The pathway tools software

Peter D. Karp [*], *Suzanne Paley and Pedro Romero*

Bioinformatics Research Group, SRI International, 333 Ravenswood Ave, Menlo Park, CA, 94025, USA

Received on January 24, 2002; revised and accepted on March 29, 2002

ABSTRACT

Motivation: Bioinformatics requires reusable software tools for creating model-organism databases (MODs).

Results: The Pathway Tools is a reusable, production-quality software environment for creating a type of MOD called a Pathway/Genome Database (PGDB). A PGDB such as EcoCyc (see http://ecocyc.org) integrates our evolving understanding of the genes, proteins, metabolic network, and genetic network of an organism. This paper provides an overview of the four main components of the Pathway Tools: The PathoLogic component supports creation of new PGDBs from the annotated genome of an organism. The Pathway/Genome Navigator provides query, visualization, and Web-publishing services for PGDBs. The Pathway/Genome Editors support interactive updating of PGDBs. The Pathway Tools ontology defines the schema of PGDBs. The Pathway Tools makes use of the Ocelot object database system for data management services for PGDBs. The Pathway Tools has been used to build PGDBs for 13 organisms within SRI and by external users.

Availability: The software is freely available to academics and is available for a fee to commercial institutions. Contact ptools-support@ai.sri.com for information on obtaining the software.

Contact: pkarp@ai.sri.com

Keywords: Bioinformatics; model organism database; genome analyses; metabolic pathways.

INTRODUCTION

The Pathway Tools is a software environment for creating a type of model-organism database (MOD) called a Pathway/Genome Database (PGDB). A PGDB integrates information about the genes, proteins, metabolic network, and genetic network of an organism. The Pathway Tools provides two different modalities for interacting with a PGDB: it provides a graphical environment that allows users to visualize the contents of a PGDB and to interactively update a PGDB; and it provides a sophisticated ontology and database API that allow programs to perform complex queries, symbolic computations, and data mining on the contents of a PGDB. For example, the software has been used for global studies of the *E. coli* metabolic network (Ouzounis and Karp, 2000) and genetic network (Karp, 2001).

The origins of Pathway Tools stem from its development for the EcoCyc project in the mid 1990s. In 1997 we began to generalize the software so that it could be applied to manage genome data from other organisms besides *E. coli*, and so that it could simultaneously manage data from multiple organisms to facilitate comparative genomics studies.

MODEL ORGANISM DATABASES

Although there has been a trend whereby every MOD project has developed its own unique software and database environment (Flybase Consortium, 1998; Ball *et al.*, 2000; Blake *et al.*, 2001; Stein *et al.*, 2001), it is not practical for this trend to continue because (a) the large number of new MODs that are emerging as a result of genome sequencing projects makes it prohibitively expensive for each group to create its own software environment, compared to the cost of reusing existing software; (b) MOD software environments contain algorithms of such complexity that they have been and will be impossible for some groups to reproduce, or would take years of effort to reproduce (examples include graphical layout of individual metabolic and signaling pathways, and layout of the complete metabolic map of an organism); (c) a proliferation of incompatible software environments will make comparative studies across multiple MODs difficult.

MODs serve several important functions in the post-genomic era. An MOD serves as an electronic reference source for the genome sequence of an organism, and for the interpretation of that sequence. It also serves as a central repository for integration of the evolving interpretation of a genome as new wet-lab and computational predictions of gene function are produced. Ongoing revision of genome annotations is an important undertaking because all genome annotations to date are both incomplete (omitting function predictions for

[*] To whom correspondence should be addressed.

many genes) and contain errors. A MOD is a device for integrating the many different predictions of gene function made throughout the scientific community, and for disseminating the most accurate current annotation of a genome to the community. A MOD is also a resource for supporting functional-genomics studies of an organism, such as through the Pathway Tools tool for painting gene-expression data on the full metabolic pathway map of the organism. We contrast the MOD approach, in which biologist experts for an organism manually curate the genome annotation, with the many databases of genome annotations that are derived from purely computational predictions, which are likely to be less reliable.

