CIMS Technical Report 2005-858

Remembrance of Experiments Past: Analyzing Time Course Datasets to Discover Complex Temporal Invariants*

Marco Antoniotti[1] Naren Ramakrishnan[2] Deept Kumar[2]
Marina Spivak[1] Bud Mishra[1,3]

[1] Courant Institute of Mathematical Sciences,
New York University, New York, NY 10012, USA.
[2] Department of Computer Science, Virginia Tech,
Blacksburg, VA 24061, USA.
[3] NYU School of Medicine,
New York, NY 10016, USA.

February 2, 2005

Abstract

Motivation: Current microarray data analysis techniques draw the biologist's attention to targeted sets of genes but do not otherwise present global and dynamic perspectives (e.g., invariants) inferred collectively over a dataset. Such perspectives are important in order to obtain a process-level understanding of the underlying cellular machinery, especially how cells react, respond, and recover from stresses.

Results: We present GOALIE, a novel computational approach and software system that uncovers formal temporal logic models of biological processes from time course microarray datasets. GOALIE 'redescribes' data into the vocabulary of biological processes and then pieces together these redescriptions into a Kripke-structure model, where possible worlds encode transcriptional states and are connected to future possible worlds. This model then supports various query, inference, and comparative assessment tasks, besides providing descriptive process-level summaries. An application of GOALIE to characterizing the yeast (*S. cerevisiae*) cell cycle is described.

Availability: GOALIE runs on Windows XP platforms and is available on request from the authors.

1 Prologue

Microarray technologies today constitute a popular approach for characterizing cellular transcriptional states by simultaneously measuring mRNA abundance of many thousands of genes. The measured (absolute or relative) gene expression levels while the cell is subjected to a particular ambient condition (e.g., temperature shift, desiccation, or starvation—just a few commonly favored by the biologists) can be readily studied by contemporary statistical methods, visualization techniques, and data mining algorithms.

*This work was supported by grants from NSF's ITR and NGS programs, Defense Advanced Research Projects Agency's (DARPA) BioCOMP program, and the US air force (AFRL).

Typically, such methods draw the biologist's attention to targeted sets of genes, e.g., those that vary in a well correlated manner [ESBB98], are under similar regulatory control [SSR$^+$03], or that have consistent functional annotation or ontological categorizations; yet, information in the full dataset complement, most of it abandoned by these techniques, contains a richer and more detailed picture. For instance, how does a cell marshall resources and respond to a given stress? What events must transpire, in what (partial or total) order, to successfully mobilize defense mechanisms? To obtain such global and dynamic perspectives on transcription states, we must bring together quantitative analysis of microarray datasets with formal models for characterizing the temporal evolution of biological processes.

1.1 Motivating Example

Fig. 1 depicts the well-known state diagram representing the regulation of cell cycle in budding yeast. The M (mitosis) phase is closely followed by cytokinesis and the G1 phase (gap 1), during which the cell grows but does not replicate its DNA. There is then a phase of synthesis (S), i.e., DNA replication, followed by G2 (gap 2). Relative to each other, the gaps constitute the most time in the cell cycle. Since entry to S is carefully controlled, we have broken down G1 into an early-mid part (G1 (I)) during which the cell grows in size and a later part (G1 (II)) beyond which the cell is committed to undergoing one full cycle. G1 (II) effectively acts as a checkpoint to ensure sufficient availability of nutrients, polypeptide mating factors, and significant growth in cell size. If these conditions are not met, then the cell enters a quiescent phase (G0) and might attempt to continue the cell cycle at a later stage.

A formal way to reason about such dynamical systems is to encode their properties in the vernacular of temporal logic. Temporal logics are traditionally defined in terms of Kripke structures (V, E, Π) [CGP99]. Here (V, E) is a directed graph having the reachable states of the system as vertices and state transitions of the system as edges. In the above example, there are six states. Π is a labeling of the states of the system with properties that hold in each state. To obtain a Kripke structure from a reachability graph such as Fig. 1, one first needs to fix a set of atomic propositions AP, which denote the properties of individual states. For instance, we can define a proposition p to be 'cell size large enough for division.' p is hence not true in states M, G1(I), and G0. It, however, becomes true in G1(II). Once we have defined a vocabulary of such propositions, we replace the state symbols (M, G1(I), etc.) with the set of atomic propositions that hold in that state. Thus this is a map Π from the set of states to the power set of AP. Such a resulting structure is a Kripke structure.

Given a formal Kripke structure, we can reason about its properties, perform symbolic model checking, and answer queries about pathways. For instance, if we consider the additional propositions q meaning 'cytokinesis takes place', r meaning 'DNA replication takes place,' and s meaning 'cell is in quiescence,' we can pose the question 'Beginning from when q is true, is there a way to reach a state where r is true, without passing through a state where p is true?' (the answer is 'no'). As another example, 'Beginning from when q is true, is there a way to reach a state where r is true without passing through a state where s is true?' (the answer is 'yes'). As is evident, Kripke structures constitute a powerful mechanism to reason about temporal characteristics of biological systems.

Upon a Kripke structure we can also impose a procedure for labeling the possible worlds with more complex temporal formulæ by appropriately combining other temporal sub-formulæ that have been shown valid inductively. One can then reduce these models to more comprehensible structures by projection and collapsing operations, while maintaining a bisimulation equivalence [CGP99]. Because time has a specific topological interpretation, one can easily mix descriptions of fast operations with slow operations while focusing only on the major biological events and their temporal order. Most importantly, one can query this model to see if a particular biological property holds; one can examine a counter-example to a postulated query when it is falsified; or one may ask for hypothetical properties when certain new properties are speculated to hold true. For instance, by observing carcinoma and sarcoma cancer cells co-cultivated, we may summarize a property such as 'certain processes in sarcoma cells are not activated until certain extra-cellular factors in carcinoma cells are made available.'

