Title: HyBrow - A Prototype System for Computer-Aided Hypothesis Evaluation

Authors:
Racunas S. A.[#], The Department of Electrical Engineering, Pennsylvania State University, University Park, PA 16802, USA

Shah N. H.[#][*], The Huck Institute of Life Sciences, Pennsylvania State University, University Park, PA 16802, USA

Albert I., The Huck Institute of Life Sciences, Pennsylvania State University, University Park, PA 16802, USA

Fedoroff N. V. The Huck Institute of Life Sciences and the Department of Biology, Pennsylvania State University, University Park, PA 16802, USA

Abstract

Experimental design, hypothesis-testing, and model-building in the current data-rich environment require the biologist to collect, evaluate and integrate large amounts of information of many disparate kinds. Developing a unified framework for the representation and conceptual integration of biological data and processes is a major challenge in bioinformatics because of the variety of available data and the different levels of detail at which biological processes can be considered. We have developed the HyBrow (Hypothesis Browser) system as a prototype bioinformatics tool for designing hypotheses and evaluating them for consistency with existing knowledge. HyBrow consists of a modeling framework with the ability to accommodate diverse biological information sources, an event-based ontology for representing biological processes at different levels of detail, a database to query information in the ontology, and programs to perform hypothesis design and evaluation. We demonstrate the HyBrow prototype using the galactose gene network in *Saccharomyces cerevisiae* as our test system, and evaluate alternative hypotheses for consistency with stored information.

[#] These two authors contributed equally to this work.

[*] Corresponding author: nigam@psu.edu
211 Wartik Labs,
University Park
PA – 16801 USA.
Ph: (814) 863 5720; Fax: (814) 863 1357

Availability: www.hybrow.org

Introduction

To expand understanding of a biological system, an experimentalist 1) formulates hypotheses about relationships that exist within that system, 2) gathers information from various repositories about the components of the system, 3) evaluates the hypotheses to assess whether they are supported or contradicted by this information, 4) revises hypotheses as needed, 5) perturbs the system in informative ways, and 6) integrates all available information to deepen understanding of how the system works. Understanding grows as hypotheses are accumulated.

Current investigations of signal transduction and gene regulation generate large volumes of data, making it increasingly difficult to assemble and organize all of the information needed to test hypotheses. To complicate matters, the various kinds of data reside in a wide array of repositories and are stored in different formats. To help biologists make effective use of increasing amounts of diverse data, we are developing the HyBrow (Hypothesis Browser) system to aid in the hypothesis formulation and evaluation cycle.

Most bioinformatics tools are designed to perform specific analytical functions. These tools carry out tasks such as identifying patterns, categorizing information, and probing data sources for similarities. However, information synthesis[1] has remained solely the purview of the biologist. Before designing a system to support the tasks of formulating and evaluating hypotheses for consistency with prior knowledge, we must address several issues. We must specify a representation for hypotheses that is both machine-understandable and accessible to the experimental biologist. We must choose a conceptual model and methods for storing existing information about the biological system in that model. Finally, we must decide on a framework that supports the evaluation of hypotheses with respect to the stored information.

Organisms are (and contain) complex systems which are incompletely understood. Such systems are more readily represented by describing the system's dynamics in terms of the events that occur in the system than by trying to write differential equations for all of the system's constituent reactions[2]. In a previous publication, we described the development of a framework for conceptualizing biological processes in terms of events[3]. *Biological events* are changes in a biological system for which we can obtain experimental evidence. In order to represent hypotheses about a biological process in a machine-understandable format, it is necessary to create a vocabulary of objects (agents) and processes, and define the relationships in which these entities can participate. We refer to this vocabulary as the hypothesis ontology and in our current work we construct and populate such an ontology for a simple test system. We describe biological events by naming the agents from the ontology (such as proteins and nucleic acids) and the processes (such as "binds") that connect them. We use the term *hypothesis event* to represent an abstract biological event. Thus, an hypothesis event consists of an acting agent (a "subject," such as a protein), a relationship (a "verb," such as induce, repress …), a target agent (an "object," a gene, protein…), the experimental and cellular contexts in which the event takes place, and a set of associated conditions (such as the presence or absence of other agents) which can accompany the event. This event-based framework[3], together with our hypothesis ontology, allows us to represent hypotheses in a *formal language* that specifies the time and context-dependent relationships among the system's objects and processes[3][4].