MODs will also serve as a critical foundation for systems biology research (Karp, 2001). Systems biology seeks to define theories that explain the complex manner in which the molecular parts of a biological system give rise to the behaviour of that system. An MOD is a catalog of both the molecular parts of an organism, and of the molecular interactions among those parts, and is therefore a vehicle for storing systems-biology theories, for testing those theories for internal consistency and for consistency with external data, and for refining these theories to be consistent with external data.

DEFINITIONS

Pathway/Genome Database: A PGDB describes the genome of an organism (its chromosome(s), genes, and genome sequence), the product of each gene, the biochemical reaction(s) catalysed by each gene product, the substrates of each reaction, and the organization of reactions into pathways. A PGDB can also describe the genetic network of an organism: its promoters, operons, transcription factors, and transcription-factor binding sites. A PGDB is a type of MOD.

EcoCyc Database: A PGDB for the organism *E. coli* (Karp *et al.*, 2002). The majority of the information in EcoCyc is derived from the biomedical literature.

MetaCyc Database: A PGDB containing metabolic data for many different organisms (Karp *et al.*, 2002). The goal of MetaCyc is to contain broad coverage of experimentally elucidated metabolic pathways from many different organisms, rather than to attempt to model the complete pathway complement of any particular organism. MetaCyc contains a broad base of well-established pathways that are used by the PathoLogic program to predict the pathway complement of a particular organism, which is modelled within a separate PGDB for that organism. The majority of the information in MetaCyc is derived from the biomedical literature.

BsubCyc Database: BsubCyc is a PGDB for *Bacillus subtilis* that was generated by the PathoLogic program and is available through URL http://biocyc.org.

BioCyc Knowledge Library: The collection of PGDBs at URL http://biocyc.org is called the BioCyc Knowledge Library. EcoCyc, MetaCyc, and BsubCyc are all component databases within the BioCyc Library.

OVERVIEW OF PATHWAY TOOLS CAPABILITIES

The following high-level capabilities are provided by the four main software components of the Pathway Tools, and by the Ocelot object database system used in conjunction with the Pathway Tools.

The **Pathway/Genome Navigator** provides query, visualization, and analysis services for PGDBs, both as a local application and as a Web server. These services allow scientists to find information quickly, to display that information in familiar graphical forms, and to disseminate a PGDB to the scientific community via the Web. The Navigator provides a platform for pathway-based analysis of functional-genomics data with a tool for painting gene-expression data on a full metabolic map of the cell.

The **PathoLogic** program allows a user to create a new PGDB quickly using the annotated genome of an organism as the starting point. PathoLogic generates a new PGDB that contains the genes, proteins, biochemical reactions, and predicted metabolic pathways of the organism.

The **Pathway/Genome Editors** provide a PGDB developer with interactive forms for editing the contents of a PGDB to refine its content, such as creating a new metabolic pathway, defining an interaction between a transcription factor and a newly discovered binding site for the factor, or modifying the function of a transport protein.

The **Pathway Tools ontology** defines a rich set of classes, attributes, and relationships for high-fidelity modelling of biological data such as metabolic pathways, enzyme function, DNA regions such as genes and promoters, and mechanisms of gene regulation.

The **Ocelot object database** system provides database management (DBMS) services for the Pathway Tools. It provides powerful operations for distributed development of PGDBs that go far beyond the relational databases and flat files commonly used in bioinformatics. For example, (1) It records a history of all DB updates so the developer can determine who made what DB changes on what date; (2) Ocelot does not require an expensive or hard-to-install commercial DBMS such as Oracle, although use of Oracle as a DB back-end increases the power of Ocelot; and (3) It provides schema-evolution services that are valuable because the complexity of bioinformatics schemas implies that they are always evolving.

Pathway Tools is production-grade software: It is easy to install and includes extensive documentation (170 pages in length). An informational Web site at URL http://bioinformatics.ai.sri.com/ptools/ provides examples and further information about the software. Seven different users outside SRI are using the software to build PGDBs, such as for *Plasmodium falciparum* and *Arabidopsis thaliana*—see URLs http://plasmocyc.stanford.edu and http://aracyc.stanford.edu.

SYSTEM ARCHITECTURES

The Pathway Tools can be configured in several combinations to provide a system architecture that is appropriate to the task at hand. See Figure 1 for an illustration of the following configurations.

