MONOD, a Collaborative Tool for Manipulating Biological Knowledge

Running Head: MONOD: Biological Knowledge Management System

David Soergel*[1,2], Kirindi Choi[1], Ty Thomson[3], Jay Doane[1], Brian George[1], Ross Morgan-Linial[1], Roger Brent[1], and Drew Endy[3]

1) The Molecular Sciences Institute, 2168 Shattuck Avenue, Berkeley, California 94704, 2) Biophysics Graduate Group, University of California, Berkeley, and 3) Division of Biological Engineering, MIT 68-580, 77 Mass. Ave., Cambridge, Massachusetts 02139 *) to whom correspondence should be addressed. Email: soergel@berkeley.edu

Summary

We describe an open source software tool called MONOD, for Modeler's Notebook and Datastore, designed to capture and communicate knowledge generated during the process of building models of many-component biological systems. We used MONOD to construct a model of the pheromone response signaling pathway of *Saccharomyces cerevisiae*. MONOD allowed the accumulation, documentation, and exchange of data, valuations, assumptions, and decisions generated during the model building process. MONOD thus helped preserve a record of the steps taken on the path between from the experimental data to the computable model. We believe that MONOD and its successors may streamline the processes of building models, communicating with other researchers, and managing and manipulating biological knowledge. "Collaborative annotation"-- fine-grained, structured, searchable communication enabled by software tools of this type-- could positively affect the practice of biological research.

Introduction

A great deal is known about the components and functions of intracellular biological systems [1,2]. One tool for understanding multi-component systems is to construct, simulate and analyze computable representations, or models, of them [3]. The task of constructing computable models of biological systems would be aided by a greater ability to capture and document the assumptions and decisions that arise during model construction, and by enabling groups of scientists to participate in and review the process of model building and scientific discovery.

To better understand the current process of building models of natural biological systems, first consider how experimental scientists write papers on their work. They use their judgment, a combination of explicit reasoning, tacit knowledge [4], and intuition, to selectively include some experimental results, while excluding others. Scientists shape the included results into an explanatory narrative that, in many cases, incorporates an explicit rationalization for how the system they describe came to be the way it is [5]. In the biological literature, such narratives are typically expressed in natural language, and make use of accompanying diagrams or cartoons, figures and tables of experimental data that support key points, and descriptions of the methods used to generate and analyze the experimental data. These narratives typically do not describe the criteria used to include data and, by definition, do not incorporate excluded data.

Next, consider that modelers of biological systems use their judgment to select facts from these narratives to include in their models. Modelers interpret and synthesize experimental results, make assumptions, resolve ambiguities and contradictions, and iteratively test and refine their models to match experimental data as closely as possible. They typically represent facts and interpretations using some mathematical formalism suitable for simulation, for example, sets of elementary rate equations. Model-building produces its own narratives, in the sense that, in building the model, the modeler develops chains of reasoning supporting a mental representation of the system. As with narratives of experimental results, the resulting models typically do not include descriptions of the criteria used to include data, nor, again, by definition, do they incorporate excluded data.

Modeling narratives are thus frequently both private and implicit. For example, a reaction rate used in a particular model might come from a modeler's best estimate, based on measured values of related reactions. In many cases, the reasoning and judgment that the modeler used to arrive at such an estimate might in the long run be more valuable than the model itself. But as long as it is recorded only in a personal lab notebook or computer file, the work that went into the estimate cannot be retrieved, examined, or refuted.

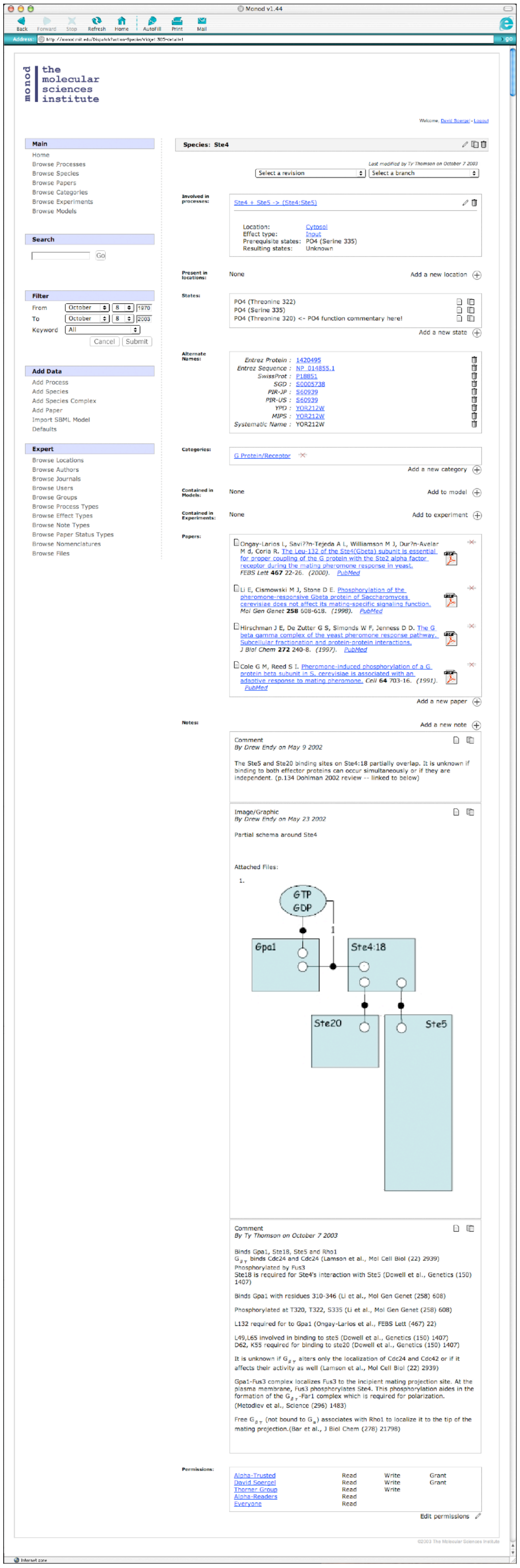
This inability to examine intermediate assumptions makes it difficult to evaluate and compare models to one another, or to the experimental data on which they are based. It also makes it difficult to alter or reuse models without inadvertently violating some of the assumptions, or to update them consistently in the face of new experimental results – a difficulty that increases with the number of components included in the model. Further, this current process of model-building in isolation does not facilitate a seemingly desirable outcome, the ability of investigators to work with preexisting models in a common framework, searching, simulating, linking, and merging them as appropriate [6].

