GOALIE, A Common Lisp Application to Discover Kripke Models: Redescribing Biological Processes from Time-Course Data*

Marco Antoniotti[1], Naren Ramakrishnan[2], and 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] Dept. of Cell Biology, NYU School of Medicine,
New York, NY 10016, USA.

Abstract

GOALIE is a Common Lisp application that redescribes numerical gene expression value measurements into formal temporal logic models of biological processes. It finds extensive uses in the analysis of microarray and other high-throughput biological datasets. GOALIE incorporates several statistical, logical, and ontological modules, connected together through an architecture that exploits various features of several Common Lisp libraries in order to smoothly integrate with popular bioinformatics formats and databases—the most notable example being the Gene Ontology (GO) and the GO database [GO, GODb].

1 Introduction

GOALIE is a novel computational approach and software system that uncovers formal temporal logic models of biological processes from time course microarray datasets. GOALIE 'redescribes' numerical data into the vocabulary of biological processes and then pieces together these redescriptions into a Kripke-structure model, where possible worlds encode transcriptional states, connect by transition relations to future possible worlds, and derive labeling with propositions corresponding to an ontology. Such a model, once constructed by GOALIE, supports various query, inference, and comparative assessment tasks, in addition to presenting descriptive process-level summaries.

In more details, GOALIE takes as inputs numerical measurements from time-course microarray experiments, and begins by clustering overlapping (possibly weighted) time-windows of the dataset. These "timed" clusters constitute the skeleton of a Kripke Model, which, in the process of being unveiled by the redescription algorithm, gets decorated with suitable controlled vocabulary terms – e.g. Gene Ontology (GO) terms - thus giving descriptive meaning to the numerical measurements of the mRNA abundance present in the cell. The outcome is a graph of temporally related terms—the resulting Kripke structure—which can be "read" directly by a biologist. In the case of the GO, the terms used are those pertaining to the *biological processes* ontology. As such, to the biologists, they convey important information about the temporal evolution of various processes in the biological system.

Implementing this application in Common Lisp considerably sped up the construction of those software components which deal with the manipulation of temporal logic formulæ (which constitute a sub-language).

*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).

The numerical procedures involved in the computation of the various statistical parameters (*Jaccard*'s coefficients, p-values and Fisher Exact Test measuring the goodness of the result of set-theoretical operations) also benefited from the concurrent use of different compilers to track down possible inefficiencies.

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 next 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.

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 visiting any 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 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.'

1.2 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 [BJ04] or on reconciling multiple ways of characterizing datasets [RKM$^+$04]. Researchers have since begun integrating microarray information with other sources of data, primarily to elucidate transcriptional regulatory programs [SSR$^+$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.

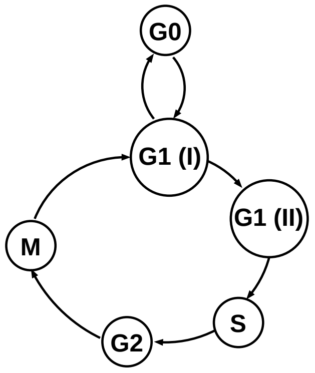


Figure 1: Cell cycle regulation in *S. cerevisiae.*, depicting M, G1 (early-mid and late parts), G0, S, and G2 stages.

At the other end of the spectrum, researchers begin with a process-level view of biological processes and aim to support query, inference, and simulationtasks. 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 [APUM03], thus offering a dynamicalsystems perspective on the functioning of molecular machinery. There are even projects that emphasize temporal logic and qualitative models of biological systems and pathways [BdJGP03, CRCD+04, RSS01] 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]) 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. The approach presented here borrows several conceptual frameworks from our prior work in redescription mining [RKM+04] and model checking algorithms [APUM03], but also deviates significantly by getting closer to biological insights in the form of temporal invariants. 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.

2 Architecture, Implementation, and Methods

The internal architecture of GOALIE is depicted in Figure 2. GOALIE relies on a few basic building blocks to perform its functions. It contains an interface to the GO database, a small library to manipulate a traditional temporal logic language with an S-expression syntax, and a small library of statistical functions for the computation of statistical significance, e.g., p-values and Fisher Exact Test.

The input of GOALIE is a set of "cluster files" in a simple comma separated values (CSV) format. GOALIE does not compute the clusters by itself[1]. From these files the "gene accessions" are associated to the corresponding terms by accessing an instance of the MySQL-based GO database. All the data is loaded and represented internally as a set of hash-tables indexing both gene accessions and GO terms, which are `INTERN`ed as symbols in a separate CL package. The data is loaded into the internal hash-tables in a lazy fashion. Once loaded, the data is ready to be manipulated by GOALIE redescription algorithms.

[1]A Common Lisp library of well-known clustering algorithms – k-means, hierarchical, self-organizing-maps, etc. – would be welcome.

