Systems Biology via Redescription and Ontologies (I): Finding Phase Changes with Applications to Malaria Temporal Data

Samantha Kleinberg† Kevin Casey† Bud Mishra†‡

April 8, 2008

†Courant Institute of Mathematical Sciences, New York University, New York, New York, United States of America
‡New York University School of Medicine, New York, New York, United States of America
Corresponding Author: Samantha Kleinberg, samantha@cs.nyu.edu, 715 Broadway 10th floor, New York, NY 10012, USA
Word count: 5,383 text, 204 abstract
Number of Figures and Tables: 3 figures, 1 table. All figures may be used as cover images.

Abstract

Biological systems are complex and often composed of many subtly interacting components. Furthermore, such systems evolve through time and, as the underlying biology executes its genetic program, the relationships between components change and undergo dynamic reorganization. Characterizing these relationships precisely is a challenging task, but one that must be undertaken if we are to understand these systems in sufficient detail. One set of tools that may prove useful are the formal principles of model building and checking, which could allow the biologist to frame these inherently temporal questions in a sufficiently rigorous framework. In response to these challenges, GOALIE (Gene Ontology Algorithmic Logic and Information Extractor) was developed and has been successfully employed in the analysis of high throughput biological data (e.g. time-course gene-expression microarray data and neural spike train recordings). The method has applications to a wide variety of temporal data, indeed any data for which there exist ontological descriptions. This paper describes the algorithms behind GOALIE and its use in the study of the Intraerythrocytic Developmental Cycle (IDC) of *Plasmodium Falciparum*, the parasite responsible for a deadly form of chloroquine resistant malaria. We focus in particular on the problem of finding phase changes, times of reorganization of transcriptional control.

Summary: This paper demonstrates the utility of the GOALIE system, an ontology-based model-checking algorithm devised for the purpose of analyzing large complex time-course data. Using GOALIE applied to gene-expression time-course data, one is able to successfully recover the main structure of the life cycle of the Malaria parasite *P. Falciparum* in a completely automated manner. GOALIE accomplished this with prior knowledge of the underlying biology limited to ontological descriptions and without the use of frequency based methods.

Keywords: information theory, microarray data, model checking, ontology, redescription, timecourse data

List of Abbreviations:CTL (Computation Tree Logic), FFT (Fast Fourier Transform), GOALIE (Gene Ontology Algorithmic Logic and Invariant Extractor), GO (Gene Ontology), HKM (Hidden Kripke Model), IDC (Intraerythrocytic Developmental Cycle), LTL (Linear temporal logic), MEA (multi-neuronal electrode array), MPI (Message Passing Interface), ORF (open reading frame), *P. Falciparum* (*Plasmodium Falciparum*), *S. Cerevisiae* (*Saccharomyces cerevisiae*), SEB (Staphylococcus enterotoxin B), STEM (Short Time-series Expression Miner), rRNA (ribosomal RNA), tRNA (transfer RNA)

1 Introduction

"If we describe a game of chess, but do not mention the existence or role of the pawns, one may say we have provided an incomplete description of the game. However, it can also be said that what we have done is given a complete description of a simpler game"(See Wittgenstein [Wittgenstein 1934]). This is essentially the problem we face in the analysis of large biological systems, where we may not have a complete description of either the players or their roles. One way to mitigate this difficulty in the context of systems-biological data analysis is by combining our knowledge of gene expression patterns and biological processes so that information about one may shed light on the other.

This paper shows that, by inferring biological rules from studying the visible interactions, one can provide a description of the dynamics of the system with no prior knowledge of the system's underlying structure, aside from the functional annotations of individual genes. Thus, the paper makes contributions to several fields: (1) To information theory, e.g. rate distortion theory, by defining parsimonious phenomenological models in biology, (2) To systems biology, e.g. model checking of biochemical systems, by devising hidden Kripke models in terms of successive temporal states that are indiscernible in standard clustering methods, and (3) To philosophy of discourse, e.g. redescription and ontology, by showing how to automatically translate static ontologies to dynamic ones.

1.1 Motivating Example

Up to half a billion new cases of malaria are reported annually. The parasite *Plasmodium falciparum*, a strain of *Plasmodium*, is responsible for a deadly form of drug-resistant malaria in humans, resulting in as many as two million deaths each year, and leading to many of the hundreds of millions of malaria episodes worldwide. While great gains have been made in the fight against malaria via drugs, vector control and public health, a long-term solution to the disease remains yet to be found. With no present malaria vaccine, the disease continues to affect the lives and economies of many nations, taking a particularly devastating toll in many developing countries. The genomic information of *P. falciparum*, recently sequenced, is hoped to provide insight into the function and regulation of *P. falciparum*'s over 5,400 genes and should bolster the search for future treatments as well as a possible vaccine.