(1) Distributed Concurrent PGDB Development Configuration: A PGDB is persistently stored in an Oracle DBMS that is queried via a network by multiple concurrent developers. PGDB objects are incrementally faulted across the network as the user refers to them during queries and edits. Modified objects are saved back to Oracle. This configuration runs under Sun Solaris.

(2) Nonconcurrent Development Configuration without Oracle: The Oracle DBMS is difficult to install and configure; therefore, some users prefer a configuration in which PGDBs are stored in disk files. A PGDB is loaded from disk (which typically takes about 1 minute) when a user starts a session, and is saved to disk in its entirety when a developer saves updates. Only one user at a time can update a PGDB in this configuration. File PGDBs can easily be converted to Oracle PGDBs at a later time. This configuration runs under Sun Solaris.

(3) In-Memory PGDB Delivery Configuration: When SRI delivers a collection of PGDBs to scientists for read-only query and visualization access, we load all PGDBs into the Pathway Tools binary program file. This configuration does not require Oracle either, and it performs even better than configuration (2) because all PGDBs are already in virtual memory when the user executes the program. This configuration runs under Sun Solaris, and it runs on a PC under Microsoft Windows (Windows-98 and above), in which case it runs as an application that interacts with a single user.

Note: The Sun Solaris version of the Navigator can run as either an X-windows application, or as a Web server.

THE PATHWAY / GENOME NAVIGATOR

The Navigator software provides a scientist user with the ability to interrogate a PGDB and visualize the results of a query in an intuitive, graphical fashion. It also provides analysis operations, such as whole-metabolic-map comparisons across multiple organisms. The Navigator contains several visualization tools, each of which has been carefully crafted to generate a window of information for a given bioinformatics data type, such as operons or metabolic pathways. These visualization tools are as follows.

The **pathway visualization** produces automated layouts of metabolic pathways at multiple levels of semantic resolution. The tool depicts pathways of different topologies using different layouts (linear, circular, and tree structured). By varying the resolution of a pathway drawing the user can include or suppress information such as the chemical structures of substrates and the genes associated with each enzyme. The pathway drawing tool can depict feedback inhibition within a pathway, and the chromosomal locations of all genes that code for enzymes within a pathway. For sample displays see URL http://biocyc.org:1555/ECOLI/new-image?type=PATHWAY&object=COMPLETE-ARO-PWY.

The **overview visualization** shows the entire metabolic map for an organism in a single screen. For an example display see URL http://biocyc.org:1555/CAULO/new-image?type=OVERVIEW. This tool provides many different operations on the entire metabolic map of an organism, such as:

- Find and highlight individual entities by name, such as metabolites, enzymes, or pathways
- Species comparisons, such as: Highlight on the diagram those reaction steps that are shared with one or more of the other organisms for which PGDBs are present
- Highlighting of reaction steps where the enzyme is subject to substrate-level regulation by a given ligand
- Highlighting of all reaction steps where a particular transcription factor controls the expression of the genes coding for the enzymes that catalyse those reactions
- Pathway-based visualization of gene-expression data sets—each reaction is coloured according to the expression levels of the underlying genes (Karp *et al.*, 1999), using a user-defined colour scheme.

The **reaction visualization** displays a biochemical transformation, including enzymatic, transport, and signal-transduction reactions. The **enzyme visualization** shows the subunit structure, cofactors, activators, and inhibitors of the enzyme. It also depicts the reactions that the enzyme catalyses, and lists the pathways in which those reactions are involved. For an example display see URL http://biocyc.org:1555/ECOLI/new-image?type=ENZYME&object=PRAI-IGPS.

The **gene visualization** lists the nucleotide position of the gene, the chromosome on which the gene is present, the gene product, and the pathway(s) in which that product is involved. The nucleotide sequence of the gene and the amino-acid sequence of its product may be retrieved in one mouse click. A schematic diagram of the operon containing the gene is displayed.