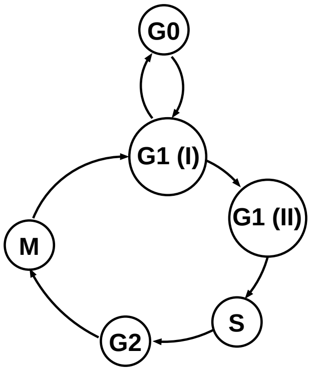


Figure 1: Cell cycle regulation in *S. cerevisiae.*, depicting M, G1 (early-mid and late parts), G0, S, and G2 stages.

2 Basic Approach

How do we obtain Kripke structures in the first place? Typically, only well understood model systems or experimental conditions afford such formal definitions. Notice that this is a problem of both defining a state transition diagram as well as providing a labeling for the states using a vocabulary. In this paper, we present the first approach to automatically infer Kripke structures from time course microarray datasets. Specifically, we bring together our prior work in 'redescription mining' [RKM$^+$04] and 'model checking' algorithms for systems biology [APUM03], yielding a novel approach that presents *global* and *dynamic* perspectives on transcriptional states.

A redescription is a shift-of-vocabulary, or a different way of communicating a given aspect of information. Redescription mining is a technique to find sets (here, of genes) that afford multiple definitions. The inputs to redescription mining are the universal set of ORFs in a given organism, and various subsets (called *descriptors*) defined over this universal set. Such subsets could be based either on prior biological knowledge or defined by the outputs of algorithms operating on gene expression data. Example descriptors can be: 'genes localized in cellular compartment nucleus,' 'genes up-expressed two-fold or more in heat stress,' 'genes encoding for proteins that form the Immunoglobin complex,' and 'genes involved in glucose biosynthesis.' The goal of redescription mining is to connect these diverse vocabularies, by relating set-theoretic constructs formed over the descriptors. For instance, we might find that 'genes expressed in the desiccation experiment except those participating in universal stress response' are the same as 'genes significantly expressed 2-fold positively or negatively in the salt stress experiment.' This redescription relates a set difference on one side to a set union on the other. Such equivalence relationships help bridge diverse ways of qualifying information by pointing out regions of similarity and overlap. The reasoning is that sets that indeed afford multiple descriptions are likely to exhibit interesting behavior and worthy of further investigation. See [RKM$^+$04] for more details.

Redescription mining thus exhibits traits of supervised as well as unsupervised learning, because while the biologist defines the permissible vocabulary (via the descriptors), the algorithm automatically identifies subsets that can be defined in at least two ways. More generally, each descriptor can be thought of as an atomic Boolean formula in a specific universe of biological discourse and redescriptions connect descriptors (and expressions involving them) through a metric quantifying their common supports.

To see how we can use redescriptions to infer Kripke structures, consider one vocabulary based on expression levels in given time points/intervals and another vocabulary based on the GO biological process taxonomy. Redescription in this scenario is equivalent to labeling time-dependent expression clusters (states) with atomic symbols based on GO categories (propositions). To obtain state transitions, we perform redescription again, but this time helping connecting states defined over one time slice to states defined in the neighboring (successive) time slice. Essentially, we have used descriptors defined in a propositional temporal logic and performed redescriptions both within and across intervals of time; by subsequently piecing together these redescriptions into a Kripke-structure model, we obtain a global picture of the temporal nature of the underlying biological processes. This approach effectively integrates our earlier work on model-checking methods [APP$^+$04, APP$^+$03, APUM03] with the data-driven emphasis of redescriptions.

The above ideas have been implemented in the GOALIE software system, an environment that uses the GO biological process ontology to automatically extract temporal invariants. The key research contribution afforded by GOALIE is integrating data-driven reasoning about time course datasets with model-building capabilities. See Fig. 4 for a sneak peek into the type of models constructed by GOALIE. While we have successfully tested this approach with many time course datasets, we present an application to the classical yeast cell cycle dataset of Spellman et al. [SSZ$^+$98] both since this dataset is familiar to the ISMB audience and because it allows us to evaluate our results in the context of a well studied system.

3 Related Work

We primarily survey related work under two themes, namely analyzing microarray datasets and formal models of biological systems.

The literature on interpreting data from large-scale microarray experiments is vast. Classical unsupervised learning techniques identify gene clusters of coordinated activity, which researchers often supplement with post-analysis of upstream regulatory elements or functional enrichment analysis. In some cases, special emphasis has been placed on clustering time course datasets [PNRH03, BJ04]. Researchers have since begun integrating microarray information with other sources of data, primarily to elucidate transcriptional regulatory programs [SSR$^+$03] or to obtain network models of gene regulation [YIJ04]. This thread of data-driven research continues unabated, as more and more experimental data is made available online and researchers find new ways to integrate diverse sources of evidence.

At the other end of the spectrum, researchers begin with a process-level view of biological processes and aim to support query, inference, and simulation tasks. Baral et al. [BCT$^+$04] present a knowledge-based logic and reasoning system for signaling systems. Classical mathematical biologists build ODE models or hybrid system models of well-studied regulatory, metabolic, and signaling pathways [BI99, APUM03], thus offering a dynamical systems perspective on the functioning of molecular machinery. The yeast cell cycle [CCCN$^+$04, BAF$^+$05] and the MAPK signaling pathway [BRI02, BAF$^+$05] are examples of key processes that have been modeled in this manner. There are even projects that emphasize temporal logic and qualitative models of biological systems and pathways [RSS01, CRCD$^+$04, BdJGP03] but they typically concentrate on reasoning and model checking with a given model, and not model inference. The few works that do focus on model inference and model estimation (e.g., [ASR97, Wig03, FP04]) do so at the level of kinetic parameters and rate constants for biochemical pathways, far removed from the abstract forms of representation we are considering here.