Here, we sought to bridge the gap between the biological literature and computable models, and to facilitate the documentation and exchange of the expert knowledge used to construct models. We did this by developing a software tool, MONOD, that combines the structure of a limited formal language with the *ad hoc* flexibility of natural language narratives. This hybrid approach to knowledge representation may be particularly appropriate for rapid communication of ongoing work within communities of scientific researchers.

Description of MONOD and how users interact with it

MONOD stands for "Modeler's Notebook and Datastore". The name deliberately evokes the memory of Jacques Monod, a pioneering molecular biologist whose abstraction of the relevant details from his work with bacterial lactose metabolism enabled him and François Jacob to devise the "operon model", which correctly describes how many bacterial genes are regulated [7]. MONOD (in the current version, 1.5) is an interactive web application: it runs on a server, and users interact with it through a standard web browser (Figure 1). Data are stored in a conventional relational database. MONOD makes use of numerous open source software products to map data to object-oriented data structures for the programmer and to present it to the user through the web interface. The scope of an individual MONOD installation can vary from one that serves a single user to one that serves a research community.

The first implementation and populated database focuses on the signal transduction pathway that governs the response of budding yeast (*Saccharomyces cerevisiae*) to mating pheromone. The early steps of this pathway are effected by biochemical reactions among a small number of proteins. The database



Species: Ste4

Last modified by Ty Thomson on October 7 2003

Involved in processes: Ste4 + Ste5 -> (Ste4:Ste5)

Location: Cytosol
Effect type: Input
Prerequisite states: PO4 (Serine 335)
Resulting states: Unknown

Present in locations: None

States:

PO4 (Threonine 322)
PO4 (Serine 335)
PO4 (Threonine 320) <- PO4 function commentary here!

Alternate Names:

Entrez Protein :	1420495
Entrez Sequence :	NP_014855.1
SwissProt :	P18851
SGD :	S0005738
PIR-JP :	S60939
PIR-US :	S60939
YPD :	YOR212W
MIPS :	YOR212W
Systematic Name :	YOR212W

Categories: G Protein/Receptor

Contained in Models: None

Contained in Experiments: None

Papers:

Ongay-Larios L, Savi??n-Tejeda A L, Williamson M J, Dur?n-Avelar M d, Coria R. The Leu-132 of the Ste4(Gbeta) subunit is essential for proper coupling of the G protein with the Ste2 alpha factor receptor during the mating pheromone response in yeast. *FEBS Lett* **467** 22-26. *(2000).* *PubMed*

Li E, Cismowski M J, Stone D E. Phosphorylation of the pheromone-responsive Gbeta protein of Saccharomyces cerevisiae does not affect its mating-specific signaling function. *Mol Gen Genet* **258** 608-618. *(1998).* *PubMed*

Hirschman J E, De Zutter G S, Simonds W F, Jenness D D. The G beta gamma complex of the yeast pheromone response pathway. Subcellular fractionation and protein-protein interactions. *J Biol Chem* **272** 240-8. *(1997).* *PubMed*

Cole G M, Reed S I. Pheromone-induced phosphorylation of a G protein beta subunit in S. cerevisiae is associated with an adaptive response to mating pheromone. *Cell* **64** 703-16. *(1991).* *PubMed*

Notes:

Comment
By Drew Endy on May 9 2002

The Ste5 and Ste20 binding sites on Ste4:18 partially overlap. It is unknown if binding to both effector proteins can occur simultaneously or if they are independent. (p.134 Dohlman 2002 review -- linked to below)

Image/Graphic
By Drew Endy on May 23 2002

Partial schema around Ste4

Attached Files:

1.

Comment
By Ty Thomson on October 7 2003

Binds Gpa1, Ste18, Ste5 and Rho1
$G_{\beta\gamma}$ binds Cdc24 and Cdc24 (Lamson et al., Mol Cell Biol (22) 2939)
Phosphorylated by Fus3
Ste18 is required for Ste4's interaction with Ste5 (Dowell et al., Genetics (150) 1407)

Binds Gpa1 with residues 310-346 (Li et al., Mol Gen Genet (258) 608)

Phosphorylated at T320, T322, S335 (Li et al., Mol Gen Genet (258) 608)

L132 required for to Gpa1 (Ongay-Larios et al., FEBS Lett (467) 22)

L49,L65 involved in binding to ste5 (Dowell et al., Genetics (150) 1407)
D62, K55 required for binding to ste20 (Dowell et al., Genetics (150) 1407)

It is unknown if $G_{\beta\gamma}$ alters only the localization of Cdc24 and Cdc42 or if it affects their activity as well (Lamson et al., Mol Cell Biol (22) 2939)

Gpa1-Fus3 complex localizes Fus3 to the incipient mating projection site. At the plasma membrane, Fus3 phosphorylates Ste4. This phosphorylation aides in the formation of the $G_{\beta\gamma}$-Far1 complex which is required for polarization. (Metodiev et al., Science (296) 1483)

Free $G_{\beta\gamma}$ (not bound to G_{α}) associates with Rho1 to localize it to the tip of the mating projection.(Bar et al., J Biol Chem (278) 21798)

Permissions:

Alpha-Trusted	Read	Write	Grant
David Soergel	Read	Write	Grant
Thorner Group	Read	Write	
Alpha-Readers	Read		
Everyone	Read		

©2003 The Molecular Sciences Institute

Figure 1. A sample detail page from the MONOD web interface showing information that has been entered about the protein Ste4, including reactions in which it participates, states of posttranslational modification, links to external databases, keywords, citations, text and image annotations, and access permissions. Users can add to or edit the information on this page through web forms, for instance by clicking on the button labelled “add a new note”.

includes information about molecular species and interactions between them (such as binding, dissociation, and posttranslational modification). This structure closely matches the natural language descriptions used by biologists to describe intracellular signal transduction pathways. Data representation is "fine grained": particles of entered information can be quite small, but multiple pieces of data are easily linked together.

Features of MONOD relevant to an individual user of the program

MONOD allows a user to define a set of molecular species and associated reactions. A species can be a monomeric molecule, for example "protein A", or a complex of molecules, for example "protein A bound to protein B". The interface provides forms for describing states of post-translational modification such as phosphorylation, and allows reactions to depend on such states and to change them.

The species detail page displays links to the reactions in which the species participates (whether as an input, output, or catalyst); keywords with which the species is associated (e.g. kinase cascade or gene expression); links to models, experiments, and journal papers that include the species; any described post-translational modification states; species abundance and subcellular localization; and user notes or commentary. Alternate names may be recorded for each species and associated with specific systems of nomenclature. For proteins in *S. cerevisiae*, links to detail pages in external databases such as the Saccharomyces Genome Database (SGD) [8], Entrez [9,10], SwissProt [11], PIR [12], YPD [13], and MIPS [14] are automatically established and displayed.