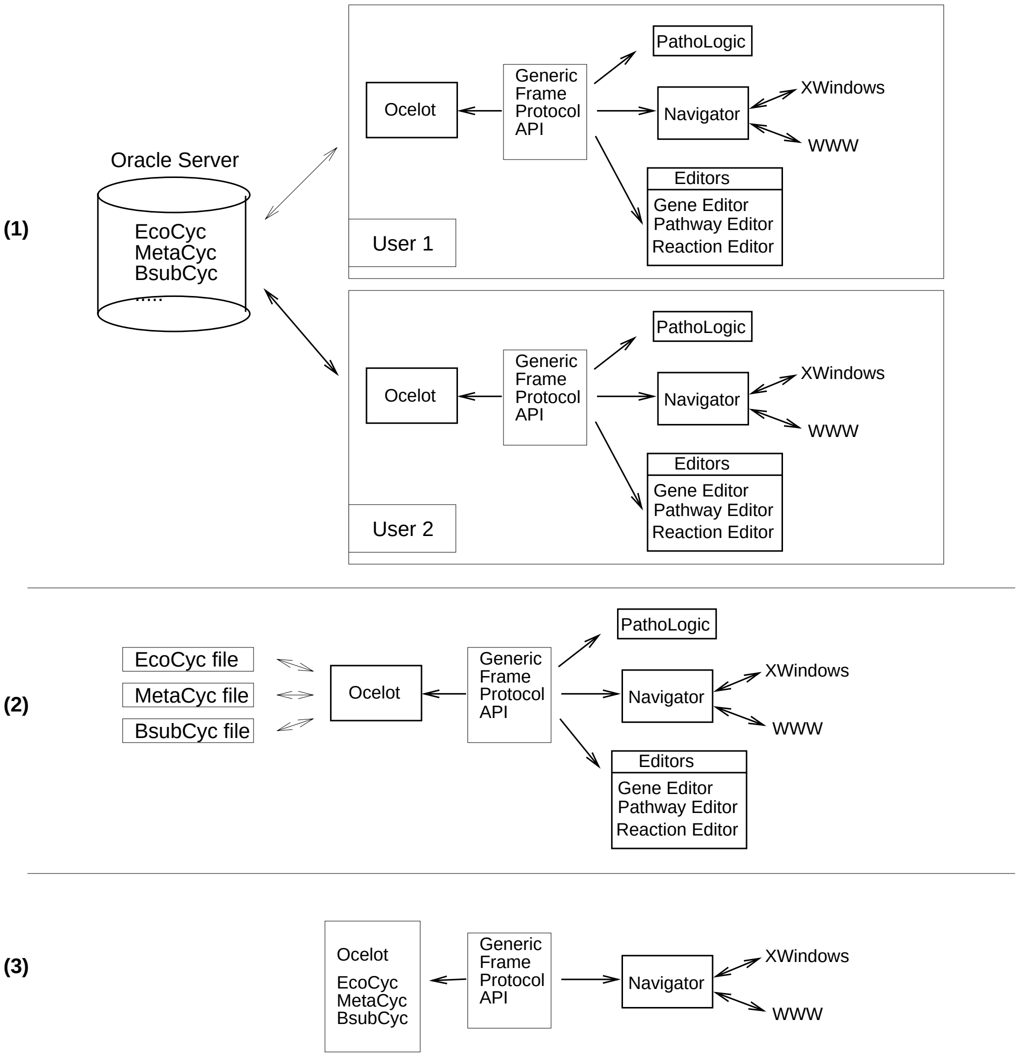

Fig. 1. The alternative software architectures of the Pathway Tools. Configuration (1) shows two Pathway Tools users accessing the same PGDBs in a single Oracle server, via a local network or the Internet. Configuration (2) shows a Pathway Tools configuration in which a user accesses PGDBs in disk files. Configuration (3) shows PGDBs stored in the binary program file for the Pathway Tools.

The **chromosome viewer** allows browsing of chromosomes at multiple levels of resolution, showing positions of coding regions and gene names. Sequence regions such as promoters, transcription-factor binding sites, genes, and transcription terminators are displayed.

The Web version of the Navigator allows users to submit protein or nucleic-acid sequences for Blast search against the genome of an organism in a PGDB. The Pathway Tools post-processes the Blast results to contain links to the PGDB entries for genes that were matched in the Blast search.