Transmitted by mosquitoes, the protozoan *Plasmodium falciparum* exhibits a complex life cycle involving a mosquito vector and a human host. Once the infection is initiated via sporozoites injected with the saliva of a feeding mosquito, *P. falciparum*'s major life cycle phases commence. These phases are: liver stage, blood stage, sexual stage, and sporogony. The blood stage is characterized by a number of distinct and carefully programmed substages which include the ring, trophozoite and schizont; these are referred to collectively as the intraerythrocytic developmental cycle (IDC).

This study presents our results of the analysis of the Intraerythrocytic Developmental Cycle (IDC) of *P. Falciparum* as previously described by Bozdech

et al. in [Bozdech et al. 2003]. *P. Falciparum* is a strain of the human malaria parasite that was recently sequenced. This new information allows one the opportunity to gain further insight into the role of *P. falciparum*'s approximately 5,400 genes, the majority of whose functions remain unknown. It has been shown that a large percentage of the genome is active during the IDC and that the regulation pattern is such that as one set of genes is deactivated, another is being turned on, causing what the authors of [Bozdech et al. 2003] refer to as a continuous cascade of activity, in which transcriptional regulation is controlled in a tightly timed choreography. The malaria parasite was chosen for this study due to the simplicity of its regulation pattern, making it a good candidate for determining whether we are able to replicate known results. Yet, traditional approaches to understand the structure of the temporal relations among these key processes have been difficult, and required tedious manual intervention. In this paper, we demonstrate GOALIE's ability to automatically reconstruct the main features of the system, including the cascade of gene expression, as well as the stages of the IDC and their associated processes. Fig. 1 depicts the IDC stages as found by GOALIE. We find that in most cases, genes remain in the same clusters throughout the time course, further supporting the results of [Bozdech et al. 2003].

Bozdech et al. conducted their investigation with the help of Fourier analysis, using the frequency and phase of the gene profiles to filter and categorize the expression data. They used the FFT (Fast Fourier Transform) data to eliminate noisy genes and those that lacked differential expression. Most of the profiles registered a single low frequency peak in the power spectrum, which the authors used to classify the expression profiles. Classified in this way, the cascading behavior of the genes involved in the IDC was clear. Our method reproduced this cascade of expression in an automated manner and without relying on the implicit assumptions of the frequency based methods. To recover the underlying structure of the system, we employed an approach that combined information theoretic techniques developed by engineers with the redescription theoretic techniques of philosophers.

1.2 Related work

Many prior methods for analyzing microarray data have focused on clustering, that is, on breaking the data up into similarly behaving groups[Bar-Joseph 2004]. For temporally ordered data, this step has often required clustering the entire time course experiment into sets of genes (forcing genes to remain in the same cluster throughout the evolution of the system) or clustering by function, using an ontology such as the Gene Ontology (GO)[Ashburner et al. 2000] (grouping genes responsible for similar functions together). These methods are limited by their failure to account for the fact that correlations in expression activity between genes are dynamic and that coexpression changes with time. As conditions change, genes may be expressed similarly for a brief period before diverging. Thus, what is necessary is a system for finding critical time points at which transcriptional control is reorganized. These may then be used to de-

scribe the biological events under study, taking into account both expression levels and functional descriptions. This approach focuses biologists' attention on smaller sets of genes and processes that are likely to be interesting and that may warrant further exploration.

Related tools tend to be focused on a specific problem, such as STEM [Ernst and Bar-Joseph 2006], which was developed for the study of short time series, and GoMiner [Zeeberg et al. 2003, Zeeberg et al. 2005] which has recently expanded to include time course and multiple microarray experiments. The dominant paradigm of our tool differs significantly from these: namely, by utilizing information theory and temporal logic we are able to create a compact representation of the data that is easily visualized and manipulated and that summarizes the key elements in the data from a biological, rather than purely numerical perspective.

2 Materials and Methods

2.1 Temporal redescription approach

To address these problems, we developed GOALIE (Gene Ontology Algorithmic Logic and Information Extraction), which combines ideas from information theory, model checking and logic to provide a temporal redescription of large scale time course experiments. This method is based on the translation of genes into a controlled vocabulary, such as the Gene Ontology (GO) [Ashburner et al. 2000], and then a stitching together of these translations to form a picture of the biological system as it evolves over time.

We begin our analysis by partitioning the entire time course dataset into (possibly non-uniform) windows in time. These windows are defined by $[T_s, T_e]$, their start and end times. Each window contains all of the genes in the dataset for a continuous subset of the time points. We use a clustering approach based on rate distortion theory[Casagrande et al. 2007] to find the start and end points of these windows. Based on this clustering, we track biological processes as they move across windows throughout the experiment.