HyBrow's event-based framework includes methods to evaluate such formal language hypotheses for internal consistency and for agreement with existing knowledge[3]. Consistency of an hypothesis with observed data and prior knowledge is evaluated by applying *constraints* and *rules. Constraints* specify classes of forbidden events. *Rules* are the operations performed upon available information in order to enforce the constraints. Rules generate judgments of *support* or *conflict*, depending upon whether or not an assertion is supported by existing knowledge. The framework also includes *neighborhood functions* to establish similarity between hypotheses. These facilitate hypothesis revision through the automatic generation of "neighboring" hypotheses that are variants of an original hypothesis. Neighborhood functions use biologically acceptable notions to generate sets of variant events for events that conflict with existing data or prior knowledge. We examine these variants to find more fitting events and replace conflicted events with superior variants to produce hypotheses that better fit the stored information.

HyBrow's event-based framework makes it possible for the biologist to deal strictly with experimental evidence and to avoid the unintended assertions that are common artifacts of statistical and equation-based approaches. In HyBrow's framework, hypotheses and evaluation methods are directly compatible with the way information is conceptualized by biologists, making it easier to tap the expertise of experienced biologists. Finally, and most importantly, HyBrow's framework makes it possible to bring together many kinds of data and information in a unified formal language. The inability to combine information sources has been a stumbling block for computational models of biological systems, leading current information integration efforts to focus on only one or two categories of information[5, 6].

In this paper, we demonstrate HyBrow's information synthesis capability using the galactose metabolic and regulatory network, which we chose because abundant data and information of many different types are publicly available for this system[7]. We designed a small hypothesis ontology appropriate for the GAL system. We specified the formal grammar that describes how to combine terms from the ontology into hypotheses. We designed a database to store yeast GAL data structured in the ontology and developed hypothesis composition, visualization and evaluation software.

To test the HyBrow prototype, we evaluated and ranked hypotheses about the GAL system. During evaluation, HyBrow assayed all stored data for conflicting or supporting evidence for each statement in each hypothesis. HyBrow modified hypotheses which contained errors to generate variants with fewer flaws. Finally, HyBrow combined the resulting determinations of conflict and support to generate evaluations and rankings for all of the original and variant hypotheses.

Implementation

Hypothesis ontology. Common ontologies for biological objects and processes[8, 9] are being developed to support the intercommunication of diverse databases as well as enable automated annotation and extraction of information from the literature[10, 11]. Ontologies also provide a foundation for the construction of higher level models of biological systems[12, 13]. Models vary from abstract Boolean[14] and Bayesian networks[5] to highly specific[15] and quantitative models[16]. Currently, most databases do not store information in an explicit ontology that facilitates modeling, and groups that design ontologies[12, 13] do not store all relevant data structured in those ontologies. Hence, efforts aimed at integrating diverse information sources need to choose or design a representation scheme and convert existing data to that representation. Although specialized ontologies exist[12, 17], there is a need for an ontology that allows users to represent biological processes in an event-based manner. Such an ontology should be compatible with existing ones so that hierarchical relationships can be made between terms in existing ontologies [12, 17] and this "hypothesis ontology."

We used Protégé[18] to design a small hypothesis ontology for representing GAL system information in HyBrow's event-based conceptual framework[3]. For guidance, we relied upon the principles used to design the Rzhetsky and the Bioprocess ontologies[12, 13]. Our ontology (Fig.1) accommodates currently available literature data, extracted primarily from YPD[19] at a coarse level of resolution. An event consists of an acting agent (the "subject," such as gene, RNA, protein), a target agent (the "object," such as a gene, protein, complex), a relationship (the "verb," such as induce, repress, bind), a context in which the event takes place, and an optional set of associated conditions (such as the presence or absence of other agents) which accompany the event. The construction of events from elements of the ontology, event sets from events, and hypotheses from event sets is governed by a context-free grammar. Events that occur in the same context are combined to form *event sets* and an *hypothesis* consists of event sets linked by logical and temporal operators. An hypothesis must contain at least one event set, which must contain at least one event. Please refer to www.hybrow.org for a formal specification of this grammar.