Other information present in each of the preceding displays includes object names, synonyms, comments, citations, histories, and links to other Web-accessible DBs. Every display lists objects that are related to the entity being visualized (e.g. a reaction display lists the substrates of the reaction and the enzyme(s) that catalyse it); if the user clicks on a related object, the visualization for that object is displayed.

Queries supported by the Navigator include (1) Query objects by exact or substring match to a synonym for the object name; (2) Query genes, reactions, pathways, and compounds using stored classification hierarchies; and (3) Show the position of a gene on the genomic map.

PATHOLOGIC PATHWAY PREDICTOR

The PathoLogic program creates a new PGDB from an input file that describes the annotated genome of an organism S. PathoLogic performs both a conversion process and an inference process. The conversion process transforms flatfile descriptions of genes and gene products into a PGDB representation of that information. The conversion process is itself of value because it transforms the genome annotation into a form that can be queried and displayed using the Navigator, and that can be further updated using the Pathway/Genome Editors. The PathoLogic inference process predicts the metabolic-pathway complement of S from its genome by comparison to the MetaCyc pathway DB (Karp *et al.*, 2002). The input to PathoLogic can be in GenBank format, with one input file per chromosome or plasmid.

The MetaCyc DB contains more than 450 pathways from all domains of life. For each pathway P in MetaCyc, PathoLogic evaluates the evidence that P occurs in organism S by computing how many enzymes in P are present in S, based on the existing set of functional annotations for S. The algorithmic evaluation of pathway evidence differentiates enzymes that are unique signatures for a given pathway from enzymes that are used in multiple pathways and thereby provide weaker evidence for the presence of the pathway. PathoLogic generates a set of HTML reports that outline the evidence for the presence of MetaCyc pathways in S in which pathways are grouped by functional pathway categories derived from the Pathway Tools ontology, such as amino-acid biosynthesis and energy metabolism—see URL http://biocyc.org:1555/PSEUDO/pwys.html for an example.

PathoLogic provides a rapid means of creating a PGDB whose contents can be adjusted manually later as new information about the genome comes to light. The algorithm used by the program is as follows (see (Paley and Karp, 2002) for more details).

Inputs to PathoLogic:

- F = A list of input files, one per replicon (chromosome or plasmid) in the organism S. Each file in F contains the following information for each gene on that replicon, when known:
  - The start and end position of the gene on the chromosome
  - The gene name
  - The type of the gene product (e.g. rRNA or protein)
  - The gene product (the name of the predicted function(s) for the gene product)
  - The EC number(s) for the reaction(s) catalysed by the gene product, for those gene products that are enzymes
- The MetaCyc pathway DB

Major steps in PathoLogic processing:

1. Initialize the schema of the new PGDB by replicating the MetaCyc schema.
2. Create a PGDB object for each replicon in the set F. Then create a PGDB object for each gene in each replicon, and create a PGDB object for the product of each gene.
3. Determine the reaction catalysed (if any) by each gene product in the organism. If an EC number was assigned to a gene product, then rely on the EC number to identify the reaction. Otherwise, match the function name (product name) provided for each gene against an extensive dictionary of enzyme names that is available to PathoLogic.
4. Match the list of reactions now known to be catalysed by the organism from step (3) against all pathways from MetaCyc, and import into the new PGDB those MetaCyc pathways, reactions, and substrates for which significant evidence exists (Paley and Karp, 2002).

PathoLogic is configured to err on the side of more liberal pathway predictions on the assumption that it is better to bring questionable pathway predictions to the attention of the scientist user for further study than to omit those pathways from consideration.

SRI has applied PathoLogic to 11 microbial genomes. The resulting PGDBs are available through the SRI web site at URL http://biocyc.org. The number of pathways predicted by PathoLogic in these organisms ranges from a low of 38 pathways for *M. pneumoniae* to a high of 156 pathways for *Agrobacterium tumefaciens*. It typically requires about two weeks of effort to create a new PGDB. The automated phases of PathoLogic processing require only a few hours. The most time consuming parts of the process are usually preparing an input file that correctly follows one of the two formats required by PathoLogic (such as Genbank format), and performing literature research on those enzymes whose names are unrecognized by PathoLogic to determine what biochemical reactions they carry out. The accuracy of PathoLogic pathway predictions are assessed in (Paley and Karp, 2002).