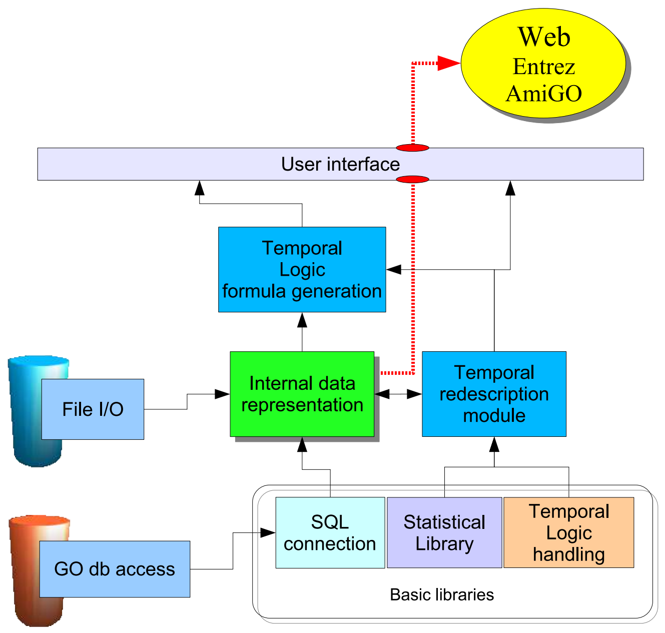


Figure 2: GOALIE Software Architecture.

The graphical user interface presents the user with a summary view of the relationships among the clusters, and allows the user to access the AmiGO web-interface and the Entrez portal.

2.1 GOALIE operations

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 as a conjunction 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.

2.1.1 Initialization and Labeling

Given an $m \times n$ gene expression dataset $\mathcal{D}$ with m genes whose values have been sampled (uniformly or non-uniformly) at n time instants $(t_1 < t_2 < \cdots < t_n)$, we aim to exploit the locality inherent in biological processes by first examining correlations among genes over short windows of time. Specifically, we organize the time interval into k overlapping windows:

$$[T'_1, T_1], \quad \text{where} \quad T'_1 = t_1;$$
$$[T'_k, T_k], \quad \text{where} \quad T_k = t_n.$$

Notice that each time window $[T'_i, T_i]$ induces a submatrix $\mathcal{D}_i$ of the original dataset with m genes and $(n-k)$ time points. The task of mining Kripke structures is then decomposed into clustering each $\mathcal{D}_i, 1 \leq i \leq k$ (yielding the states of the Kripke structure), imposing a labeling function on these states, and tracking these assignments across states in neighboring windows, to identify the state transitions. We employ a k-means algorithm for clustering and combine a Fisher exact test [BS04, ZFW$^+$03] with an empirical Bayes approach to yield a probabilistic labeling algorithm. Given a cluster $C_{i,j}$ (one of the clusters of $\mathcal{D}_i$ and a sample over $\mathbf{X_i}$; note that the cluster number j is arbitrary) 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 $G - C_{i,j} - G_g$, where G is the set of all m genes. 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. At this point, we have effectively imposed the 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 $\neg p \in \Pi(s) \equiv p \notin \Pi(s)$.

2.1.2 Chasing Clusters across Time Windows

Our next step is to 'chase' these propositions across time windows. Essentially, we are performing redescription again, but this time helping connect states defined over one time window to states defined in a neighboring time window. For a value $0 \leq \theta \leq 1$, we say that two clusters $C_{i,j}$ and $C_{i',j'}$ in time windows i and i' are *θ-equivalent* if the Jaccard's coefficient between $C_{i,j}$ and $C_{i',j'}$ is $\geq \theta$. The Jaccard's coefficient $\mathcal{J}(X,Y)$ 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 clusters are perfectly redescribable) and 0 if the sets are disjoint. Here, we qualify the cardinalities and memberships of the clusters not as sets of genes contained but as sets of GO labels assigned. Let l' be the smallest index of the time window such that for an appropriately chosen θ, clusters in time windows 1, 2, ..., are *θ-equivalent* to clusters in time windows l', $l'+1$, ..., 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 $i' - i \equiv 0 \mod (l'-1)$ and the Jaccard's coefficient between $C_{i,j}$ and $C_{i',j'}$ is at least θ.

2.1.3 Inferring Temporal Relationships

Once the vertices of $\mathbb{G}$ are 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.