Contexts specify *where* events occur in the cell and *under what genetic conditions* they occur. Our *contexts* are derived from established ontologies. For example, terms for specifying physical locations in the cell come from the cellular component division of the Gene Ontology. We currently support genes, proteins, mRNA, small molecules, and complexes of proteins, small molecules, and mRNA as agents in our prototype. We define three main categories of relationships: logical (e.g. induce), biochemical (e.g. phosphorylate) and physical (e.g. bind). The key design principle is that the ontology describes a regulatory system in an event-based way consistent with our evaluation framework. Our current hypothesis ontology allows representation of events such as: 'Gal4p binds to the promoter of the gal1 gene in the presence of galactose in wild type *S. cerevisiae*'. Depending on the resolution of the ontology, this approach can represent anything from simple protein phosphorylation to the entire cell cycle[20] . Formal presentation of the complete ontology is available at www.hybrow.org

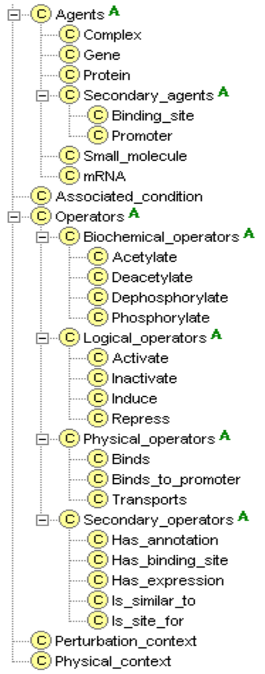


Fig.1 An overview of the ontology used to represent data in an event-centered way. 'Operators' are the relationships that can exist between agents.

Inference rules and constraints. We have defined constraints and the rules that determine whether or not a constraint is satisfied for each relationship expressible with terms from our ontology. We define several categories of constraints. *Ontology constraints* determine what agents can participate in which types of biological relationships. For example, a gene cannot transport a gene, but a protein can transport a small molecule or another protein. *Data constraints* determine what data values are valid for a particular relationship. For example, for the relationship 'protein A binds to the promoter of *gene B'*, it is acceptable for protein A to be annotated as localized in the nucleus or cytoplasm but not on the cell membrane. *Existence constraints* require an agent's presence before it can enter a relationship. For example, a protein cannot perform its function when its gene has been deleted. *Temporal constraints* govern the transmission of modifications made to an agent by previous events. For example, event 'X phosphorylates Y' implies that in all subsequent events Y is phosphorylated (unless a dephosphorylation event occurs).

Each rule has divisions that correspond to the different constraints that exist in HyBrow. The first division deals with ontology constraints, the second, with constraints on annotation data in Gene Ontology (GO)[9] format, the third deals with literature-extracted information structured in the ontology, and the fourth with constraints on the specific data type(s) for a relationship, such as promoter sequence in the case of the *Binds to promoter* relationship. For each constraint that is violated in any division, the event is assigned a 'conflict.' For each constraint that is satisfied, the event is assigned a 'support'. If a constraint is neither violated nor supported, a 'cannot comment' is assigned. Divisions 1, 2 and 3 can be generalized because they have a common structure for different relationships, and the operations to be performed on the data are very similar. Division 4 is very specific because of the different ways in which different data types for each relationship must be used. There are additional general divisions that enforce existence and temporal constraints. For example, the rule for protein A *binds to promoter* of gene B has the following divisions: 1) check if A is a protein or a protein-complex and if B is a gene; 2) check whether protein A is annotated a) to have the molecular function of a transcriptional activator or repressor, b) to be involved in the biological process of transcriptional regulation and c) to have a nuclear localization; 3) determine whether the literature reports the postulated event; 4) search the promoter of gene B for a binding site for protein A; 5) ensure that the event is not postulated in a genetic context where the gene for protein A is knocked out.

Rules are coded in Perl as hierarchical function libraries to keep the rule set extensible and flexible. Most of the constraints enforced by the generalized divisions are stored in database tables which are queried at run time, allowing flexibility for changing the stringency of the constraints. A more detailed description of the rule library is provided at www.hybrow.org

Database and information gathering. At the heart of HyBrow is the idea that disparate kinds of information can be represented in a unified formal language. Biological information residing in the published literature and electronic databases is expanding at an accelerating rate. Retrieving information and translating it into our ontology presents several problems because the information is in different repositories and in different storage formats. Further, only a fraction of the published literature is available electronically. The problem of automating extraction of information from the literature is being addressed by a number of research groups, but is far from solved[12, 21, 22]. The most promising approach appears to be MedScan[11], which can parse literature abstracts to identify 'biological events'. But information extraction is still largely manual, practiced by annotators who read papers for relevant concepts and information.