We hence posit that in spite of important and impressive progress, existing methods do very little to close the disconnect between experimental results and the biological insights concealed in the data. Our goal in this paper is to present an approach that *directly* yields such insights in the form of temporal invariants from analysis of time-course gene expression datasets, using multi-level redescription analyses as building blocks. This helps overcome both the inability of traditional data mining methods to express their results at the level of relationships between biological processes and the inadequacies of current simulation-based methods to exploit the wealth of data arising from high-throughput sources.

4 Methods

We ease the transition from detailed time course data to temporal formulæ at the level of abstract biological processes by using redescriptions at multiple levels. Given a time course microarray dataset, we begin by a traditional cluster analysis performed on overlapping time windows across the dataset. These clusters essentially constitute the states of the Kripke structure. Each cluster, in each time window, is then redescribed into a union of (many) GO biological process categories. Labels so obtained can be interpreted propositionally as 'genes involved in process p behave concertedly in this state,' or even 'process p persists in this state,' depending on the specific label. The descriptions of GO categories, so obtained, are then related *across time points* using a redescription across time windows. At this stage, we allow one-to-many ('scatter') as well as many-to-one ('gather') redescriptions. Combinations of these multiple-window trends are finally summarized into a Kripke structure, yielding important insight about the global transcriptional activity. We hasten to add that what follows is undoubtedly only one way to organize the computational pipeline, and other possibilities to configure each stage might readily suggest themselves to the reader.

4.1 Clustering and Initial Redescription

Given an $m \times n$ gene expression dataset with m genes whose values have been sampled (uniformly or non-uniformly) at n time instants ($0 \leq t_1 < t_2 < \cdots < t_n \leq T$) over a sufficiently long interval of time $[0, T]$, we aim to exploit the locality information inherent in biological processes by examining correlations among genes over short windows of time. Thus, we tile the time interval $[0, T]$ using k overlapping time-windows:

$$\begin{array}{rll} [T'_0, T_1], & \text{where} & 0 = T'_0; \\ [T'_i, T_{i+1}], & \text{where} & T'_{i-1} < T'_i < T_i, 1 \leq i < k; \\ & \text{and} & T_k = T. \end{array}$$

Each time window $[T'_{j-1}, T_j] = [t_{j,1}, t_{j,2}, \ldots, t_{j,l}]$ then induces an $m \times l$ submatrix of the original dataset and can be subjected to clustering using a variety of metrics (e.g., Euclidean or correlation distances), resulting in a partition into some fixed number (K) of disjoint subsets (clusters):

$$C_{1,j}, C_{2,j}, \ldots, C_{K,j}.$$

These clusters, across all k windows, are used as descriptors for redescription analysis. To ensure that we redescribe into a controlled vocabulary, as mentioned before, we adopt the 8517 biological process ontology terms created by the the Gene Ontology$^{\text{TM}}$ (GO) Consortium; these terms describe broad biological activities, such as *mitosis* or *purine metabolism*, that are accomplished by ordered assemblies of molecular functions.

We begin by redescribing each cluster, by itself, into a union of GO biological process terms. Redescriptions with this particular syntactic bias are the purview of functional enrichment analysis, and there are many ways to obtain such patterns. For instance, we can mine classification rules [AMS97] from singular GO terms to the targeted cluster and attempt to cover the cluster using the antecedents of these rules. One problem with this approach is that it is difficult to determine *a priori* the minimum support threshold to be used for mining the rules. An approach to tile regions by spatially clustering association rules [LSW97] has also been proposed, but this does not apply here since the GO terms do not exhibit any spatial coherence. A second idea is to adopt a greedy sequential set covering strategy where we successively (partially) cover the cluster, remove the genes covered, and repeat. In this paper, we combine a Fisher exact test [BS04, ZFW$^+$03] with an empirical Bayes approach to yield a probabilistic labeling algorithm. Given a cluster $C_{i,j}$ and a GO descriptor G_g, we first construct the contingency table comprising the four disjoint sets: $C_{i,j} \cap G_g$, $C_{i,j} - G_g$, $G_g - C_{i,j}$, and $G - C_{i,j} - G_g$. Equipped with these sets, we can determine a p-value for how accurately the GO term can be used as a label for that cluster. Once the labels for a possible cluster have been ordered by their p-values, an empirical Bayes approach [CLC00] can be used to retain only those labels that satisfy an appropriate false discovery rule, e.g., the Benjamini-Hochberg test. The resulting union of the labels is then characterized by the Jaccard's coefficient that it shares with cluster $C_{i,j}$. The Jaccard's coefficient betwen two sets X and Y is defined by:

$$\frac{|X \cap Y|}{|X \cup Y|}$$

i.e., the ratio of the size of the intersection to the size of their union. The Jaccard's coefficient is 1 if the sets are identical (i.e., the cluster is perfectly covered) and 0 if the sets are disjoint.

4.2 Chasing Clusters Across Time Windows

Notice that, at this stage, we have effectively imposed a labeling function Π on clusters; if we view the clusters as transcription states, then given any state s and any proposition p, we already know whether $p \in \Pi(s)$ or $\neg p \in \Pi(s) \equiv p \notin \Pi(s)$. Our next step is to 'chase' these propositions across time windows using yet another level of redescription.

For a value $0 \leq \theta \leq 1$, we say that two clusters $C_{i,j}$ and $C_{i',j'}$ $(1 \leq i, i' \leq K)$ in time windows j and j' are *θ-equivalent* if the Jaccard's coefficient between $C_{i,j}$ and $C_{i',j'}$ is $\geq \theta$. Let l' be the smallest index of the time window such that for an appropriately chosen θ, clusters in time windows 0, 1 etc. are *θ-equivalent* to clusters in time windows l', $l'+1$ etc. respectively. With these definitions, one can introduce a directed graph $\mathbb{G} = (V, E)$, whose vertices are the clusters, connected by directed edges going from vertex $C_{i,j}$ to $C_{i',j'}$ if and only if $j' - j \equiv 1 \mod l'$ and the Jaccard's coefficient between $C_{i,j}$ and $C_{i',j'}$ is at least θ. We can think of this stage of the process as akin to episode mining in event sequences [MTV97].