We connect the clusters to form a graphical representation of the temporal formulae found to be true within the system. This Hidden Kripke Model (HKM), which results from connecting the clusters across neighboring windows, provides a structure for generating and testing temporal logic formulae. We may discover simple properties of the system such as those that hold throughout (e.g. a gene is continuously expressed), and temporal relationships between genes (e.g. A is expressed and then B is expressed). These can also be combined to form testable hypotheses such as "Once A is true, is it possible to get to a state where C is true without going through B?" All such rules are implicit in the HKM, and are not explicitly returned as the number of generated formulae may be so large as to obscure their meaning[1].

[1]Future work includes support for directly querying the HKM using syntax similar to that of database queries.

We have used this core methodology to successfully reconstruct the yeast (*S. Cerevisiae*) cell cycle [Spellman et al. 1998, Kleinberg et al. 2006], for the study of a host-pathogen interaction dataset of Staphylococcus enterotoxin B (SEB) infection of human kidney cells, and more recently, in the analysis of synthetic multi-neuronal electrode array (MEA) data[Kleinberg et al. 2008].

2.2 Methods in detail

The main features of our approach are model building through lossy compression and redescription and subsequent model checking. We first use information theory to derive a compressed representation (clustering) of the expression data, we then "redescribe" the data using the vocabulary provided by the Gene Ontology. Redescription is accomplished by labeling the clusters with their functional enrichments (a common practice in microarray analysis). This condensed representation summarizes each cluster by the statistically most relevant processes controlled by its genes.

2.2.1 Rate Distortion Theory

We are interested in deriving a redescription that captures the dynamics of the data set with respect to some ontological labeling. We would like a concise description of the data that minimizes some measure of the distortion or disagreement between our description and the gene expression profiles, and that highlights the points in time during which significant process level reorganization occurs. We desire a formalism that we can use to represent such distortions precisely, allowing us to specify an objective function that we can minimize, thus obtaining an optimal partition of our data. We call the problem of finding this compressed representation, as well as the "interesting" time points, the "time course segmentation problem".

In rate distortion theory [Cover and Thomas 1991, Cilibrasi and Vitányi 2005], one desires a compressed representation Z of a random variable X that minimizes some measure of distortion between the data elements $x \in X$ and their prototypes $z \in Z$. Taking $I(Z;X)$, the mutual information between Z and X, to be a measure of the compactness or degree of compression of the new representation, and defining a distortion measure $d(x,z)$ that measures "distance" between clusters and data elements, one can frame the clustering problem as a trade-off between compression and average distortion. One balances the desire to achieve a compressed description of the data with the precision of the clustering, as measured by the average distortion, and finds the appropriate balance that maintains enough information while eliminating noise and inessential details.

This trade-off is characterized mathematically as an optimization problem:

$$\mathcal{F}_{min} = I(Z;X) + \beta\langle d(x,z)\rangle \tag{1}$$

where mutual information and average distortion are defined to be:

$$I(Z;X) = \sum_{x,z} p(z|x)p(x)log\frac{p(z|x)}{p(z)} \tag{2}$$

$$\langle d(x,z)\rangle = \sum_{x,z} p(x)p(z|x)d(x,z) \tag{3}$$

and

$$d(x,z) = \sum_{x_1} p(x_1|z)d(x_1,x) \tag{4}$$

This is simply the weighted sum of the distortions between the data elements and their prototypes. The problem is characterized in terms of minimization as we are attempting to use as few possible clusters, while also minimizing the distortion. That is, if we put all elements in one cluster, then the number of clusters will be minimized, but the distortion will be very high. This is why we must minimize the function as a whole.

More recently, Slonim et al. [Slonim et al. 2005] have discussed a modification to rate distortion clustering for which only relations between data elements are used in the distortion function, rather than an explicit mention of cluster prototypes. We have used a similar approach in our graph search based approach to the time course segmentation problem.

We focus on the problem of compressing a given time-course data set into a series of clustered windows. The functional above captures the compression/ precision trade-off inherent in the clustering problem and when combined with a shortest path graph search algorithm (as described in Section 2.3.1), it allows one to use an iterative method, to find a numerical solution to our time course segmentation problem. The trade-off is controlled by the Lagrange parameter β that sets the balance between compression and preservation of relevant information, as β becomes large we focus on precision, as β tends to zero we focus more on compression. Setting the segmentation problem up in this way allows us to find both an optimal windowing of our data, as well as optimal clusters of genes within the windows. From this compressed representation, we can create an optimal redescription. These functions are computed on the raw data, with no noise correction or discretization. Evaluation of the quality of clustering can be done visually, by creating rate distortion curves that depict the trade-off between compression and distortion, or by measuring the coherence of clusters, how they relate to qualitative groups such as by GO annotations. Additionally, when the correct clustering is known, as in the case of synthetically generated examples or well studied systems, we may measure how well the clusterings agree by using a distance measure based on conditional entropy.