MONOD uses a general representation for reactions and other processes. A *process* consists of a set of *effects*, where each effect describes the participation of one molecule of a species in the process and the role it plays (i.e., input, output, or catalyst), taking into account the modification state of the molecule and its location within the cell. This structure provides a consistent framework to represent different kinds of intracellular processes, including enzymatic reactions, protein complex formation, passive and active transport processes, and diffusion. For example, in MONOD a hypothetical dimerization reaction $A + B \rightarrow C$ is a process with three effects: it removes one molecule of A (from the plasma membrane, say, and only if it is phosphorylated), and removes one molecule of B (from the cytosol), and ejects one molecule of C (into the plasma membrane, and in a certain modification state). Similarly, a hypothetical nuclear export process has four effects: it removes a molecule from the nucleus, places it in the cytosol, and hydrolyses ATP in the process, and this occurs only if a transport protein is present in the nuclear membrane (a requirement represented as a fourth effect).

To every reaction, species, state, or other object, the user can attach annotations, including citations and keywords. This capability is quite general: annotations can consist of text and attached files (such as images, movies, data tables, or other file types), annotations can be searched, and annotations can be annotated. For instance, a user might want to explain how an estimate for the number of molecules of a certain protein in an average cell was derived. The detail page for that estimate will show the annotation explaining the underlying experiments or reasoning, along with any supporting citations or graphics. Further users might add competing estimates for the same value, each with its own distinct annotations and citations. In the specific case of citations, MONOD automatically imports full references from PubMed, given a PubMed ID or journal, volume and page. The user can also select references to be imported using the integrated PubMed search tool. MONOD will download PDF versions of the selected papers when available, gaining access through the user’s electronic journal subscriptions as necessary. In the specific case of keywords, for example "journal club" or "G protein", the user can access the "keyword detail" page to browse journal club papers, or entries involving G proteins. Once data and annotations have been entered, users can search the database using a standard text search box, and browse the database contents by clicking on links between records.

MONOD also represents entire models, defined here as specific subsets of the entire set of species and reactions stored in the database. MONOD imports and exports these models using the Systems Biology Markup Language (SBML) Level 2 [15,16], a standard XML-based exchange format. This feature allows models developed in other environments (for example, graphical model building programs) to be loaded into MONOD, and conversely allows models developed in MONOD to be exported to simulators and visualization programs. MONOD also communicates within the BioSpice Dashboard environment (Kenneth Koster, personal communication; http://www.biospice.org), a software package that allows investigators to graphically specify and automate data flows among a large collection of programs useful for simulating and analyzing biological systems.

Features of MONOD relevant to interaction among multiple users

MONOD provides a medium for structured communication among its users. By allowing users to make entries into the system visible to others, the program allows investigators to construct models collaboratively and to discuss them. For example, suppose researchers disagree on the value of a reaction rate; in this case, the disagreement will be recorded, and will become apparent when browsing the accumulated entries in MONOD. The program can also be used as a typical web discussion forum, allowing users to reply to one another's postings. Access to such discussions (indeed, to any data or annotations in the system) can be restricted to specific groups of users through a fine-grained privilege system. The program requires users to log in with a username and password, and tracks this information throughout each session. When creating or editing a record, the user can specify which individuals or groups may view the entry, modify it, or grant privileges on it to others. For example, a user might enter an idea as a private annotation. She might subsequently release it to designated individuals and, still later, to all users of that instance of MONOD, thereby "publishing" the information within that microcosm. MONOD also incorporates a revision control system, similar in concept to the Concurrent Versions System (CVS) [17] that is widely used to coordinate the development of software projects. Like CVS, MONOD retains every revision of every record, along with a time and date stamp and the name of the user who made the revision. Normally, users only see the most current revisions of records, but they may choose to view any record from any time in the past. The revision control system allows the program to capture disagreements, and allows users to explore the history of such disagreements by studying branches of the revision tree. As researchers modify different records over time, this aspect of the program will provide a primary record of how the understanding of a biological system develops.

SIDEBAR

Example in practice: Alpha pathway ODE model

We used MONOD to capture biological knowledge about the yeast pheromone signal transduction pathway and to document assumptions used in building a computational model of the pathway. We subsequently built a chemical reaction model in Matlab, consisting of a series of nonlinear ordinary differential equations (ODEs), describing dynamic changes in the amounts of each molecular species over time. Currently the pheromone response model includes 300 rate equations, 108 different rate constants, and 1749 distinct elementary chemical reactions.

To do so, we first assembled information and assumptions from the literature into MONOD in the form of annotations linked to the species and reactions being studied. With the information assembled, we examined how it might be used to construct an ODE model. This examination helped us identify gaps in our knowledge of the pathway, and increased our awareness of how many of the needed assumptions went beyond the available evidence.

We then used MONOD to help construct a model of the pheromone response pathway. We illustrate this process by describing the work involved in representing the first reaction in the pathway: the binding of pheromone to the cell surface receptor Ste2. In cells not exposed to pheromone, the receptor is bound to the G protein, and the α subunit (Gpa1) of this trimeric protein is bound to GDP. The binding of pheromone to Ste2 stimulates the exchange of GDP for GTP on Gpa1 and also Gpa1's dissociation from its other two partners. At an extracellular pH lower than 7 and in low ionic strength, pheromone binds to the receptor whether or not the receptor is bound to the G protein [18]. For the model, we had to decide whether to represent the binding affinity as a binary switch, dependent on the presence of the G protein (assuming a pH greater than 7 or high ionic strength) or as constant under all circumstances. Since the yeast are grown in the lab at low pH and ionic strength, we decided to posit that the binding affinity between pheromone and Ste2 was constant. We then modeled the interaction as a second-order association reaction and first-order dissociation reaction. We assigned the interaction K_d value of 4 nM, which we selected as a consensus from the literature [18,19,20,21,22]. We similarly arbitrarily estimated the association rate as 3×10^5 $(M\cdot s)^{-1}$ (within the range of experimentally observed rates of 10^5 to 10^7 $(M\cdot s)^{-1}$ [23]), which resulted in a dissociation rate of 1.2×10^{-3} s^{-1}.

We proceeded through the rest of the pathway in a similar manner, recording each step of the process in MONOD. While this narrative account of the process of model writing is linear, our actual work was iterative, with many cycles of recording and synthesizing results extracted from the literature, filling in the gaps with assumptions, and attempting to assess the likelihoods of different proposed mechanisms. MONOD supported this style of working by enabling us to browse and search accumulated knowledge of the pathway and by allowing incremental annotations as the model developed.

