig. 4a it appears that, even in the cell cycle case study, the algorithm was able to extract a satisfactory reconstruction of the timing of the selected patterns in human genes. The genes *CCNE*, *E2F*1, *CDC6* and *PCNA* are found to be synchronized at the G_1/S boundary, as it is suggested in the literature [27]. This first module is found to precede the DNA metabolism genes *RRM2* and *RAD51*, which are synchronized at the beginning of phase S. They also have a co-occurrence connection with gene *CDC2* which instead peaks in mitosis. This second module and the one made up of genes showing a peak in G_2 (*CCNA2*, *TOP2A* and

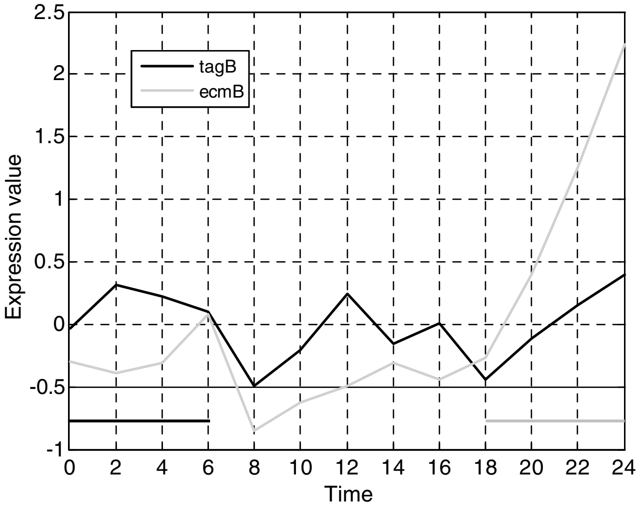

Fig. 7. *tagB* and *ecmB* gene expression profiles. Time series corresponding to the gene expression profiles of the two pre-stalk genes *tagB* and *ecmB*; the two genes are supposed to be activated during the same developmental phase in *Dictyostelium*, but the algorithm derives a false positive precedence connection between the two. The segments corresponding to the intervals which triggered the detection of the rule $\{tagB\} \rightarrow_P ecmB$ (black segment for *tagB* and grey segment for *ecmB*) visually explain the reason why the connection was detected.

CDC2) are found to temporally precede all the genes peaking in M phase and in the M/G_1 transition. Note that for these last two groups no synchronized gene sets are found, which means that none of the extracted rules is able to connect in the same antecedent two genes peaking into the same phase. Moreover, all the extracted rules involve as antecedents genes with peaks in phases S and G_2, i.e. no direct connection appears between genes peaking in M and genes with a peak at M/G_1 boundary. The explanation for this behavior comes from the analysis of Fig. 4b. The intervals related to genes showing peaks in phase M and in the transition M/G_1 are very close to each other, preventing the algorithm from finding any precedence. This is anyway reasonable, since the two considered phases are very close in time, and the expression peaks may not be as separated as in previous conditions. This picture highlights how the information on the intervals can help in the interpretation of results that may be not clear at a first glance. Moreover, in the general situation in which no prior knowledge on the timing of events in a process is available, the only way to represent a PTN is according to Fig. 4b, i.e. by ordering nodes exploiting the information on the intervals.