2.2.2 Hidden Kripke Models

One of the components of this methodology is the use of temporal logic in the form of Hidden Kripke Models (HKMs). A Kripke structure is defined by (S, S_0, L, R):

- S, a finite set of states;
- $S_0 \subseteq S$, the set of initial states;
- $L : S \rightarrow 2^{AP}$, a labeling of the states with the set of atomic propositions true within that state; and
- $R \subseteq S \times S$, a transition function between states.

Kripke structures[Clarke et al. 1999] are models for modal logic for which vertex-labeled directed graphs are defined by their vertices (V) (i.e. the reachable states of the system), edges ($E \subseteq V \times V$) (i.e. the transitions between the states) and properties (P) (i.e., the labels affixed to the states indicating the properties that hold true within them). In our case, the vertices correspond to clusters, edges to connections between clusters and the properties correspond to the ontological categories from GO. We introduce the terminology "Hidden" Kripke Models by analogy to Hidden Markov Models, in that the states described by our Kripke structures are not known a priori.

Using this framework, we can ask questions about pathways through time, using propositional temporal logic. Computation Tree Logic (CTL), is comprised of propositions, Boolean connectives and modal operators [Emerson 1997]. The main feature of CTL that differs from other propositional temporal logics (e.g., LTL) is the provision for branching time. That is, an event does not have to hold for every possible traversal of the system. We have the modal operators A, which means "for all paths" and E, which means "exists a path." For example, we may ask "starting when q is true, is it possible to reach r without going through p?" In the case of the *P. falciparum* data, we can make queries to test hypotheses such as "A transcription U translation." This logical formula, which uses the always and until operators, means that there is no path in the HKM in which translation occurs and is not preceded by transcription. If we replaced the A with E in the preceding formula, this modified query would inquire whether there is at least one path in which the formula is true. More detailed examples may be found in [Antoniotti et al. 2003].

2.3 Computation Steps

2.3.1 Time Series Segmentation

Generally, we would like to cluster our data in both the genes and in time. In other words, we would like a procedure that yields windows in time that capture intervals of concerted gene activity, in which the genes are clustered into a number of groups of co-expressed elements. From such a compressed representation, we can produce a redescription that has a number of locations equal to the number of time windows, and for which the dynamics are less complex because we derive them from the clustered data rather than from individual genes.

Let $T = \{T_1, T_2, \ldots, T_n\}$ be a sequence of time points at which a given system is sampled, and l_{min} and l_{max} be the minimum and maximum window lengths respectively. For each time point $T_a \in T$, we define a candidate set of

windows starting from T_a as $S_{T_a} = \{W_{T_a}^{T_b} | l_{min} \leq T_b - T_a \leq l_{max}\}$, where $W_{T_a}^{T_b}$ is the window containing the time points $T_a, T_{a+1}, \ldots, T_b$. Each of these windows may then be clustered and labeled with a score based on its length and the cost associated with the clustering functional defined in (Equation 1). Following scoring, we formulate the problem of finding the lowest cost windowing of our time series in terms of a graph search problem and use a shortest path algorithm to generate the final set of (non-overlapping) time windows that fully cover the original series.

To score the windows, we use a variant of rate distortion clustering and a pairwise distortion function based on Pearson correlation. We aim to maximize compression (by minimizing the mutual information between the clusters and data elements), while at the same time forcing our clusters to have minimal distortion (as described in [Slonim et al. 2005]).

We perform model selection by iterating over the number of clusters while optimizing (line search) over β. This procedure results in a fairly complete sampling of the rate-distortion curves. We trace the various solutions for different model sizes while tuning β and choose the simplest model that achieves minimal cost in the target functional. In this way, we obtain a score for each window that is the minimum cost in terms of the trade-off between compression and precision. This method is computationally expensive and run times can be substantial, $O(N^5 \cdot N_c)$, where N is the number of time points in the window and N_c is the number of clusters. For this reason we have developed a parallel implementation that uses the Message Passing Interface (MPI) [Forum 1994] to execute on a cluster of nodes, and used that implementation in this study.

Once the scores are generated, we pose the problem of finding the lowest cost windowing of the time series as a graph search problem. We consider a graph $G = (V, E)$ for which the vertices are time points $V = \{T_1, T_2, \ldots, T_n\}$, and the edges represent windows with associated scores. Each edge $e_{ab} \in E$ represents the corresponding window $W_{T_a}^{T_b}$ from time point T_a to time point T_b, and has an initially infinite positive cost. The edges are labeled with the costs for the windows they represent, each edge e_{ab} gets assigned a cost $(\mathcal{F}_{ab} \cdot length)$ where $\mathcal{F}_{ab}$ is the minimum cost found by the clustering procedure and length is the length of the window $(b - a)$. Our original problem of segmenting the time series into an optimal sequence of windows can now be formulated as finding the minimal cost path from the vertex T_1 to the vertex T_n. The vertices on the path with minimal cost represent the points at which our optimal windows begin and end. We use a shortest path algorithm and generate a windowing that segments our original time series data into a sequence of optimal windows which perform maximal compression in terms of the clustering cost functional.