In this work, we adopted different approaches to gather and structure data in our ontology. For data with standardized representation formats, we designed user agents to access the existing public repositories and retrieve desired information. For example, we designed a user-agent to retrieve promoter sequences from the *S. cerevisiae* Promoter Database[23]. In most cases, we were able to access well-annotated information from the Saccharomyces Genome Database[24] directly. We used YPD[19] to access curated literature information about *S. cerevisiae* genes and proteins. We designed a form-based layout for gathering biological information from YPD reports and filled in predefined table fields compatible with the ontology from specific fields of the YPD report. This process is easily automated if direct access to database tables is obtained and can be extended to frame-based 'loading forms' like the EcoCyc database[25]. For quantitative data, such as that from microarray expression profiling experiments, we converted the data to our table format using custom Perl scripts. If microarray data are structured in the MAGE object model,[26] this task is more straight-forward.

We designed a MySQL database and mapped our ontology onto the database for easy extension as our ontology evolved. We created a table in the database for each class in the ontology, at the finest level of resolution. The table has fields for properties, called 'slots' and ‘facets’ in the ontology, of the relevant class. This creates more tables than would be present in a well-normalized relational database. However, a prototyping effort requires the backend to be easily modified in response to changes in the ontology. The backend also contains tables to store constraints used during evaluation. A detailed database schema is found at www.hybrow.org

User interfaces. An important feature of HyBrow is that it is easy for the user to construct a machine-readable formal language hypothesis. We have created two interfaces for this purpose: a visual interface (Fig.2 left panel) and a widget interface (Fig. 2 right panel). Our visual interface allows users to construct hypotheses using a visual notation constructed in accordance with proposed conventions[27]. This interface allows users to draw diagrams which are then automatically translated into hypotheses. The widget interface allows the user to write hypotheses in English-like “sentences” constructed using subject/verb/object selection menus. A user can construct portions of an hypothesis using different interfaces and then combine them. Details on how to use the tools are provided at www.hybrow.org. Hypotheses are saved to local files and then submitted for evaluation via the web.

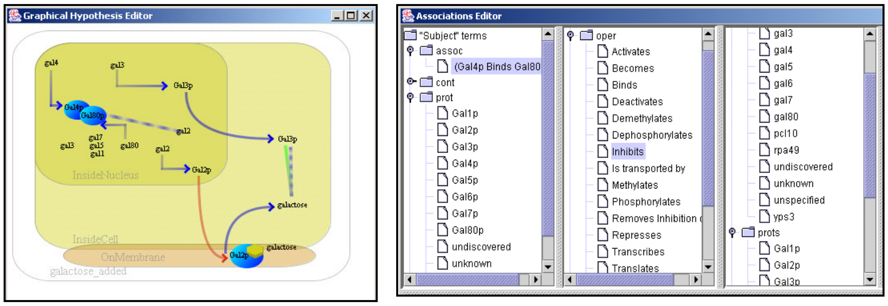
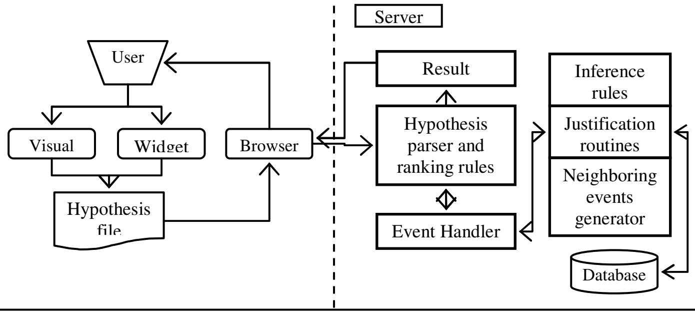


Fig. 2 Screen shots of the visual and widget interface used to construct hypotheses.