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

```
(EP "sister chromatid cohesion" :UNTIL (AND "G2 phase" "G2 specific transcription"))
(EVENTUALLY (EP (AND "G2 phase" "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 (the EP operator) 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.

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 asymptotic worst-case complexity 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 [GOda] is an important resource about GO terms and their relationships. The GoMiner [ZFW$^+$03] software is used to produce 'gene accession'-to-GO associations. We also employed the K-means algorithm of the Genesis system [SQT02] to produce the initial clustering of each time window.

2.2 Generation of Temporal Descriptions

The time-windowed cluster graph $\mathbb{G}$ is internally augmented with a set of *chain-info* data structures that keep track of "extent" over time of union-sets of GO categories. This internal data structure allows for a relatively efficient construction of moderately complex CTL formulæ over $\mathbb{G}$. The overall complexity of the naïve construction process is exponential in the number of time windows. However, the largest dataset we have seen so far (the *P. falciparum* dataset) has only 48 time points, which can be subdivided in less than 10 windows, thus limiting the combinatorial explosion of the problem.

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 case holds only when *biological processes* GO categories[2] 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 3 and to Fig. 5, example temporal logic formulæ generated by GOALIE are shown and explained in Fig. 3. Although we have not encountered this difficulty 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

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

evolution of the system.

3 Analysis Examples

GOALIE has been tested on the well known yeast (*S. cerevisiae*) cell cycle dataset of Spellman et al., [SSZ$^+$98], a host-pathogen interaction dataset of *Staphylococcus enterotoxin B* (SEB) infection of human kidney cells[3], and the *P. falciparum* datasets described by Bozdech et. al. [BLP$^+$03].

As an example, we show an analysis produced by GOALIE (see Figure 4). First, consider the connection between time course window 2 to time course window 3: connection 2:4 to 3:2, and 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.

Next, we may consider the connection between time course window 3 to time course window 4, connection 3:2 to 4:4, observed in the analysis of the Spellman's yeast cell-cycle dataset (the notation: '$L : N$,' with L and N positive integers, denotes cluster N in time course window L.) The connection shows how G2 phase in cell-cycle gets initiated: as we enter Cluster 4:4, 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.

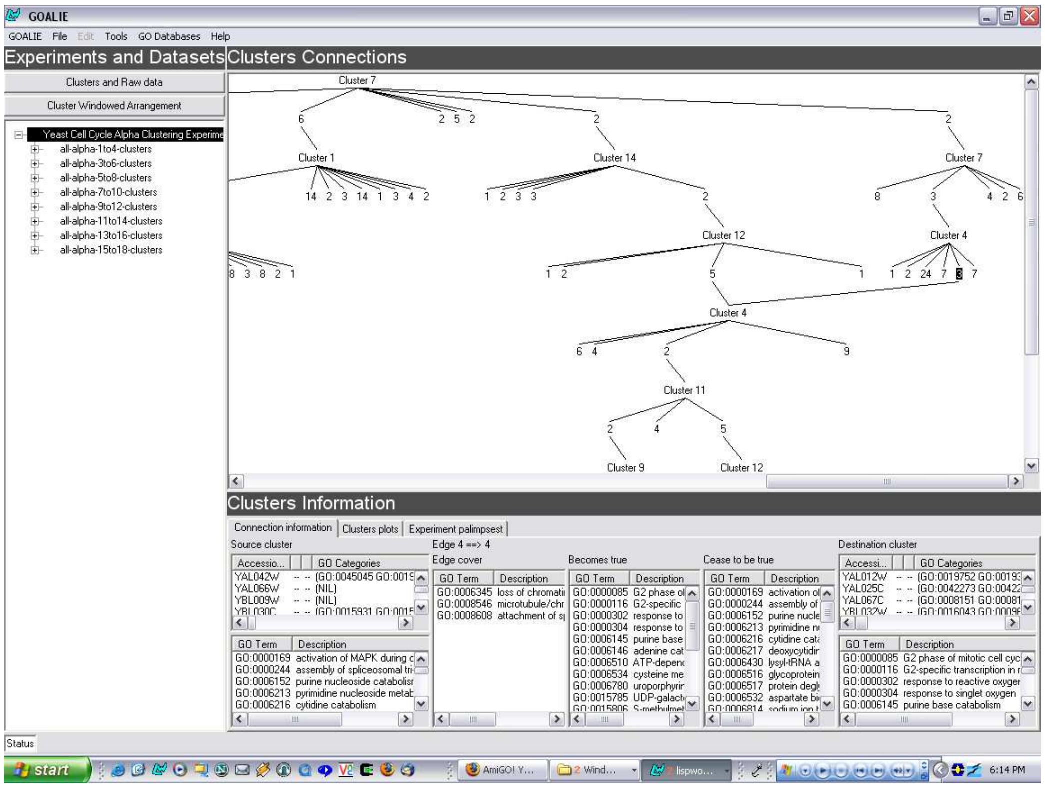
Following the chains downward, eventually we find connections that exhibit the expected transition from G2 to M phases, e.g., in transition from Time Course Window 7 to Time Course Window 8, Connection 7:11 to 8:12.

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 a scatter-gather behavior (see Fig. 5). 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.

4 Ongoing and Future Work

At this time there are many open problems in GOALIE that need a careful and in-depth examination: 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. We are currently casting GOALIE's algorithms in the context of information bottleneck theory [FMST01] in order to find clusterings that are both descriptive in terms of GO process ontology terms [SK02] and that are indicative of connections to clusters in neighboring time windows. Second, we will need to incorporate known regulatory relationships among proteins, microRNAs, and genes to arrive at such models. We will need to understand how to introduce modes other than time: e.g., spatial variation, variations in