END SIDEBAR

Development now in progress

Enabling a network of peer-to-peer MONOD servers.

In the present version of MONOD, collaboration is possible only between users of the same instance of the program, and remote users need to access the single server that runs it. If MONOD proves to be a useful knowledge sharing mechanism and the number of people using it increases, we could imagine creating a single, central, MONOD server. However, at present we believe that a linked network of distributed servers will be more secure, faster, and more reliable. In this vision, a future MONOD network would grow organically as more labs establish and maintain individual servers. This development path should better address security concerns, since private data can be stored on a local server under the physical control of laboratory that generates it, thereby ensuring local control of who accesses the data (for example, any users outside the lab group might be prevented from accessing certain data, perhaps with an exception for one trusted collaborator who backs up the data on a remote server). This architecture should also give better performance, since users would interact primarily with their local servers, which will intelligently cache remote data. In this architecture, an individual interacting with a local server will have the illusion of accessing a single worldwide database.

Extension of the database schema to handle binding sites on proteins, protein complexes, and quantitative parameters.

Knowledge representation programs like MONOD require a specific database schema that reflects the structure of that specific knowledge. MONOD's current representation of biological knowledge is "reaction-centric", rather than explicitly involving space or geometry. But the current database schema does not yet formally represent several important aspects of even this limited view of intracellular processes. In particular, while it is possible to represent association reactions at the level of molecular species, MONOD does not yet track individual sites on proteins that form complexes with other proteins.

That deficiency requires the representation of multi-protein complexes as distinct molecular species, and the enumeration of all the different sequences or pathways of association reactions that are able to produce those complexes. But, given even a small number of complex forming proteins and different modification states, the very large numbers of possible complexes make enumeration impractical [24].

To solve this problem, we are making use of lessons learned from the development of Moleculizer, a stochastic reaction network generator and simulator [25]. Moleculizer uses its representation of individual binding sites to construct complexes only as those arise in the course of a simulation. Future versions of MONOD will represent binding sites in the same way, allowing the program to automatically assemble all complexes that are consistent with the set of possible binding site interactions and known constraints on those interactions.

We are also trying to solve the related problems of the representation of different protein conformational states, modification states, and "states of activity". Some protein conformations and modification states occur only when a protein is a member of certain complexes. Moreover, one "state of activity", complex membership, cannot sensibly be attributed to individual proteins but rather must be shared by all members of a protein complex (Ron Maimon, personal communication). Finally, we are extending the schema of the MONOD database to accommodate the quantitative parameters and rate laws required for quantitative simulations.

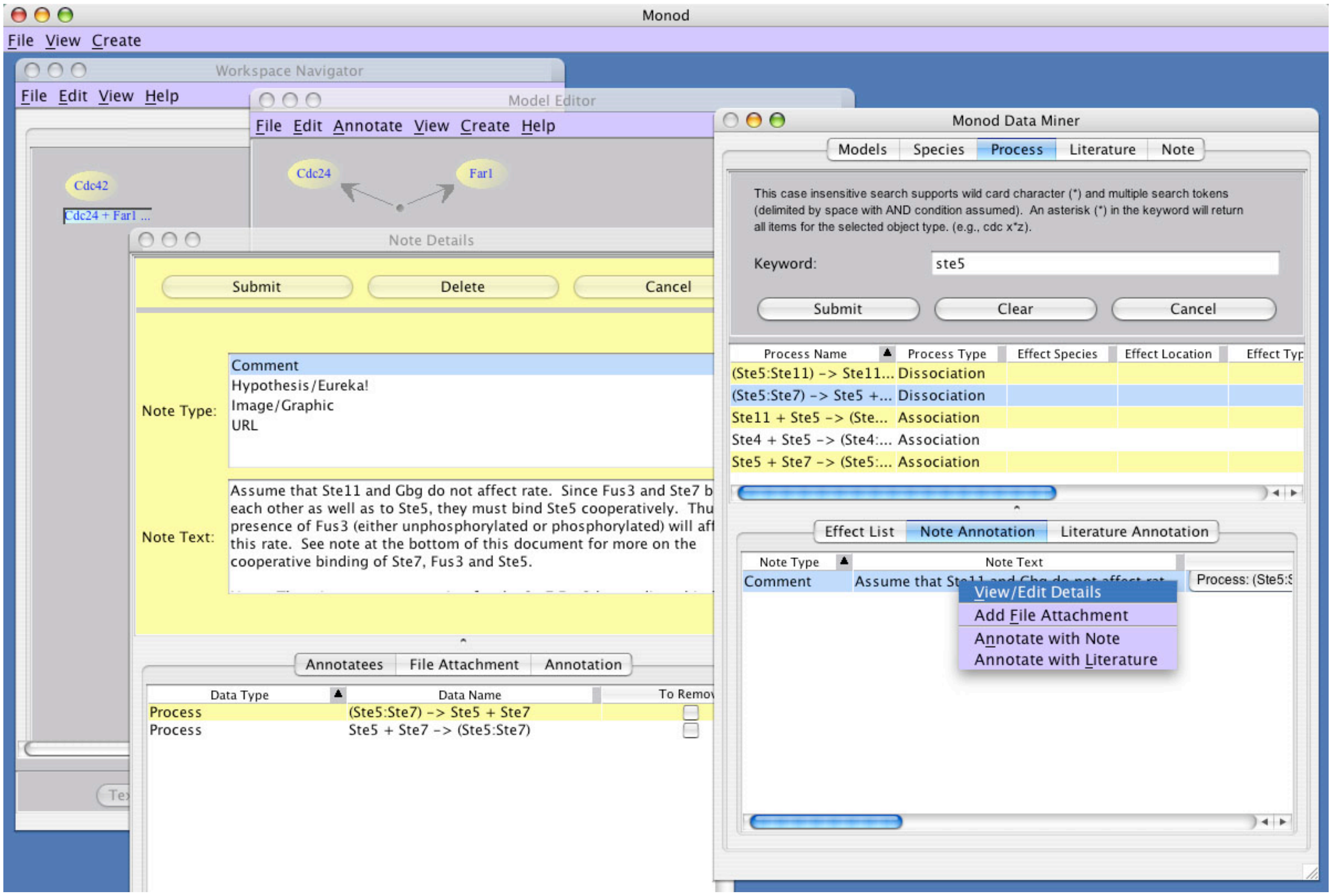
Integration of MONOD with other software tools.