4.3 Inferring Temporal Relationships

Now, since the vertices of $\mathbb{G}$ are further labeled by propositions from a universe of discourse, we can view these labeled graphs as Kripke structures, whose vertices are the possible worlds and whose edges are temporal transitions between possible worlds. Since the atomic propositions are chosen from a controlled vocabulary, we can combine these propositions to create formulæ in a propositional temporal logic (CTL) to describe the complex dynamic interactions among the genes. In this case, the truth properties of more complex temporal formula can be computed inductively by the rules shown in Table 1.

We will sometimes allow this graph to be manipulated by graph rewriting rules that will allow projection and collapsing, by allowing clusters along certain directed paths to be combined into bigger clusters or near-by clusters within a time-window to be combined (e.g., if $C_{i,j}$ and $C_{i',j}$ have Jaccard's $> \theta$, they may be replaced by a new cluster $C_{i,j} \cup C_{i',j}$). However, these rewrite rules must be further constrained so that certain 'bisimulation-like' relations hold between uncollapsed and collapsed graphs. These and other such enhancements to the basic algorithm will be discussed in future versions of this work.

5 Software and Implementation

GOALIE is a software tool that embodies these ideas and allows the user to interactively conduct the redescription of clusters, explore chains of GO descriptors across time windows, and track the validity of temporal formulæ, to see if they change state. The system provides hyperlinks to external web sites (e.g., related to definitions of GO categories, public repositories of experimental datasets) as well as visualizations and query interfaces.

See Fig. 2 for a screenshot of GOALIE. In the left part of the view there is a list of overlapping windows from a time course experiment. The right part is divided in two: the top view shows the cluster relationship graph, where the clusters are vertices of the graph and clusters between two successive windows are joined by an edge, if they are not disjoint; the edge is labeled by the cardinality of the set of common GO terms. Notice that the numbering of clusters is arbitrary and merely denotes a descriptor. The bottom part shows the list of GO categories associated with a cluster in a window, or to a connection between two

Syntax	Semantics	Meaning
Base Formulæ		
p	$p \in \Pi(s)$	A proposition
$f_1 \vee f_2$	$f_1 \in \Pi(s)$ or $f_2 \in \Pi(s)$	A disjunction
$f_1 \wedge f_2$	$f_1 \in \Pi(s)$ and $f_2 \in \Pi(s)$	A conjunction
$\neg f$	$f \notin \Pi(s)$	A negation
$f_1 \Rightarrow f_2$	$f_1 \notin \Pi(s)$ or $f_2 \in \Pi(s)$	An implication
Temporal Formulæ		
$\mathbf{EX}(f)$	$f \in \Pi(s')$ and s' is a successor state of s	f will be true in some next state
$\mathbf{AX}(f)$	$f \in \Pi(s')$ for every s' successor of s	f will be true in all the next states
$\mathbf{E}[f_1\,\mathbf{U}\,f_2]$	If $s_0, s_1, \ldots, s_n$ is a sequence of states and at each of them $f_1 \in \Pi(s_i)$ for $i < n$ and $f_2 \in \Pi(s_n)$	There is a sequence of states where f_1 holds *until* f_2 will.
$\mathbf{A}[f_1\,\mathbf{U}\,f_2]$	For any sequence of states $s_0, s_1, \ldots, s_n$ at each of them $f_1 \in \Pi(s_i)$ for $i < n$ and $f_2 \in \Pi(s_n)$	There is a sequence of states where f_1 holds *until* f_2 will.
$\mathbf{EF}(f)$	There is *a sequence* of states where f will *eventually* hold (this is actually an abbreviation for $\mathbf{E}[\mathrm{True}\,\mathbf{U}\,f]$)	This formula represents a *potential* event.
$\mathbf{AF}(f)$	For *any sequence* of states f will *eventually* hold (this is actually an abbreviation for $\mathbf{A}[\mathrm{True}\,\mathbf{U}\,f]$)	This formula represents a *necessary* event.
$\mathbf{EG}(f)$	There is *a sequence* of states, f will *always* hold (this is actually an abbreviation for $\neg\,\mathbf{AF}(\neg f)$)	The formula f will always hold on some path.
$\mathbf{AG}(f)$	For *all sequences* of states, f will *always* hold (this is actually an abbreviation for $\neg\,\mathbf{EF}(\neg f)$)	This formula states a *global* and *invariant* property of the system.