The distinctive features of a PTN can be further elucidated by comparing the network in Fig. 4a (PTN reconstructed by our algorithm) with the one depicted in Fig. 5a, which represents the result of an algorithm for dynamic Bayesian networks reconstruction. The comparison enlightens a clear difference in the meaning of the edges connecting genes in the two networks. In the PTN of Fig. 4a, all the edges connect genes peaking earlier to genes peaking later in the cell cycle. We can therefore say that connections in a PTN have a clear *temporal* interpretation, as they link genes which are temporally related to each other, giving an insight into the temporal sequence of events (patterns) taking place into the observed process. On the contrary, looking at the DBN in Fig. 5a, we can observe arrows going from genes of later phases to genes peaking in earlier phases of the cell cycle. Besides that, also edges connecting genes peaking into the same cell cycle phase can be noticed (e.g. the link connecting *CDC2* to *CCNA2*). These edges cannot therefore be interpreted in terms of temporal meaning, but they are the representation of a probabilistic relationship which may express a causal connection or a physical interaction between the nodes. An example to illustrate this point is given by the arrow connecting *CDC2* to *CCNE1* in Fig. 5a. From a temporal viewpoint, *CCNE1* peaks earlier than *CDC2*; this sequence of temporal events is described in Fig. 4a by the precedence connection which links *CCNE1* to the module of synchronized genes *RRM2*, *RAD51* and *CDC2*. The edge which directly connects *CDC2* to *CCNE1* in Fig. 5a is instead of a different nature: we can interpret it as an interaction between the two genes, which is documented in the BioGRID database (http://www.thebiogrid.org/).

Besides a qualitative visual comparison of the two networks, we also performed a quantitative analysis to assess the capability of the algorithms in reconstructing temporal and biological interactions relying on a set of known temporal and causal relationships. To this aim, we carried out a comparative evaluation based on *precision* and *recall*. These indexes, coming from the information retrieval field, are defined as follows:

$$Precision = \frac{TP}{TP + FP};$$
$$Recall = \frac{TP}{TP + FN},$$

In the above definitions *TP* indicates the true positives, *FP* the false positives and *FN* the false negatives. In general, the determination of the values for *TP*, *FP* and *FN* depends on the definition of a *target* application which allows the user to define the *positives* class; in our analysis context, this translates into the construction of a sort of ideal network, whose connections are then compared to the ones in the real PTN and DBN.

We have herein considered two cases, which allow us to formally evaluate the differences between the two methodologies and to further support the qualitative observations already pointed out in the previous paragraphs. As a first application we evaluated the capability of the algorithms of reconstructing biological interactions documented in the BioGRID database; to this aim we extracted from such repository all the interactions occurring among the analyzed set of cell cycle genes and considered them as the *positives*. We found a total number of 14 validated biological interactions.

For both the PTN and the DBN we then defined:

- *TP*: number of connections in the network which are supported by the literature;
- *FP*: number of links found by the algorithm but not reported in BioGRID;
- *FN*: number of connections in BioGRID which are not found in the network.

In the second case, we considered *precedence* as the target concept and evaluated the mapped links accordingly. To correctly establish the number of *TP*, *FP* and *FN* we needed to define which are the connections we would expect to find in an ideal network derived relying on the timing of peaks published in the literature [27]; we will call these links the *expected connections*. In more detail, an expected connection is defined as an edge linking a gene peaking in one phase to all the genes peaking into the following two phases of the cell cycle. Following this strategy, we expect each gene of phases G1/S, S and G2 to be linked to eight genes (four in the immediately following phase and four in the next one), genes in phase M to be connected with the four genes of phase M/G1, while genes peaking in phase M/G1 not to show any connection; this results in a total number of 112 expected connections. For both the PTN and the DBN we thus defined:

- *TP*: number of edges in the network which belong to the set of expected connections;
- *FP*: number of links drawn in the network but not included into the expected connections set;
- *FN*: number of expected connections not present in the network.

To perform the evaluation on the PTN we herein considered only precedence edges, since the positive class was defined only in terms of precedence relationships.

The results in Tables 5 and 6 allow us to point out some interesting observations. First, as regards the reconstruction of biological interactions between the genes, small values for precision and recall are found for both the methodologies. As a matter of fact, a small number of true positives and a high number of false positives are visualized by both the networks. As we anyway expected, the number of false positives is much higher in the case of the PTN than in the DBN. This is expected, since the links in a PTN do not describe causal relationships, while they represent the temporal sequence of specific events occurring during the observed process. On the contrary, DBNs are aimed at extracting causal interactions and they have better performance in highlighting such kind of relationships from experimental data.