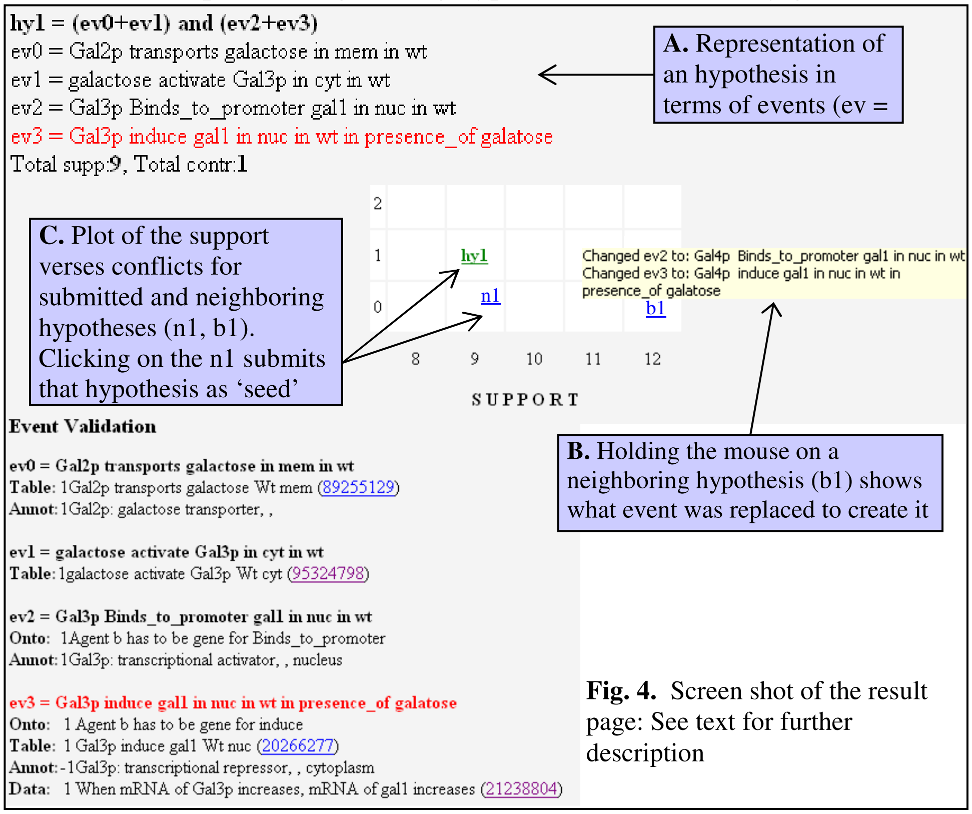
The hypothesis evaluation process. The hypothesis evaluation process is illustrated diagrammatically in Fig. 3. When HyBrow receives an hypothesis, it checks the connections between events and event sets for conformity with the hypothesis grammar. If the hypothesis passes these tests for syntax, each event is then checked for validity using the appropriate rule for the relationship proposed in the event. For each event, a *support*, *conflict* or *cannot comment* result corresponding to each of the four divisions of the inference rules is returned. Finally, the support and conflict calls are tallied based upon the logical structure of the hypothesis. Each ‘and’ between event sets leads to the inclusion of results from both sets in the final tally. For each ‘or’ connection, the ‘better’ set is chosen using a hierarchical set of rules. [Sample rules: *1)* an event set with conflicts it is better than an event set with more conflicts and worse than one with fewer conflicts. *2)* An event set for which all events have at least some support is better than an event set for which at least one event is not supported. *3)* If one event set's support is a strict superset of another event set's support, the superset is superior., …] We apply these rules sequentially until one of the rules returns a clear decision.

Fig.3 The visual or the widget interface is used to design hypotheses, which are sent to the server via a browser. The hypothesis parser is the entry point for the system and it uses the event handler, which manages the event library, ranking, justification and event neighborhood generation. The database stores the different data structured into ‘events’.



For each event, the hypothesis evaluation process finds all conflicts with existing knowledge and indexes them, along with their sources. These are reported to the user to allow the user to identify specific problems with the hypothesis and the conflicting data source. For each event that has a conflict, a set of variant events is generated using biologically motivated heuristics, such as replacing the acting agent with agents that share a sequence similarity or share a similar cellular localization with the original agent. Neighboring hypotheses that share the logical structure of the original are generated by replacing conflicting events with the best variant events. These neighboring hypotheses are then evaluated, and if a better (more supported, less conflicted) hypothesis is found, it is presented to the user.