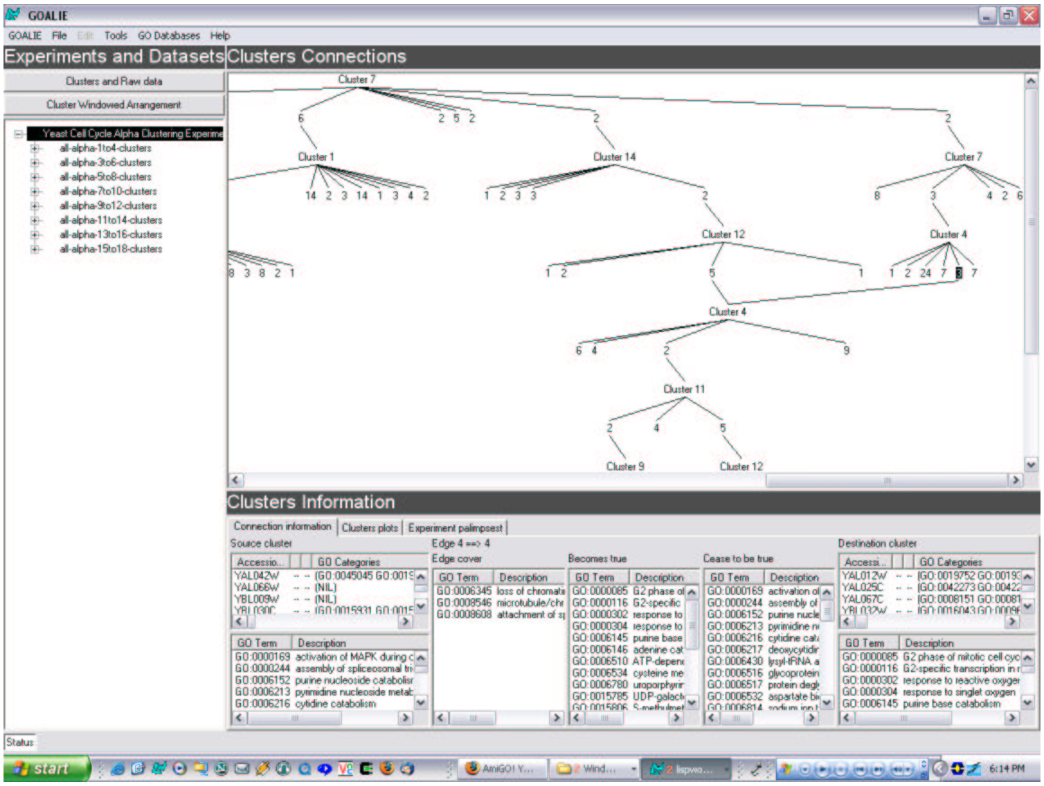
Table 1: Syntax and informal Semantics for CTL under the assignment Π.

clusters. In the example shown in Fig. 2, the viewer displays the GO categories associated to each cluster in the left and right part of the connection, with a list of what is maintained between the two clusters, which GO categories appear anew in the target cluster, and which disappear from the source cluster.

In the visualizer, not all edges are shown at once, but are revealed as the user navigates through the graph. Thus if one wishes to follow a directed path across the time-windows from top to bottom, then the most informative directed paths are those whose edges have relatively large numbers associated with them. Most likely, these paths depict a set of biological processes that persist over time. Other common patterns one discovers manifest as follows: in one time window a set of processes appear in a coordinated fashion, persist for few successive time windows, and then *scatter* into many different sub-processes; or the reverse phenomenon, i.e., many subprocesses *gather* in a time-window and then persist. One such scatter-gather example in the next section examines such phenomena in the context of yeast cell cycle processes.

The overall architecture of GOALIE is that of a system that must *generate* and *maintain* a large set of relationships among several sets. The relationships are uniformly organized in a directed acyclic graph. In particular, the generation algorithms must use appropriate heuristics to tame the complexities inherent in the management of large sets of well formed logical formulæ. A simple counting argument on the structure of CTL formulæ shows that the number of formulæ grows double exponentially as a function of depth d, with a lower-bound of $\Omega\left(2^{2^d}\right)$ – too large to consider even for small values of d. Fortunately, this does not pose a serious problem in realistic biological examples, as most of these formulæ are not significant and are pruned out.

GOALIE relies on a number of public domain tools in its functionality. The GO database [GOd] is an important resource about GO terms and their relationships. The GoMiner [ZFW+03] software is used to produce ORF-to-GO associations. We also employed the K-means algorithm of the Genesis system [Stu04] to produce the initial clustering of each time window.



Source cluster

Accessio...			GO Categories
YAL042W	--	--	(GO:0045045 GO:001
YAL066W	--	--	(NIL)
YBL009W	--	--	(NIL)
YBL030C	--	--	(GO:0015931 GO:001

GO Term	Description
GO:0000169	activation of MAPK during c
GO:0000244	assembly of spliceosomal tri
GO:0006152	purine nucleoside catabolisr
GO:0006213	pyrimidine nucleoside metab
GO:0006216	cytidine catabolism

Edge 4 ==> 4

Edge cover

GO Term	Description
GO:0006345	loss of chromati
GO:0008546	microtubule/chr
GO:0008608	attachment of s

Becomes true

GO Term	Description
GO:0000085	G2 phase of
GO:0000116	G2-specific
GO:0000302	response to
GO:0000304	response to
GO:0006145	purine base
GO:0006146	adenine cat
GO:0006510	ATP-depenc
GO:0006534	cysteine me
GO:0006780	uroporphyrir
GO:0015785	UDP-galact
GO:0015806	S-methylmet

Cease to be true

GO Term	Description
GO:0000169	activation of
GO:0000244	assembly of
GO:0006152	purine nucle
GO:0006213	pyrimidine n
GO:0006216	cytidine cat
GO:0006217	deoxycytidir
GO:0006430	lysyl-tRNA a
GO:0006516	glycoprotein
GO:0006517	protein degl
GO:0006532	aspartate bi
GO:0006814	sodium ion t

Destination cluster

Accessi...			GO Categories
YAL012W	--	--	(GO:0019752 GO:0019
YAL025C	--	--	(GO:0042273 GO:0042
YAL067C	--	--	(GO:0008151 GO:0008
YBL032W	--	--	(GO:0016043 GO:0009

GO Term	Description
GO:0000085	G2 phase of mitotic cell cyc
GO:0000116	G2-specific transcription in r
GO:0000302	response to reactive oxyger
GO:0000304	response to singlet oxygen
GO:0006145	purine base catabolism

Figure 2: A screenshot of the GOALIE tool. The left part of the tool depicts the various time slices utilized in the study. The top right part depicts a snapshot of interactive exploration using redescriptions. The bottom right part identifies propositions that remain true when going from a source cluster to a destination cluster, as well as the propositions that become true, and those that cease to be true. Notice that Cluster 7 in the first window has been 'chased' to yield a chain through successive time windows (Clusters 7, 4, 4, 11, and 12 respectively). The links between clusters are labeled with the cardinality of GO terms in common. For instance, the first edge in this chain involves 2 common GO terms, the second involves 3 common GO terms, and so on.