Currently MONOD allows for import and export of data via SBML. We wish to allow the export of MONOD models to the Moleculizer stochastic reaction network generator and simulator. For these purposes, the current version of SBML (Level 2) is insufficient, and we are modifying MONOD to work with the planned SBML Level 3. We are also working to enable MONOD to archive results of simulation runs from Moleculizer and eventually from other simulators. Finally, we plan to allow the use of Gene Ontology (GO) terms [26,27] and ontology keywords from future versions of Textpresso [28] to enable improved searching, improved linking to external databases, and improved integration of imported data and models.

Development of a graphical user interface.

We initially implemented MONOD as a web application because doing so allowed remote users to access it easily via standard browsers. But we found that the web interface is cumbersome; many mouse clicks are required to accomplish any given task, and the resulting delays are frustrating to the user. The functionality this type of interface can offer is inherently limited. For this reason, we are developing a desktop graphical user interface (GUI) client, called MONOD Desktop (a beta version is now available). This client communicates with a backing MONOD server, but provides a more fluid and dynamic user interface [29,30], with contextual pop-up menus, drag-and-drop capabilities, and better navigation.

MONOD Desktop includes four primary components: a search interface; an annotation editor, allowing connection of a single annotation to multiple species or processes by drag-and-drop; a diagrammatic model editor using the Kohn molecular interaction notation [31,32]; and a "workspace navigator" for quick access to favorite items and work in progress (Figure 2).



Note Type: Comment
Hypothesis/Eureka!
Image/Graphic
URL

Note Text: Assume that Ste11 and Gbg do not affect rate. Since Fus3 and Ste7 b each other as well as to Ste5, they must bind Ste5 cooperatively. Thu presence of Fus3 (either unphosphorylated or phosphorylated) will aff this rate. See note at the bottom of this document for more on the cooperative binding of Ste7, Fus3 and Ste5.

Data Type ▲	Data Name	To Remov
Process	(Ste5:Ste7) -> Ste5 + Ste7	□
Process	Ste5 + Ste7 -> (Ste5:Ste7)	□

This case insensitive search supports wild card character (*) and multiple search tokens (delimited by space with AND condition assumed). An asterisk (*) in the keyword will return all items for the selected object type. (e.g., cdc x*z).

Keyword: ste5

Process Name ▲	Process Type	Effect Species	Effect Location	Effect Typ
(Ste5:Ste11) -> Ste11...	Dissociation			
(Ste5:Ste7) -> Ste5 +...	Dissociation			
Ste11 + Ste5 -> (Ste...	Association			
Ste4 + Ste5 -> (Ste4:...	Association			
Ste5 + Ste7 -> (Ste5:...	Association			

Note Type ▲	Note Text
Comment	Assume that St

Figure 2. A view of MONOD Desktop, a graphical user interface for working with information on a MONOD server. This interface allows searching, browsing, and editing data more fluidly than is possible through the web interface, and additionally allows users to draw diagrams of reaction networks.

Discussion

We have described the development and first three years of work on MONOD. This software tool embodies a number of ideas for knowledge representation articulated by one of the authors [33] and by others [34,35] during the 1990s. The program facilitates user-guided extraction from literature of information relevant for building quantitative models and qualitative representations of intracellular events. It provides a shared repository of knowledge organized around a network of relationships between molecular species, the reactions they undergo, their modification states, cellular compartments in which they are found, citations, and various other properties. It affords an extraordinarily flexible means for affixing comments and supporting information (including images, PDFs, and links to external databases) to this data structure. The information handled can be as detailed as the user wishes. MONOD thus exemplifies a data structure that allows storage, modification, and retrieval of knowledge at an organizational level intermediate between the rigid framework provided by a more conventional database and the free form allowed in the natural language literature. While MONOD is at present best suited to represent signal transduction pathways, we are working work to extend its data schema.

The initial motivation for building MONOD was to aid in constructing computable models of biological behavior. The resulting program has some features in common with projects such as ProcessDB [36] and SigPath [37], but its design more explicitly supports the documentation of reasoning and inferences, the interactive elaboration and discussion of ideas, and collaboration. We have demonstrated in one test case that MONOD serves as a useful repository of information used to build a computational model and as a tool to document judgments and choices made during the course of building the model. This form of documentation helped us organize our work during the iterative construction of the model, helped us interpret the results of simulations based on the model, and we believe it will be a useful reference as we and others revise it.

Taken together, MONOD's communications, security, and revision control components, as well as its fine-grained structure, also make it a novel medium for structured communication and discourse. Use of MONOD should help integrate the efforts of multiple researchers with different expertise, and help bring about models that are larger, more detailed, more transparent, and more up-to-date than those commonly produced by single individuals. Accumulated entries in MONOD may, for some purposes, complement the role now filled by review articles, by allowing users to quickly survey current thinking on specialized topics. These sorts of communication capabilities might complement those provided by journal publications, email, phone calls, conference talks, and personal conversations.

For the possible positive outcome of MONOD's data structure and communications ability to be realized, individual users must perceive personal benefit from entering data into it. At present we imagine three such benefits. First, an individual building a model needs to keep track of what he or she knows about the system being modeled, so a tool that supports model building should be valuable per se. Second, as is true in scientific writing, the process of formalizing knowledge in MONOD should help the users see connections between facts or gaps in current knowledge that were previously overlooked. Third, increased understanding and productivity resulting from structured communications with labmates, collaborators, and the community at large might motivate researchers to participate, in the same way that they are motivated now to use email or to attend conferences.