PATHWAY / GENOME EDITORS

The purpose of a PGDB is to accurately model the biochemical functions of all known gene products of the organism, and the organization of those functions into larger pathways. New information about the organism modelled in a PGDB that comes to light from both single-gene studies, and from functional-genomics studies of the organism, will require updates to the PGDB to reflect additional functions for some genes, changes to the functions in other genes, and pathway additions, pathway deletions, and modifications to pathway structures.

The Pathway/Genome Editors consist of a set of interactive editing tools for PGDB curation. The Editors support creation of new PGDB objects and modification of existing PGDB objects. These tools can be invoked quickly through a single mouse operation from the Navigator when the user sees, within the Navigator, a PGDB object that needs to be updated. When the user exits from the editing tool, the modified version of the object is then displayed within the Navigator so that the user can verify the update.

The Editors are in daily use by members of the EcoCyc project at SRI International, The Institute for Genomic Research (TIGR), UC San Diego (UCSD), and the National Autonomous University of Mexico (UNAM). Biologists typically become proficient at using these tools after a day of training and several days of hands-on experience.

The Editors provided in the Pathway Tools are a gene editor, a protein editor, a transcription-unit (operon) editor, a reaction editor, a chemical-compound editor, and a pathway editor.

THE PATHWAY TOOLS ONTOLOGY

The power of a MOD depends strongly on its ontology. A MOD cannot store information in a queryable form that its schema cannot accommodate, and if its schema is poorly designed, a MOD will distort information. The Pathway Tools ontology is designed to capture many genomics datatypes, and it has been honed during the process of entering thousands of objects into the EcoCyc and MetaCyc databases. When we encounter a biological situation that the ontology cannot accurately capture, we ask whether the ontology should be extended to capture that situation. In most cases the ontology is extended, although in some cases that occur infrequently we prefer to not extend the ontology so as to limit its complexity. Because the ontology is described in more detail in (Karp, 2000), this section outlines only its salient features.

Classes in the Pathway Tools ontology are arranged in a hierarchy that reflects their generalization–specialization relationships. The following classes define the following types of bioinformatics data. Genomics data are defined by classes such as `Chromosomes`, `Plasmids`, and `Genes`. Gene products are defined by classes such as `Polypeptides`, `Protein-Complexes`, `rRNAs`, and `tRNAs`. Metabolites are defined by the class `Chemicals`. Genetic regulatory sites are defined by classes such as `Promoters`, `DNA-Binding-Sites`, and `Terminators`. Molecular interactions are defined by the classes `Reactions` and `Pathways`.

The ontology associates, with each class, slots that allow us to encode attributes of a biological object, and relationships among those objects. For example, slots on class `Protein-Complexes` allow us to specify the molecular weight and subunits of a protein complex, and to link the complex to enzymatic, transport, or DNA-binding reactions that it may catalyse or for which it may serve as a substrate. The class Reactions has slots that specify the objects that are reactants and products of a reaction. We can describe a large range of biological interactions by allowing those reactants and products to be small-molecule metabolites, proteins, tRNAs, and DNA binding sites. The ontology also allows us to tag reaction substrates with information about their cellular location, which permits us to describe transport events that move substrates from one cellular compartment to another. Other slots of class `Reactions` define the EC number, pathways in which the reaction is a member, whether the reaction occurs spontaneously, the enzyme(s) that catalyse the reaction, and the equilibrium constant of the reaction.

THE OCELOT OBJECT DATABASE

Ocelot (Karp *et al.*, 1999) is an object database that has been used within the EcoCyc project for over 7 years. It is a mature and stable database manager. Ocelot combines the expressive power of frame knowledge representation systems (Karp, 1992) developed within the artificial intelligence (AI) community (whose object data model is far superior to the relational data model for representing biological data) with the scalability, multiuser access capabilities, and robust operation of relational database systems.

Persistent storage for Ocelot DBs can be provided either by an Oracle DB, or by disk files (see Figure 1). The latter configuration provides an easy and low-cost way to begin a PGDB project; it requires neither purchase of an Oracle license, nor installation of Oracle. A project can switch to the Oracle configuration as its complexity grows. Ocelot DB files also provide a means of exchanging entire PGDBs among different organizations.