5.1 Generation of Temporal Descriptions

As a simple example (not yet integrated in the visualization tool), the system can find all the connections which exhibit a constant set of GO categories. These paths indicate that certain GO categories persist throughout the time course measurements. Incidentally, this holds when only *biological processes* GO categories[1] are considered.

Another example of the formulæ generation capabilities of the system involves how we can build an "until" CTL formulæ by analyzing the connections between clusters. These formulæ are of the form: *some GO categories remain active,* ***until*** *some other GO categories become active.* Since we have been considering the biological processes hierarchy so far, we can rephrase the CTL "until" formulæ as *some* ***process*** *persists in the cell* ***until*** *some other* ***process*** *is activated.*

In reference to the upcoming example in Section 6 and to Fig. 4, example temporal logic formulæ generated by GOALIE are shown and explained in Fig. 3. Although we have not encountered this diffi-

[1] Recall that the GO ontology is subdivided into three broad categories: biological processes, cellular components, and cellular function.

```
                                       *   *   *
Exists_path(‘sister chromatid cohesion’ Until (‘G2 phase’ And ‘G2 specific transcription’))
Eventually(Exists_path((‘G2 phase’ And ‘G2 specific transcription’)
                           Until ‘G2/M specific transcription’))
                                       *   *   *
```

Figure 3: GOALIE has all the pre-processed information available to automatically generate these two temporal logic formulæ. The first one states that there exists a directed path connecting a sequence of clusters in successive time windows such that the GO category ‘sister chromatid cohesion’ holds *until* the cell enters G2 phase. The second formula states, albeit obviously, the following: “the cell, after dwelling in G2 phase, enters M phase.” Although this is a well known feature of the cell cycle, it is interesting as it derives automatically from numerical expression matrices and a static ontological annotation.

culty yet, larger data sets might cause GOALIE to generate many more formulæ, which would necessitate heuristics to constrain the number of generated formulæ. Criteria such as novelty, as studied in the data mining community, can be used to filter formulæ that may suggest new interpretations of the data and of the processes involved.

Finally, GOALIE can easily incorporate more traditional *query based* model checking technology [APP+03, APUM03] that can be used by a biologist to formulate Natural Language or CTL queries about the temporal evolution of the system.

6 Experimental Results

We tested our ideas and the GOALIE tool on the well known Spellman et. al. yeast cell cycle data-set [SSZ+98]. The data set comprises several time course microarray measurements of gene expression levels under a number of cell synchronization treatments (α-factor, Cdc15, elutriation). In the following we describe our observations on a dataset prepared from the α-factor time course data using the full set of more than 6000 genes.

We partitioned the α-factor time course data in windows of 4 time points, yielding 8 windows. Each window was partitioned into 15 clusters giving a total number of 120 clusters. These 120 clusters were redescribed at a p-value of 0.05; the redescriptions across windows were then computed using a stringent Jaccard's coefficient $\theta = 0.8$. Fig. 2 shows part of the resulting temporal relationship in a directed acyclic graph connecting the clusters in different time course windows. In the following paragraphs we use the notation: ‘$L : N$,’ with L and N positive integers, to denote cluster N in time course window L.

Time Course Window 1 to Time Course Window 2: Connection 1:15 to 2:4. By inspecting the first cluster in the first window (Cluster 1:15), we note that there is only one connection worth following to a cluster in the second window (Cluster 2:4). The criteria for this choice is that the edge connecting the two clusters is labeled (among many others) by the GO categories “positive regulation of sister chromatid cohesion” (GO:0045876), and that the category “regulation of S phase of mitotic cell cycle” (GO:0007090) labels Cluster 2:4, i.e. it gets activated in the second time course window.

Time Course Window 2 to Time Course Window 3: Connection 2:4 to 3:2. Next, we concentrate on Cluster 2:4 in the second time-course window. There are two connections worth following to time-course window 3: one to Cluster 3:2 and one to Cluster 3:4. Consider the first one to Cluster 3:2. We note that GO category “regulation of S phase of mitotic cell cycle” (GO:0007090) is maintained across the connection, while the category “positive regulation of sister chromatid cohesion” (GO:0045876) becomes inactive.

We will discuss the other relevant connection to Cluster 3:4 in a short while.

Time Course Window 3 to Time Course Window 4: Connection 3:2 to 4:4. The next connection we take into consideration shows how the GO categories “G2 phase of mitotic cell cycle”

(GO:0000085), “G2-specific transcription in mitotic cell cycle” (GO:0000116), “microtubule/chromatin interaction” (GO:008546), and “attachment of spindle microtubules to kinetochore” (GO:008608) become active in Cluster 4:4.

Time Course Window 4 to Time Course Window 5: Connection 4:4 to 5:11. Again there is only one relevant connection to take into consideration. The GO categories “G2 phase of mitotic cell cycle” (GO:0000085), and “G2-specific transcription in mitotic cell cycle” (GO:0000116) remain active, while the two categories “microtubule/chromatin interaction” (GO:008546), and “attachment of spindle microtubules to kinetochore” (GO:008608) become inactive.

Time Course Window 5 to Time Course Window 6: Connection 5:11 to 6:10. In this step we note how the G2 related categories become inactive, while M phase related activities are initiated. The GO categories “G2 phase of mitotic cell cycle” (GO:0000085), “G2-specific transcription in mitotic cell cycle” (GO:0000116) become inactive. The GO categories “G2/M-specific transcription in mitotic cell cycle” (GO:0000117), “positive regulation of sister chromatid cohesion” (GO:0045876) become active.