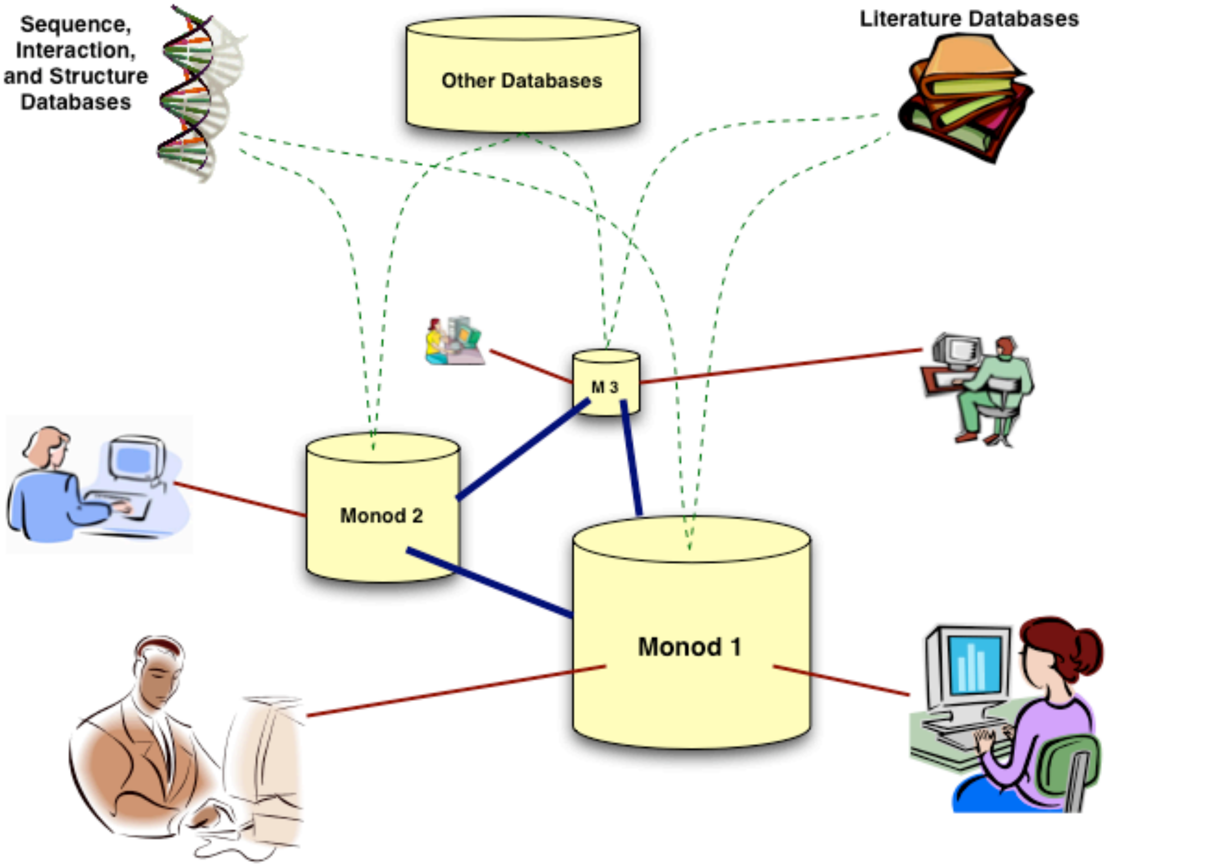


Figure 3. We are in the process of extending the Monod structure so that it enables secure peer-to-peer exchange of data among different instances of its relational database. The figure shows a future network in which analysts interact with multiple, secure, instances of Monod as the instances of Monod exchange data with one another. We show analysts working on information resident in Monod and other online databases, but note that analysts can also input into Monod and work with information from other sources such as personal communications, telephone conversations, and personal print magazine subscriptions. Although the current Monod database schema and user interface focus on models of intracellular signal transduction pathways, the underlying technologies will support work in other domains—so that the analysts shown here could be tracking the progress of bills through a session of a state legislature; or they might be tracking incidence data, current research, and public health measures for a disease caused by a new pathogen.

We hope that the use of tools like MONOD might contribute to the development of a more collaborative culture in biology (Figure 3). Although it is by no means clear (or desirable) that most biology will evolve toward a "big science" model, it does seem clear that some biology—from genome sequencing work, to most work in corporations—will require teams of investigators. In general, we imagine that technology that facilitates interaction, communication, fact checking, and attribution of ideas will have a positive effect on this "conventional" biological culture. It has already had a positive effect on the coalition research effort for which it was designed and built. Moreover, as the new discipline of design-based engineering of biological systems (sometimes called "synthetic biology") emerges [38,39], we imagine that biological engineers will use tools like MONOD to facilitate collaborative work on projects, just as software engineers now use collaboration tools such as SourceForge [40]) and CVS. For engineering purposes, tools to help structure communication are particularly helpful, because successful engineering of biological, mechanical, or electronic systems benefits from communication that employs standard formal representations of their behavior (as required, for example, by the MIT Registry of Standard Biological Parts (http://parts.mit.edu)).

MONOD offers a means of documenting and communicating about research to those provided by the formal academic literature. One set of complementary capabilities may lie in its communication functions. In the traditional style of scholarly communication, the accretion of knowledge proceeds by a natural language discussion, carried out over a period of years, and mainly in print. In comparison, communication through MONOD should lower the bar to providing structured commentary and new information, and should facilitate interaction among researchers in different spatial locations and positions within the scientific hierarchy. This communication and elaboration of information can occur in near real time. Furthermore, most of the knowledge scientists generate is never published in the formal literature. This knowledge includes but is not limited to strong negative results, weak negative results, weak positive results, experience with laboratory techniques that may be useful but did not generate published results, and confirmations of previous findings. This kind of knowledge is now propagated primarily as an oral culture, or worse, is lost entirely. MONOD and tools like it should allow researchers to make this information available to the community without stigma. We expect that the aggregation of such results and experiences from many researchers will show how valuable this information might become.

Another set of complementary capabilities might lie in MONOD's data structure. In contrast to the large, slowly updated blocks of information contained in individual journal papers and books, MONOD represents information as a large number of smaller connected pieces, each of which can be independently updated [33,41,42]. In Eric Raymond's metaphor, we might think of books and journal articles as "cathedrals" and MONOD as a "bazaar" [43]. MONOD is in this sense like a Wiki, a system for collaboratively editing a set of interlinked web pages [44] (see also http://www.wiki.org and http://www.wikipedia.org for a large-scale example), with the distinction that MONOD benefits from an organizational structure specific to its problem domain of molecular biology. While information in MONOD will often be rooted in primary literature, its fine-grained data representation makes it easy to browse and to search, and these attributes may make biological knowledge more accessible to those not comfortable reading the textbooks and primary literature (for example, students and people coming from engineering backgrounds). But we also note that nothing in the fine grained data representation prevents future investigators from selecting a set of linked results, submitting those to future "journal editors" for peer review, "publishing" the linked results to the database, and having a higher value ascribed to these bodies of work.