2.3.2 Connecting clusters across windows

After computing the clusters, we use ontology relationships between clusters to connect those in neighboring windows. For each cluster in each window, we use the Fisher-Exact test with Benjamini-Hochberg correction to determine the GO terms enriching the cluster. Then, for two clusters in neighboring windows, we

compute the Jaccard coefficient to determine whether they should be connected. The Jaccard coefficient is the ratio of the intersection of the sets divided by their union. Two clusters, C_i and C_j, are then θ-*equivalent* if their computed coefficient between the sets of GO ids labeling each cluster is $\geq \theta$. Then, when constructing the cluster graph, we place an edge between C_i and C_j if they reside in neighboring slices of time and are θ- *equivalent* for some θ. In the case of $\theta = 1$, the clusters are described by identical processes from one window to the next, while at the other extreme, $\theta = 0$, the clusters have no common labels.

3 Results

3.1 Software

The GOALIE software is divided into two sequential parts, an initial clustering application that employs rate distortion theory to provide a segmentation of the data set and a second application that performs redescription and visualization. The clustering software performs the segmentation of the time course data and outputs the cluster files for each time window. The redescription and visualization software has two main parts: the experiment information displays, and the graph view of the generated HKM. Using the graph view one may select GO terms and genes of interest. The graph is organized such that each vertical grouping of clusters represents a temporal window, with each vertex displayed as a cluster and connections between vertices representing ontology terms persisting between clusters (i.e., across critical time points). Also included are tools to facilitate visualization of clusters and cluster-cluster connections. These include: scaled Venn diagrams that depict the intersection of genes in pairs of clusters, plots of expression activity for each gene in each cluster, integration with the GO database to view the GO terms associated with each gene and the ability to browse the ontology.

In this study we analyzed the overview dataset provided by Bozdech et al. [Bozdech et al. 2003]. There were 3719 oligonucleotides (represented by 2714 unique open reading frames (ORFs)) for which 1878 (approximately 50 percent) had a total of 6943 associated GO terms. While the ontological descriptions are a large component of our tool, it is possible to reconstruct the system with sparsely annotated data. Further, the use of GOALIE for redescription and visualization facilitates hypothesis generation with respect to the function of unlabeled genes (i.e. genes for which there are no associated ontological labels).

3.2 Cluster Graph

The main output display of GOALIE is the cluster graph. This is the visual display of the HKM and all of its associated information. For the dataset studied here, there are 4-5 clusters per window, and five windows. By studying the cluster centroid graphs (mean profiles for the expression patterns of the genes in each cluster), we can visually verify the cascade of genes as described

in [Bozdech et al. 2003]. In figure 2, the thickness of the red edges (cluster connections) denotes that many of the terms selected (those related to biosynthesis, glycolysis, translation, and transcription), traveled along the same paths through time (i.e. they were in the connections between the clusters connected by the edges). This inference is consistent with the earlier semi-manual data analysis presented in [Bozdech et al. 2003].

3.2.1 Windows

The windowing of the data, discovered using our rate distortion theory based segmentation method, corresponds well to the main stages of the *P. Falciparum* IDC as described in [Bozdech et al. 2003]. When the segmentation is run on the overview dataset, critical time points 7, 16, 28 and 43 drop out of the method as points at which the amount of compression that can be accomplished on the data changes significantly. These critical points signal times at which major functional reorganization of gene expression is likely to be taking place. Bozdech et al. note that the 17th and 29th hour time points correspond to the ring-to-trophozoite and trophozoite-to-schizont stages of the IDC, which agrees well with our automated method. As one may verify visually from the plotted data, notches in the aggregate profile of the expression data occur at roughly these locations, which are also the locations found via frequency analysis [Bozdech et al. 2003] to be transitions between major functional stages (i.e., ring/trophozoite and trophozoite/schizont). The first critical time point produced by our clustering, at hour 7, corresponds to the end of the previous merozoite invasion. The last critical time point produced by our clustering, at hour 43, corresponds to the final portion of the schizont stage overlapping with the early portion of the next period.

Below we use the notation $W : C$ to denote the Cth cluster in the Wth window. (See figure 2.)

1:1 This cluster is about to enter the ring stage. It comprised 631 ORFs and is labeled by ontology terms related to biosynthesis, glycolysis, and transcription.