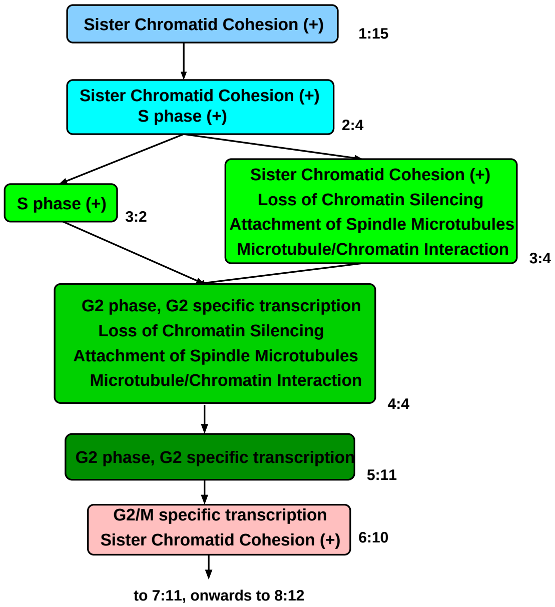
Following the chains downward, eventually we find connections that exhibit the expected transition from G2 to M phases, e.g., in transition from Time Course Window 7 to Time Course Window 8, Connection 7:11 to 8:12.

Going back to the transition between Time Course Window 2 to Time Course Window 3 we can follow another set of connections. The initial connection between Cluster 2:4 and Cluster 3:4 is relevant because of the presence of the “positive regulation of sister chromatid cohesion” (GO:0045876) GO category. Following down one level we find the connection between Cluster 3:4 to Cluster 4:4 with GO categories “loss of chromatine silencing” (GO:0006345), “attachment of spindle microtubules to kinetochore” (GO:008608), “microtubule/chromatin interaction” (GO:008546) staying active between the levels, while “G2 phase of mitotic cell cycle” (GO:0000085), “G2-specific transcription in mitotic cell cycle” (GO:0000116) become active in Cluster 4:4 (as expected). When we summarize these directed paths, we discover the scatter-gather behavior mentioned before (see Fig. 4). The genes participating in this phenomenon appear to move from cluster to cluster, only to end up ultimately behaving in a coordinated manner, after certain key events occur in the cell cycle.

7 Discussion

We believe that the research described in this paper represents preliminary but foundational steps towards extracting temporal invariants of biological processes. The collection of a body of invariants, such as shown in Fig. 3, may reveal interconnections among a set of basic biological functions whose coordinated unraveling could be ‘life itself.’ Evolutionary processes, developmental processes, aging, processes involved in cancer progression, immune reactions, cell cycle, circadian clock, apoptosis, signal transduction, each holds its unique story of a set of processes creating, persisting and annihilating each other. In some cases, for instance, in cancer, cataloging progression of genomic changes (e.g., amplification of oncongenes and deletion of tumor suppression genes) as well as changes in the processes involved in cell proliferation, apoptosis, methylation, angiogenesis and motility, may have enormous value in understanding the disease, variations among different forms of cancer, and the design of therapeutic interventions. We hope to create such studies from existing and future data sets.

Basic to such redescription analysis would be an algorithm to reveal a ‘hidden Kripke model (HKM),’ composed of a set of hidden states or possible worlds, transitions among the states, and the states labeled with logical propositions. At first glance, this may appear to be a variation of the classical hidden Markov model (HMM), an approach enjoying unusual popularity among bioinformaticists. There are several basic differences: there are no obvious emission alphabets that can be observed, rather true logical propositions from a universe of discourse must be inferred or redescribed; no system architecture can be assumed *a priori*, the transitions themselves must be inferred from the structure and the semantics of the possible



Sister Chromatid Cohesion (+)
1:15
Sister Chromatid Cohesion (+)
S phase (+)
2:4
S phase (+)
3:2
Sister Chromatid Cohesion (+)
Loss of Chromatin Silencing
Attachment of Spindle Microtubules
Microtubule/Chromatin Interaction
3:4
G2 phase, G2 specific transcription
Loss of Chromatin Silencing
Attachment of Spindle Microtubules
Microtubule/Chromatin Interaction
4:4
G2 phase, G2 specific transcription
5:11
G2/M specific transcription
Sister Chromatid Cohesion (+)
6:10
to 7:11, onwards to 8:12

Figure 4: One portion of the mined Kripke structure of the yeast cell cycle. Each state in the diagram is labeled with GO descriptor propositions on the inside and time slice/cluster numbers on the outside.

worlds. Once the HKM has been inferred, however, it is expected to be equally powerful in discovering invariants, in predicting dynamic properties of unannotated genes, predicting behavior of a cell, or even an organ or an organism at a system level under various conditions. The underlying theoretical questions are deep and challenging.

In the short term, there are many loose ends in GOALIE to tie up: for instance, we have not provided any optimizing rule for choosing the window size or number of clusters, while recognizing that such choices do affect the model inferred and the invariants discovered. Our “next” operation connecting states models a persistence of process, not any causal relationship; we will need to incorporate known regulatory relationships among proteins, microRNAs, and genes to arrive at such models. We will need to understand model collapsing that can yield a succinct description and yet maintain the truth properties. We will need to understand how to introduce modes other than time: e.g., spatial variation, variations in temperature, nutrients, light and ambient stresses, and more generally, any arbitrary partial order space of control variables. Finally, we must investigate possible multi-modal logics that can express invariants in such rich spaces.

References

[AMS97] K. Ali, S. Manganaris, and R. Srikant. Partial Classification using Association Rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*, pages 115–118, 1997.

[APP$^+$03] M. Antoniotti, F. C. Park, A. Policriti, N. Ugel, and B. Mishra. Foundations of a Query and Simalation System for the Modeling of Biochemical Processes. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2003)*, pages 116–127, Jan 2003.

[APP$^+$04] M. Antoniotti, C. Piazza, A. Policriti, M. Simeoni, and B. Mishra. Taming the Complexity of Biochemical Models through Bisimulation and Collapsing: Theory and Practice. *Theoretical Computer Science*, Vol. 325(1):pages 45–67, 2004.