The Oracle configuration provides Ocelot with multiuser update capabilities for distributed DB development, and it permits incremental (faster) saving of DB updates. The Oracle configuration also allows Ocelot to maintain a history of all DB transactions—DB curators can examine the history of all updates to an object to determine when a change was made, and by whom. This feature helps curators diagnose mistakes within a DB.

In the EcoCyc project a geographically distributed set of collaborators uses Ocelot in its Oracle configuration; however, we found that querying Oracle across transcontinental Internet connections could be quite slow. Therefore, we added a disk-caching facility to Ocelot: all PGDB objects retrieved from Oracle are also saved in a local disk file, and that file is searched before queries are issued to Oracle, resulting in terrific speedups. The disk cache is also appropriately purged of objects that have been updated in Oracle by other users.

Ocelot DBs can be queried using short Lisp programs that employ the Ocelot API, which is called GFP (Generic Frame Protocol) (Karp and Gruber, 1995). Learning the amount of Lisp required to write GFP-based queries is no more difficult than is learning SQL—we have trained several biologists in 2 to 3 days to write GFP queries.

PERFORMANCE

Some bioinformaticians have criticized object DBMSs as having insufficient performance for their applications. Although this may be true for some applications, we have found the performance of Ocelot to be more than adequate. The SRI Web site at biocyc.org contains EcoCyc, MetaCyc, and 11 other PGDBs, with each DB containing approximately 10 000–20 000 objects. That Web site is driven by a single-processor 500 MHz SunBlade–100 with 640 MB of memory, which can answer most queries (such as displaying a metabolic pathway or reaction) in under 1 sec, demonstrating that the software performs well on a relatively small machine. We see no obstacles to expanding significantly both the size of individual DBs, and the number of DBs that are active simultaneously.

To give a sense of how the configurations in Figure 1 affect the performance of complex queries that compute across a DB, consider a query that searches across all 2700 reaction objects in EcoCyc. Under configuration (1) and with the disk cache disabled, that query takes 10 sec when the Oracle server is accessible via a local-area network. With disk cache enabled the query takes 7 sec (the disk cache not only increases performance across the Internet, it also increases performance across a local-area network). Under configuration (3) that query takes 750 msec. Executing that query (or any other query against reactions) a second time under configuration (1) takes 750 msec because all data are now cached in the Ocelot address space.

COMPARISON TO RELATED WORK

Many software tools have been developed for publishing genomes on the Web and share some features of the Pathway/Genome Navigator (Mural *et al.*, 1999; Ball *et al.*, 2000; Stein *et al.*, 2001; Blake *et al.*, 2001; Peterson *et al.*, 2001). However, most are not production-grade software in the sense that they have never been installed at other sites, and do not provide detailed documentation for database developers at other sites. These tools typically do not provide an assortment of editing tools, do not have an analog of PathoLogic for metabolic pathway prediction, do not have a rich ontology, and do not support the wide range of bioinformatics datatypes that the Pathway Tools supports. For example, most of these tools provide no support for metabolic or genetic networks.

KEGG (Kanehisa and Goto, 2000) provides some limited support for metabolic networks. KEGG pathways are hand-drawn images—KEGG does not provide automated pathway drawing software, nor software for editing the database definition of a KEGG pathway. The KEGG ontology is quite limited; it encodes the relationships among a gene, an enzyme it encodes, a reaction the enzyme catalyses, and the substrates and pathways of a reaction. No other information is provided in KEGG flatfiles about these entities, and it is not clear if any DBMS underlies KEGG. Thus, KEGG does not provide tools with which users can customize pathways or other data to create a MOD.

AceDB (Thierry-Mieg *et al.*, 1999) was the first reusable software for building MODs; however, many projects that were using AceDB have abandoned it. Among its shortcomings were that it used a home-brewed database manager that allowed write access to a DB by only one user at a time, and its DBMS is tightly intertwined with its graphical interface. AceDB also lacks interactive editing tools, and an analog to PathoLogic.