After evaluation, the user is shown 1) the support and conflict totals, 2) the least conflicted, most supported event sets that fit the logical structure of the hypothesis, 3) a support-conflict scatter plot of neighboring hypotheses automatically generated from the user submitted hypothesis, and 4) a list of all events that had conflicts, the data that triggered the conflicts, an explanation of *why* the rules interpret that data as a conflict, and a reference to the original article or data source. The results pages (Fig. 4) allow a user to gauge the 'fitness' of his/her hypothesis in the light of all stored data. Iterative refinement of the hypothesis allows the user to reconcile all existing data into a single coherent representation whose level of detail depends on the resolution of the ontology used for constructing hypotheses.



hy1 = (ev0+ev1) and (ev2+ev3)
ev0 = Gal2p transports galactose in mem in wt
ev1 = galactose activate Gal3p in cyt in wt
ev2 = Gal3p Binds_to_promoter gal1 in nuc in wt
ev3 = Gal3p induce gal1 in nuc in wt in presence_of galatose
Total supp:**9**, Total contr:**1**

Changed ev2 to: Gal4p Binds_to_promoter gal1 in nuc in wt
Changed ev3 to: Gal4p induce gal1 in nuc in wt in presence_of galatose

Event Validation

ev0 = Gal2p transports galactose in mem in wt
Table: 1Gal2p transports galactose Wt mem (89255129)
Annot: 1Gal2p: galactose transporter, ,

ev1 = galactose activate Gal3p in cyt in wt
Table: 1galactose activate Gal3p Wt cyt (95324798)

ev2 = Gal3p Binds_to_promoter gall in nuc in wt
Onto: 1Agent b has to be gene for Binds_to_promoter
Annot: 1Gal3p: transcriptional activator, , nucleus

ev3 = Gal3p induce gall in nuc in wt in presence_of galatose
Onto: 1 Agent b has to be gene for induce
Table: 1 Gal3p induce gal1 Wt nuc (20266277)
Annot: -1Gal3p: transcriptional repressor, , cytoplasm
Data: 1 When mRNA of Gal3p increases, mRNA of gal1 increases (21238804)

A. Representation of an hypothesis in terms of events (ev =

C. Plot of the support verses conflicts for submitted and neighboring hypotheses (n1, b1). Clicking on the n1 submits that hypothesis as 'seed'

B. Holding the mouse on a neighboring hypothesis (b1) shows what event was replaced to create it

Fig. 4. Screen shot of the result page: See text for further description

Test runs with sample hypotheses. In order to test the prototype, we composed hypotheses about the GAL system and ranked them. The GAL system consists of genes that transport and metabolize galactose and the regulatory network that controls whether the pathway is on or off[28]. The process involves three types of proteins: a) a permease (Gal2p) that transports galactose (encoded by gal2 gene). b) proteins that utilize intracellular galactose; galactokinase (encoded by gal1), uridylyltransferase (encoded by gal7), epimerase (encoded by gal10), and phosphoglucomutase (encoded by gal5); and c) the regulatory proteins Gal3p, Gal4p and Gal80p, which exert transcriptional control over the genes encoding the transporter, the enzymes, and to some extent, their own genes[7]. HyBrow successfully identified the hypothesis that best explained the current understanding about GAL system regulation[7, 28]. For 6 of the 7 events that had conflicts, HyBrow was also able to successfully suggest corrections that increased agreement with stored information. All hypotheses used and explanations of their evaluations can be found at www.hybrow.org.

Here we describe the evaluation of a simple illustrative hypothesis: "Gal2p transports galactose into the cell at the cell membrane. In the cytoplasm, galactose activates Gal3p. Gal3p binds to the promoter of gal1 gene and induces its transcription in the presence of galactose". This hypothesis was decomposed into events as shown in Fig.4A. On evaluation, HyBrow reported support from literature and GO annotation for event number 0 (ev0), support from literature for ev1, support from ontology constraints and annotation for ev2 and support from the ontology, literature and data divisions for ev3. It reported a conflict for ev3 (which is marked in red) from the annotation rule division because Gal3p is annotated to be primarily in the cytoplasm[24]. HyBrow then searched for variant events. For ev3 it found an event (Gal4p binds to promoter of gal1) with higher support and for ev4 it found the more meaningful event (Gal4p induces gal1 in nucleus in wt in presence of galactose) with the same support but no conflict. These events were inserted in place of the original events to create a neighboring hypothesis that is better than the original hypothesis (Fig.4B, C).