1:2 This cluster is about to enter the ring stage. In this cluster there are 835 ORFs, which are primarily involved in translation and tRNA and rRNA processing.

1:0 and 1:3 are at the end of the last cycle.

2:3 and 2:1 These clusters followed from 1:1 and 1:2, and have expression in a "hump" shape, corresponding to the ring stage.

2:0 This cluster shows the overlap from one stage to the next, forming the cascade of genetic activity. It is in the Early Trophozoite stage. This transition comprised 957 ORFs, which agrees quite closely with 950 ORFs found by Bozdech et al.

3:3 This cluster contains 1400 genes, those that were involved in the ring stage, which is now winding down.

3:0 This cluster contains Trophozoite ORFs (379), while 3:2 contains 1400 genes expressed later in this stage.

4:3 and 4:0 These clusters contain ORFs which were involved in the late Trophozoite stage.

4:2 This cluster contains ORFs expressed in the late trophozoite stage and 4:1 contains 669 ORFs that are beginning the schizont stage. These clusters have a total of 1161 ORFs (as compared to 1,050 as found by Bozdech et al.)

5:3 This cluster comprised solely ORFs from 4:2 and 4:1 which are completing the schizont stage.

5:1 This cluster contains 524 ORFs that are highly expressed in the late schizont stage and which have early-ring stage annotations. This is consistent with prior findings of "approximately 550 such genes" [Bozdech et al. 2003].

3.3 Gantt chart view

A second way one may interpret the results is by using Gantt Charts[Clark 1952], bar graphs for visualizing data with a temporal component. In GOALIE, these graphs are available for each ontology term within the dataset. They contain one bar per window, color coded to show the processes' overall expression level in that window. This expression (i.e., up, down, normal, inactive — colored red, green, yellow and black respectively) is computed using the cluster centroids for each cluster in which the ontology term and its descendants appear. These charts facilitate summarization of the data, as users may choose to view the graphs for all terms or a selected subset of terms. Note that there is some information loss in this process, but the charts are intended to help make sense of the cluster graph. Allowing users to get an overall sense for how a process is regulated is helpful to that end. For example, in the case of the IDC (a chart depicting a small subset of its GO terms is shown in fig 3.), we see that "DNA replication initiation" is up-regulated in windows 3 and 4. This is consistent with our identification of those windows as the Trophozoite and Schizont stages, as replication was identified as a process active during these stages in [Bozdech et al. 2003].

4 Discussion

We had developed GOALIE (Gene-Ontology for Algorithmic Logic and Invariant Extraction), a systems biology application, with the aim of extracting global and dynamic perspectives (e.g., invariants) that could be inferred collectively over a temporal gene-expression dataset. Such perspectives are important in order to obtain a process-level understanding of the underlying cellular machinery; especially how cells respond to environmental cues. GOALIE uncovers formal temporal logic models of biological processes by redescribing time course microarray data into the vocabulary of biological processes and then piecing these redescriptions together into a Kripke structure. In such a model, possible worlds encode transcriptional states and are connected to future possible worlds by state transitions. An HKM (Hidden Kripke Model) constructed in this manner then supports various query, inference, and comparative assessment

tasks, besides providing descriptive process-level summaries. The formal basis for GOALIE is a multi-attribute information bottleneck (IB) formulation, where only the most relevant information is retained about states and their transitions while at the same time compressing the number of syntactic signatures used for representing the data.

Because its input data is purely syntactic, without any explicit signal about why a gene would respond coordinately with other genes and why it must do so at a particular instant after sensing an external event, it may appear surprising that a phenomenological model recovered by GOALIE would even possess any functional semantics. The ontologies, even though nonspecific, incomplete and rudimentary, are able to bestow a skeletal labeling to the possible worlds in the dynamic model and thus, focus our attention to the set of tasks that must be orchestrated precisely to perform a biological function. Because of this attractive feature, GOALIE is expected to be an ideal tool for additional annotation of other unknown genes and consequent expansion of our biological knowledge. Similarly, GOALIE could also seek to augment the underlying phenomenological model with causal rules and thus, shift from its focus on the proximate questions of "how" to ultimate questions of "why." [Friedman et al. 2000, Kleinberg and Mishra 2008]

We also suspect that what is true of the biological examples presented here may also hold for many other domains: e.g., financial domains with syntactic variables: prices and volumes of stocks, and information retrieval domains with syntactic variables: click streams or hyper-links. The GOALIE system is designed to be highly inter-operable in a domain-agnostic manner and will seek to extract meanings in many natural and artificial universes, such as these and others.