SUMMARY

The Pathway Tools software supports creation of new PGDBs using its PathoLogic component. The Pathway/Genome Editors support interactive refinement of PGDBs by MOD curators. The Pathway/Genome Navigator provides Web publishing of PGDBs, and supports many forms of interactive queries, visualization, and analysis. The Pathway Tools have been used at SRI to construct 13 different PGDBs; construction of another 8–10 databases is under way at SRI and by other scientists.

ACKNOWLEDGEMENTS

This work was supported by grant 1-R01-RR07861-01 from the National Center for Research Resources, by grant DE-FG03-01ER6321 from the Department of Energy, and by DARPA contract N66001-01-C-8011.

REFERENCES

Ball,C.A., Dolinski,K., Dwight,S.S., Harris,M.A., Issel-Tarver,L., Kasarskis,A., Scafe,C.R., Sherlock,G., Binkley,G., Jin,H., Kaloper,M., Orr,S.D., Schroeder,M., Weng,S., Zhu,Y., Botstein,D. and Cherry,J.M. (2000) Integrating functional genomic information into the *Saccharomyces* Genome Database. *Nucleic Acids Res.*, **28**, 77–80.

Blake,J.A., Eppig,J.T., Richardson,J.E., Bult,C.J. and Kadin,J.A. (2001) The Mouse Genome Database (MGD): Integration nexus for the laboratory mouse. *Nucleic Acids Res.*, **29**, 91–94.

Flybase Consortium (1998) FlyBase: a *Drosophila* database. *Nucleic Acids Res.*, **26**, 85–88.

Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**, 27–30.

Karp,P. and Gruber,T. (1995) The Generic Frame Protocol. Available via World Wide Web URL http://www.ai.sri.com/∼gfp/spec/paper/paper.html.

Karp,P., Krummenacker,M., Paley,S. and Wagg,J. (1999) Integrated pathway/genome databases and their role in drug discovery. *Trends Biotechnol.*, **17**, 275–281.

Karp,P.D. (2000) An ontology for biological function based on molecular interactions. *Bioinformatics*, **16**, 269–285.

Karp,P.D. (1992) The design space of frame knowledge representation systems. *Technical Report 520*. SRI International AI Center, URL ftp://www.ai.sri.com/pub/papers/karp-freview.ps.Z

Karp,P.D. (2001) Pathway databases: A case study in computational symbolic theories. *Science*, **293**, 2040–2044.

Karp,P.D., Chaudhri,V.K. and Paley,S.M. (1999) A collaborative environment for authoring large knowledge bases. *J. Intelligent Information Systems*, **13**, 155–194.

Karp,P.D., Riley,M., Paley,S. and Pellegrini-Toole,A. (2002) The MetaCyc database. *Nucleic Acids Res.*, **30**, 59–61.

Karp,P.D., Riley,M., Saier,M., Paulsen,I.T., Paley,S. and Pellegrini-Toole,A. (2002) The EcoCyc database. *Nucleic Acids Res.*, **30**, 56–58.

Mural,R.J., Parang,M., Shah,M., Snoddy,J. and Uberbacher,E.C. (1999) The Genome Channel: A browser to a uniform first-pass annotation of genomic dna. *Trends Genet.*, **15**, 38–39.

Ouzounis,C. and Karp,P.D. (2000) Global properties of the metabolic map of *Escherichia coli*. *Genome Res.*, **10**, 568.

Paley,S. and Karp,P.D. (2002) Evaluation of computational metabolic-pathway predictions for *H. pylori*. in press.

Peterson,J.D., Umayam,L.A., Dickinson,T., Hickey,E.K. and White,O. (2001) The Comprehensive Microbial Resource. *Nucleic Acids Res.*, **29**, 123–125.

Stein,L., Sternberg,P., Durbin,R., Thierry-Mieg,J. and Spieth,J. (2001) WormBase: Network access to the genome and biology of *Caenorhabditis elegans*. *Nucleic Acids Res.*, **29**, 82–86.

Thierry-Mieg,J., Thierry-Mieg,D. and Stein,L. (1999) ACEDB: The ACE Database Manager. *Bioinformatics Databases and Systems*. Kluwer Academic Publishers, Norwell, MA, pp. 265–278.