When a submitted event contains a perturbation, such as the deletion of a gene, HyBrow identifies the agents disabled because of the perturbation and *infers* a conflict with events that depend on those agents. For example, if the submitted event is: "Gal3p induce gal1 in nucleus in gal3-K/O" HyBrow reports a conflict. (The event "Gal3p not induces gal1 in nucleus in gal3-K/O" gets support). Some of the inferences suggested by HyBrow are obvious for the small GAL system, but HyBrow's ability to automate the process offers a substantial advantage for systems containing large numbers of genes and proteins.

Discussion

HyBrow supports the construction, "proofreading," and evaluation of hypotheses expressed in familiar diagram or intuitive text-based formats to aid in synthesizing data into working models. HyBrow's methodology is evaluation-based. Thus, unlike systems that construct statistical or equation-based models, HyBrow is able to provide explicit reasons (and references) for its output. However, HyBrow does not force the user to accept all of the data, nor does it judge the validity of stored information. Rather, it gives the user links to the exact source of each conflict, leaving it up the user to judge the relative merits of information sources. The user can choose to ignore conflicts from data sources deemed unreliable

HyBrow differs fundamentally from existing efforts such as EcoCyc, modeling biological processes as "workflows" and Genome Knowledgebase (GKB). EcoCyc is designed using an explicit ontology for biological function and facilitates functional querying. However, it lacks the notion of an *hypothesis* or a formal framework to evaluate and rank alternative statements about a biological process[29]. Modeling of biological processes as "workflows" by Altman's group includes some of HyBrow's features, but the underlying conceptual model (which uses hybrid Petri nets) does not support hypothesis neighborhoods and the analysis of models has to be done manually[13]. GKB is an effort to structure biological knowledge in an event-centered data model[20]. It is not a modeling framework by itself, but serves as a public source of structured data which efforts like ours can use.

In our test runs, HyBrow identified the least conflicted hypothesis accurately and suggested valid 'corrections' for events with conflicts. We believe that we can build upon this success and plan to extend and strengthen HyBrow in several ways. Currently, improvements to hypotheses are suggested using neighboring events generated using simple heuristics, while our conceptual framework supports neighborhood functions that create similar event sets from a given event set[3]. Extending HyBrow to use neighborhoods of event sets as well as of events requires new evaluation routines to track all of the biochemical and other modifications that an event set generates and to ensure that the neighboring event sets satisfy them. In future work, we will explore biological notions of similarity between event sets and modify our neighborhood functions and evaluation routines accordingly. The current rule library contains rules for 'extrapolations' in the presence of perturbations such as gene knock-outs and constitutive over-expression. In the future, we will include extrapolations for more categories of perturbations. Our current implementation can only propagate temporal constraints about the presence or absence of biochemical modifications. We intend to propagate constraints about activation/inhibition and induction/repression in an attempt to model *how* an event affects down-stream agents. To that end, we will extend the current ontology to include 'modification state' and 'activation state' descriptors for agents; events will then be able to modify these state descriptors. Finally, HyBrow can identify events that are frequently specified, but for which evaluation was not possible. Identifying such events will allow HyBrow to aid experiment design. For instance, if many users include an event in their hypotheses and there is no experimental evidence for it, HyBrow can indicate a need to obtain such data.

Conclusion and Summary

Our implementation of HyBrow for the GAL system demonstrates that ontology driven, event-based modeling of biological processes is feasible and that structuring data in HyBrow's event-based framework facilitates computer-aided hypothesis evaluation. HyBrow can accommodate both more data and more types of data as they become available. Moreover, its constraints can be elaborated as understanding about the biological system grows. We believe that the approach we have developed for this HyBrow prototype can significantly inform experimentation by integrating large amounts of information for the evaluation of hypotheses.