The capability of PTNs to extract temporal relationships is on the contrary clearly pointed out by the results in Table 6, obtained by considering precedence links as the target. Herein the differences between the two methods are marked by an high difference in the values of precision and recall, coming from different values for *TP* and *FP*. In particular, the number of *TP* relations is high in the PTN while small in the DBN and, even more important, the number of false positives in the DBN is very high with respect to the one for the PTN. This observation further confirms the considerations coming from a first visual inspection of the two networks and strongly underlines the clear differences between the aims of the two networks herein compared.

5. Conclusions

The method presented in this paper represents a paradigmatic shift from the approaches reported in the literature for the analysis of gene expression time series. Traditional clustering algorithms focus on the detection of groups of time series with similar and, usually, synchronized behavior. On the contrary, algorithms for deriving gene regulatory networks, look for causal, deterministic or probabilistic relationships between genes, on the basis of the observation of their time course. The presented approach is aimed at finding temporal relations between specific patterns that the gene may assume over time. The goal is therefore to provide a view of the phenomena under observation which may highlight synchronization and temporization of events. The method is by nature knowledge-based, since it is oriented towards the identification of precedence among interesting temporal patterns, where the interestingness is dependent on the research goals and targets. However, the method can be easily coupled with temporal clustering for deriving the most interesting or frequent patterns occurring in the data. In this case, it is possible either to apply a TA-oriented clustering method already presented by the authors [8] or to exploit a standard clustering approach and then define the patterns to be further analyzed.

One of the main advantages of the proposed methodology is related to its generality. As a matter of fact, it is possible to search for precedence relationships between any temporal pattern which can be represented with a TA. Being TAs very general, this allows exploring any set of temporal relations which may be of interest in biological or clinical research, thus exploiting the capabilities of the rule extraction algorithm on a wide spectrum of applications. A potential drawback is related to the need to carefully specify a large number of design parameters, including those for the basic TAs and the ones involved in the definition of the temporal relationships. This often requires a well-established knowledge of the analyzed problem and a good user's control on the algorithm structures. Besides that, an increase in the number of the considered interesting events could lead to a consequent complexity of the network obtained, making more difficult to interpret the extracted results.

At this regard, the development of a user friendly software tool could be a potential solution to overcome the above mentioned limitations. A graphical interface able to automatically suggest default values for the design parameters and a navigation tool which allows non-expert users to explore even sophisticated networks would help in the interpretation of the results.

Coupled with other methods for gene expression temporal data analysis, the proposed approach can improve the insights in the process under observation. Authors' hope is that the PTN method may provide researchers with a new viewpoint which may help to reveal the complex nature of molecular dynamics.

Acknowledgments

This work is part of the PRIN project "Dynamic modeling of gene and protein expression profiles: clustering techniques and regulatory networks" funded by the Italian Ministry of University and Research. We gratefully acknowledge Carlo Combi for his contribution to the formalization of the temporal rule mining algorithm. We thank Riccardo Porreca for his help in programming the Matlab code and in performing preliminary data analyses. We are in debt with Blaž Zupan, Tomaž Curk, Janez Demšar of the Faculty of Computer Science of the University of Ljubljana and Gad Shaulsky of the Baylor College of Medicine for the knowledge provided on *Dictyostelium* developmental program.

References

[1] de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol 2002;9(1):67–103.

[2] D'Haeseleer P, Liang S, Somogyi R. Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 2000;16(8):707–26.

[3] Liang S, Fuhrman S, Somogyi R. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. Pac Symp Biocomput 1998;98:18–29.

[4] D'haeseleer P, Wen X, Fuhrman S, Somogyi R. Linear modeling of mRNA expression levels during CNS development and injury. Pac Symp Biocomput 1999;4:41–52.

[5] Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. J Comput Biol 2000;7:601–20.

[6] Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet 2003;34(2):166–76.

[7] Eisen M, Spellman PT, Botstein D, Brown PO. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 1998;95:14863–8.