[3]Work in progress in collaboration with Dr. Jett's laboratory at Walter Reed Army Institute of Research. Unpublished results.



Source cluster

Accessio...			GO Categories
YAL042W	--	--	(GO:0045045 GO:001S
YAL066W	--	--	(NIL)
YBL009W	--	--	(NIL)
YBL030C	--	--	(GO:0015931 GO:0015

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 sp

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-depend
GO:0006534	cysteine me
GO:0006780	uroporphyrir
GO:0015785	UDP-galacto
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 cata
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:00081
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 4: 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 two common GO terms, the second involves three common GO terms, and so on.

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.

Availability: GOALIE has been built on top of LispWorks (LispWorks Ltd. Cambridge, UK) and runs on Windows XP, Linux and MacOS X platforms. It is currently available on request from the authors under the DARPA Open Source License terms.

References

[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.

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

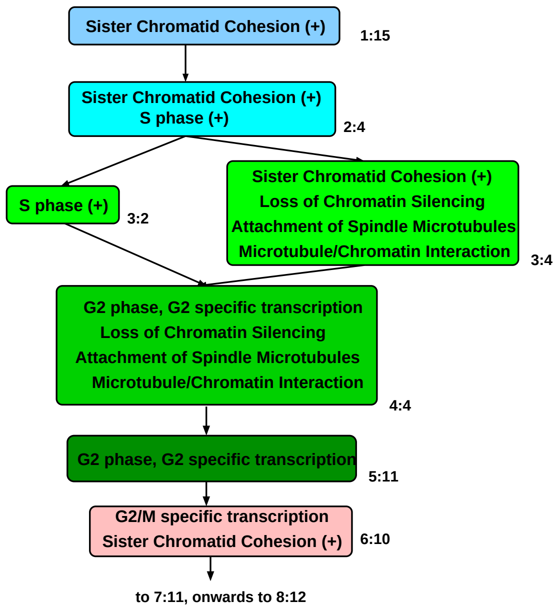

Sister Chromatid Cohesion (+)
1:15
Sister Chromatid Cohesion (+)
S phase (+)
2:4
S phase (+)
3:2
Sister Chromatid Cohesion (+)
Loss of Chromatin Silencing
Attachment of Spindle Microtubules
Microtubule/Chromatin Interaction
3:4
G2 phase, G2 specific transcription
Loss of Chromatin Silencing
Attachment 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 5: 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.

[ARK+05] M. Antoniotti, N. Ramakrishnan, D. Kumar, M. Spivak, and B. Mishra. Remembrance of Experiments Past: Analyzing Time Course Datasets to Discover Complex Temporal Invariants. Technical Report CIMS TR2005-858, Bioinformatics Group, Courant Institute of Mathematical Sciences, New York University, February 2005.

[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.

[BCT+04] C. Baral, K. Chancellor, N. Tran, N. Tran, A. Joy, and M. Berens. A Knowledge Based Approach for Representing 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.

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

[BLP+03] Z. Bozdech, M. Llinas, B. L. Pulliam, J. Zhu, and J. L. DeRisi. The Trascriptome of the Intraerythrocytic Developmental Cycle of *Plasmodium falciparum*. *PLoS Biology*, 1(1), October 2003.

[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.

[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.

[DOS] DARPA BioCOMP Open Source License. https://biospice.org/visitor/documents/BioCOMPLicense.pdf.

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

[FMST01] N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate Information Bottleneck. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 152–161, 2001.

[GO] Gene Ontology Consortium Site. `http://www.geneontology.org`.

[GOda] GO Database. Web site at `http://www.godatabase.org/dev/database/`.

[GODb] GO Database Site. Website at `http://www.godatabase.org/dev`.

[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. Potts, and R. F. Helm. 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. ACM, August 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.

[SK02] J. Sinkkonen and S. Kaski. Clustering Based on Conditional Distributions in an Auxiliary Space. *Neural Computation*, Vol. 14(1):pages 217–239, 2002.

[SQT02] A. Sturn, J. Quackenbush, and Z. Trajanoski. Genesis: Cluster Analysis of Microarray Data. *Bioinformatics*, Vol. 18(1):pages 207–208, 2002. Available at: http://genome.tugraz.at.

[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:pages 3273–3297, 1998.

[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.