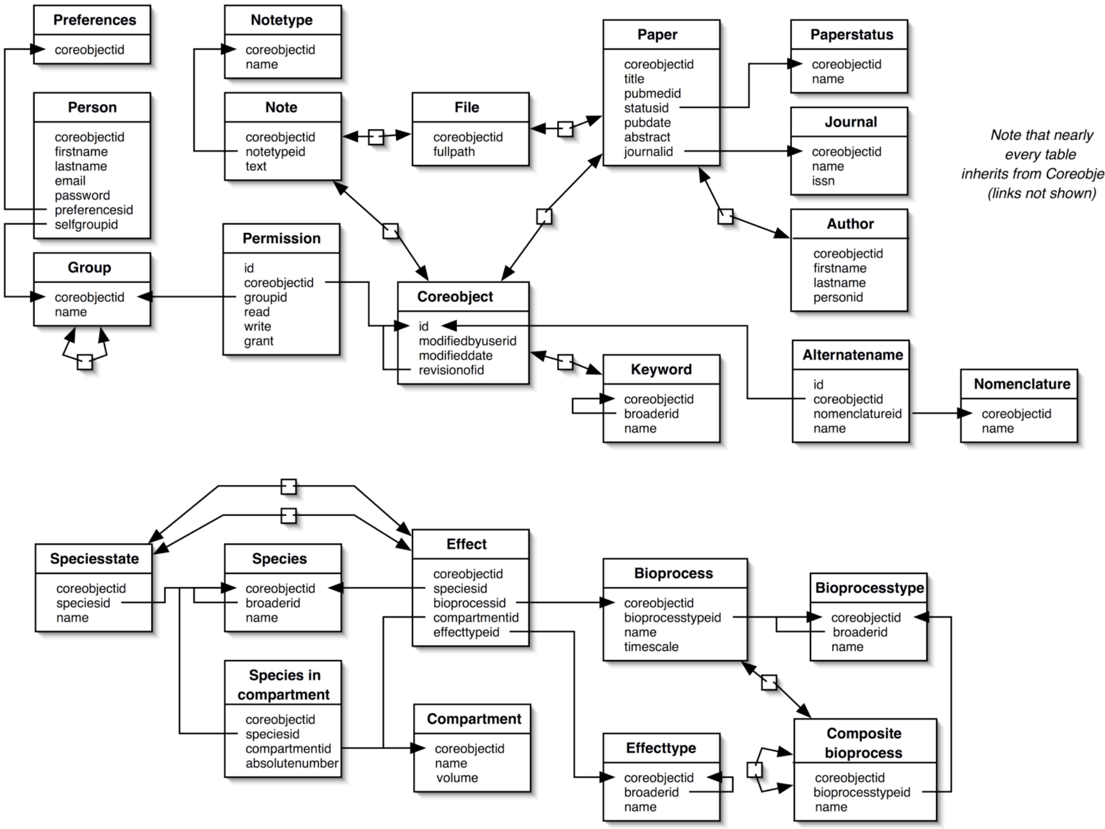
This data-structure and communication ideas embodied in MONOD may have application outside of biology. It is possible that other kinds of knowledge-based activities, including the detection and management of infectious diseases, the work of ecologists tracking changes in population structures, the work of news-gathering organizations, or the work of government and nonprofit agencies on international development issues, would benefit from the use of collaborative knowledge management systems of this type.

Methods

We wrote MONOD in Java 1.4 using a number of well-known and well-tested open source software entities: the PostgreSQL relational database management system (http://www.postgres.org), the Apache web server (http://www.apache.org), the Resin-EE Java application server (http://www.caucho.com), Enterprise Java Beans (EJBs) (http://java.sun.com/products/ejb), Xdoclet (http://www.xdoclet.org), and the eXtensible Stylesheet Language Transformations (XSLT) (http://www.w3.org/TR/xslt). In MONOD, these software entities work together to translate web-based activities into commands that store and retrieve information in the relational database. All components of the system are freely downloadable and will run on Linux, Mac OS X, and Windows.

The MONOD Desktop GUI is written in Java Swing, and can be started directly from a web page using the Java Web Start technology (http://java.sun.com/products/javawebstart). It communicates with the MONOD server using Resin’s Hessian binary RPC protocol (http://www.caucho.com/hessian).

We show the schema of the database that now underpins MONOD in the entity-relationship diagram in figure 4. The most important point about this schema is that it is adapted to descriptions of biological systems that can be reduced to named molecular species and the reactions they undergo. At the same time, because many generic functions are associated with a “Coreobject” table from which all others inherit, we and others can extended this schema to include different kinds of biological knowledge. If, for example, we were to add a “Sequence” table (also inheriting from “Coreobject”) to represent nucleic acid sequences, then sequence records would automatically support all of the generic functions: textual annotations, citations, user permissions, and so on.



Note that nearly every table inherits from Coreobject! (links not shown)

Figure 4. The version 1.5 database schema. Coreobjects provide generic functionality, such as user permissions, revision control, annotations (Notes), citations (Papers), and keywords. Most other tables, such as Species and Bioprocess, inherit this functionality. Small squares denote many-to-many link tables, and plain arrows denote many-to-one relationships.

MONOD was first released in April 2002, under the GNU Lesser General Public License (LGPL) [45]. Its source code may be freely downloaded (from http://monod.molsci.org) and modified. As of this writing, the current version is 1.5.

Contributions

MONOD was designed and written by David Soergel with significant programming assistance from Brian George and Ross Morgan-Linial. Drew Endy initially articulated the need for a tool like MONOD, and has helped direct the software development effort by providing active feedback throughout the process. Roger Brent provided continuing comment and encouragement, as well as input to the definition of functionality. Kirindi Choi wrote MONOD Desktop (the GUI), and Jay Doane is developing the peer-to-peer architecture. Ty Thomson researched and constructed the model of the Alpha pheromone pathway in yeast and provided valuable feedback to the developers. The manuscript was written by Soergel, Brent, Endy, and Thomson, who guarantee the veracity of this report.

Acknowledgements

We are grateful to Larry Lok for ongoing helpful input on many topics, especially assembly of protein complexes, and for extensive discussion of his unpublished work on Moleculizer; to Ron Maimon for pointing out the special representation problems pertinent to membership in such complexes; to Ken Koster for communication of the details of the Analyzer API used in the BioSpice Dashboard; to Scott Ferguson for incorporating our patches into the Resin-EE application server; to Paul Rabinow for useful discussions on knowledge production, to Kirsten Benjamin, Orna Resnekov and Dagobert Soergel for critical reading of the manuscript; and to Alejandro Colman-Lerner, Chris Hong, Sri Kosuri, Dan Moisa, Patrick Paul, Birgit Schoeberl, and Jonathan Webb for testing and user feedback.

Work was funded via a grant to R.B. from the DARPA IPTO Bio-Comp program.

References

1 Alberts B, Johnson A, Lewis J, Raff M, Roberts K, and Walter P. (2002) Molecular Biology of the Cell, 4th edition. New York: Garland Science Publishers.

2 Lodish HF, Berk A, Matsudaira P, Kaiser CA, Krieger M, Scott MP, Zipursky L, Darnell J. (2003) Molecular Cell Biology, 5th edition. New York: W. H. Freeman and Company.