Bibliography

1. Kuchinsky, A., et al. *Biological Storytelling: a software tool for biological information organization based upon narrative structure.* in *Advanced Visual Interfaces*. 2002. Trento, Italy.
2. Ho, Y.C., *Special Issue on Discrete Event Dynamical Systems: Editorial.* Proc IEEE, 1989. **77**(1): p. 24-38.
3. Racunas, S.A., et al. *A Contradiction-Based Framework for Testing Gene Regulation Hypotheses*. in *IEEE Bioinformatics*. 2003. Stanford University, Palo Alto, California: IEEE Computer Society.
4. Sudkamp, T.A., *Languages and machines*. 1988, Reading: Addison-Wesley.
5. Hartemink, A.J., et al., *Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks.* Pac Symp Biocomput, 2001: p. 422-33.
6. Segal, E., et al., *Discovering molecular pathways from protein interaction and gene expression data.* Bioinformatics, 2003. **19**(90001): p. 264i-272.
7. Ideker, T., et al., *Integrated genomic and proteomic analyses of a systematically perturbed metabolic network.* Science, 2001. **292**(5518): p. 929-34.

8. Schulze-Kremer, S., *Ontologies for molecular biology.* Pac Symp Biocomput, 1998: p. 695-706.
9. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.* Nat Genet, 2000. **25**(1): p. 25-9.
10. Fleischmann, W., et al., *A novel method for automatic functional annotation of proteins.* Bioinformatics, 1999. **15**(3): p. 228-33.
11. Novichkova, S., et al., *MedScan, a natural language processing engine for MEDLINE abstracts.* Bioinformatics, 2003. **19**(13): p. 1699-1706.
12. Rzhetsky, A., et al., *A knowledge model for analysis and simulation of regulatory networks.* Bioinformatics, 2000. **16**(12): p. 1120-8.
13. Peleg, M., et al., *Modelling biological processes using workflow and Petri Net models.* Bioinformatics, 2002. **18**(6): p. 825-37.
14. Akutsu, T., et al., *Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function.* J Comput Biol, 2000. **7**(3-4): p. 331-43.
15. McAdams, H.H., et al., *Simulation of prokaryotic genetic circuits.* Annu Rev Biophys Biomol Struct, 1998. **27**: p. 199-224.
16. Sveiczer, A., et al., *Modeling the fission yeast cell cycle: quantized cycle times in wee1-cdc25Delta mutant cells.* Proc. Natl. Acad. Sci. USA, 2000. **97**(14): p. 7865-7870.
17. Karp, P.D., *An ontology for biological function based on molecular interactions.* Bioinformatics, 2000. **16**(3): p. 269-85.
18. Crubézy, M., et al., *Protege 2000*. 2003, Stanford University: Palo Alto.
19. Proteome, *Yeast Proteome Database*. 2001, Proteome.
20. *Genome Knowledge Base*. 2003, Cold Spring Harbor Laboratory, The European Bioinformatics Institute, and The Gene Ontology Consortium.
21. Andrade, M.A., et al., *Automated extraction of information in molecular biology.* FEBS Lett, 2000. **476**(1-2): p. 12-7.
22. IBM, *Discovery Link*. 2002, IBM Corporation.
23. Zhu, J., et al., *SCPD: a promoter database of the yeast Saccharomyces cerevisiae.* Bioinformatics, 1999. **15**(7-8): p. 607-11.
24. Cherry, J.M., et al., *SGD: Saccharomyces Genome Database.* Nucleic Acids Res, 1998. **26**(1): p. 73-9.
25. Karp, P.D., et al., *Ecocyc the resource and the lessons learned*, in *Bioinformatics Databases and Systems*, S. Letovsky, Editor. 1999, Kluwer Academic Publishers: New York. p. 47-62.
26. Spellman, P.T., et al., *Design and implementation of microarray gene expression markup language (MAGE-ML).* Genome Biol, 2002. **3**(9): p. RESEARCH0046.
27. Cook, D.L., et al., *A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems.* Genome Biol, 2001. **2**(4).
28. Lohr, D., et al., *Transcriptional regulation in the yeast GAL gene family: a complex genetic network.PG - 777-87.* Faseb J, 1995. **9**(9).
29. Karp, P.D., et al., *The EcoCyc Database.* Nucleic Acids Res, 2002. **30**(1): p. 56-8.