[8] Sacchi L, Bellazzi R, Larizza C, Magni P, Curk T, Petrovic U, et al. TA-clustering: cluster analysis of gene expression profiles through Temporal Abstractions. Int J Med Inform 2005;74(7-8):505–17.

[9] Ferrazzi F, Magni P, Bellazzi R. Random walk models for bayesian clustering of gene expression profiles. Appl Bioinformatics 2005;4(4):263–76.

[10] Ernst J, Bar-Joseph Z. STEM: a tool for the analysis of short time series gene expression data. BMC Bioinformatics 2006;7:191.

[11] Shahar Y. A framework for knowledge-based temporal abstraction. Art Int 1997;90:79–133.

[12] Shahar Y, Musen MA. Knowledge-based temporal abstraction in clinical domains. Artif Intell Med 1996;8(3):267–98.

[13] Larizza C, Moglia A, Stefanelli M. M-HTP: a system for monitoring heart transplant patients. Artif Intell Med 1992;4:111–26.

[14] Combi C, Chittaro L. Abstraction on clinical data sequences: an object-oriented data model and a query language based on the event calculus. Artif Intell Med 1999;17(3):271–301.

[15] Bellazzi R, Larizza C, Riva A. Temporal abstractions for interpreting diabetic patients monitoring data. Intel. Data Anal. 1998;2:97–122.

[16] Allen JF. Towards a general theory of action and time. Art Int 1984;23:123–54.

[17] Bellazzi R, Larizza C, Magni P, Bellazzi R. Temporal data mining for the quality assessment of hemodialysis services. Artif Intell Med 2005;34(1):25–39.

[18] Höppner F, Klawonn F. Learning rules about the development of variables over time. In: Leondes CT, editor. Intelligent systems—techniques and applications, 4. Boca Raton: CRC Press; 2002. p. 201–28.

[19] Kam PS, Fu AWC. Discovering temporal patterns for interval-based events. Proc 2nd Int Conf Data Warehousing and Knowledge Discovery (DaWaK) 2000:317–26.

[20] Winarko E, Roddick JF. Discovering richer temporal association rules from interval-based data. Proc Int Conf Data Warehousing and Knowledge Discovery (DaWaK) 2005:315–25.

[21] Sacchi L, Bellazzi R, Larizza C, Porreca R, Magni P. Learning rules with complex temporal patterns in biomedical domains. Proc AIME 2005:23–32.

[22] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. Proc Int Conf Very Large Databases. Morgan Kaufmann 1994:487–99.

[23] Dechter R, Meiri I, Pearl J. Temporal constraint networks. Art Int 1991;49:61–95.

[24] Van Driessche N, Shaw C, Katoh M, Morio T, Sucgang R, Ibarra M, et al. A transcriptional profile of multicellular development in *Dictyostelium discoideum*. Development 2002;129(7):1543–52.

[25] Van Driessche N, Demsar J, Booth EO, Hill P, Juvan P, Zupan B, et al. Epistasis analysis with global transcriptional phenotypes. Nat Genet 2005;37(5):471–7.

[26] Loomis WF. Role of PKA in the timing of developmental events in *Dictyostelium* cells. Microbiol Mol Biol Rev 1998;62(3):684–94.

[27] Whitfield ML, Sherlock G, Saldanha AJ, Murray JI, Ball CA, Alexander KE, et al. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. Mol Biol Cell 2002;13:1977–2000.

[28] Keogh E, Chu S, Hart D, Pazzani M. Segmenting time series: a survey and novel approach. In: Data mining in time series databases. World Scientific Publishing Company, 2004;1–22.

[29] Smith VA, Yu J, Smulders TV, Hartemink AJ, Jarvis ED. Computational inference of neural information flow networks. PloS Comput Biol 2006;2(11):e161.

[30] Bernard A, Hartemink AJ. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. Pac Symp Biocomput 2005:459–70.