[APUM03] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Reasoning about Biochemical Processes. *Cell Biochemistry and Biophysics*, 38:271–286, 2003.

[ASR97] A. Arkin, P. Shen, and J. Ross. A Test Case of Correlation Metric Construction of a Reaction Pathway from Measurements. *Science*, Vol. 277(5330):pages 1275–1279, Aug 1997.

[BAF$^+$05] P.E. Barbano, M. Antoniotti, J. Feng, M. Spivak, and B. Mishra. A Coherent Framework for Multi-resolution Analysis of Biological Networks with Memory: RAS pathway, Cell Cycle and Immune System. *PNAS*, 2005. to appear.

[BCT$^+$04] C. Baral, K. Chancellor, N. Tran, N. Tran, A. Joy, and M. Berens. A Knowledge Based Approach for Rpresenting and Reasoning about Signaling Networks. In *Proceedings of the Twelfth International Conference on Intelligent Systems for Molecular Biology (ISMB'04)*, pages 15–22, 2004.

[BdJGP03] G. Batt, H. de Jong, J. Geiselmann, and M. Page. Analysis of Genetic Regulatory Networks: A Model-Checking Approach. In *Proceedings of the 17th International Workshop on Qualitative Reasoning (QR'03)*, Aug 2003.

[BI99] U.S. Bhalla and R. Iyengar. Emergent Properties of Networks of Biological Signaling Pathways. *Science*, Vol. 283(5400):pages 381–387, 1999.

[BJ04] Z. Bar-Joseph. Analyzing TIme Series Gene Expression Data. *Journal of Computational Biology*, Vol. 20(16):2493–2503, 2004.

[BRI02] U.S. Bhalla, P.T. Ram, and R. Iyengar. MAP Kinase Phosphatase as a Locus of Flexibility in a Mitogen-Activated Protein Kinase Signaling Network. *Science*, Vol. 297(5583):1018–1023, 2002.

[BS04] T. Beissbarth and T.P. Speed. GOstat: Finding Statistically Overrepresented Gene Ontologies within a Group of Genes. *Bioinformatics*, Vol. 20(9):pages 1464–1465, Jun 2004.

[CCCN$^+$04] K.C. Chen, L. Calzone, A. Csikasz-Nagy, F.R. Cross, B. Novak, and J.J. Tyson. Integrative Analysis of Cell Cycle Control in Budding Yeast. *Mol Biol Cell*, Vol. 15(8):pages 3841–3862, 2004.

[CGP99] E. M. Clarke, O. Grunberg, and D. A. Peled. *Model Checking.* MIT Press, 1999.

[CLC00] B.P. Carlin, T.A. Louis, and B. Carlin. *Bayes and Empirical Bayes Methods for Data Analysis.* Chapman and Hall/CRC, 2000.

[CRCD$^+$04] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schachter. Modeling and Querying Biomolecular Interconnection Networks. *Theoretical Computer Science*, Vol. 325(1):pages 25–44, Sep 2004.

[ESBB98] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster Analysis and Display of Genome-Wide Expression Patternst. *PNAS*, Vol. 95(25):pages 14863–14868, 1998.

[FP04] C.D. Fabbro and C. Piazza. Preprocessing Biochemical Traces. In *Proceedings of the Workshop on Genomes: Information, Structure, and Complexity*, Sep 2004.

[GOd] Go database. Web site at `http://www.godatabase.org/dev/database/`.

[LSW97] B. Lent, A.N. Swami, and J. Widom. Clustering Association Rules. In *Proceedings of the Thirteenth International Conference on Data Engineering (ICDE'97)*, pages 220–231, 1997.

[MTV97] H. Mannila, H. Toivonen, and A.I. Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, Vol. 1(3):pages 241–258, 1997.

[PNRH03] T.L. Phang, M.C. Neville, M. Rudolph, and L. Hunter. Trajectory Clustering: A Non-Parametric Method for Grouping Gene Expression Time Courses with Application to Mammary Development. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 351–362, Jan 2003.

[RKM$^+$04] N. Ramakrishnan, D. Kumar, B. Mishra, M. Pott, and R.F. Helms. Turning CARTwheels: An Alternating Algorithm for Mining Redescriptions. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 266–275, Aug 2004.

[RSS01] A. Regev, W. Silverman, and E. Shapiro. Representation and Simulation of Biochemical Processes using the pi-calculus Process Algebra. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 459–470, 2001.

[SSR$^+$03] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module Networks: Identifying Regulatory Modules and their Condition-Specific Regulators from Gene Expression Data. *Nature Genetics*, Vol. 34(2):pages 166–176, 2003.

[SSZ$^+$98] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyers, K. Anders, M. B. Eisen, P. O. Brown, D. Bolstein, and B. Futcher. Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast *Saccharomyces Cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[Stu04] A. Sturn. Genesis web site. Web site at `http://genome.tugraz.at`, 2004.

[Wig03] C.H. Wiggins. Process Pathway Inference via Time Series Analysis. *Experimental Mechanics*, Vol. 43(3):pages 361–370, 2003.

[YIJ04] C.-H. Yeang, T. Ideker, and T. Jaakkola. Physical Network Models. *Journal of Computational Biology*, Vol. 11(2/3):pages 243–262, 2004.

[ZFW$^+$03] B.R. Zeeberg, W. Feng, G. Wang, M.D. Wang, A.T. Fojo, M. Sunshine, S. Narasimhan, D.W. Kane, W.C. Reinhold, S. Lababidi, K.J. Bussey, J. Riss, J.C. Barrett, and J.N. Weinstein. GOMiner: A Resource for Biological Interpretation of Genomic and Proteomic Data. *Genome Biology*, Vol. 4(4), 2003.