3 Endy D, Brent R (2001) Modelling cellular behaviour. Nature 409: 391-395.

4 Polanyi, M (1967) The tacit dimension. Garden City, New York: Anchor Books, Doubleday and Company.

5 Kipling, R (1902) Just So Stories (US edition, 1912). New York: Doubleday and Company .

6 Takahashi K, Kaizu K, Hu B, Tomita M (2004) A multi-algorithm, multi-timescale method for cell simulation. Bioinformatics 20: 538-546.

7 Jacob F, Perrin D, Sanchez C, Monod J (1960) L'operon: groupe de genes a expression coordennee par un operateur. Comptes rendus de l'Académie des Sciences 250: 1727-1729.

8 Cherry JM, Adler C, Ball C, Chervitz SA, Dwight SS, et al. (1998) SGD: Saccharomyces Genome Database. Nucleic Acids Res 26: 73–79.

9 Schuler GD, Epstein JA, Ohkawa H, Kans JA (1996) Entrez: molecular biology database and retrieval system. Methods Enzymol 266: 141–162.

10 Ostell J. (2002) The Entrez Search and Retrieval System. In: McEntyre J, editor. The NCBI handbook [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=handbook.chapter.588

11 Boeckmann B, Bairoch A, Apweiler R, Blatter M, Estreicher A, et al. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res 31(1): 365-370.

12 Wu CH, Yeh LS, Huang H, Arminski L, Castro-Alvear J, et al. (2003) The Protein Information Resource. Nucleic Acids Res 31: 345-347.

13 Hodges PE, McKee AH, Davis BP, Payne WE, Garrels JI. (1999). The Yeast Proteome Database (YPD): a model for the organization and presentation of genome-wide functional data. Nucleic Acids Res 27: 69-73.

14 Mewes HW, Amid C, Arnold R, Frishman D, Guldener U, et al. (2004) MIPS: analysis and annotation of proteins from whole genomes. Nucleic Acids Res. 32 Database issue: D41-44.

15 Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19: 524-531.

16 Finney A, Hucka M (2003) Systems biology markup language: Level 2 and beyond. Biochem Soc Trans, 31(Pt 6): 1472-1473.

17 Cederqvist P, Pesch R, Grubbs D, et al. (1993) Version Management with CVS. http://www.cvshome.org

18 Weiner JL, Guttierez-Steil C, Blumer, KJ (1993) Disruption of receptor-G protein coupling in yeast promotes the function of an SST2-dependent adaptation pathway. J Biol Chem 268: 8070-8077.

19 Dosil M, Schandel KA, Gupta E, Jenness DD, Konopka JB (2000) The C terminus of the Saccharomyces cerevisiae alpha-factor receptor contributes to the formation of preactivation complexes with its cognate G protein. Mol Cell Bio 20: 5321-5329.

20 David NE, Gee M, Andersen B, Naider F, Thorner J, et al. (1997) Expression and purification of the Saccharomyces cerevisiae alpha-factor receptor (Ste2p), a 7-transmembrane-segment G protein-coupled receptor. J Biol Chem 272: 15553-15561.

21 Clark CD, Palzkill T, Botstein D (1994) Systematic mutagenesis of the yeast mating pheromone receptor third intracellular loop. J Biol Chem 269: 8831-8841.

22 Jenness DD, Burkholder AC, Hartwell LH (1985) Binding of alpha-factor pheromone to Saccharomyces cerevisiae a cells: dissociation constant and number of binding sites. Mol Cell Biol 6: 318-320

23 Janin J, Chothia C (1990) The structure of protein-protein recognition sites. J Biol Chem 265:16027-16030.

24 Morton-Firth CJ (1998) Stochastic Simulation of Cell Signalling Pathways (PhD thesis). Cambridge, UK: University of Cambridge.

25 Lok WL, Brent R (2004) Automatic generation of reaction networks with Moleculizer 1.0. Nature Biotechnology, submitted

26 Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 25: 25-29.

27 Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, et al. (2004) The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res 32 Database issue: D258-261.

28 Müller HM, Kenny EE, Sternberg PW (2004) Textpresso: An ontology-based information retrieval and extraction system for biological literature. PLoS Biol 2: e309, 1-15

29 Mandel T (1997) The Elements of User Interface Design. New York: John Wiley.

30 Raskin J (2000) The Humane Interface: New Directions for Designing Interactive Systems. Boston: Addison-Wesley.

31 Kohn KW (1999) Molecular interaction map of the mammalian cell cycle control and DNA repair systems. Mol Biol Cell 10: 2703-2734.

32 Kohn KW (2001) Molecular interaction maps as information organizers and simulation guides. Chaos 11: 84-97.

33 Soergel, David (1998) OrganISM, an abstract representation paradigm for intelligent information networks. Stanford, CA: Stanford University, Symbolic Systems Program. http://www.davidsoergel.com/organism.html.

34 Soergel, Dagobert (1994) Information Structure Management. A unified framework for indexing and searching in database, expert, information-retrieval, and hypermedia systems. In: Fidel R, Hahn TB, Rasmussen ER, Smith PJ, editors. Challenges in Indexing Electronic Text and Images. Medford, NJ: Learned Information. pp. 111-156.

35 Soergel, Dagobert (2000) Design of an integrated information structure interface. A unified framework for indexing and searching in database, expert, information retrieval, and hypermedia systems. (updated and enlarged). College Park, MD: University of Maryland, College of Information Studies. http://www.dsoergel.com/cv/D11.html.

36 Chasson AK, Phair RD (2001) ProcessDB: A cellular process database supporting large-scale iterative kinetic modeling in cell biology. 2nd International Conference on Systems Biology (ICSB2001), Pasadena, CA.
37 Campagn F, Neves S, Chang CW, Skrabanek L, Ram PT, et al. (2004) Quantitative information management for the biochemical computation of cellular networks. *Sci STKE*, *2004(248)*, PL11.

38 Gibbs WW (2004) Synthetic life. Sci Am, 290(5), 74-81.

39 Brent R (2004) A Partnership between biology and engineering. Nat Biotech 22: 1211-1214

40 OSDN: Open Source Development Network, Inc. (1999-2004) SourceForge Collaborative Software Development Tools. http://www.sourceforge.net.

41 Soergel, Dagobert (1977) An automated encyclopedia -- a solution of the information problem? International Classification 4(1): 4-10; 4(2): 81-89.

42 Bush V (1945) As We May Think. The Atlantic Monthly 176: 101-108.

43 Raymond ES (1999) The cathedral and the bazaar. San Francisco: O'Reilly and Associates.

44 Leuf B, Cunningham W (2001) The Wiki Way: Collaboration and Sharing on the Internet. Boston: Addison-Wesley.

45 Free Software Foundation (1991, 1999) GNU Lesser General Public License. http://www.gnu.org/copyleft/lesser.html.