More narrowly, this paper demonstrated that using GOALIE, one is able to successfully recover the main structure of the IDC of Malaria parasite P. Falciparum in a completely automated manner. As highlighted earlier, GOALIE accomplished this feat with only prior knowledge of the underlying biology limited to ontological descriptions and without the use of frequency based methods. Even in the case of data that is not fully described by GO terms, it is shown that one is still able to discover its characteristic processes. Future work will include the examination of unannotated genes to determine novel functional characteristics, as well as a study of the causal relations between genes to facilitate richer descriptions of the underlying biology. GOALIE is currently available for Windows XP on request from the authors.

References

[Antoniotti et al. 2003] Antoniotti M et al (2003) Cell Biochem Biophys 38(3):271

[Ashburner et al. 2000] Ashburner M, Ball CA, Blake JA et al(2000) Nat Genet 25:25

[Bar-Joseph 2004] Bar-Joseph Z (2004) Bioinformatics 20(16):2493

[Bozdech et al. 2003] Bozdech Z, Llinas M, Pulliam BL et al (2003) PLoS Biology 1(1)

[Casagrande et al. 2007] Casagrande A et al (2007) In: Anai H, Horimoto K, Kutsia T (eds) AB'07: Algebraic Biology. 2nd International Conference on Algebraic Biology, Castle of Hagenberg, Austria, July 2007. Lecture notes in computer science, vol 4545. Springer, Berlin Heidelberg New York, p 51.

[Cilibrasi and Vitányi 2005] Cilibrasi R, Vitányi P (2005) IEEE Trans Inf Theory 51(4):1523

[Clark 1952] Clark W (1952) Pitman and Sons, London

[Clarke et al. 1999] Clarke EM, Grunberg O, Peled DA (1999) MIT Press

[Cover and Thomas 1991] Cover TM, Thomas JA (1991) Wiley-Interscience, New York

[Emerson 1997] Emerson EA (1997) DIMACS Series in Discrete Mathematics 31:185

[Ernst and Bar-Joseph 2006] Ernst J, Bar-Joseph Z (2006) BMC Bioinformatics 7(1):191

[Forum 1994] Forum, MPI (1994) Int J Supercomputer Applications 8(3/4):159

[Friedman et al. 2000] Friedman, N et al (2000) Computational Biology 7(3/4):601

[Kleinberg et al. 2006] Kleinberg S et al (2006) In: Minai A, Braha D, Bar-Yam Y (eds) Proceedings of ICCS'06. 6th International Conference on Complex Systems, Boston, MA, June 2006.

[Kleinberg et al. 2008] Kleinberg S et al (2008) CIMS Technical Report TR2007-907.

[Kleinberg and Mishra 2008] Kleinberg S, Mishra B (2008) Inferring Causation in Time Course Data with Temporal Logic (Submitted)

[Slonim et al. 2005] Slonim N, Atwal GS, Tkačik G et al (2005) Proc Natl Acad Sci U S A 102(51):18297

[Spellman et al. 1998] Spellman PT, Sherlock G, Zhang MQ et al (1998) Molecular Biology of the Cell 9:3273

[Wittgenstein 1934] Wittgenstein L (1934) s.n.

[Zeeberg et al. 2003] Zeeberg B, Feng W, Wang G et al (2003) Genome Biology 4:R28

[Zeeberg et al. 2005] Zeeberg BR, Qin H, Narasimhan S et al (2005) BMC Bioinformatics 6(1):168

5 Legends

Table 1 Correspondence of windows to IDC stages.

Figure 1. Summary of IDC as recovered by GOALIE. A more detailed graphic with annotations can be found at: http://bioinformatics.nyu.edu/Projects/GOALIE/malaria/index.sh

Figure 2. GOALIE's output of the HKM of *P.Falciparum* IDC as a graph of clusters.

Figure 3. Gantt chart view of selected GO terms. Each bar represents a window of time, with up-regulated terms labeled in red, down regulated terms in green and terms not enriching any cluster in the window labeled with black.

6 Figures/Tables

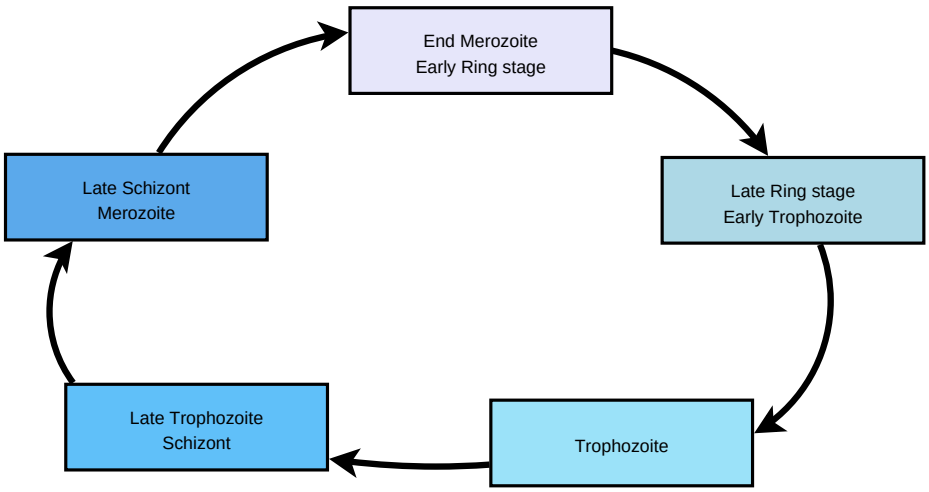

Figure 1:

Window	Time period(in hours)	Stage
1	1-7	End of Merozoite Invasion and Early Ring
2	7-16	Late Ring stage and Early Trophozoite
3	16-28	Trophozoite
4	28-43	Late Trophozoite and Schizont
5	43-48	Late Schizont and Merozoite

Table 1:

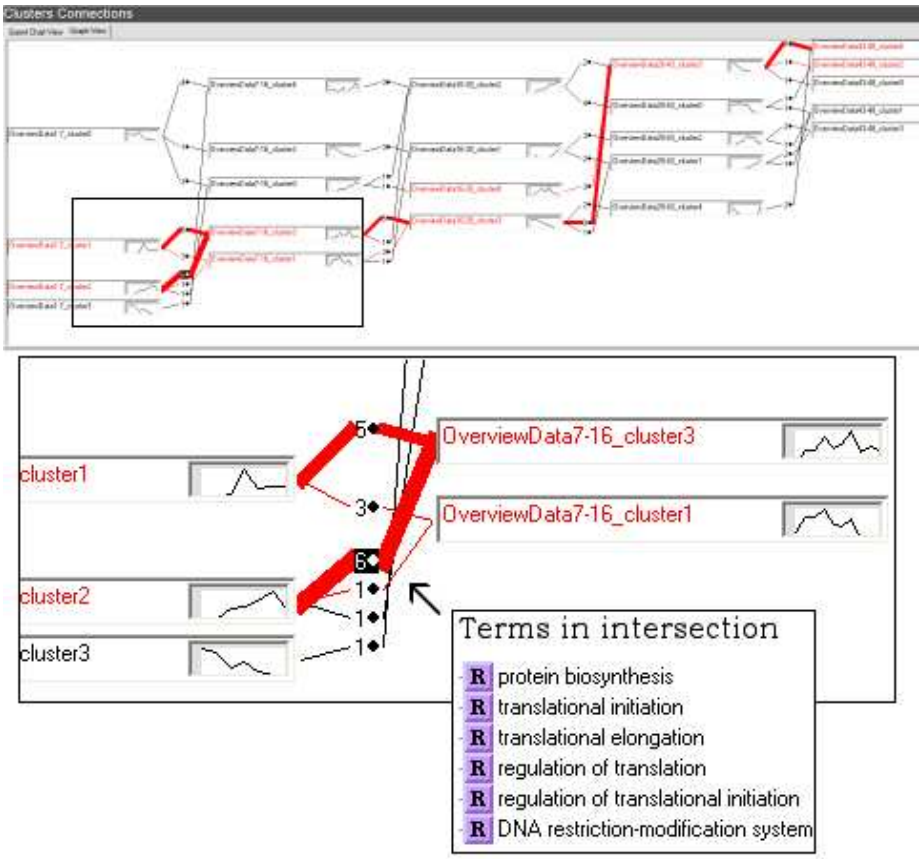

Figure 2:

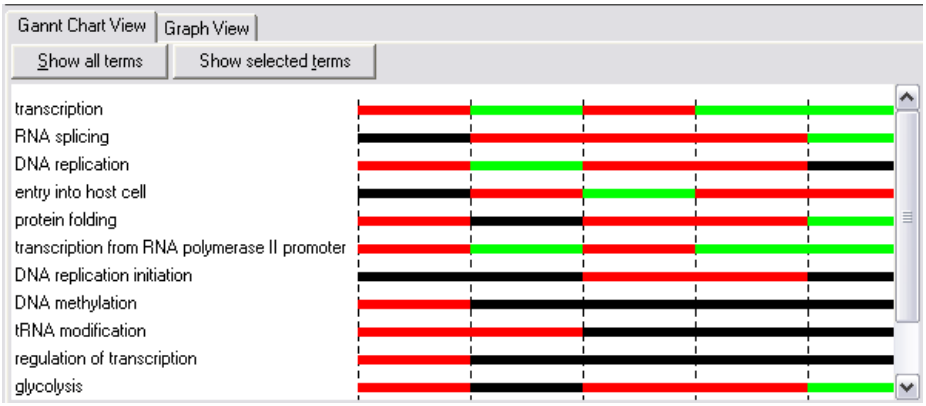

Figure 3: