BRIEFINGS IN BIOINFORMATICS. VOL 11. NO 1. 40–79
Advance Access published on 2 December 2009

doi:10.1093/bib/bbp043

Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology

Peter D. Karp, Suzanne M. Paley, Markus Krummenacker, Mario Latendresse, Joseph M. Dale, Thomas J. Lee, Pallavi Kaipa, Fred Gilham, Aaron Spaulding, Liviu Popescu, Tomer Altman, Ian Paulsen, Ingrid M. Keseler and Ron Caspi

Submitted: 8th June 2009; Received (in revised form): 25th August 2009

Abstract

Pathway Tools is a production-quality software environment for creating a type of model-organism database called a Pathway/Genome Database (PGDB). A PGDB such as EcoCyc integrates the evolving understanding of the genes, proteins, metabolic network and regulatory network of an organism. This article provides an overview of Pathway Tools capabilities. The software performs multiple computational inferences including prediction of metabolic pathways, prediction of metabolic pathway hole fillers and prediction of operons. It enables interactive editing of PGDBs by DB curators. It supports web publishing of PGDBs, and provides a large number of query and visualization tools. The software also supports comparative analyses of PGDBs, and provides several systems biology analyses of PGDBs including reachability analysis of metabolic networks, and interactive tracing of metabolites through a metabolic network. More than 800 PGDBs have been created using Pathway Tools by scientists around the world, many of which are curated DBs for important model organisms. Those PGDBs can be exchanged using a peer-to-peer DB sharing system called the PGDB Registry.

Keywords: *Genome informatics; Metabolic pathways; Pathway bioinformatics; Model organism databases; Genome databases; Biological networks; Regulatory networks*

INTRODUCTION

Pathway Tools [1–3] is a software environment for management, analysis and visualization of integrated collections of genome, pathway and regulatory data. Pathway Tools handles many types of information beyond pathways, and its capabilities are very

Corresponding author. Peter D. Karp. Bioinformatics Research Group, Artificial Intelligence Center, SRI International, 333 Ravenswood Ave, AE206, Menlo Park, CA 94025, USA. Tel: 650 859 4358. Fax: 650 859 3735 E-mail: pkarp@ai.sri.com

Peter D. Karp is the director of the Bioinformatics Research Group at SRI International. He received the PhD degree in Computer Science from Stanford University.

Suzanne Paley is a computer scientist in the Bioinformatics Research Group at SRI International.

Markus Krummenacker is a scientific programmer in the Bioinformatics Research Group at SRI International. He has worked on Pathway Tools for over 8 years and has interests ranging from computers to molecular life sciences and nanotechnology.

Mario Latendresse is a Computer Scientist at SRI International. He received the PhD degree in Computer Science from Université de Montréal.

Joseph M. Dale is a computer scientist in the Bioinformatics Research Group at SRI International.

Thomas J. Lee is a Senior Research Engineer at SRI International. He received his MS degree in Computer Science from the University of Wisconsin (Madison).

Pallavi Kaipa is a scientific programmer in the Bioinformatics Research Group at SRI International.

Fred Gilham has been a Software Engineer at SRI International for 20 years. He received the Masters Degree in Computer Science from Stanford University.

Aaron Spaulding is a computer scientist in the Artificial Intelligence Center at SRI International.

Ian T. Paulsen is Professor in Genomics at Macquarie University, Sydney, Australia. He received his PhD in Microbiology from Monash University, Melbourne, Australia.

Ingrid M. Keseler is a Scientific Database Curator in the Bioinformatics Research Group at SRI International. She received an M.S. degree in Microbiology from the University of Georgia and a Ph.D. in Biochemistry from Stanford University.

Ron Caspi is a curator of the MetaCyc database. He received his PhD degree in Marine Biology from the Scripps Institution of Oceanography, UC San Diego.

extensive. The software has been under continuous development within the Bioinformatics Research Group within SRI International since the early 1990s. Pathway Tools serves following several different use cases in bioinformatics and systems biology:

- It supports development of organism-specific databases (DBs) [also called model-organism databases (MODs)] that integrate many bioinformatics datatypes.
- It supports scientific visualization, web publishing and dissemination of those organism-specific DBs.
- It performs computational inferences including prediction of metabolic pathways, prediction of metabolic pathway hole fillers and prediction of operons, which can be used for genome analysis.
- It provides visual tools for analysis of omics datasets.
- It provides tools for analysis of biological networks.
- It provides comparative analyses of organism-specific DBs.
- It supports metabolic engineering.

This article provides a comprehensive description of Pathway Tools. It describes both what the software does, and how it does it. Where possible it references earlier publications that provide more algorithmic details. However, in some cases those earlier publications are outdated by new developments in the software that are described here. This article also emphasizes new aspects of the software that have not been reported in earlier publications.

Pathway Tools is focused around a type of MOD called a Pathway/Genome Database (PGDB). A PGDB integrates information about the genes, proteins, metabolic network and regulatory network of an organism.

Pathway Tools has several components. The PathoLogic component allows users to create a new PGDB from the annotated genome of an organism. PathoLogic generates a new PGDB that contains the genes, proteins, biochemical reactions and predicted metabolic pathways and operons of the organism.

The Pathway/Genome Editors let PGDB developers interactively refine the contents of a PGDB, such as editing a metabolic pathway or an operon, or defining the function of a newly characterized gene.

The Pathway/Genome Navigator supports querying, visualization and analysis of PGDBs. The Navigator can run as a local desktop application and as a web server. The Navigator allows scientists to find information quickly, to display that information in familiar graphical forms and to publish a PGDB to the scientific community via the web. The Navigator provides a platform for systems-level analysis of functional-genomics data by providing tools for painting combinations of gene expression, protein expression and metabolomics data onto a full metabolic map of the cell, onto the full genome, and onto a diagram of the regulatory network of the cell.

Pathway Tools includes a sophisticated ontology and DB application programming interface (API) that allows 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 *Escherichia coli* metabolic network [4] and genetic network [5].

Pathway Tools is seeing widespread use across the bioinformatics community to create PGDBs in all domains of life. The software has been licensed by more than 1700 users to date. As well as supporting the development of the EcoCyc [6] and MetaCyc [7] DBs at SRI, and SRI's BioCyc collection of 500 PGDBs [7], the software is in use by genome centers, by experimental biologists, and by groups that are creating curated MODs for bacteria (such as the National Institute of Allergy and Infectious Diseases Bioinformatics Resource Centers PATRIC, BioHealthBase, Pathema and EuPathDB), for fungi (such as the *Saccharomyces* Genome Database and the *Candida* Genome Database), mammals (such as the Jackson Laboratory's MouseCyc) and for plants (such as *Arabidopsis thaliana*). See Section 9 for a more detailed listing of available PGDBs.

The organization of this article is as follows. Section 'Pathway Tools use cases' articulates in more detail the use cases for which Pathway Tools was designed. 'Creating and curating a PGDB' section relates how a new PGDB is created, and describes the computational inference procedures within Pathway Tools. It summarizes the interactive editing capabilities of Pathway Tools, and the associated author crediting system. It also describes tools for automatic upgrading of a PGDB schema, and for bulk updating of the genome annotation within a PGDB. 'The pathway Tools schema' section describes the schema of a PGDB. 'Visualization and querying of PGDBs' section relates the querying and visualization facilities of Pathway Tools.

'Computational access to PGDBs' section summarizes the mechanisms for importing and exporting data from Pathway Tools, and for accessing and updating PGDB data via APIs. 'Systems biology analyses' section describes multiple Pathway Tools modules for performing systems analyses of PGDBs including a tool for interactively tracing metabolites through the metabolic network, tools for performing network reachability analysis and for identifying dead-end metabolites, a tool for predicting antimicrobial drug targets by identifying metabolic network choke points and a set of comparative analysis tools. 'Software and DB architecture' section describes the software architecture of Pathway Tools. 'Survey of pathway tools compatible DBs' section lists the large family of PGDBs that have been created by Pathway Tools users outside SRI International, and describes a peer-to-peer data sharing facility within Pathway Tools that allows users to easily exchange their PGDBs. 'Comparison with related software environments' section compares Pathway Tools to related efforts.

PATHWAY TOOLS USE CASES

This section articulates the objectives for which Pathway Tools was designed. Please note that when we assert that Pathway Tools supports a given type of use case, it does not mean that Pathway Tools provides every type of computational tool needed in that area. For example, omics data analysis is a huge field, and although Pathway Tools contributes novel and useful omics data analysis capabilities, it does not provide every omics data analysis method, and in fact it is intended to be used in conjunction with other omics analysis tools (such as for data normalization). Similarly, although Pathway Tools can contribute a number of useful capabilities to metabolic engineering, by no means does it solve every problem in metabolic engineering. 'Limitations and future work' section summarizes the limitations of Pathway Tools.

Development of organism-specific DBs

Organism-specific DBs (also known as MODs) describe the genome and other information about an organism [8–19]. We posit that every organism with a completely sequenced genome and an experimental community of significant size requires an organism-specific DB to fully exploit the genome sequence. Such DBs should provide a central information resource about the genome, molecular parts and cellular networks of the organism, and as such they must be able to capture a diverse range of information types. A critical role of organism-specific DBs is to integrate information that is scattered across the biomedical literature, both to assemble as a complete picture of the organism as possible, and to permit cross-checking and validation of isolated pieces of information. These DBs should both direct and accelerate further scientific investigations.

Pathway Tools facilitates rapid initial computational construction of organism-specific DBs, followed by manual refinement of the PGDB, to produce an extremely rich and accurate DB in minimal time. Our approach tracks experimental versus computationally inferred information whenever possible. Rapid construction of PGDBs is achieved by importing an annotated genome into a PGDB in the form of a Genbank file, and by applying several computational inference tools to infer new information within the PGDB, such as metabolic pathways. Scientists can then employ the Pathway/Genome Editors to correct and supplement computational inferences when necessary, and to perform ongoing manual curation of the PGDB if desired. Further details of our approach can be found in 'Creating and curating a PGDB' section.

The Pathway Tools DB schema (for definition see 'The Pathway Tools schema' section) is significant in both its breadth and its depth: It models an unusually broad set of bioinformatics datatypes ranging from genomes to pathways to regulatory networks, and it provides high-fidelity representations of those datatypes that allow PGDBs to accurately capture complex biology.

We note that Pathway Tools can also be used to construct metabolic DBs such as MetaCyc and PlantCyc [7, 20] that are not organism specific.

Visualization and web publishing of organism-specific DBs

To speed user comprehension of the complex information within PGDBs, the Pathway/Genome Navigator provides many scientific visualization services including a genome browser, visualization of single metabolic pathways and entire metabolic maps, visualization of single operons and of entire regulatory networks and visualization of chemical compounds and reactions (see 'Visualization and querying of PGDBs' section for more details). These visualization tools operate within a web server, permitting developers of PGDBs to publish their PGDBs to the scientific community through a web site. This form of PGDB publishing supports interactive querying and browsing by individual scientists using a three-tiered series of web query interfaces (see 'Query tools' section) including a quick search, a set of object-specific query tools and a tool for interactively constructing queries whose power is comparable with that of SQL.

We have developed other publishing paradigms to support computational analysis and dissemination of PGDBs. Pathway Tools APIs exist in three languages [21]. Web services access to PGDBs is under development now. PGDBs can be exported in several formats and imported into the BioWarehouse DB integration system [22]. Finally, users can easily share and exchange PGDBs using a peer-to-peer DB sharing system that we have developed.

Extend genome annotations with additional computational inferences

Pathway Tools extends the paradigm of genome analysis. After traditional analyses such as gene calling and gene function, predictions, are performed by external software packages; Pathway Tools provides additional computational genome analyses that layer additional information above the traditional genome annotation. Pathway Tools predicts the operons of the organism. It predicts the metabolic pathways of the organism. It also predicts which genes in the organism code for missing enzymes in the predicted metabolic pathways, thus using pathway information to predict additional gene functions. See 'Creating and curating a PGDB' section for more details.

Analysis of omics data

Pathway Tools was the first software system to provide pathway-based analysis of gene expression data [23]. Pathway Tools provides three genome-scale viewers for animated visualization of omics datasets in the context of the full metabolic network [3], full transcriptional regulatory network and full genome (see 'System-level visualization of metabolic networks, system-level visualization of regulatory networks and system-level visualization of genome maps' sections for more details).

Symbolic systems biology workbench

As well as serving as an online reference for researchers, a PGDB is a computational model of the organism. It must be possible to employ such models to test and extend our understanding of the organism, by checking models for internal consistency, and for their consistency with experimental data. Both this use case and the next are concerned with PGDBs as models.

A symbolic systems biology workbench supports users in developing global analyses of a biological system that are symbolic (qualitative) in nature. 'Symbolic computing is concerned with the representation and manipulation of information in symbolic form. It is often contrasted with numeric representation' [24]. General examples of symbolic computation include string matching for DNA and protein sequences, symbolic algebra programs (e.g. Mathematica and the Graphing Calculator), compilers and interpreters for programming languages, DB query languages, web crawlers and many Artificial Intelligence (AI) algorithms (e.g. expert systems and symbolic logic).

A strong motivation for applying symbolic computing techniques in systems biology is that these techniques can provide insight in areas where numerical techniques fail because of the unavailability of quantitative system parameters.

An example symbolic systems biology problem is: Let *C* be the set of carbohydrates for which *E. coli* has transporters. Which members of *C* are not the inputs to a degradative metabolic pathway in *E. coli*? Which members of *C* are not consumed by any metabolic reaction in *E. coli*? These questions are of interest because they may indicate incomplete or incorrect knowledge of a cell's transport and metabolic networks. Pathway Tools assists users in answering queries of this sort by providing a rich schema (ontology) for PGDBs that makes a wide array of information accessible for computational analysis, and by providing a rich library of Lisp functions (callable through the Lisp, Perl and Java APIs) for computing symbolic relationships among information in a PGDB, such as for retrieving all transported substrates in the cell.

Analysis of biological networks

Pathway Tools includes programs for symbolic analysis of biological networks (see 'Systems biology analyses' section for more details) that rely on the detailed biological network ontology underlying

Pathway Tools. That ontology provides high-fidelity representations of a wide range of metabolic and regulatory interactions (see 'Metabolites, reactions and pathways' and 'Pathway Tool regulation ontology' sections).

Two tools perform consistency checking of metabolic networks: (i) The software identifies dead-end metabolites, which are metabolites that are only synthesized by the metabolic network, or only consumed by the metabolic network, and are not transported into or out of the cell. Although occasionally dead-end metabolites are biologically valid, usually they reflect errors or incompleteness of our knowledge of a metabolic network. (ii) The software performs forward qualitative propagation of metabolites through the metabolic network [25], which we call *reachability analysis*. The intuition here is that by tracing the paths of metabolites from a known growth medium through the metabolic network, we should be able to reach essential compounds that the cell must be producing from those starting metabolites. Failure to reach those essential compounds (such as the amino acids and cell wall components) usually indicates gaps in the metabolic network model and indicates the need for further model curation or basic research.

Pathway Tools indirectly supports a two-phased pathway-based paradigm for drug discovery. Phase I is the search for essential *in vivo* metabolic pathways: pathways whose function is essential for microbial growth in the host. Phase II is the search for targets within essential *in vivo* pathways. Both phases are supported by a Pathway Tools module that predicts choke point reactions within the metabolic network as likely drug targets [26].

Comparative analyses of organism-specific DBs

Pathway Tools provides a suite of comparative analysis operations that can be applied to multiple user-selected PGDBs (see 'Comparative tools' section for more details). Pathway Tools emphasizes comparisons at the functional level, rather than the sequence level. Example comparisons include (i) highlighting on the cellular overview of one organism the reactions that it shares (or does not share) with one or more other organisms; (ii) a tabular comparison of the reaction complements of several organisms, organized by substrate type (e.g. small molecules, RNAs, and proteins) or by number of isozymes per reaction; (iii) a comparison of the pathway complements of several organisms, where the tabular pathway comparison is organized by a pathway ontology; (iv) a table showing which genes have orthologs in which PGDBs; and (v) a comparison of the genome organization of orthologs using the genome browser.

Metabolic engineering

Metabolic engineering is a discipline that seeks to modify the metabolic network of an organism in a desired fashion, such as to achieve overproduction of desired end products, or degradation of specified compounds [27]. Pathway Tools is designed to assist metabolic engineers in several respects. Its inference capabilities aid in rapid characterization of a host organism for metabolic engineering. Its editing tools permit refinement of that metabolic model. Its omics analysis capabilities aid metabolic engineers in understanding the activity levels of different portions of the metabolic network under different growth conditions. It provides a tool for tracing metabolites forward and backward through the metabolic network to aid understanding of the metabolic fate of specific molecules (see 'Metabolite tracing' section for more details).

CREATING AND CURATING A PGDB

The life cycle of a PGDB typically includes the following three types of procedures.

(1) Initial creation of the PGDB: PGDB creation starts with one or more input files describing the functionally annotated genome of an organism. The PathoLogic component of Pathway Tools transforms the genome into an Ocelot [28] DB structured according to the Pathway Tools schema. Next the user applies one or more computational inference tools within PathoLogic to the genome to infer new information such as metabolic pathways. For several of the PathoLogic inference tools, we have created graphical user interfaces that allow the user to review the inferences made by these tools, and to accept, reject or modify those inferences.
(2) PGDB curation: Manual refinement and updating of a PGDB is performed using the Pathway/Genome Editors. This phase can last for years, or for decades, as in the case of EcoCyc [8]. Curation can be based on information found

about the organism in the experimental literature, on information from in-house experiments or on information inferred by the curator, perhaps with help from other computational tools. PGDB curation is multidimensional [8], involving addition and/or deletion of genes or metabolic pathways to/from the PGDB; changing gene functions; altering the structure of metabolic pathways; authoring of summary comments for genes or pathways; attachment of MultiFun or Gene Ontology (GO) terms to genes and gene products; entry of chemical structures for small molecules; defining regulatory relationships; and entry of data into many different PGDB fields including protein molecular weights, pIs and cellular locations.

(3) Bulk updating of a PGDB: A PGDB developer might run an external program that predicts cellular locations for hundreds of genes within the genome, and want to load those predictions into the PGDB. Or, although most users of Pathway Tools keep their authoritative genome annotation within the PGDB, some groups store the authoritative genome annotation in another genome data management system, and want to periodically import the latest genome annotation into Pathway Tools. Another type of bulk PGDB update is that applied by the Pathway Tools consistency checker, which scans a PGDB for noncompliant data (for example, see 'Consistency checker and aggregate statistics' section), and either repairs the problem automatically, or notifies the user of problems. In addition, most of the individual components within PathoLogic that were used to initially create a PGDB can be run again at a later date to take advantage of updated information.

The following subsections describe the Pathway Tools components for addressing these procedures.

PathoLogic PGDB creation

PathoLogic performs a series of computational inferences that are summarized in Figure 1. These inferences can be performed in an interactive mode, in which the user guides the system through each step, and can review and modify the inferences made by the system using interactive tools. PathoLogic can also execute in a batch mode in which all processing is automated. In batch mode, PathoLogic can process hundreds of genomes.

The input to PathoLogic is the annotated genome of an organism. PathoLogic does not perform genome annotation; its input must supply the genome sequence, the locations of genes and identified functions of gene products. The sequence is supplied as a set of FASTA-format files, one per replicon. The annotation is supplied as a set of files in Genbank format or PathoLogic format, each of which describes the annotation of one

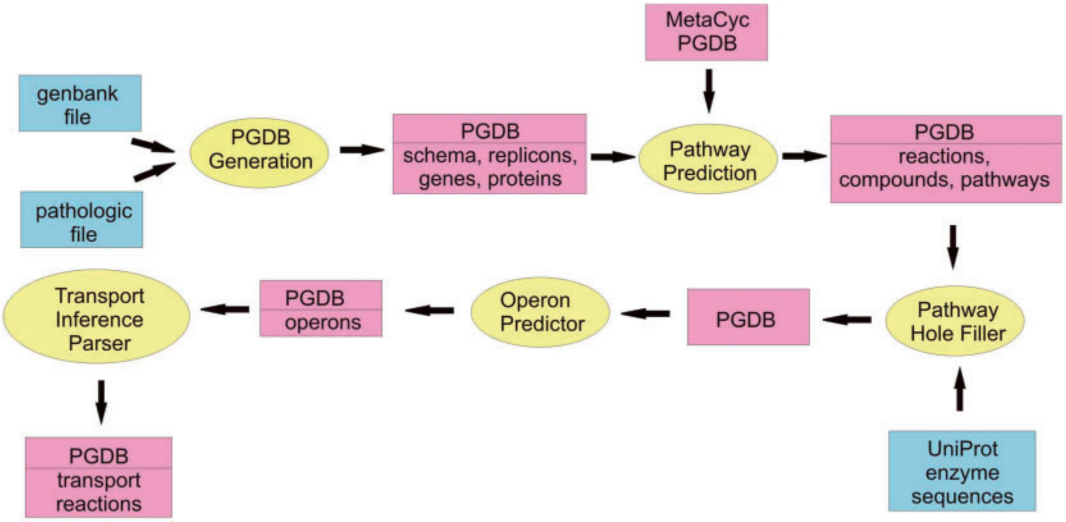


Figure 1: Inputs and outputs of the computational inference modules within PathoLogic. The initial input to PathoLogic is either a Genbank or a PathoLogic-format file. The boxes labeled "PGDB" all indicate that a PGDB is an input to or an output from some processing step; the notations at the bottom of the PGDB boxes indicate what types of data have been added by the previous processing step, for example, the Transport Inference Parser (TIP) adds transport reactions to a PGDB.

replicon (chromosome or plasmid), or of one contig for genomes that are not fully assembled.

The annotation specified in a Genbank or PathoLogic file can include the start and stop positions of the coding region for each gene, and intron positions. It can also include a description of the function of the gene product as a text string, one or more Enzyme Commission (EC) numbers and one or more GO terms. The annotation can also include a gene name, synonyms for the gene name and the product name, links to other bioinformatics DBs, and comments.

PathoLogic initializes the schema of the new PGDB by copying from MetaCyc into the new PGDB, the definitions of the approximately 3200 classes and 250 slots (DB attributes) that define the schema of a PGDB.

PathoLogic next creates a PGDB object for every replicon and contig defined by the input files, and for every gene and gene product defined in the input files. It populates these new objects with data from the input files, such as gene names and their sequence coordinates and gene product names. As a result of these operations, the new PGDB now mirrors the information in the input files.

PathoLogic inference of metabolic pathways

Pathway Tools predicts the metabolic pathway complement of an organism by assessing what known pathways from the MetaCyc PGDB [29] are present in the annotated genome of that organism's PGDB. This inference is performed in two steps that are described and evaluated further in Paley and Karp [30] and Karp *et al.* [2].

Step 1

Enzymes in the PGDB are assigned to their corresponding reactions in MetaCyc, thus defining the reactome of the organism. PathoLogic performs this assignment by matching the gene-product names (enzyme names), the EC numbers and the GO terms to MetaCyc reactions assigned to genes in the genome. The program can use whatever combination of these three information types is available in a given genome. For example, the *fabD* gene in *Bacillus anthracis* was annotated with the function 'malonyl CoA-acyl carrier protein transacylase.' That name was recognized by PathoLogic as corresponding to the MetaCyc reaction whose EC number is 2.3.1.39. PathoLogic therefore imported that reaction and its substrates into the *B. anthracis* PGDB, and created an enzymatic-reaction object linking that reaction to that of *B. anthracis* protein.

Although hundreds of such enzyme-reaction assignments are performed automatically by PathoLogic, it typically does not recognize on the order of 20% of the enzyme names in a genome. Therefore, PathoLogic includes an interactive tool that presents names of putative metabolic enzymes (all proteins whose name ends in 'ase', with exclusion of certain nonspecific and nonmetabolic enzyme names) to the user, and aids the user in assigning those enzymes to reactions in MetaCyc. For example, PathoLogic provides an operation that runs an inexact string comparison search between the enzyme name and all enzyme names in MetaCyc, which sometimes allows the user to identify a match based on scrambled word orders within complex enzyme names.

Step 2

Once the reactome of the organism has been established in the preceding manner, PathoLogic imports all MetaCyc pathways that contain at least one reaction in the organism's reactome into the new PGDB. Once imported, PathoLogic then attempts to prune out those pathways that are likely to be false positive predictions. That pruning process considers both the fraction of reaction steps in the pathway that has assigned enzymes, and how many of the reactions with assigned enzymes are unique to that pathway (as opposed to being used in additional metabolic pathways in that organism). The remaining pathways are those that are predicted to occur in the organism under analysis.

As MetaCyc has grown in size, we have seen a significant increase in the number of false positive predictions made by PathoLogic; thus, we have recently altered the pruning procedure to prune a predicted pathway from organism *X* if organism *X* is outside the expected taxonomic distribution of that pathway. MetaCyc records curated information about the expected taxonomic groups in which a pathway is expected to occur based on experimental observations of that pathway to date. For example, many pathways are expected to occur in plants only. This rule has significantly increased the accuracy of PathoLogic.

PathoLogic inference of operons

The Pathway Tools operon predictor identifies operon boundaries by examining pairs of adjacent genes *A* and *B* and using information such as intergenic distance, and whether it can identify a functional relationship between *A* and *B*, such as membership in the same pathway [31], membership in the same multimeric protein complex, or whether *A* is a transporter for a substrate within a metabolic pathway in which *B* is an enzyme.

PathoLogic inference of pathway holes

A pathway hole is a reaction in a metabolic pathway for which no enzyme has been identified in the genome that catalyzes that reaction. Typical microbial genomes contain 200–300 pathway holes. Although some pathway holes are probably genuine, we believe that the majority are likely to result from the failure of the genome annotation process to identify the genes corresponding to those pathway holes. For example, genome annotation systems systematically under-annotate genes with multiple functions, and we believe that the enzyme functions for many pathway holes are unidentified second functions for genes that have one assigned function.

The pathway hole filling program PHFiller [32] (a component of PathoLogic) generates hypotheses as to which genes code for these missing enzymes using the following method. Given a reaction that is a pathway hole, the program first queries the UniProt DB to find all known sequences for enzymes that catalyze that same reaction in other organisms. The program then uses the BLAST tool to compare that set of sequences against the full proteome of the organism in which we are seeking hole fillers. It scores the resulting BLAST hits using a Bayesian classifier that considers information such as genome localization, that is, is a potential hole filler in the same operon as another gene in the same metabolic pathway? At a stringent probability score cutoff, our method finds potential hole fillers for ~45% of the pathway holes in a microbial genome [32].

PHFiller includes a graphical interface that optionally presents each inferred hole filler to the user along with information that helps the user evaluate the hole fillers, and allows the user to accept or reject the hole fillers that it has proposed.

PathoLogic inference of transport reactions

Membrane transport proteins typically make up 5–15% of the gene content of organisms sequenced to date. Transporters import nutrients into the cell, thus determining the environments in which cell growth is possible. The development of the PathoLogic TIP [33] was motivated by the need to perform symbolic inferences on cellular transport systems, and by the need to include transporters on the Cellular Overview diagram. The motivating symbolic inferences include the problems of computing answers to the following queries: What chemicals can the organism import or export? For which cellular metabolites that are consumed by metabolic reactions but never produced by a reaction is there no known transporter (meaning that the origin of such metabolites is a mystery, and indicates missing knowledge about transporters or reactions that produce the compound)?

To answer such queries, we must have a representation of transporter function that is computable (ontology based). Pathway Tools has such a representation, in which transport events are represented as reactions in which the transported compound(s) are substrates. Each substrate is labeled with the cellular compartment in which it resides, and each substrate is a controlled-vocabulary term from the extensive set of chemical compounds in MetaCyc [7]. The TIP program converts the free-text descriptions of transporter functions found in genome annotations (examples: ‘predicted ATP transporter of cyanate’ and ‘sodium/proline symporter’) into computable transport reactions.

TIP performs the following operations that are explained more fully in Lee *et al.* [33]. Starting with the full set of monomeric proteins encoded by the genome, TIP first identifies the likely transport proteins by searching for proteins that include various keywords indicative of transport function (such as ‘transport’ and ‘channel’), and that lack certain counter-indicator keywords (such as ‘regulator’). Then, for each such identified transport protein *T*, the program performs these steps.

(1) It identifies the reaction substrates of *T*. The program parses the descriptions of transporter function to find the names of small molecules from the dictionary of compound names in MetaCyc.
(2) It determines the energy coupling for *T* (e.g. is *T* a passive channel, or an ATP-driven

transporter?) Energy coupling is inferred by a number of rules that include analysis of keywords and identified substrates.

(3) It assigns a compartment to each substrate of *T* by searching for keywords such as 'uptake', 'efflux', 'symport', and 'antiport'.

(4) It constructs a multimeric protein complex for *T* if so indicated. Most transporters are multimeric systems. A multimeric complex will be created for *T* if its gene is located within an operon containing other proteins annotated as transporting the same substrate, and if all proteins share the energy coupling mechanism of ATP or of the phosphotransferase system.

(5) It constructs a transport reaction for *T* by defining a new reaction object within the PGDB with appropriate reactants and products. If the coupling mechanism is phosphoenol pyruvate, the program creates a product that is a phosphorylated form of the transported substrate.

An evaluation showed that 67.5% of TIP predictions were correct; the remainder had an error in the substrate, in the directionality of transport, or in the energy coupling [33]. TIP includes a graphical interface that allows the user to interactively review and revise its predictions.

Pathway/Genome Editors

The Editors support PGDB curation through interactive modification and updating of all the major datatypes supported by Pathway Tools. They can be invoked quickly from every Navigator window through a single mouse operation so that a user who sees within the Navigator an object that needs to be updated can quickly invoke an editing tool to make the required change. When the user exits from the editing tool, the modified version of the object is then displayed within the Navigator.

The Editors allow the user to invoke an external spelling checker (ispell) to check spelling within comment fields.

Curators typically become proficient at these tools after a day of training and a few weeks of experience.

The editing tools included in Pathway Tools are as follows:

- Gene editor: This supports editing of gene name, synonyms, DB links and start and stop position within the sequence.
- Protein editor: This supports editing of protein attributes as well as of protein subunit structure and protein complexes (Supplementary Figure S12), and also allows users to assign terms from the GO and MultiFun controlled vocabularies. Pathway Tools can store, edit and display features of interest on a protein; see 'Pathway Tools protein feature ontology' section for more details. When editing a protein feature, the user selects a feature type (e.g. phosphorylation site), defines the location of the feature on the sequence, a bound or attached moiety where appropriate, a textual label, an optional comment, citations and sequence motif. The feature location can be specified either by typing in the residue number(s) or by selecting a portion of the amino acid sequence with the mouse. In addition, the sequence can be searched for specific residue combinations, which may include wild cards.
- Reaction editor: This supports editing of metabolic reactions, transport reactions and signaling reactions.
- Pathway editor: This allows users to interactively construct and edit a metabolic pathway from its component reactions (Supplementary Figure S11).
- Regulation editor: This allows definition of regulatory interactions including regulation of gene expression by control of transcription initiation, attenuation and by control of translation by proteins and small RNAs (Supplementary Figure S13). This editor also allows creation of operons and definition of their member genes, as well as specifying the positions of promoters and transcription factor binding sites.
- Compound editor: This supports editing of compound names, citations and DB links. Pathway Tools has been interfaced to two external chemical structure editors: Marvin [34] and JME [35]. A chemical compound duplicate checker runs whenever chemical structures are entered or modified, to inform the user if the resulting structure duplicates another compound in that user's PGDB or in MetaCyc.
- Publication editor: This supports entry of bibliographic references.
- Organism editor: This supports editing information about the organism described by a PGDB, including species name, strain name and synonyms, and taxonomic rank within the NCBI Taxonomy.

Author crediting system

Often, many curators collaborate on a given PGDB, and it is desirable to attribute their contributions accordingly. This not only helps to find out who should be asked if questions about particular entries arise, but more important, it will provide an incentive for high-quality contributions, because contributors will be able to clearly demonstrate their accomplishments.

The editing tools for the most important objects thus support attaching credits of several kinds. When an object such as a pathway is first created, by default, a 'created' credit is attached to the object, along with a timestamp. The curator is described by an author DB object, and a DB object describing the author's organization. The author frame records the name, email address and the organization(s) with which the curator is affiliated. Editing tools exist for authors and organizations, and substring search allows convenient retrieval. A given credit for an object can be attached to either authors, organizations or both, in a flexible manner. Every author and organization has a 'home' page that lists all the objects that have been credited.

Other kinds of credit are 'revised' when a curator substantially edits an object that was created some time ago, and a 'last curated' flag can be set to indicate when a curator has last researched the literature available for a given object. The last-curated flag is useful for those objects about which almost nothing is known, to distinguish between the case where no curator ever looked at the object, versus where an extensive search was performed but still nothing new was found.

Credits are included with pathways exported to a file, which allows exchange of pathway contributions between PGDBs, complete with proper credit attribution. An additional kind of credit called 'reviewed' can be used when such external contributions have been reviewed by a receiving curator, or to also attribute reviews of various objects by invited, external domain experts.

Bulk PGDB updating

During the PGDB life cycle, a number of types of PGDB updates are required that would be extremely onerous to perform if the user were forced to perform them manually, one at a time. Therefore, Pathway Tools provides several facilities for performing bulk updates of a PGDB. The most general facility is that users can write their own programs to perform arbitrary types of updates through the Pathway Tools APIs in the Perl, Java and Lisp languages (see 'Computational access to PGDBs' section).

Some groups choose to store the authoritative version of their genome annotation in a DB external to the PGDB, such as groups that developed their own genome DB system prior to adopting Pathway Tools. Such users need the ability to update their PGDB with data from a revised genome annotation without overwriting or otherwise losing any manual curation that has been added to the PGDB. Pathway Tools provides an interface for doing just that. It takes as input one or more update files, either in GenBank format or PathoLogic file format. The files can contain either a complete revised annotation for the organism, or they can contain just the information that has changed. The software will parse the update files and determine all differences between the new data and the old. Types of changes that are detected include new genes, as well as updated gene positions, names, synonyms, comments, links to external DBs and updated functional assignments. None of the changes will be propagated automatically. Instead, a pop-up dialog will summarize different classes of changes. For example, it will list the number of new genes, the number of genes with name changes, the number of previously unassigned genes that now match a reaction and the number of previously assigned genes that now match a different reaction. For each class of changes, the curator has the option of either accepting all updates (e.g. creating DB objects for all the new genes), or of checking each proposed update. Once this phase is complete and any changes to functional assignments have been made, the software will re-run the pathway inference procedure described in 'Pathologic inference of metabolic pathways' section, identify any new pathways that are inferred to be present and any existing pathways that no longer have sufficient evidence and allow the curator to review those changes.

Consistency checker and aggregate statistics

Pathway Tools contains an extensive set of programs for performing consistency checking of a PGDB to detect structural defects that sometimes arise within PGDBs. Also included in this component are tools for computing and caching aggregate statistics for a PGDB, such as computing the molecular weights of all proteins from their amino acid sequences. The statistics are cached so that they can be displayed quickly. At SRI, we run these programs as part of the quarterly release process for EcoCyc and MetaCyc.

Roughly half of the programs automatically repair PGDB problems that they find. Such problems could be caused by user data entry errors, or by errors in Pathway Tools itself. Example checks include to ensure that inverse relationship links are set properly (e.g. that a gene is linked to its gene product, and that the product links back to the gene); make sure pathways do not contain duplicate reactions; validate and update GO term assignments with respect to the latest version of GO; perform formatting checks in comment text; search gene reading frames for internal stop codons; and to remove redundant bonds from chemical structures.

The other checker programs generate listings of every error detected, and allow the user to click on each problematic object in the listing to enter the editor for that object to repair it.

Schema upgrading

Most new releases of Pathway Tools include additions or modifications to the Pathway Tools schema. Schema changes are made to model the underlying biology more accurately (such as adding support for introns and exons), extend the datatypes within Pathway Tools (such as adding support for features on protein sequences) or to increase the speed of the software. Because each new version of the software depends on finding data within the fields defined by the associated version of the schema, existing user PGDBs created by older versions of the software will be incompatible with these new software versions.

Therefore, every release of Pathway Tools contains a program to upgrade PGDBs whose schema corresponds to the previous version of the software, to the new version of the software. When a user opens a PGDB under a new version of the software, the software detects that the schema of the PGDB is out of date, and offers to run this schema upgrade program for the user. For users who have not upgraded the software for several releases, several upgrade operations are performed consecutively. Example upgrade operations include adding new classes to the PGDB from the MetaCyc PGDB, adding new slots to PGDB classes, deleting PGDB classes, moving data values from one slot to another and moving objects from one class to another. The schema upgrade leaves the user's curated data intact.

Every new release of Pathway Tools includes a new version of the MetaCyc DB, which, in addition to providing new data content, typically contains updates and corrections to existing pathways, reactions and compounds. Pathway Tools includes an option to propagate such updates and corrections to an existing organism PGDB. However, because we do not want to override any manual edits made to a PGDB, this tool does not run automatically. Much like the tool for incorporating a revised genome annotation, described in 'Bulk PGDB updating' section, this tool organizes the changes into logical groups (such as all compounds with newly added structures, or all reactions with changed reaction equations), and allows the user to either accept an entire group of changes, or to examine and confirm each member of a group.

THE PATHWAY TOOLS SCHEMA

Conceptualizations of biological information are used within the Pathway Tools schema. The schema is a key part of Pathway Tools' ability to make many types of biological data accessible to computational analysis.

The Pathway Tools schema defines a set of classes and a set of slots. Classes describe types of biological entities, such as genes and pathways, and are arranged in a class–subclass hierarchy. Many of the important Pathway Tools classes are shown in Figures 2 and 3. DB *slots* store properties of the classes and objects within a PGDB. Slots store attributes of PGDB objects, and relationships between PGDB objects. Figure 4 provides an overview of the relationships among PGDB classes. For example, user queries can follow the relationship from a gene to the protein that it codes for, from a protein to a reaction that it catalyzes and from a reaction to a metabolic pathway in which it is a component, to answer questions such as 'find all metabolic pathways in which the products of a given gene play a role'.

Every PGDB object has a stable unique identifier (ID)—a symbol that uniquely identifies that object within the PGDB. Example unique IDs include TRP (an identifier for a metabolite), Rxn0-2382 (an identifier for a reaction) and Pwy0-1280 (an identifier for a pathway). Relationships within a PGDB are implemented by storing object IDs

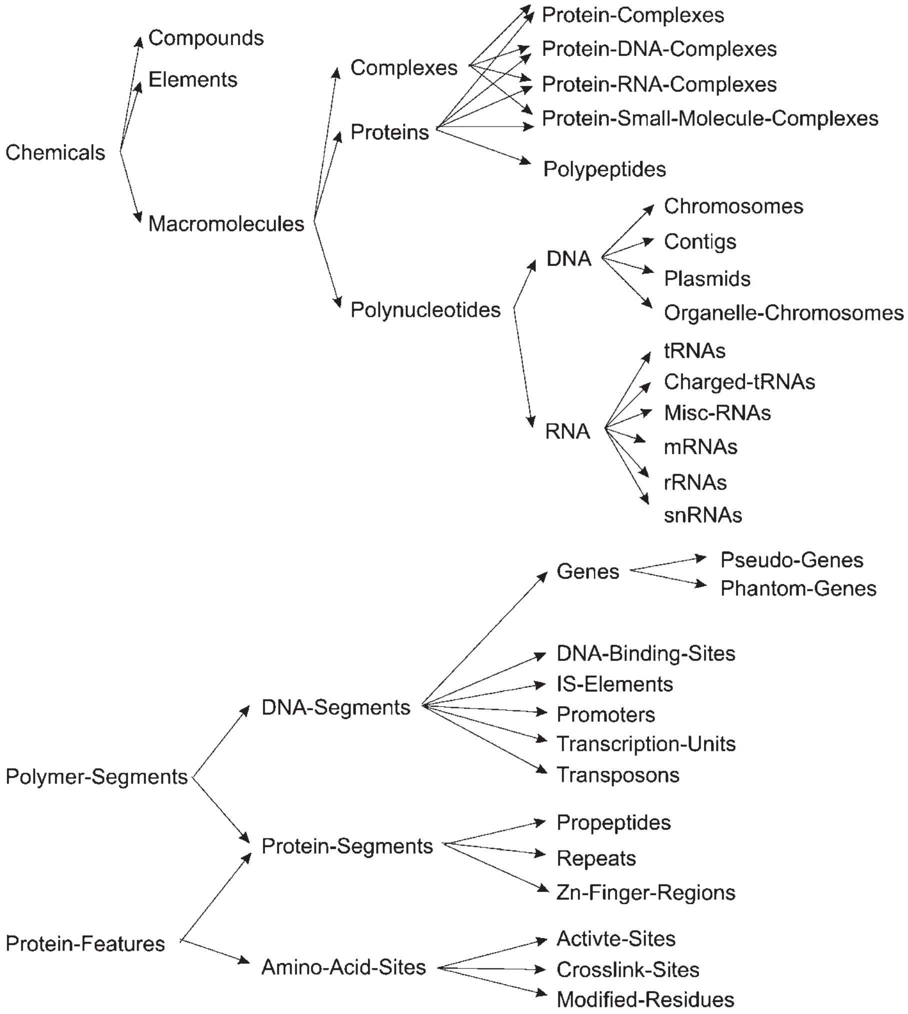


Figure 2: First set of major classes within the Pathway Tools schema, shown in a class–subclass hierarchy. Many of these classes have many subclasses that are not shown.

within slots. For example, to state that the TRP (L-tryptophan) object is a reactant in the reaction Rxn0-2382, a slot of Rxn0-2382 called LEFT (meaning reactants) contains the value TRP. Many PGDB relationships exist in both forward and backward directions, for example, the TRP object contains a field called APPEARS-IN-LEFT-SIDE-OF that lists all reactions in which TRP is a reactant. The slots LEFT and APPEARS-IN-LEFT-SIDE-OF are called inverses.

Metabolites, reactions and pathways

There are two alternative ways in which one might choose to represent the metabolic network in a computer: as a simple listing of all metabolic reactions that occur in the cell, or by partitioning the reaction list into a carefully delineated set of metabolic pathways that describe small, functionally linked subsets of reactions. Which approach is preferred? Both approaches have value, and they are not mutually exclusive; therefore, Pathway Tools supports both views of metabolism in a PGDB.

Pathway Tools conceptualizes the metabolic network in three layers. The first layer consists of the small molecule substrates upon which metabolism operates. The second layer consists of the reactions that interconvert the small molecule metabolites. The third layer consists of the metabolic pathways whose components are the metabolic reactions of the second layer. Note that not all reactions in the second layer are included in pathways in the third layer, because some metabolic reactions have not been assigned to any metabolic pathway by biologists.

Scientists who choose to view the metabolic network within a PGDB solely as a reaction list can operate on the second layer directly without interference from the third layer. But for a scientist

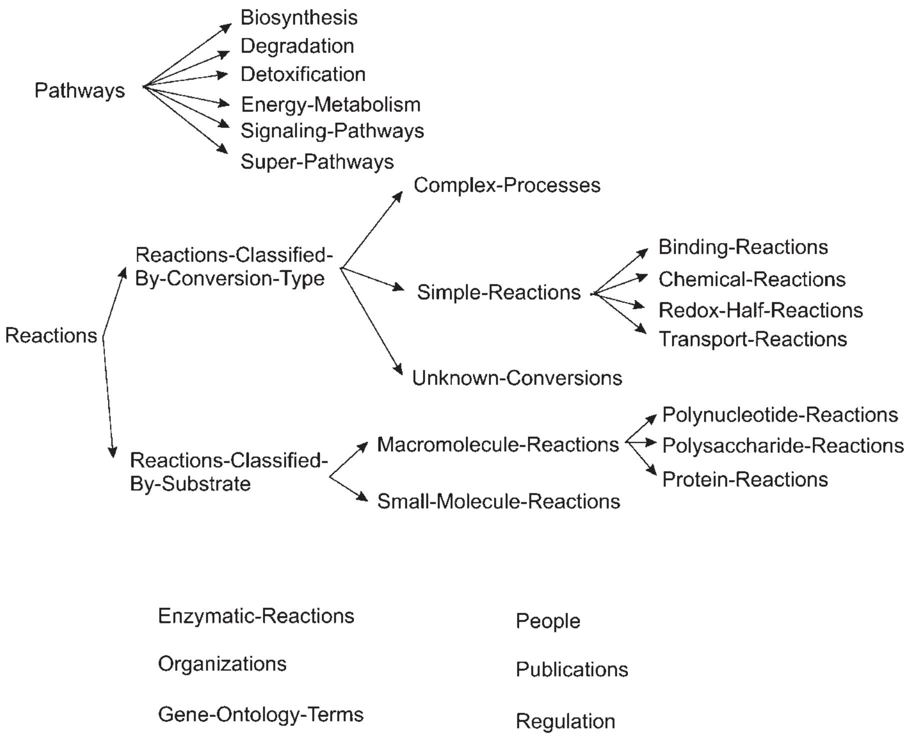


Figure 3: Second set of major classes within the Pathway Tools schema, shown in a class–subclass hierarchy. The classes shown at the bottom have no subclasses.

for whom the pathway definitions are important, the pathway layer is available in the PGDB.

The pathways in PGDBs are modules of the metabolic network of a single organism. Pathway boundaries are defined by considering the following factors. Pathways are often regulated as a unit (based on substrate-level regulation of key enzymes, on regulation of gene expression and on other types of regulation). Pathway boundaries are often defined at high connectivity, stable metabolites [36]. Pathway conservation across multiple species is also considered, as are pathway definitions from the experimental literature [29].

The compounds, reactions and pathways in levels 1–3 are each represented as distinct DB objects within a PGDB. The relationships among the metabolic datatypes in a PGDB are depicted by the blue region of Figure 4.

The pathway for biosynthesis of L-tryptophan shown in Figure 5 is represented within a PGDB as shown in Figure 6. An object representing the pathway, at the top of the figure, is connected via slot Reaction List to objects representing every reaction within the pathway. One of those reactions (Rxn0-2382) is shown in the figure. It is connected, via slots Left and Right, to objects that represent each substrate of the reaction. In addition, it is connected to the enzymatic reaction object EnzRxn0-3701, which in turn is connected to an object representing the enzyme complex that catalyzes the reaction, Cplx0-2401. Note that every slot shown in this diagram has an inverse slot, meaning a slot that represents the inverse relationship. For example, slot In-Pathway represents the relationship from a reaction to the pathway containing that reaction.

The proteome and the genome

Our discussion of PGDB definitions of metabolism thus far has not considered the metabolic enzymes, nor the genome that encodes those proteins. PGDBs define the proteome and the genome of an organism in the following manner, as depicted by the green region of Figure 4.

The proteome of the organism is described as a set of PGDB objects, one for each gene product in the organism, and one for each complex formed from two or more (identical or nonidentical) polypeptides. Furthermore, every chemically modified form of a monomer or of a multimer is encoded by a distinct PGDB object. For example, we might create one object representing an unmodified protein and another representing the phosphorylated form.

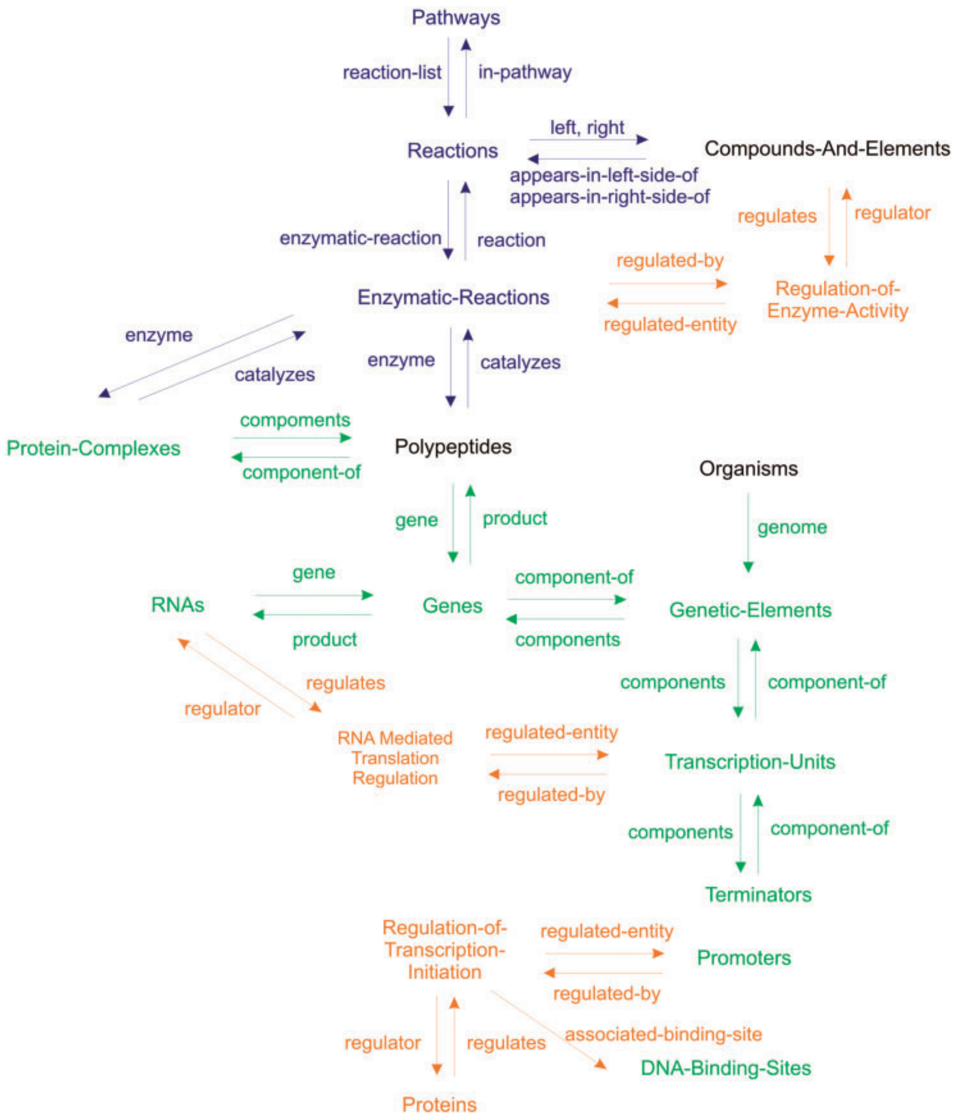


Figure 4: Major relationships among the major classes of the Pathway Tools schema. Colors indicate biological areas: blue for reaction and pathway information; green for genome and protein information; and orange for regulation.

Each protein object is in turn linked, through a slot in the object, to the metabolic reactions that it catalyzes. Proteins can also be substrates of reactions. Additional PGDB objects define features on proteins, as described in 'Pathway Tools protein feature ontology' section.

Each protein product resulting from alternatively spliced forms of a gene is also represented by a distinct protein object. Each protein object records the exons of the gene that encodes it.

Protein objects are also linked to gene objects that define the gene encoding each protein. Each gene in the genome is defined by a distinct PGDB object, as is every replicon (chromosome or plasmid) in the genome. Genes are linked to the replicon on which they reside. In addition, other features on the genome, such as operons, promoters and transcription factor binding sites, are described by PGDB objects.

The associations between enzymes and the reactions they catalyze are implemented using an intermediary object called an enzymatic reaction, as shown in Figure 6. This arrangement allows us to capture the many-to-many relationship that exists between enzymes and reactions—one reaction can be catalyzed by multiple enzymes, and multifunctional enzymes catalyze multiple reactions. The purpose of the enzymatic reaction is to encode

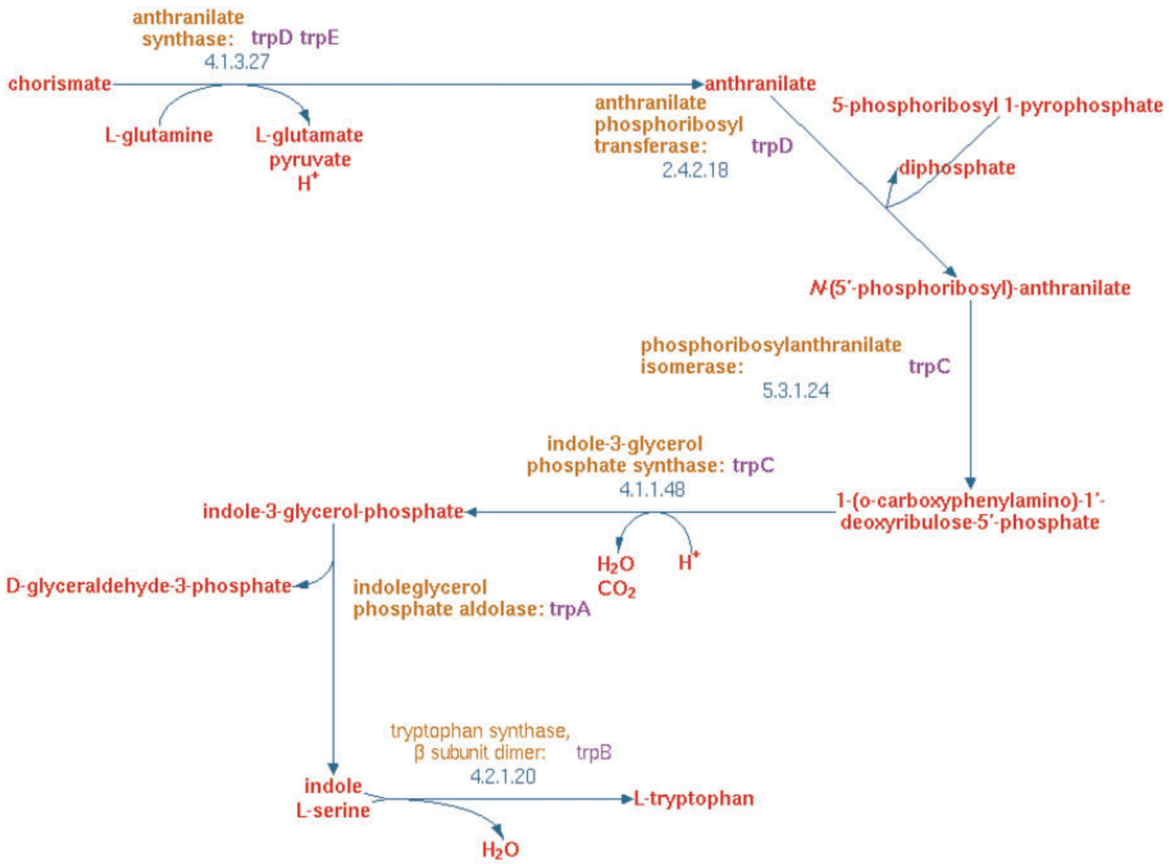


Figure 5: The biosynthetic pathway for L-tryptophan in EcoCyc.

information that is specific to the pairing of the enzyme with the reaction, such as cofactors, activators and inhibitors. Consider a bifunctional enzyme with two active sites, where one of the active sites is inhibited by pyruvate, and the second active site is inhibited by lactate. We would represent this situation with two enzymatic reactions linking the enzyme to the two reactions it catalyzes, and each enzymatic reaction would specify a different inhibitor.

Continuing the explanation of the example shown in Figure 6, the homodimer enzyme Cplx0-2401 is connected to Trypsyn–BProtein, which represents the monomer. It in turn is connected to Eg11025, which represents the gene encoding this monomer. It is connected to an object representing the *E. coli* chromosome, Ecoli-K12-Chromosome.

Pathway Tools regulation ontology

The Pathway Tools schema can represent all important regulatory interaction types within *E. coli*, ranging from substrate-level regulation of enzyme activity, to the transcriptional control of gene expression, to regulation of gene expression by small RNAs. In each case, a regulation object within a PGDB captures essential information about one regulatory interaction. A schema class called Regulation defines the class of all regulatory interactions; each of its subclasses defines a different mechanism of cellular regulation. Those subclasses are as follows.

Class `Regulation-of-Enzyme-Activity` describes substrate-level regulation of enzyme catalytic activity, such as the allosteric activation or competitive inhibition of an enzyme by a small molecule. Slots of this class link to the regulator molecule and to the regulated enzymatic reaction object (such as EnzRxn0-3701 in Figure 6). Other slots encode the polarity of regulation (activation or inhibition), the mechanism of regulation, such as whether it is allosteric, competitive or noncompetitive, and whether this is an important mode of regulation *in vivo* (since experimenters routinely test for enzyme activity *in vitro* in the presence of a number of compounds that the enzyme is not likely to encounter *in vivo*).

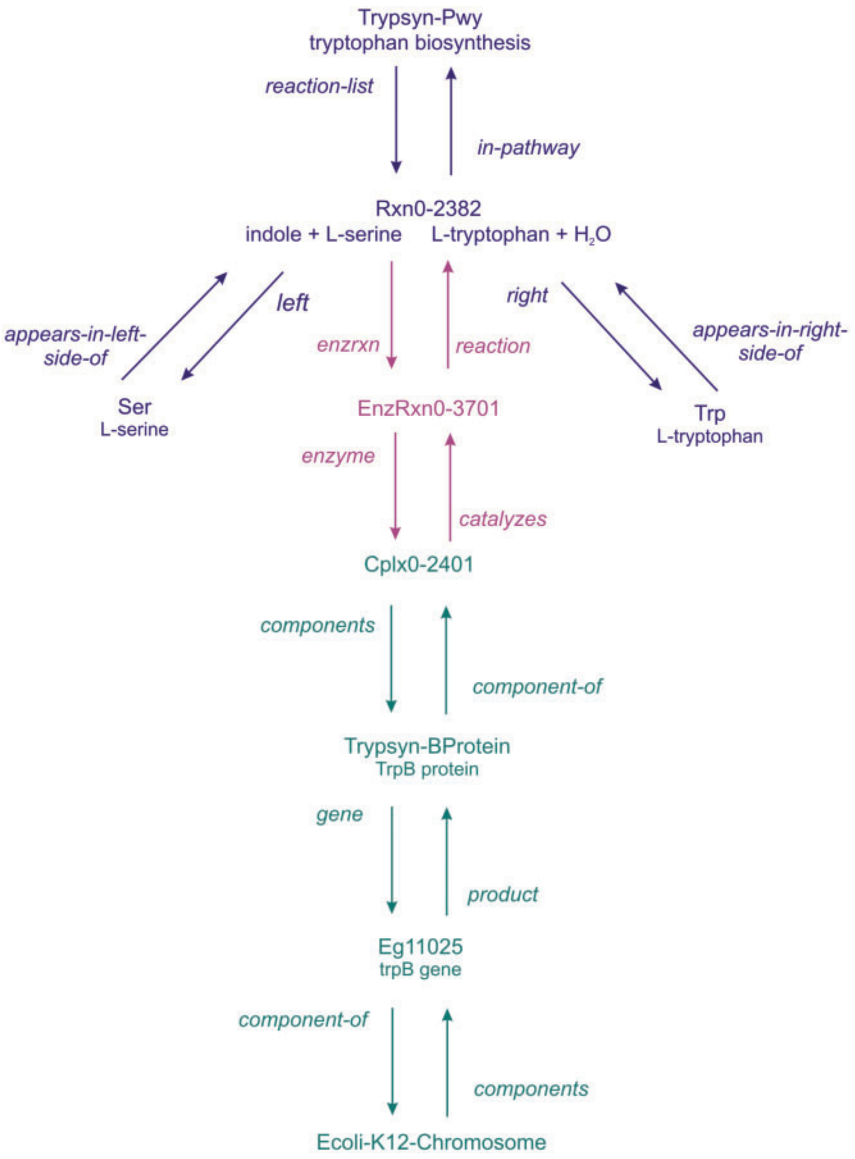


Figure 6: Some of the PGDB objects representing the pathway for L-tryptophan biosynthesis, and their connections.

Class `Regulation-of-Transcription-Initiation` describes the regulation of a bacterial promoter by a transcription factor protein. The slots (attributes) of this regulation class describe the essential information about that regulatory interaction, identifying the transcription factor, the promoter that is regulated, and the binding site to which the regulator binds. Each of the preceding entities is represented by a PGDB object.

Class `Transcriptional-Attenuation` describes the regulation of premature termination of transcription. This class is further divided into six subclasses, each describing a different attenuation mechanism (e.g. ribosome mediated, protein mediated and RNA mediated). The slots of these classes identify the regulated terminator region, the regulator (a protein, RNA or small-molecule, depending on the type of attenuation) and the regulator binding site if one exists, all of which are represented by PGDB objects. They also identify additional relevant sequence regions where appropriate, such as the antiterminator region, the anti-antiterminator region and the ribosome pause site.

Class `Regulation-of-Translation` describes regulation of the translation of an mRNA transcript to the corresponding protein. This class is divided into two subclasses to distinguish between regulation by a protein and regulation by a small

RNA. The slots of these classes identify the regulated transcription unit (which corresponds to a single transcript), the regulator protein or RNA and the mRNA binding site where the regulator binds. An additional slot indicates whether regulation is by direct interference with the translation machinery, by processing the mRNA transcript to promote or inhibit its degradation before translation, or both.

Regulation of protein activity by chemical modification, such as by phosphorylation, is represented by a reaction that converts the unmodified form of the protein to the modified form.

Pathway Tools evidence ontology

MOD users want to know the type(s) of evidence that support assertions within a MOD, and they want to know the strength of that evidence. We have developed an evidence ontology [37] that can encode information about *why* we believe certain assertions in a PGDB, the *sources* of those assertions and the *degree of confidence* scientists hold in those assertions (although in practice the latter field is rarely populated). An example assertion is the existence of a biological object described in a PGDB—we would like to be able to encode the evidence supporting the existence of a gene, an operon or a pathway that is described within a PGDB. Has the operon been predicted using a computational operon finder? Or is it supported by wet-lab experiments? Our evidence ontology builds upon and substantially extends the GO evidence ontology, which applies only to gene products.

Evidence about object existence in PGDBs is recorded as a structured *evidence tuple*. An evidence tuple allows us to associate several types of information within one piece of evidence. Each *evidence tuple* is of the form

```
Evidence-code : Citation : Curator :
      Timestamp : Probability
```

where Evidence-code is a unique ID for the type of evidence, within a hierarchy of 48 evidence types described in Karp *et al.* [37] and Pathway Tools Evidence Ontology [38]. `Citation` is an optional citation identifier such as a PubMed ID that indicates the source of the evidence. For computational evidence, the citation refers to an article describing the algorithm used. `Curator` identifies the curator who created this evidence tuple; `Timestamp` encodes when this evidence tuple was created. `Probability` is an optional real number indicating the probability that the assertion supported by this evidence is correct, such as a probability provided by an algorithm.

The Pathway Tools editors allow users to manually enter evidence codes, and the PathoLogic pathway and operon predictors annotate objects that they create with appropriate computational evidence codes. The Navigator supports display and querying of evidence codes.

Pathway Tools Cell Component Ontology

The Cell Component Ontology (CCO) is a controlled vocabulary of terms describing cellular components and compartments, and relationships between these terms [39]. It was developed to provide a controlled vocabulary of terms for annotating the subcellular locations of enzymes, and compartments involved in transport reactions, in PGDBs. CCO spans all domains of life, and includes terms such as cytoplasm, cell wall and chloroplast. The ontology currently contains 150 terms. CCO includes many terms and their definitions from the GO [40], but substantially extends GO.

Pathway Tools protein feature ontology

We have developed an ontology of protein features in order to identify and represent post-translational modifications, binding sites, active sites, conserved regions and other regions of interest on a protein. Starting from the list of feature types described in the UniProt User Manual [4], with some suggested additions from the SRI EcoCyc and MetaCyc DB curators, we created an ontology of 40 feature classes.

Features fall into two major classes. For amino acid site features, the feature location is a list of one or more amino acid residue numbers (or residue types, if the feature is associated with a generic protein whose precise sequence is unspecified). For protein segment features, the feature location is a range defined by its starting and ending residue numbers.

Feature types that are classified as binding features (either covalent or noncovalent) permit specification of an attached group. The attached group could be a compound or compound fragment, as in the case of a protein that binds a small molecule. The attached group can also be another protein feature, as in the case of disulfide bond or other cross-link between two features on different proteins, or any other type of molecule or binding site (such as a DNA binding site).

A different protein object is created in a PGDB for each biologically relevant modified form of a protein, and a single feature may be linked to multiple forms of the same protein. Some feature types are capable of existing in multiple states. For example, an amino acid modification feature can be in either the modified or the unmodified state (as in the case of a phosphorylation feature, which will be in the modified state when associated with the phosphorylated protein and the unmodified state when associated with the unphosphorylated protein), and a binding feature can be in either the bound or unbound state (as in the case of a metal-binding feature whose state indicates whether or not the metal ion is bound to the protein). We consider the state to be not an attribute of the feature, but rather an attribute of the pairing between a particular form of a protein and the feature. Thus, a reaction may convert a protein with a feature in the unmodified state to another form of the protein with the same feature in the modified state, making it clear that the only change was to the state of a single feature. Feature states may also be left unspecified—this enables us to avoid the combinatorial explosion of different protein forms that would otherwise result when a protein has multiple modification features, and a change in state of one feature does not depend on the state of other features.

VISUALIZATION AND QUERYING OF PGDBs

The Pathway/Genome Navigator component of Pathway Tools provides mechanisms for interrogating PGDBs, and for visualizing the results of those queries. We begin by describing the query tools. We then describe visualization tools for individual biological entities (such as genes and pathways), followed by systems-level visualization tools that graphically display the entire metabolic network, entire regulatory network and entire genome map of an organism.

The Navigator runs as both a desktop application and a web server. The desktop mode is faster, and has more overall functionality (see [42] for details), but the web mode has some functionality not present in the desktop mode.

Query tools

Version 13.0 of Pathway Tools, released in March 2009, introduced a completely redesigned web-based query interface. It provides a three-tiered query paradigm, meaning that three different types of query tools are available, each of which represents a different tradeoff between ease of use and query power. For example, the quick search is designed to provide a fast and simple way for new or casual users to find general information in the site. Statistics from our web logs presented below support the notion that the simpler search tools are used more frequently.

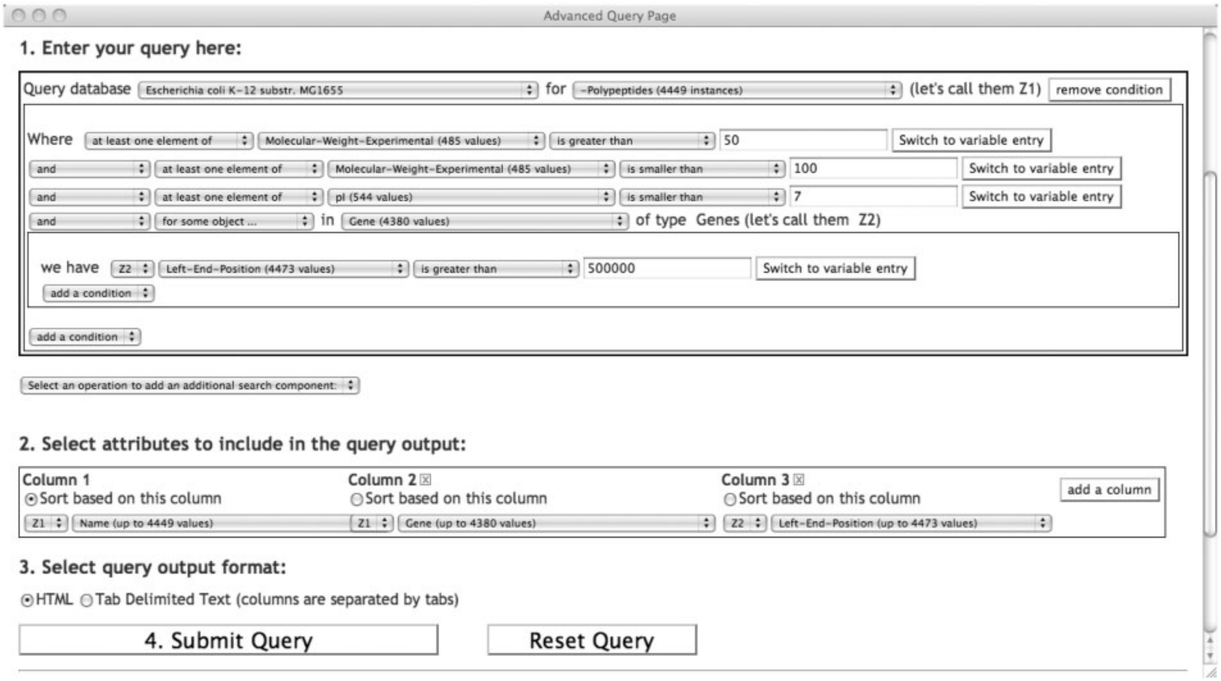
The ‘Quick Search’ box that appears at the top of most web pages generated by a Pathway Tools server is extremely easy to use. The user enters a search term and selects the organism whose PGDB the user wants to query. Pathway Tools searches that PGDB for objects whose primary name or synonyms contain the search term as a substring, and presents the list of results, organized by object type. The user can click on an object name to navigate to the display page for that object. A total of 62 349 quick searches were performed at BioCyc.org in May 2009.

A set of intermediate-level query tools provides the ability to construct more powerful and precise searches against objects of a single class. One such query page exists for genes, proteins and RNAs (Supplementary Figure S1); there are additional query pages for pathways, for reactions and for chemical compounds. A total of 3476 object searches were performed at BioCyc.org in May 2009.

Finally, the next section describes a tool called the Structured Advanced Query Page (SAQP) that allows advanced users to construct extremely powerful searches (that are approximately as powerful as provided by the SQL language). The graphical interactive nature of this web form makes these searches much easier to construct than using the SQL language. Six hundred and seventy four SAQP searches were performed at BioCyc.org in May 2009.

Structured advanced query page

The SAQP enables a biologist to search a large number of DBs in a precise manner. The queries can be as simple as looking up a gene given a name, or as complex as searching several DBs and several object types interconnected by several relations. The SAQP allows biologists to formulate



1. Enter your query here:

Query database Escherichia coli K-12 substr. MG1655 for -Polypeptides (4449 instances) (let's call them Z1)

Where at least one element of Molecular-Weight-Experimental (485 values) is greater than 50
and at least one element of Molecular-Weight-Experimental (485 values) is smaller than 100
and at least one element of pI (544 values) is smaller than 7
and for some object ... in Gene (4380 values) of type Genes (let's call them Z2)

we have Z2 Left-End-Position (4473 values) is greater than 500000

2. Select attributes to include in the query output:

Column 1 ● Sort based on this column Z1 Name (up to 4449 values)
Column 2 ☒ ○ Sort based on this column Z1 Gene (up to 4380 values)
Column 3 ☒ ○ Sort based on this column Z2 Left-End-Position (up to 4473 values)

3. Select query output format:

● HTML ○ Tab Delimited Text (columns are separated by tabs)

Figure 7: A query for the *E. coli* polypeptides whose experimental molecular weight lies between 50 and 100 kDa, whose pI is smaller than 7, and whose gene is located after the first 500 kb of the genome. An output column is used to include the gene (or sometimes genes) producing each polypeptide using the second variable Z2.

queries whose power and expressiveness closely approach SQL, but without having to learn SQL. The SAQP translates a formulated query into BioVelo, an OQL-like language [43], before sending it to the Web server.

The following explanation presents the elements of this Web user interface using one example. Figure 7 shows an example query against the class of protein monomers (polypeptides) in the EcoCyc DB.

Step 1: select DB and class

The first step in building a query is to specify at least one DB and the class of objects to search.

Step 2: specify conditions

Most queries include one or more conditions on the desired objects within the class. By clicking the button labeled `add a condition` in the initial blank SAQP, a *where* clause is added—visually boxed—in the search component. This operation adds a selector for an *attribute* (e.g. `name`) of the objects and a selector for a relational operator (e.g. `contains the substring`). It also adds a *free text box* to enter a number or string. Several other relational operators are provided, such as `is equal to`, `is not equal to` and `is a substring of`. Regular expression matching is also available as an operator, such as to allow wildcards within query strings.

This new field forms an *atomic condition*. Additional atomic conditions can be added to the query by using the button labeled 'add a condition'.

When clicking the drop-down selector for a relational operator, the list of relational operators provided is compatible with the type of the selected attribute. In the case of the attribute `name`, the selectable operators are for strings, since the *type* of the attribute `name` is string. This notion of type extends to all biological objects such as genes, proteins, metabolic pathways, reactions and compounds. Thus, the user can select only those operators that are compatible with the selected attribute. The query in Figure 7 has three atomic conditions to filter the selected polypeptides.

Quantifiers on relations within the SAQP allow a join-like capability. For example, imagine that we want to extend the query with an additional restriction that depends on the *gene encoding the polypeptide,* not on the polypeptide itself.

To do so, the user would add an `and` condition, and then select the `gene` attribute, which represents the gene encoding the polypeptide. We then select the quantifier operator for some object..., meaning that we want to define a condition that applies to some of the genes in the gene attribute of this polypeptide (although in the majority of cases only one gene will be present).

At this point, the SAQP adds a new indented query clause, to allow a condition to be defined on the gene. We have specified a constraint that its nucleotide coordinate must lie after the first 500 kb of the genome. Since several attributes and logical connectors can be specified in this new clause, forming a complex condition by itself, the Web interface draws a box around this condition and introduces it with the we have keyword. A new unique variable, named Z2, is also introduced. This variable represents every value of the gene attribute.

Step 3: define query results
The section titled Select attributes to include in the query output allows the user to describe the contents of the query results by selecting the attributes to display for each result object. The result of a query is always a table of at least one column. The tables have zero or more rows, one for each query result, and each column is a selected attribute. A new column can be added by clicking the button add a column. In the case of Figure 7, three columns are specified, two using variable Z1 (for the polypeptides) and one using the Z2 variable for the genes encoding them.

The selector provided in each column contains the list of accessible attributes for the object class selected for this query. When more than one search component is specified or a subquery is used with a quantifier, a variable selector is provided to select the desired variable. The interface provides the number of possible objects having at least one value for each attribute.

The output table produced by the SAQP can be formatted in two possible styles: tabulated and HTML. For the tabulated format, column entries are separated by a tab. It can be used as input to such software as Excel. The HTML format is the preferred format to navigate and analyze the results using a web browser.

Step 4: submit query
Once the user submits its query, a web page similar to Figure 8 is returned. The rows of the resulting table can be sorted based on any user-selected column. It can be resorted at will on any column in the output page.

Visualization tools for individual biological entities

(i) Genes: Gene-display windows list information such as the map position of the gene on the chromosome, the GO and MultiFun ontology class(es) to which the gene was assigned, and the regulation of the gene. The gene product is listed; for enzymes and transporters, the display shows reactions it catalyzes and their associated pathway(s).

(ii) Reactions: Reaction display and editing applies to metabolic, transport and protein signaling reactions (Supplementary Figure S2). The reaction display shows the one or more enzymes that catalyze the reaction, the gene(s) that code for the enzymes and the pathway(s) that contain the reaction. The display shows the EC number for the reaction, and the reaction equation.

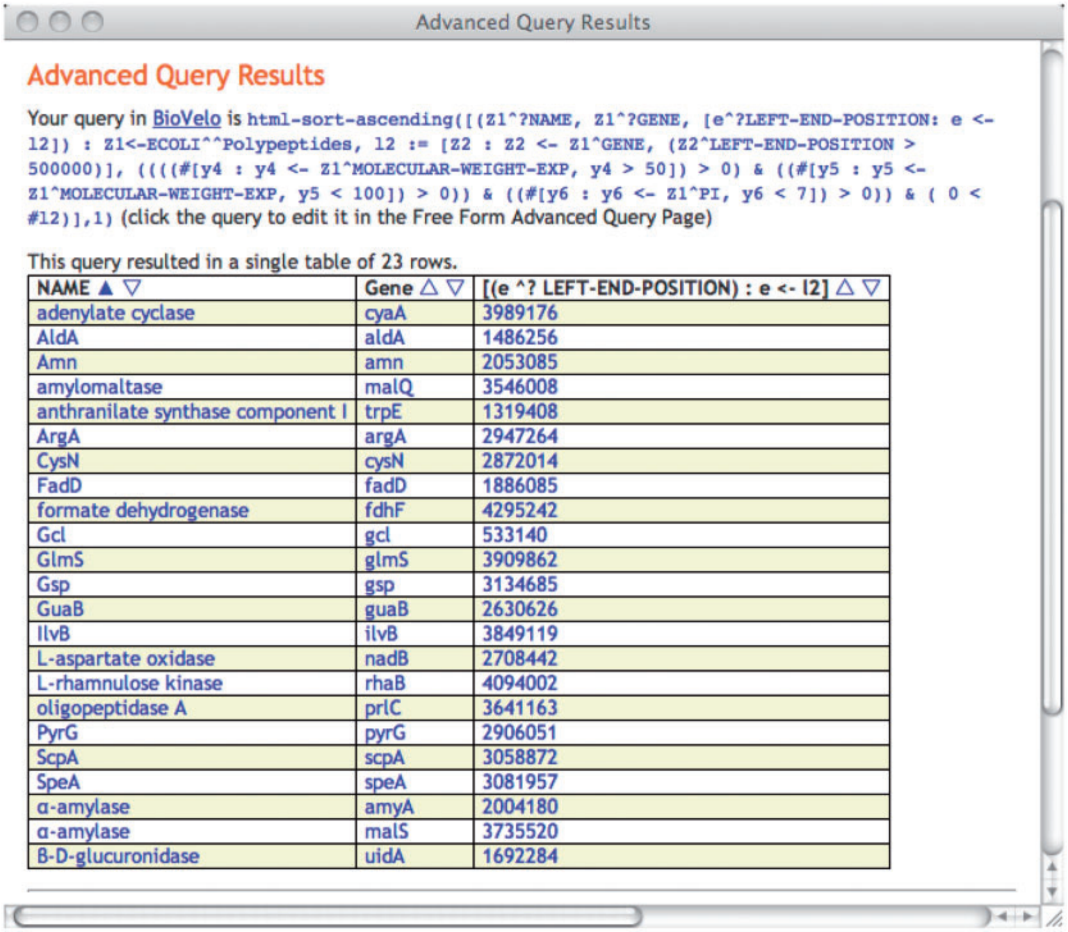
(iii) Proteins: Common to all protein types is the ability to display information about protein regions (such as phosphorylation sites and active sites) using a protein feature ontology that we developed.

  (a) Enzymes: The software displays the reaction catalyzed by the enzyme and the name of the pathway that contains that reaction (if any) (Supplementary Figure S3); the activators, inhibitors and cofactors required by the enzyme; and comments and citations for the enzyme.

  (b) Transporters: The software displays the transport reaction catalyzed by the transporter (Supplementary Figure S4).

  (c) Transcription factors: The software displays diagrams for all operons controlled by the transcription factor (the regulon for the transcription factor) (Supplementary Figure S5).

(iv) Pathways: All pathway visualizations are computed automatically using pathway-layout algorithms. Pathway Tools can draw pathways at multiple levels of detail, ranging from a skeletal view of a pathway that depicts the compounds only at the periphery of the pathway and at internal branch points, to a detailed view that shows full structures for every compound, and



Advanced Query Results

Your query in BioVelo is `html-sort-ascending([(Z1^?NAME, Z1^?GENE, [e^?LEFT-END-POSITION: e <- l2]) : Z1<-ECOLI^^Polypeptides, l2 := [Z2 : Z2 <- Z1^GENE, (Z2^LEFT-END-POSITION > 500000)], ((((#[y4 : y4 <- Z1^MOLECULAR-WEIGHT-EXP, y4 > 50]) > 0) & ((#[y5 : y5 <- Z1^MOLECULAR-WEIGHT-EXP, y5 < 100]) > 0)) & ((#[y6 : y6 <- Z1^PI, y6 < 7]) > 0)) & (0 < #l2)],1)` (click the query to edit it in the Free Form Advanced Query Page)

This query resulted in a single table of 23 rows.

NAME ▲ ▽	Gene △ ▽	[(e ^? LEFT-END-POSITION) : e <- l2] △ ▽
adenylate cyclase	cyaA	3989176
AldA	aldA	1486256
Amn	amn	2053085
amylomaltase	malQ	3546008
anthranilate synthase component I	trpE	1319408
ArgA	argA	2947264
CysN	cysN	2872014
FadD	fadD	1886085
formate dehydrogenase	fdhF	4295242
Gcl	gcl	533140
GlmS	glmS	3909862
Gsp	gsp	3134685
GuaB	guaB	2630626
IlvB	ilvB	3849119
L-aspartate oxidase	nadB	2708442
L-rhamnulose kinase	rhaB	4094002
oligopeptidase A	prlC	3641163
PyrG	pyrG	2906051
ScpA	scpA	3058872
SpeA	speA	3081957
α-amylase	amyA	2004180
α-amylase	malS	3735520
B-D-glucuronidase	uidA	1692284

Figure 8: The output result of the query in Figure 7. The BioVelo query generated from the user selection of Figure 7 is also shown near the top of the page.

EC numbers, enzyme names and gene names at every reaction step (Supplementary Figure S6). The user can customize a pathway drawing to include desired elements only.

(v) Chemical compounds: The compound display shows the chemical structure for the compound (Supplementary Figure S7). It lists all reactions in which the compound appears, and it lists enzymes whose activity is regulated by the compound.

(vi) Transcription units: The display window for transcription units diagrams the transcription unit and its regulatory sites including promoters, transcription factor binding sites, attenuators and binding sites for proteins and RNAs that regulate its translation. The display contains sections describing each site within the transcription unit. The promoter section describes which sigma factor recognizes it. Sections for transcription factor binding sites describe which transcription factor it binds, ligands that influence the activity of the transcription factor and whether the effect of binding is to activate or inhibit transcription initiation. Sections for attenuators describe the signal that the attenuator senses, and show the sequence regions that form the attenuator.

System-level visualization of metabolic networks

Pathway Tools can automatically generate organism-specific metabolic charts that we call Cellular Overview diagrams [3]. The diagram can be generated as a graphic, on the computer screen, that can be interrogated interactively and used to analyze omics datasets. It can be generated as a PDF file for printing as a large-format poster. Supplementary File S1 contains such a poster for *Caulobacter crescentus*.

Figure 9 depicts the entire diagram at low resolution painted with gene expression data. It contains all known metabolic pathways and transporters of

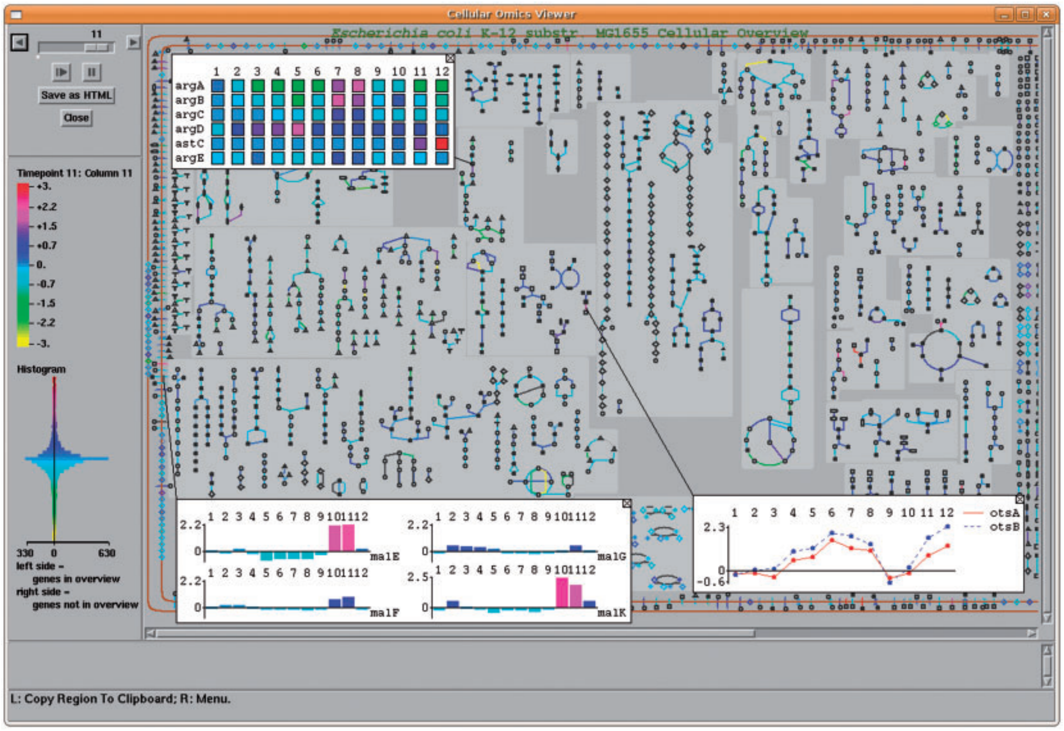


Figure 9: The PathwayTools Cellular Overview diagram for EcoCyc, painted with gene expression data. Three omics popups show expression data for individual genes in each of the three supported styles: heat map, X-Y plot, and bar graph. The top (heat map) and right (X-Y plot) popups include values for all the genes in their respective pathways. The bottom left popup (bar graph format) shows expression of all genes involved in one transport reaction.

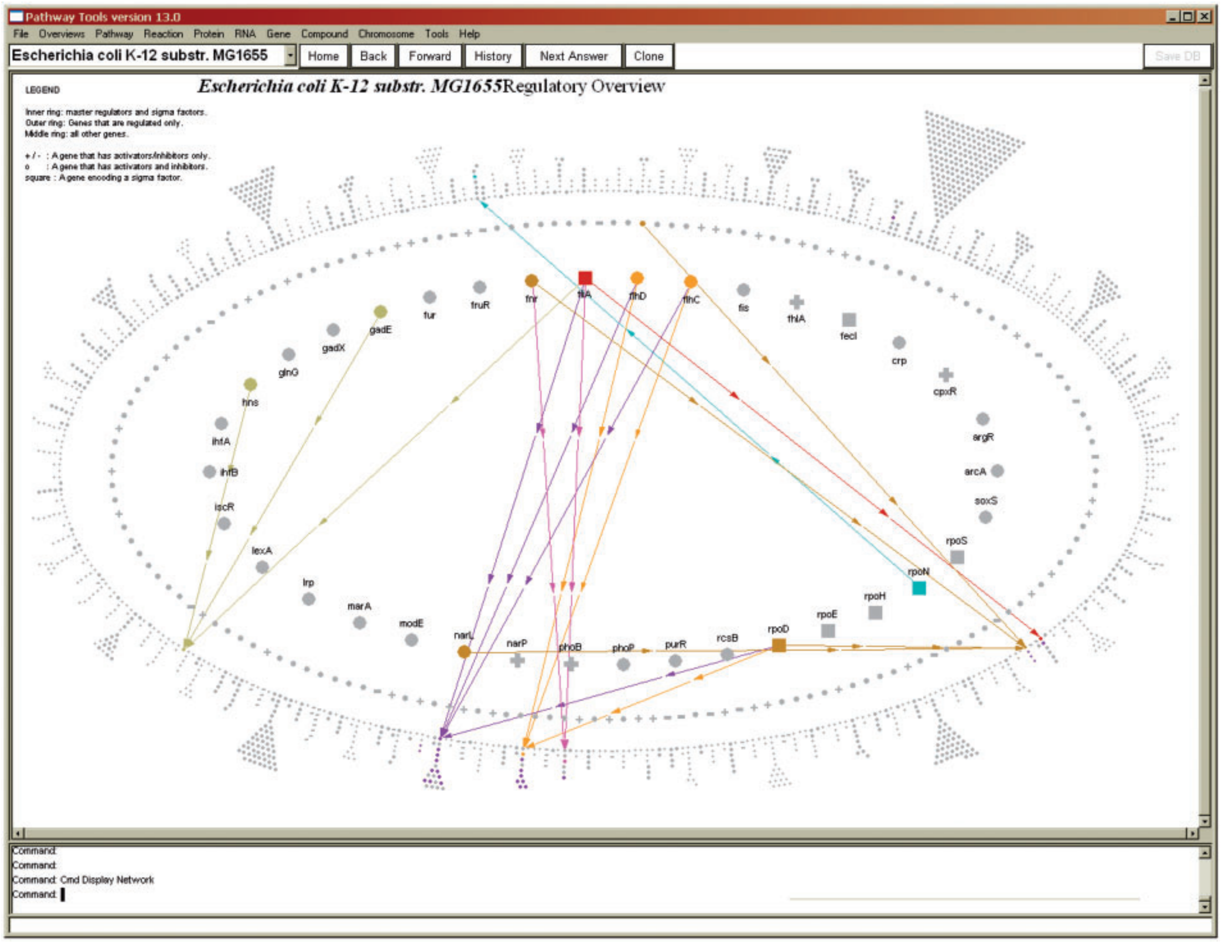
an organism (online example: [44]; example with animated display of omics data: [45]). Each node in the diagram represents a single metabolite, and each line represents a single bioreaction. Omics data (e.g. gene expression or metabolomics measurements) for a given organism can be painted onto the cellular overview to place these data in a pathway context and to allow the user to discern the coordinated expression of entire pathways [such as the tricarboxylic acid (TCA) cycle], or of important steps within a pathway. Omics data may be loaded from a data file and superimposed on the Overview diagram for that organism.

Cellular Overview diagrams are generated automatically using an advanced layout algorithm [3]. Automated layout is essential to allow the diagram to accurately depict the underlying DB content as that content evolves, without requiring time-consuming manual updates by curators that are bound to overlook some updates. In addition, automated layout allows generation of organism-specific cellular overviews that reflect the exact pathway content of each organism-specific PGDB in large PGDB collections such as BioCyc.

The Cellular Overview in the desktop version of Pathway Tools has many capabilities that are described in more detail in Paley and Karp [3]. These capabilities include semantic zooming of the diagram (where the highest magnification corresponds to the detail shown in the poster version), highlighting of user-requested elements of the diagram (such as metabolites or pathways), highlighting large biologically relevant subnetworks [such as all reactions regulated by a given transcription factor, and the results of a reachability analysis (see 'Network reachability analysis and dead-end metabolite analysis' section)] and highlighting comparative analysis results such as comparison of the metabolic networks of two or more PGDBs.

System-level visualization of regulatory networks

The Pathway Tools Regulatory Overview depicts the full transcriptional regulatory network stored in a PGDB in one screen, and allows the user to interrogate and explore relationships within



*Escherichia coli K-12 substr. MG1655*Regulatory Overview

LEGEND

Inner ring: master regulators and sigma factors.
Outer ring: Genes that are regulated only.
Middle ring: all other genes.

+ / - : A gene that has activators/inhibitors only.
o : A gene that has activators and inhibitors.
square : A gene encoding a sigma factor.

```
Command:
Command:
Command: Cmd Display Network
Command:
```

Figure 10: The Pathway Tools Regulatory Overview diagram for EcoCyc. The diagram depicts a full regulatory network as three concentric rings: the inner ring contains master regulator genes; the middle ring contains other regulators; and the outer ring contains genes that are not regulators. An arrow (edge) from gene A to gene B indicates that gene A regulates gene B. Initially, no arrows are shown; the user can interactively add arrows, such as by clicking on a gene and requesting that arrows are added to genes that it regulates, or from the genes that regulate it.

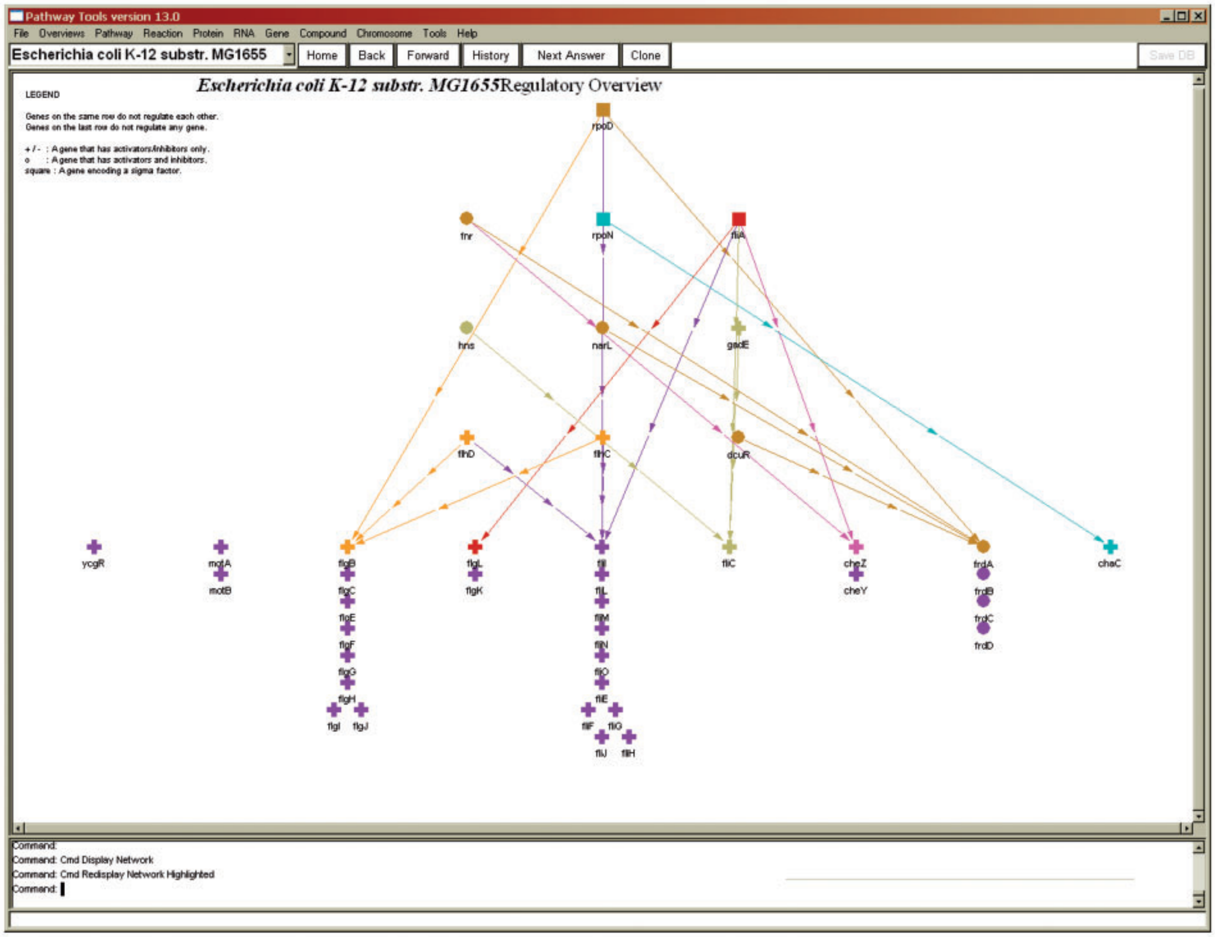
the network. Figure 10 shows the Regulatory Overview for EcoCyc, after the user has asked the system to highlight all genes annotated under GO term GO:0001539 (ciliary or flagellar motility). We can see that a few transcription factors control all *E. coli* motility genes.

The user can also request that the system display a separate diagram containing only those genes that are highlighted in the full Regulatory Overview. The resulting 'layer cake layout', shown in Figure 11, shows the regulators in a set of layers, such that no two genes in the same layer regulate one another.

System-level visualization of genome maps

The Pathway Tools genome browser displays a selected replicon, and allows the user to zoom into a region of the chromosome by gene name or by coordinates. The browser supports semantic zooming: as the user moves deeper into the genome, additional features are displayed, such as promoters and terminators. It can be used in a comparative mode that displays replicon regions centered on orthologous genes across a user-specified set of genomes to show the genomic context of those genes (e.g. [46]). In comparative mode, the user retains the ability to navigate left or right in the genome, and to zoom in and out. The genome browser can also generate large-format genome posters in PDF format; an example for *C. crescentus* is provided as Supplementary File S2.

The genome browser also supports display of tracks, meaning the ability to view positional data from external files along the genome, such as viewing predicted transcription factor binding sites.



Pathway Tools version 13.0
File Overviews Pathway Reaction Protein RNA Gene Compound Chromosome Tools Help
Escherichia coli K-12 substr. MG1655
Home Back Forward History Next Answer Clone Save DB
*Escherichia coli K-12 substr. MG1655*Regulatory Overview
rpoD
fnr
rpoN
fliA
hns
narL
gadE
flhD
flhC
dcuR
ycgR
motA
motB
flgB
flgC
flgE
flgF
flgG
flgH
flgI
flgJ
flgL
flgK
fliL
fliM
fliN
fliO
fliE
fliF
fliG
fliJ
fliH
fliC
cheZ
cheY
frdA
frdB
frdC
frdD
cheC

LEGEND

Genes on the same row do not regulate each other.
Genes on the last row do not regulate any gene.

+ / - : A gene that has activators/inhibitors only.
o : A gene that has activators and inhibitors.
square : A gene encoding a sigma factor.

```
Command:
Command: Cmd Display Network
Command: Cmd Redisplay Network Highlighted
Command: ▌
```

Figure 11: Layer cake layout of the regulatory network from Figure 10.

A user who zooms out far enough is presented with a depiction of all the genes on the replicon called the Genome Overview, shown in Figure 12. This diagram can be painted with omics data to provide a global genome view of large-scale datasets.

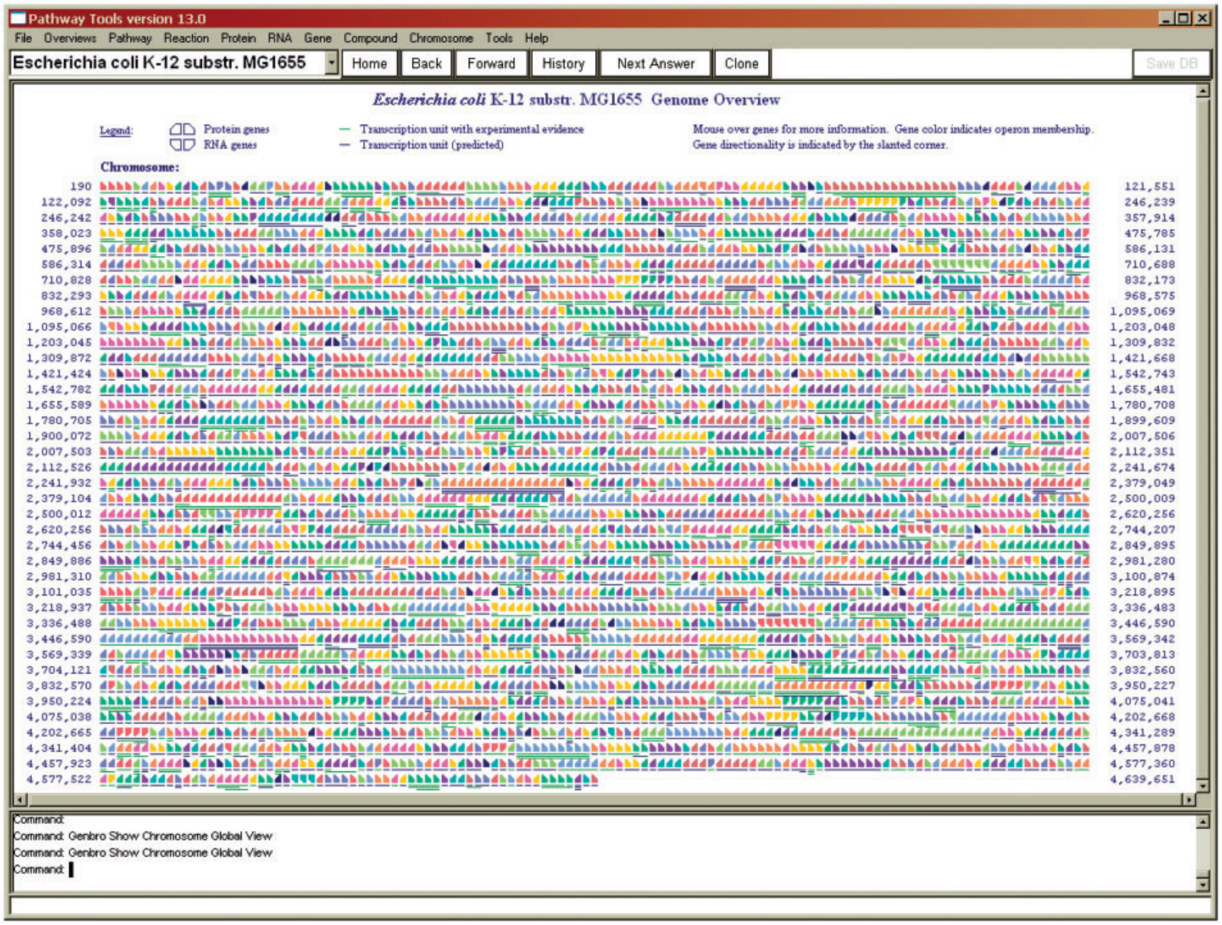
COMPUTATIONAL ACCESS TO PGDBs

In addition to the user-friendly graphical interfaces to PGDBs provided through the web and desktop versions of Pathway Tools, the software supports several formats for importing and exporting data, and allows Perl, Java and Lisp programmers to construct programs that access and update PGDB data (Figure 13).

Programmatic access through APIs: Programmers can access and update PGDB data directly [21, 47] by writing programs in the Java, Perl and Common Lisp languages. Java and Perl queries are executed using systems called JavaCyc [48] and PerlCyc [49].

Downloadable files in multiple formats: Pathway Tools can export PGDBs into several file formats that we have developed, which include tab-delimited tables and an attribute-value format [50]. Pathway Tools can also export subsets of PGDB data to other common formats including SBML [51], BioPAX [52], Genbank [53], and FASTA.

Relational DB access via BioWarehouse: For scientists who want to query PGDB data through a relational DB system, the attribute-value files exported by Pathway Tools can be loaded into SRI's BioWarehouse system [22]. BioWarehouse is an Oracle or MySQL-based system for integration of multiple public bioinformatics DBs. PGDB data can be queried through BioWarehouse alone or in combination with other bioinformatics DBs such as UniProt, Genbank, NCBI Taxonomy, ENZYME and KEGG.


Pathway Tools version 13.0
File Overviews Pathway Reaction Protein RNA Gene Compound Chromosome Tools Help
Escherichia coli K-12 substr. MG1655
Home
Back
Forward
History
Next Answer
Clone
Save DB

Escherichia coli K-12 substr. MG1655 Genome Overview

Legend: Protein genes RNA genes — Transcription unit with experimental evidence — Transcription unit (predicted) Mouse over genes for more information. Gene color indicates operon membership. Gene directionality is indicated by the slanted corner.

Chromosome:

```
      190    121,551
  122,092    246,239
  246,242    357,914
  358,023    475,785
  475,896    586,131
  586,314    710,688
  710,828    832,173
  832,293    968,575
  968,612  1,095,069
1,095,066  1,203,048
1,203,045  1,309,832
1,309,872  1,421,668
1,421,424  1,542,743
1,542,782  1,655,481
1,655,589  1,780,708
1,780,705  1,899,609
1,900,072  2,007,506
2,007,503  2,112,351
2,112,526  2,241,674
2,241,932  2,379,049
2,379,104  2,500,009
2,500,012  2,620,256
2,620,256  2,744,207
2,744,456  2,849,895
2,849,886  2,981,280
2,981,310  3,100,874
3,101,035  3,218,895
3,218,937  3,336,483
3,336,488  3,446,590
3,446,590  3,569,342
3,569,339  3,703,813
3,704,121  3,832,560
3,832,570  3,950,227
3,950,224  4,075,041
4,075,038  4,202,668
4,202,665  4,341,289
4,341,404  4,457,878
4,457,923  4,577,360
4,577,522  4,639,651
```

```
Command:
Command: Genbro Show Chromosome Global View
Command: Genbro Show Chromosome Global View
Command: ▌
```

Figure 12: Pathway Tools Genome Overview diagram for EcoCyc. Adjacent genes drawn in the same color are in the same operon. Left/right gene direction indicates transcription direction; up/down gene direction indicates genes coding for proteins versus RNAs. Horizontal lines under genes indicate transcript extents based on promoter and terminator information in the PGDB.

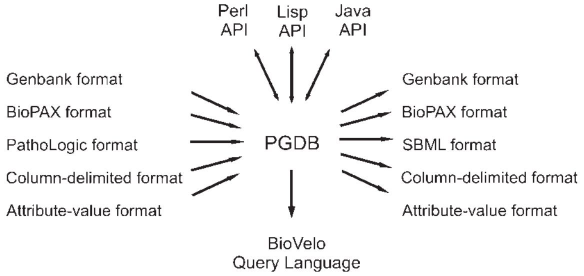


Figure 13: Pathway Tools supported formats for data import/export, APIs for data access/update and DB query language (BioVelo).

Queries using the Pathway Tools query language, BioVelo: Pathway Tools provides a powerful DB query language for querying PGDBs, called BioVelo [54]. BioVelo queries can be issued through an interactive Web form, and through APIs.

SYSTEMS BIOLOGY ANALYSES

This section describes Pathway Tools modules for performing system-level analyses of biological networks.

Metabolite Tracing

The Metabolite Tracing facility enables users to interactively trace the path of a metabolite through the metabolic network and to view it on the cellular overview diagram. Since the metabolic network is highly interconnected, there will typically be many such paths. Rather than attempting to trace all of them at once, this facility stops at branch points to allow the user to select which one or more paths should be followed. This decision is not irrevocable—at any point, the user can elect to return to one of the previously not selected branches and follow it instead.

The user specifies a starting metabolite and a trace direction (either forward or backward). The software will highlight the path from that metabolite until it reaches a branch point on the cellular overview diagram. At this point, all possible steps that can be followed will be highlighted in a different color, and a checklist of resulting compounds will appear in the control panel. The user can select which path(s) to follow either by clicking in the overview diagram or by selecting compounds in the control panel. This process proceeds interactively until a dead end is reached or the user decides to stop. The beginnings of paths that were not followed continue to be shown in another color, in case the user changes his mind about which path to follow.

Alternatively, a user can request to follow all paths from the specified metabolite, for a certain number of iterations. In this mode, for many starting metabolites, the overview diagram rapidly becomes so thoroughly colored that it is difficult to follow any single path. The user can then select from a list of metabolites encountered during the search, and ask to just show the path to that metabolite from the starting metabolite (there is no guarantee, however, that all such paths will be shown).

The overview can become hard to read when many connections exist between reactions in different pathways along a path. Thus, Pathway Tools provides a command to display a specified path in a window by itself. This command creates a temporary pathway object, consisting of just the reactions in the current path. This temporary pathway is shown in a new window and can be viewed or printed the same way as any other pathway. A sample traced path, showing it both as it appears on the overview diagram and as a pathway object, is shown in Figure 14.

Network reachability analysis and dead-end metabolite analysis

Both of the tools described here support validation of metabolic network models by computation of systems properties of those models. One application of these tools is to check whether a metabolic-network model is sufficiently well formed for flux-balance analysis [55, 56]. For example, if a model contains metabolites that are not reachable from a given growth medium, those metabolites could not be produced in a flux-balance model generated from that metabolic network.

The reachability analysis tool allows the user to ask what product metabolites are reachable through a series of reaction transformations from a specified set of input metabolites. The tool can be used to identify gaps in the metabolic network, and to identify discrepancies between experimentally determined growth media for an organism, and computationally determined growth media.

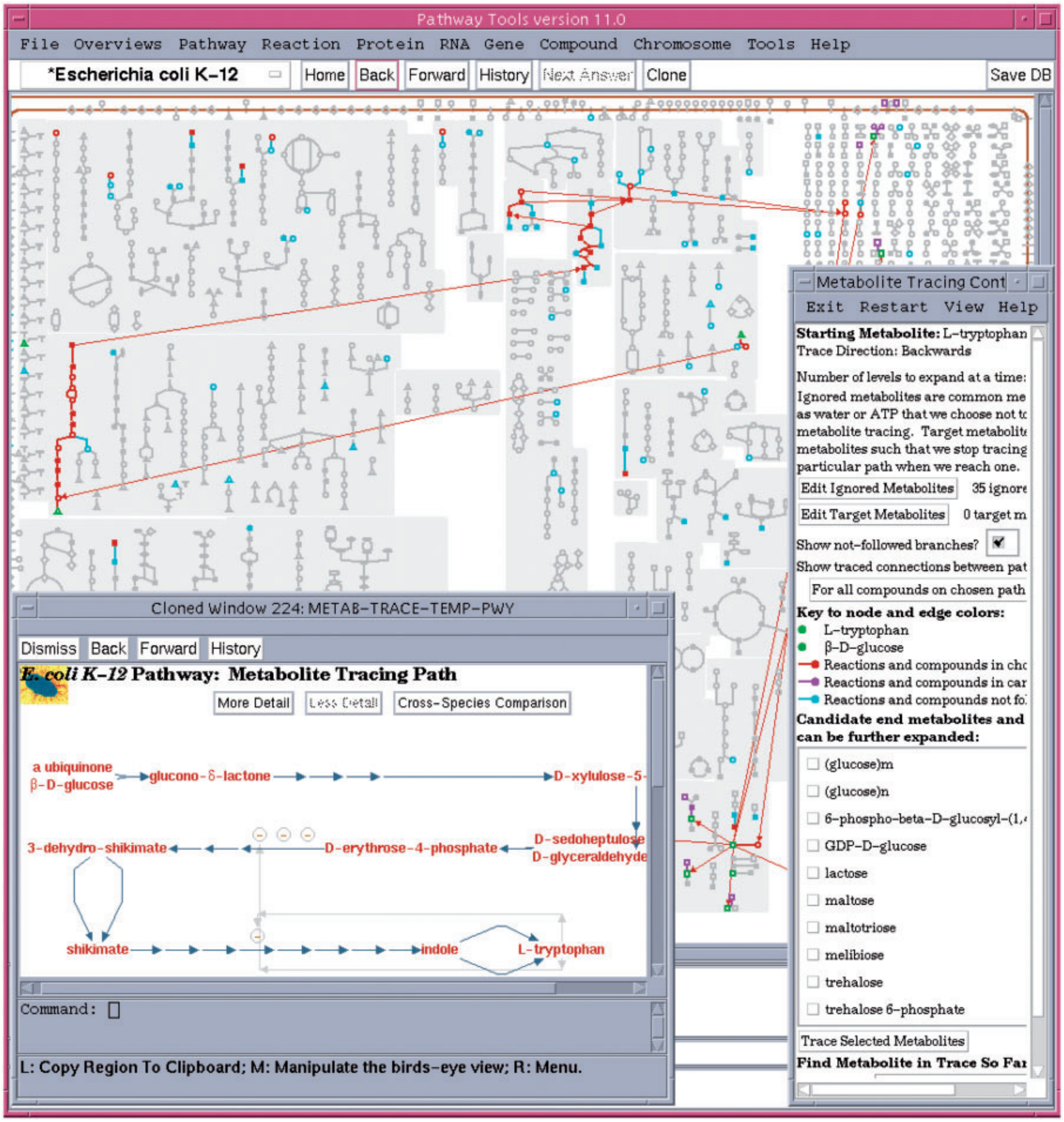
The user specifies a set of starting metabolites, using a graphical interface, which form the initial metabolite pool. Next, the system converts the reaction network within the PGDB into a system of production rules, and it repeatedly chooses an unfired rule, checks if all of its inputs are present in the metabolite pool, and if so fires the rule by adding all of its products to the metabolite pool [25]. The metabolite pool is qualitative; it includes no concentrations. This process repeats until no additional rules fire.

The results of a reachability analysis can be displayed on the cellular overview diagram as shown in Supplementary Figure S8. Furthermore, the cellular overview becomes a debugging tool: a user who is curious why a given reaction did not fire moves the mouse over a reaction line in the diagram, and the system displays a pop-up window that shows the full reaction equation and lists which reactants are present in the metabolite pool and which are not.

The reachability tool will also compute the difference between a set of expected output metabolites provided by the user, and the actual list of produced metabolites, and will track that difference over a series of reachability runs to track the user's progress in resolving unproduced compounds.

The chief limitation of the reachability tool is that its qualitative approach has difficulty with cycles in the metabolic network [25]. For example, consider ATP, which is required for its own biosynthesis (glycolysis consumes two ATP before producing four ATP). Thus, ATP must be provided as an input for all glycolysis reactions to fire, yet doing so raises the possibility that ATP could be broken down to supply carbon, nitrogen or phosphorus to the cell.

A related form of metabolic-network analysis is detection of dead-end metabolites, which informally are metabolites that are only produced by the metabolic network, or only consumed by the network.


Pathway Tools version 11.0
File Overviews Pathway Reaction Protein RNA Gene Compound Chromosome Tools Help
*Escherichia coli K-12
Home Back Forward History Next Answer Clone Save DB
Metabolite Tracing Cont
Exit Restart View Help
Cloned Window 224: METAB-TRACE-TEMP-PWY
Dismiss Back Forward History
More Detail Less Detail Cross-Species Comparison
a ubiquinone
β-D-glucose
glucono-δ-lactone
D-xylulose-5-
D-sedoheptulose
D-glyceraldehyde
D-erythrose-4-phosphate
3-dehydro-shikimate
shikimate
indole
L-tryptophan

E. coli K-12 Pathway: Metabolite Tracing Path

Command: ▯

L: Copy Region To Clipboard; M: Manipulate the birds-eye view; R: Menu.

Starting Metabolite: L-tryptophan
Trace Direction: Backwards

Number of levels to expand at a time:
Ignored metabolites are common me
as water or ATP that we choose not tc
metabolite tracing. Target metabolite
metabolites such that we stop tracing
particular path when we reach one.

Edit Ignored Metabolites 35 ignore
Edit Target Metabolites 0 target m

Show not-followed branches? ☑
Show traced connections between pat
For all compounds on chosen path

Key to node and edge colors:
- ● L-tryptophan
- ● β-D-glucose
- —● Reactions and compounds in chc
- —● Reactions and compounds in car
- —● Reactions and compounds not fo

Candidate end metabolites and can be further expanded:

- □ (glucose)m
- □ (glucose)n
- □ 6-phospho-beta-D-glucosyl-(1,
- □ GDP-D-glucose
- □ lactose
- □ maltose
- □ maltotriose
- □ melibiose
- □ trehalose
- □ trehalose 6-phosphate

Trace Selected Metabolites
Find Metabolite in Trace So Far

Figure 14: Metabolite Tracing Facility. The main window shows the cellular overview diagram, highlighting the path backward from tryptophan to glucose. The control panel, which includes a color key and options for extending or altering the trace, is superimposed on the main window, on the right. The bottom left window shows the traced path using the conventional pathway display.

We provide two tools that identify dead-end metabolites that have complementary strengths and weaknesses.

The first tool is useful when a PGDB is not expected to contain very complete or reliable information on reaction directions (The accuracy of reaction direction information is largely a function of how extensively the PGDB has been curated. Although reaction direction can be inferred from the direction of a reaction within a pathway, many reactions in MetaCyc are not part of metabolic pathway, and have no assigned reaction directions. Furthermore, a reaction that was predicted to occur in one direction in the organism based on its stored direction in MetaCyc might in fact occur in the opposite direction in that organism. A recent paper by Maranas and colleagues [57] notes that even for *E. coli* metabolic models, reaction direction information is sometimes incorrect). It computes dead-end metabolites according to the following more limited definition. A small-molecule metabolite *M* is a dead-end metabolite in the cellular compartment C if and only if all the following conditions are true:

(1) M is a substrate in only one reaction of the set of small-molecule reactions occurring in C.
(2) No other reaction exists containing a parent class of M.
(3) M is not transported into C, nor are parent classes of M.
(4) No enzyme in the PGDB uses M as a cofactor (since acting as a cofactor is an expected biological end use of synthesized compounds that might otherwise be viewed by the system as dead ends).

The second tool uses a more comprehensive definition of dead-end metabolites that requires more comprehensive information on reaction direction. A small-molecule metabolite M is a dead-end metabolite in the cellular compartment C if and only if one of the following conditions is true:

(1) M or parent classes of M are only consumed by small-molecule reactions occurring in C, and M or parent classes of M are not transported into C.
(2) M or parent classes of M are only produced by small-molecule reactions occurring in C, and M or parent classes of M are not transported out of C, and no enzyme in the PGDB uses M as a cofactor.

Prediction of network choke points

One application of a metabolic network model is to find network bottlenecks, which if blocked could kill the cell. Such bottlenecks could constitute antimicrobial drug targets. We have developed a tool for predicting these so-called choke points.

The Pathway Tools choke-point detection algorithm examines the reactions attached to a given metabolite, and processes one metabolite at a time. The first step is to assemble the list of metabolites to examine. This is done by collecting (i) all reactions that are in pathways, plus (ii) reactions that stand alone, but which use only small molecule metabolites. The reactions that came from pathways may use some macromolecular substrates, such as proteins that are modified by the reaction. From this list of reactions, the algorithm collects all of their substrate metabolites (meaning their reactant or product metabolites).

Definition [26]: A 'choke point reaction' is a reaction that either uniquely consumes a specific substrate or uniquely produces a specific product in a metabolic network, and is also balanced by at least one reaction that respectively produces or consumes that substrate. Specifically, the algorithm searches for two types of choke point reactions: (a) Reactions R_1 such that only a single reaction R_1 produces metabolite M, and at least one reaction consumes M. (b) Reactions R_2 such that only a single reaction R_2 consumes metabolite M, and at least one reaction produces M. These definitions imply that to find a choke point, all reactions involving M must be unidirectional. These choke point reactions are collected and returned as the result. Note that the definition excludes reactions directly connected to dead-end metabolites.

The resulting candidate choke point reactions can be painted onto the cellular overview to facilitate further analysis.

Comparative tools

Pathway Tools contains a rich set of operations for comparing the information in two or more PGDBs. These operations range from comparison of genome-related information to comparison of pathway information. These comparisons are of several types.

The comparative genome browser discussed in 'Systems-level visualization of genome maps' section displays replicon regions centered on orthologous genes across a set of genomes (Supplementary Figure S9).

The user can generate a comparative table for a given metabolic pathway across a specified set of organisms. For each organism, the table shows the presence of pathway enzymes and operon structures of genes within the pathway.

A global comparison of the metabolic networks of multiple PGDBs can be performed by highlighting on the Cellular Overview diagram (see 'System-level visualization of metabolic networks' section). This tool allows the user to highlight in the Cellular Overview reactions that are shared, or not shared, among a specified set of organisms.

Finally, a general comparative analysis facility allows the user to generate comparative report tables for many aspects of a PGDB. As well as being used for comparative analyses, these tools can be used to generate statistics regarding the content of a single PGDB. These tools are general in that they present their results in a standard format, and they allow the user to drill down to specific results in a consistent fashion. The initial report page shows summary statistics, but the user can drill down to compare all instances of a category by clicking on elements of a report table.

For example, consider the transporter report page in Supplementary Figure S10. Table 2 within that report summarizes the number of uptake transporters found in two organisms. A user who wants to see the actual transported substrates clicks on the text 'Compounds transported into the cell' to generate a new report page containing a table listing the union of all substrates imported by both organisms, along with an indication of which organisms transport each substrate, and which transporter is utilized. If the user clicks on a data cell within Table 2, such as the number of imported substrates in *E. coli* K-12 (156), a page is generated that lists those substrates only. Similar functionality applies to most tables in these reports.

The following report types are provided. An example comparative report is available at URL [58].

- Reaction report includes the following statistics for each selected organism:
  - number of reactions containing substrates of different types, e.g. reactions for which all substrates are small molecules, and for which some substrate is a protein or a tRNA;
  - number of reactions in each EC category;
  - number of reactions containing different numbers of isozymes.
- Pathway report includes these statistics:
  - number of pathways in each category within the MetaCyc pathway ontology;
  - number of pathways with different numbers of pathway holes.
- Compound report includes these statistics:
  - frequency with which different compounds appear in different metabolic roles (substrate, cofactor, inhibitor and activator).
- Protein report includes these statistics:
  - general statistics on number of monomers versus multimeric complexes, breakdown of multimers into heteromultimers and homomultimers;
  - statistics on multifunctional enzymes.
- Transporter report includes these statistics:
  - number of efflux versus influx transporters;
  - number of genes whose products are transporters;
  - number of unique transported substrates, both overall and broken down by efflux versus influx;
  - number of transported substrates that are substrates in metabolic pathways or are enzyme cofactors;
  - transporters with multiple substrates, and substrates with multiple transporters;
  - operon organization of transporters.
- Ortholog report includes these statistics:
  - list of all orthologous proteins across the selected organisms;
  - proteins that are shared in all selected organisms, or unique to one organism.
- Transcription Unit report includes these statistics:
  - distribution of number of genes per transcription unit;
  - distribution of number of operons into which metabolic pathway genes are distributed.

SOFTWARE AND DB ARCHITECTURE

Pathway Tools is implemented in the Common Lisp programming language (we use the Allegro Common Lisp implementation from Franz Inc., Oakland, CA, USA). We chose Common Lisp because it is a high-productivity programming environment. Because Lisp is a very high-level language, one line of Lisp code is equivalent to several lines of code in a language such as Java or C++. Therefore, the same program can be written more quickly in Lisp, with fewer bugs. A study by Gat [59] found that compiled Lisp programs generally run faster than Java programs, and that a given program can be developed two to seven times faster in Lisp than in Java [59]. Common Lisp also has a very powerful interactive debugging environment.

Lisp has powerful dynamic capabilities that are illustrated by a Pathway Tools feature called autopatch. Imagine that a Pathway Tools user site has reported a bug in the software. Once our group has found a fix for the bug, we put a patch file that re-defines the offending Lisp function(s) on the SRI web site. The next time Pathway Tools is started at remote sites, it automatically downloads the patch (in compiled form) from the SRI web site, puts the patch in an appropriate directory and dynamically loads the patch file into the running Pathway Tools.

Pathway Tools consists of 450 000 lines of Common Lisp code, organized into 20 subsystems. In addition, 7000 lines of Javascript code are used within the Pathway Tools web interface. Pathway Tools runs on the following platforms: Macintosh (32-bit, 64-bit), Linux (32-bit, 64-bit) and Microsoft Windows (32-bit).

The architecture of Pathway Tools is depicted in Figure 15. The main bioinformatics modules of Pathway Tools are the Navigator, Editors and PathoLogic, plus a chemoinformatics subsystem that includes tools such as SMILES [60] generation and parsing and a chemical substructure matcher, plus a large set of shared utilities that we call the Pathway Tools core. Pathway Tools uses an object-oriented DB system called Ocelot. The Pathway Tools user interface relies on a graph layout and display package called Grasper [61], and web and desktop graphics packages called CWEST and CLIM (the Common Lisp Interface Manager).

Ocelot is an object/relational DB management system (DBMS) developed at SRI [1, 28]. Ocelot combines the expressive power of frame knowledge representation systems [62] developed within the AI community [whose object data model is far superior to the relational data model for representing biological data Superior aspects of the object data model include the following. The object data model is better at managing very complex schemas. That is, if the same domain is represented within the object data and within the relational model, the object schema is usually much more compact and easier to comprehend. One reason is that inheritance allows the object data model to define subclasses by extending existing classes (e.g. the class Polypeptides is a subclass of the class Proteins), whereas the relational model would force attributes shared between the two tables to be duplicated in each, which both obscures the fact that the two tables are related, and complicates schema evolution. Relational normalization also increases the size of the schema by forcing the creation of new tables for every multivalued attribute, which is not required in the object data model. The object data model used by Ocelot is particularly flexible in supporting any type of schema evolution without forcing the entire DB to be reloaded (unlike relational DBMSs), which is important in bioinformatics because the complexity of biological data forces never-ending enhancements to the schema (note that not every object DBMS provides such flexibility)] with the scalability of relational DB management systems (RDBMSs). Ocelot DBs are persistently stored within an Oracle or MySQL RDBMS. Ocelot objects are faulted on demand from the RDBMS, and in addition are faulted by a background process during idle time. Objects that were modified during a user session are tracked and saved to the RDBMS during a save operation. Ocelot uses optimistic concurrency control [28]—during a save operation it checks for conflicts between the updates made by the user and updates saved by other users, since the saving user began their session or last made a save operation. This approach avoids the overhead of locking that becomes problematic in object DBs because modifications to one object often cascade to related objects and could require a large number of lock operations. The optimistic concurrency control works well in practice because curators tend to focus in different biological areas and therefore rarely update the same objects at the same time.

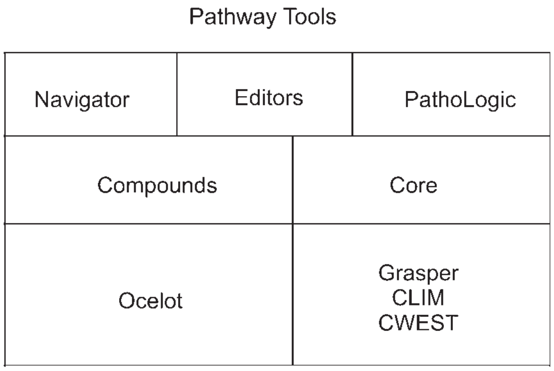


Figure 15: Pathway Tools architecture (vertical lines have no meaning other than separation of components; for example, all components in the top layer call all components in the middle layer). Each box depicts a major component of Pathway Tools. The bottom layer includes Ocelot and the graphics components. Above that layer are low-level libraries within Pathway Tools for manipulating chemical compounds and other datatypes, which in turn are called by the Navigator, Editors and PathoLogic.

Ocelot DBs can also be saved to disk files, in which case the RDBMS is not needed (Figure 16). The file persistence configuration is simpler to use, since it does not require purchase or installation of an RDBMS. It provides an easy and low-cost way to begin a PGDB project; a project can switch to an RDBMS configuration as its complexity grows. The advantage of an RDBMS configuration is that it provides Ocelot with multi-user update

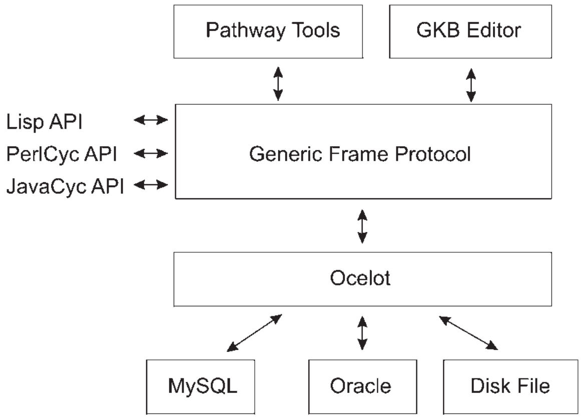


Figure 16: Storage architecture of Pathway Tools.

capabilities, and it permits incremental (and therefore faster) saving of DB updates. The RDBMS configuration also allows Ocelot to maintain a history of all DB transactions—DB curators can examine the history of all updates to a given object to determine when a given change was made, and by whom. This functionality is very useful when diagnosing mistakes within a PGDB.

Figure 17 shows the graphics architecture of Pathway Tools. The Grasper graph toolkit is used in pathway layouts, and in the cellular overview and regulatory overview. Grasper graphics, and all other graphics generated by Pathway Tools, are rendered using the CLIM Common Lisp graphics system, which is implemented using the X window system on Linux and Mac, and the native Windows API on Windows. When Pathway Tools runs as a desktop application, CLIM graphics directly update the user's screen.

Pathway Tools can also run as a web server, which is how it powers web sites such as BioCyc.org. Pathway Tools uses a somewhat nonstandard model of servicing web requests, and it does not run with an associated HTTP server such as Apache. Instead, Pathway Tools provides a fully functional web server that includes services such as compression and connection keep alive.

HTTP servers typically start a new process for each incoming web request that terminates after the request has been serviced. For bioinformatics DBs, the web server often issues a query to a relational DB server that runs as a separate process. In contrast, Pathway Tools starts one long-lived web server process that can service many thousands of web requests, with internal threads used to service overlapping requests. All DBs are stored in Pathway Tools virtual memory, and thus access is extremely fast. To date, this model has scaled to hundreds of genomes, although we are investigating other approaches to provide additional scalability.

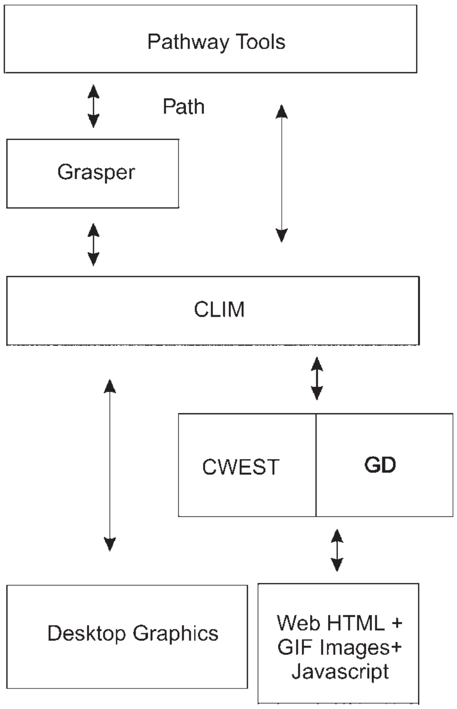


Figure 17: Graphics architecture of Pathway Tools.

Pathway Tools processes an incoming web request in the following manner. The top-level directory name within an incoming URL indicates whether the operation is requesting a static file or dynamically generated page.

- Static files: A small number of web pages, such as the home page and informational pages for BioCyc.org, are implemented as disk files. Pathway Tools can serve file-based web pages like a traditional web server.
- Dynamically generated pages: Most Pathway Tools pages are generated dynamically by querying PGDBs within the Pathway Tools virtual memory and generating query outputs and visualizations, often using the same code as for desktop mode. The CLIM graphics generated by software such as the pathway layout code are dynamically converted to HTML and GIF images using CWEST [63], which uses GD [64] to generate GIF images. The resulting HTML and GIF images are returned to the user's web browser, and the Pathway Tools web server awaits the next query. The GIF images include generated

specifications of mouse-sensitive regions and of what operations should be invoked when the user clicks on such a region.

The Pathway Tools web interface includes significant amounts of JavaScript code, and uses the Yahoo User Interface library (YUI) [65].

SURVEY OF PATHWAY TOOLS COMPATIBLE DBs

PGDBs for many hundreds of organisms from all domains of life are available for use in conjunction with Pathway Tools. This summary lists what DBs are available, and their sources.

With highly curated PGDBs available for many important organisms, it is not clear why users would consider using the uncurated (and therefore lower quality) pathway DBs available for these same organisms from other pathway DB providers such as KEGG. For example, consider the highly curated AraCyc pathway DB for *A. thaliana* [66, 67]. AraCyc contains minireview summaries for enzymes and metabolic pathways; thousands of literature references; evidence codes for enzyme functions and metabolic pathways (indicating which pathways are supported by experimental evidence); and information on enzyme subunit structure, activators, inhibitors and cofactors. KEGG contains none of the preceding information. In addition, AraCyc curators have carefully refined the metabolic reactions and pathways present in AraCyc, including removing false positive computational predictions, and to add *Arabidopsis* reactions and pathways from the biomedical literature to AraCyc. Although KEGG updates its reference pathway map diagrams periodically to contain new pathways and reactions from different organisms, the KEGG approach of computationally coloring reactions within pathway maps based on the presence of enzymes for those reactions within a genome results in significant ambiguity. If AraCyc curators are reasonably certain that a reaction or pathway is absent from *Arabidopsis*, they remove it from the DB. The KEGG model does not allow such removal, so it is never clear within KEGG whether an uncolored reaction is truly absent from an organism, or whether the gene for its enzyme has not yet been identified in the genome. This situation results in a real conundrum for a scientist who wishes to assemble the list of reactions likely to be present in *Arabidopsis* from KEGG, since there is no way to distinguish the many uncolored reactions that are likely present but for which no gene has been identified, from the many uncolored reactions that are clearly known to be absent from *Arabidopsis* (which curators have deleted from AraCyc).

Available PGDBs include the following, with curated PGDBs in bold.

- Animals
  - ***Homo sapiens* (HumanCyc) [68, 69]**
  - ***Mus musculus* (MouseCyc) [70]**
  - ***Bos taurus* (CattleCyc) [71, 72]**
- Plants
  - ***A. thaliana* (AraCyc) [66, 67]**
  - ***Medicago truncatula* [73, 74]**
  - ***Oryza sativa* and *Sorghum bicolor* [75]**
  - Multiple *Solanaceae* species [76]
- Other eukaryotes
  - ***Dictyostelium discoideum* [77]**
  - ***Saccharomyces cerevisiae* (YeastCyc) [78]**
  - ***Candida albicans* [79]**
  - ***Leishmania major* [80]**
- Microbes
  - ***E. coli* (EcoCyc) [6, 81]**
  - ***Streptomyces coelicolor* [82]**
  - ***Pseudomonas aeruginosa* (PseudoCyc) [9]**
  - **8 PGDBs for *Plasmodium, Cryptosporidium*, and *Toxoplasma* at ApiDB [83]**
  - **5 PGDBs for *Brucella suis, Coxiella burnetii*, and *Rickettsia typhi* at the PATRIC BRC [84]**
  - 500 PGDBs at BioCyc [7, 20]
  - 250 PGDBs at MicroCyc [85]
  - 48 PGDBs at the Taxonomically Broad EST DB [86]
  - 72 PGDBs at the Pathema BRC [87]

To facilitate sharing of PGDBs among multiple users, we have created a PGDB Registry that enables peer-to-peer sharing. PGDB sharing is desirable because a user whose own computer has a copy of a PGDB can use Pathway Tools functionality that would not be available through a remote Pathway Tools web server, such as functionality that exists in desktop mode only or comparative operations. Comparative analysis of two or more PGDBs is possible only when they are loaded into the same instance of Pathway Tools.

The PGDB Registry uses a server maintained by SRI that tracks the locations of available PGDBs that PGDB authors have registered for downloading. The author of a PGDB can register that PGDB by using a command within Pathway Tools that creates an entry for the PGDB in the Registry server, and places the PGDB on an FTP or HTTP server of the author's choosing. Users who want to download a PGDB from the Registry can view available PGDBs by using a Web browser (see URL in [88]) or using Pathway Tools itself. With a few mouse clicks, a user can download a PGDB from the registry using Pathway Tools.

COMPARISON WITH RELATED SOFTWARE ENVIRONMENTS

Pathway Tools stands out with respect to related software tools in the breadth of the functionality and the high level of integration that it provides. It addresses a very large number of use cases. And it provides schema, visualization and editing support for an unusually large number of datatypes in addition to pathways, including chromosomes, genes, enzymes, transporters and regulatory networks. Pathway Tools is particularly well adapted for microbes, with its support for operons and for prokaryotic gene-regulation mechanisms, and its genome browser is optimized for depicting prokaryotic genomes. The following comparison is organized according to the use cases presented in 'Pathway Tools use cases' section, although we consider the first two use cases together because they are strongly related.

Development, visualization and web publishing of organism-specific DBs

Metabolic pathway information

Other software systems for managing metabolic pathway information are KEGG [89, 90], PUMA2 [91] (inactive), Amaze [92] (inactive), GenMAPP [93, 94], PathCase [95, 96], VisANT [97], and Reactome [98–100]. KEGG, VisANT, PUMA2 and GenMAPP are based on static, predrawn pathway diagrams, a model that does not scale to produce custom pathway diagrams for tens of thousands of *different* pathways in different organisms. Nor can the static approach produce multiple views of a given pathway at different levels of detail, as can the Customize Pathway option in Pathway Tools that allows the user to choose exactly which graphical elements (e.g. gene names, EC numbers, metabolite structures, activators and inhibitors) appear in the pathway diagram.

PathCase and Reactome do have pathway layout capabilities, but the resulting diagrams bear little resemblance to those found in the biomedical literature, nor are they particularly compelling visually. They do not offer the customization or multiple-detail views offered by Pathway Tools.

Cytoscape [101] is a general tool for display of biological networks that embodies the philosophy that general graph layout techniques can satisfactorily depict any biological network. Although the Cytoscape layout algorithms are a terrific fit for display of protein interaction maps, we assert that they do not produce useful results for metabolic pathways. We believe that superior visualization results are obtained when the layout algorithm is specifically tailored to the data at hand. For example, Pathway Tools provides separate layout algorithms for circular, linear and tree-structured pathways to make the structure of those pathways stand out prominently to the biologist. Biologists developed their pre-computer depictions of metabolic pathways for important reasons, namely, to accurately depict subtleties of the data.

Most of the preceding tools lack pathway editing capabilities, exceptions being GenMAPP, Reactome, VisANT and PathCase. This limitation is a fundamental one for tools such as KEGG, for which users cannot introduce new organism-specific pathways, nor modify a reference pathway definition to customize it to a specific organism, thus eliminating the possibility of removing erroneous reaction steps from a pathway, or of adding missing reactions to a pathway.

No other tool except Reactome has analogs of our Cellular Overview diagram (which we introduced in 1999 [23, 44]), nor of our Omics Viewer capabilities. No other software system lays out its complete metabolic map diagram algorithmically as Pathway Tools does, providing the ability to generate custom diagrams for hundreds of genomes. KEGG provides a single overview metabolic map for all organisms in KEGG, as opposed to the organism-specific overviews that Pathway Tools generates through advanced layout algorithms. The KEGG diagram is not queryable or interactive as the desktop version of the Pathway Tools diagram.

All the preceding tools lack the metabolite tracing capabilities of Pathway Tools.

Genome and proteome information

Many existing bioinformatics systems include genome browsers and gene pages. A representative sample of larger systems includes GBrowse

[102, 103], IMG [104], Entrez Genome [105], CMR [106], the UC Santa Cruz Genome Browser [107], Ensembl [108] and PATRIC [109]. Here we compare the salient features of these genome browsers.

- Is there support for semantic zooming (e.g. do different visual features become visible as the user zooms in and out?)?
  - Yes: Pathway Tools and GBrowse. Limited semantic zooming (e.g. showing DNA sequence at the greatest zoom level): Entrez, UCSC and Ensembl.
  - No: IMG, CMR and PATRIC.
- Is there support for displaying custom tracks, to plot datasets against the genome?
  - Yes: Pathway Tools, GBrowse, UCSC and Ensembl.
  - No: IMG, Entrez, CMR and PATRIC.
- Is there support for wrapped multiline displays, to depict a large genome region in a space efficient manner?
  - Yes: Pathway Tools.
  - No: GBrowse, IMG, Entrez, CMR, UCSC, Ensembl and PATRIC.
- Is there depiction of introns versus exons?
  - Yes: GBrowse, Entrez, UCSC and Ensembl.
  - No: Pathway Tools, IMG, CMR and PATRIC.
- Is there support for a compact genome overview diagram, showing every gene?
  - Yes: Pathway Tools (showing multiple chromosomes) and CMR (although showing only one chromosome at a time).
  - No: GBrowse, IMG, Entrez, UCSC, Ensembl and PATRIC.
- Are there comparative genomics capabilities?
  - Yes: Pathway Tools, GBrowse (by additional tools like SynBrowse, SynView or GBrowse _syn), UCSC (by means of tracks) and Ensembl (by sequence clustering at a great zoom level).
  - No: IMG, Entrez, CMR and PATRIC.

Regulatory networks

A number of bioinformatics DBs include regulatory network information; however, the majority of these DBs and their associated software environments can represent information on transcription factor-based regulation only, such as RegTransBase [109], TRANSFAC [110], CoryneRegNet [111], ProdoNet [112], and DBTBS [113]. The exception is RegulonDB [114], which can also capture RNA-based regulation including riboswitches, attenuators and small RNA regulators.

We are not aware of tools comparable with the Regulatory Overview in being able to display and interrogate large complete cellular regulatory networks, although CoryneRegNet and ProdoNet display smaller regulatory networks. CoryneRegNet also displays omics data onto its regulatory network diagrams. Cytoscape could probably display regulatory network data using its generic graph display capabilities.

Query tools

Other bioinformatics DBs provide a subset of the three tiers of queries provided by Pathway Tools (quick search, object-specific searches and Structured Advanced Query Page). Virtually all provide a quick search. Sites providing particularly extensive object-specific searches are FlyBase [16], Mouse Genome Informatics [17], EuPathDB [115], and BioMart [116]. BioMart is used by bioinformatics DBs including WormBase, Rat Genome Database, UniProt, Reactome and Galaxy. Its underlying query language is Perl using the BioMart libraries. However, none of the preceding systems provides the query power of the Pathway Tools SAQP. For example, BioMart does not allow the user to construct arbitrary queries that perform joins (queries that combine multiple data types); it provides only the 'and' logical operator (the 'or' operator is not available); and it includes only a limited form of 'not'.

Biozon [117] (`biozon.org`) integrates several biological DBs and provides a web interface that is the closest in power to the SAQP. A query is created by first selecting an object type, entering some constraints for this type and then proceeding to another related object type if desired. That is, join operations between different types of objects are supported, making Biozon one of the few other bioinformatics web interfaces that allow joins. However, Biozon does not allow logical operators such as 'or' to be specified among all query components.

Extend genome annotations with additional computational inferences

KEGG and Reactome are the only other tools that can predict pathways from genome data. The pathway hole filler and transport inference parser tools are unique to Pathway Tools. Many genome annotation pipelines include operon predictors.

Analysis of omics data

Kono *et al*. [118] introduced a SVG-based tool for painting omics data onto individual KEGG pathway maps [118], although it does not paint onto whole-organism overview diagrams. This tool also does not produce animations as our omics viewers do. The Reactome Skypainter can paint omics data onto a human pathway overview that is customizable to other model organisms only by graying out regions of the overview (whereas Pathway Tools produces customized overviews for each organism). We argue that the utility of the Skypainter tool is compromised by its small size, and furthermore, it cannot display metabolomics data as the cellular overview can. GenMapp [119], VitaPad [120], VisANT and ArrayXPath [121] paint omics data onto single pathways, rather than onto a full metabolic overview.

Analysis of biological networks

The Palsson group has developed tools for detecting dead-end metabolites [122]. We are not aware of other groups that have developed tools for reachability analysis, although flux-balance analysis techniques are able to predict whether a metabolic network will support growth under a given growth medium [123].

Singh *et al*. [124] produced another implementation of our chokepoint method, although the availability of that software is unclear. Rahman and Schomburg [125] enhanced the chokepoint method with the additional concept of load points, which are the number of k-shortest paths and nearest neighbor links for a metabolite. Flux-balance models can also be used to predict essential reactions [123]. Kim *et al*. [126] combined flux-balance models with chokepoint analysis to predict drug targets.

Comparative analysis of organism-specific DBs

No other tools include comparative pathway analysis functionality such as that provided by Pathway Tools.

Comparative genomics is a very large area in bioinformatics. The ‘Genome and proteome information’ section compares Pathway Tools’ comparative genome browser with other tools. In general, other tools include a range of comparative genomics capabilities not found in Pathway Tools [104, 106].

Metabolic engineering

Pathway Tools contributes many relevant capabilities for metabolic engineering, such as fast development of comprehensive genome-scale models of the metabolic network of an engineered organism, and a tool for tracing the fates of metabolites through the metabolic network. It lacks the optimization and pathway design tools that have been developed by metabolic engineering researchers.

LIMITATIONS AND FUTURE WORK

Here we summarize limitations of Pathway Tools, organized by use case. Some of these limitations are being addressed in current research; many of the others will be addressed in future work.

Development of organism-specific DBs

Pathway Tools has an emphasis on prokaryotic biology, although over time we have added, and plan to add, more support for eukaryotic biology. For example, although the software can represent introns and exons internally, the genome browser does not yet depict intron/exon structure. The software can capture many types of prokaryotic regulation, but we have not attempted comprehensive coverage of eukaryotic regulation. Similarly, the ontology of cellular compartments used by Pathway Tools is oriented toward bacteria and plants, and does not describe mammalian compartments, nor can Pathway Tools define the variations in metabolic pathways across different cell types or developmental stages. Its biological sequence manipulation capabilities are limited, for example, editing of biological sequences is not supported. The editing tools within Pathway Tools are not web based, but require installation of Pathway Tools on every computer that will be used for editing.

Visualization and web publishing of organism-specific DBs

To date, Pathway Tools has scaled to manage the BioCyc collection of 500 PGDBs, which includes three vertebrate genomes (human, mouse and cattle). However, we have been concerned that the current approach of loading all PGDBs into Common Lisp virtual memory will not continue to scale. Therefore, we have investigated an alternative approach in which PGDBs are stored in the Allegro

Cache DB system from Franz Inc. Preliminary experimental results (unpublished) indicate that this approach will scale to 10 000 PGDBs with little degradation in performance.

Not all capabilities of Pathway Tools are available in both the web and desktop modes. For example, many comparative tools function in web mode only, whereas all aspects of PathoLogic are available in desktop mode only.

Extend genome annotations with additional computational inferences

We would like to see many additional bioinformatics inference tools interfaced with Pathway Tools, such as for inference of protein cellular location and regulatory network inference.

Analysis of omics data

Pathway Tools is not a general-purpose environment for analysis of omics data. Our assumption is that scientists will use one of the many other software packages for the early stages of omics data analysis (such as normalization), and provide the output of those analyses to Pathway Tools for display with the omics viewers. That said, we are working to supplement its existing omics analysis capabilities, such as with tools for computing over-representation analysis (e.g. are particular pathways over represented in a gene-expression experiment or metabolomics experiment?).

Analysis of biological networks

We would like to see many additional network analysis tools present within Pathway Tools, such for computing the scaling properties of metabolic networks [127], and functional modules within metabolic networks [128]. Pathway Tools does not perform flux-balance analysis of metabolic networks, but we are actively working on the ability to automatically generate flux-balance models from PGDBs [55, 129].

Comparative analysis of organism-specific DBs

We are not aware of striking limitations in comparative analysis.

Metabolic engineering

A natural addition to Pathway Tools to facilitate metabolic engineering would be a capability for designing novel metabolic pathways from the reaction library in MetaCyc, such as explored in Mavrovouniotis [130] and McShan *et al*. [131].

SUMMARY

Pathway Tools treats a genome as far more than a sequence and a set of annotations. Instead, it links the molecular parts list of the cell to the genome, and to a carefully constructed web of functional interactions. The Pathway Tools ontology defines an extensive set of object attributes and object relations that allows a rich conceptualization of biology to be represented within a PGDB, and queried and manipulated by the user.

Pathway Tools provides a broad range of functionality. It can manipulate genome data, metabolic networks and regulatory networks. For each datatype it provides query, visualization, editing and analysis functions. It provides MOD development capabilities including computational inferences that support fast generation of comprehensive DBs, editors that allow for refinement of a PGDB, web publishing and comparative analysis. A family of curated PGDBs has been developed using these tools for important model organisms.

The software also provides visual tools for analysis of omics datasets, and tools for the analysis of biological networks.

SOFTWARE AVAILABILITY

Pathway Tools runs on Macintosh, Windows and Linux. It is freely available to academic and government researchers; a license fee applies to commercial use. See http://BioCyc.org/download.shtml.

Key points

- The Pathway Tools software is a comprehensive environment for creation of model-organism DBs that span genome information, metabolic pathways, and regulatory networks.
- Pathway Tools inference capabilities include prediction of metabolic pathways, prediction of metabolic pathway hole fillers, inference of transport reactions from transporter functions and prediction of operons.
- The software provides interactive editing tools for use by DB curators.
- Omics data analysis tools paint genome-scale datasets onto a complete genome diagram, complete metabolic network diagram and complete regulatory network diagram.
- Other tools include comparative analysis operations, reachability and dead-end metabolite analysis of metabolic networks and interactive tracing of metabolites through a metabolic network.

SUPPLEMENTARY DATA

Supplementary Data are available online at http://bib.oxfordjournals.org/.

Acknowledgements

Pathway Tools has benefited from advice, input and contributions from many scientists during its lifetime. We particularly wish to recognize contributions from Robert Gunsalus, Monica Riley, John Ingraham, Jean-Francois Tomb and Peifen Zhang. We gratefully acknowledge guidance from Richard Gourse, Julio Collado-Vides and Charles Yanofsky, in developing the Pathway Tools treatment of cellular regulation. Lukas Mueller developed PerlCyc and has provided many helpful suggestions. Thomas Yan developed JavaCyc. Jeremy Zucker developed the SBML generation module, and contributed *many* other ideas. Christos Ouzounis was a co-developer of the original metabolic pathway prediction algorithm, contributed an early version of the import/export system, and has been a source of much sound advice. This paper draws on figures from a chapter entitled 'Cellular Organization of Metabolism' in the *Wiley Encyclopedia of Chemical Biology*, published by John Wiley and Sons Inc., with permission.

FUNDING

National Institutes of Health (grants GM75742, GM080746, GM077678 and LM009651). The contents of this article are solely the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health.

References

1. Karp P, Paley S. Integrated access to metabolic and genomic data. *J Comput Biol* 1996;**3**:191–212.
2. Karp PD, Paley S, Romero P. The Pathway Tools Software. *Bioinformatics* 2002;**18**:S225–32.
3. Paley SM, Karp PD. The Pathway Tools cellular overview diagram and omics viewer. *Nucleic Acids Res* 2006;**34**: 3771–8.
4. Ouzounis C, Karp PD. Global properties of the metabolic map of *Escherichia coli*. *Genome Res* 2000;**10**:568.
5. Karp PD. Pathway databases: a case study in computational symbolic theories. *Science* 2001;**293**:2040–4.
6. Keseler IM, Bonavides-Martinez C, Collado-Vides J, *et al*. EcoCyc: a comprehensive view of *E. coli* biology. *Nucleic Acids Res* 2009;**37**:D464–70.
7. Caspi R, Foerster H, Fulcher CA, *et al*. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res* 2008;**36**:D623–31.
8. Karp PD, Keseler IM, Shearer A, *et al*. Multidimensional annotation of the *Escherichia coli* K-12 genome. *Nucleic Acids Res* 2007;**35**:7577–90.
9. PseudoCyc Database. http://v2.pseudomonas.org:1555/.
10. Snyder EE, Kampanya N, Lu J, *et al*. PATRIC: the VBI pathosystems resource integration center. *Nucleic Acids Res* 2007;**35**:D401–6.
11. McNeil LK, Reich C, Aziz RK, *et al*. The National Microbial Pathogen Database Resource (NMPDR): a genomics platform based on subsystem annotation. *Nucleic Acids Res* 2007;**35**:D347–53.
12. Ball CA, Dolinski K, Dwight SS, *et al*. Integrating functional genomic information into the *Saccharomyces* Genome Database. *Nucleic Acids Res* 2000;**28**(1):77–80.
13. Aurrecoechea C, Heiges M, Wang H, *et al*. ApiDB: integrated resources for the apicomplexan Bioinformatics Resource Center. *Nucleic Acids Res* 2007;**35**:D427–30.
14. Chisholm RL, Gaudet P, Just EM, *et al*. Dictybase, the model organism database for *dictyostelium discoideum*. *Nucleic Acids Res* 2006;**34**:D423–27.
15. Chen N, Harris TW, Antoshechkin I, *et al*. WormBase: a comprehensive data resource for *Caenorhabditis* biology and genomics. *Nucleic Acids Res* 2005;**33**:D383–9.
16. Drysdale RA, Crosby MA, Gelbart W, *et al*. FlyBase: genes and gene models. *Nucleic Acids Res* 2005;**33**:D390–95.
17. Bult CJ, Eppig JT, Kadin JA, *et al*. The Mouse Genome Database (MGD): mouse biology and model systems. *Nucleic Acids Res* 2008;**36**:D724–8.
18. Huala E, Dickerman A, Garcia-Hernandez M, *et al*. The Arabidopsis information resource (TAIR): a comprehensive database and web-based information retrieval, analysis, and visualization system for a model plant. *Nucleic Acids Res* 2001;**29**:102–5.
19. Jaiswal P, Ni J, Yap I, *et al*. Gramene: a bird's eye view of cereal genomes. *Nucleic Acids Res* 2006;**34**:D717–23.
20. PlantCyc Web Site. http://plantcyc.org/.
21. Krummenacker M, Paley S, Mueller L, *et al*. Querying and computing with BioCyc databases. *Bioinformatics* 2005; **21**:3454–5.
22. Lee TJ, Pouliot Y, Wagner V, *et al*. BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics* 2006;**7**:170.
23. Karp P, Krummenacker M, Paley S, Wagg J. Integrated pathway/genome databases and their role in drug discovery. *Trends Biotechnol* 1999;**17**(7):275–81.
24. Introduction to Symbolic Computing. http://www.cs.sfu.ca/people/Faculty/Cameron/Teaching/384 (24 September 2009, date last accessed).
25. Romero PR, Karp P. Nutrient-related analysis of pathway/genome databases. In: Altman R, Klein T, (eds). *Pacific Symposium on Biocomputing*. World Scientific: Singapore, 2001;471–82.
26. Yeh I, Hanekamp T, Tsoka S, *et al*. Computational analysis of *Plasmodium falciparum* metabolism: organizing genomic information to facilitate drug discovery. *Genome Res* 2004; **14**(5):917–24.
27. Stephanopoulos GN, Aristidou AA, Nielsen J. *Metabolic Engineering: Principles and Methodologies*. San Diego, CA: Academic Press, 1998.
28. Karp PD, Chaudhri VK, Paley SM. A collaborative environment for authoring large knowledge bases. *J Intell Inf Syst* 1999;**13**:155–94.
29. Caspi R, Foerster H, Fulcher CA, *et al*. Meta-Cyc: a multi-organism database of metabolic pathways and enzymes. *Nucleic Acids Res* 2006;**34**:D511–16.
30. Paley S, Karp PD. Evaluation of computational metabolic-pathway predictions for *H. pylori*. *Bioinformatics* 2002;**18**(5):715–24.

31. Romero P, Karp PD. Using functional and organizational information to improve genome-wide computational prediction of transcription units on pathway/genome databases. *Bioinformatics* 2004;**20**:709–17.

32. Green ML, Karp PD. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC Bioinformatics* 2004;**5**(1):76.

33. Lee TJ, Paulsen I, Karp PD. Annotation-based inference of transporter function. *Bioinformatics* 2008;**24**: i259–67.

34. Marvin Chemical Editor. http://www.chemaxon.com/marvin/ (24 September 2009, date last accessed).

35. JME Chemical Editor. http://www.molinspiration.com/jme/index.html (24 September 2009, date last accessed).

36. Green ML, Karp PD. The outcomes of pathway database computations depend on pathway ontology. *Nucleic Acids Res* 2006;**34**:3687–97.

37. Karp PD, Paley S, Krieger CJ, Zhang P. An evidence ontology for use in pathway/genome databases. In: Altman R, Klein T, (eds). *Pacific Symposium on Biocomputing*. World Scientific: Singapore, 2004, pp. 190–201.

38. Pathway Tools Evidence Ontology. http://bioinformatics.ai.sri.com/evidence-ontology/.

39. Pathway Tools Cell Component Ontology. http://bioinformatics.ai.sri.com/CCO/.

40. Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic Acids Res* 2008;**36**:D440–44.

41. Bairoch, A, Apweiler, R, Wu, C. UniProt Knowledgebase User Manual, 2009. http://www.expasy.ch/sprot/userman.html (24 September 2009, date last accessed).

42. Comparison of BioCyc Desktop Mode and Web Mode. http://biocyc.org/desktop-vs-web-mode.shtml (24 September 2009, date last accessed).

43. Alashqur AM, Su SYW, Lam H. OQL: a query language for manipulating object-oriented databases. In: *VLDB '89: Proceedings of the 15th International Conference on Very Large Data Bases*, 1989, pp. 433–42, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

44. EcoCyc Cellular Overview. http://biocyc.org//ECOLI/NEW-IMAGE?type=OVERVIEW (24 September 2009, date last accessed).

45. Cellular Overview with Animated Display of Gene Expression Data. http://biocyc.org/ov-expr.shtml (24 September 2009, date last accessed).

46. BioCyc Comparative Genome Browser. http://biocyc.org/ECOLI/NEW-IMAGE?type=LOCUS-POSITION &object=(EG11024+C1725+ TRPA+VC1169)&orgids=(ECOLI+ECOL199310+ECOO157+VCHO) (24 September 2009, date last accessed).

47. Pathway Tools Home Page. http://brg.ai.sri.com/ptools/ptools-resources.html (24 September 2009, date last accessed).

48. JavaCyc API to Pathway Tools. http://www.arabidopsis.org/biocyc/perl/index.jsp (24 September 2009, date last accessed).

49. PerlCyc API to Pathway Tools. http://www.arabidopsis.org/biocyc/perlcyc/index.jsp (24 September 2009, date last accessed).

50. BioCyc Downloads. http://biocyc.org/download.shtml (24 September 2009, date last accessed).

51. SBML. http://www.sbml.org/ (24 September 2009, date last accessed).

52. BioPAX. http://www.biopax.org/ (24 September 2009, date last accessed).

53. Genbank Format. http://www.ncbi.nlm.nih.gov/collab/FT/#7.1.2 (24 September 2009, date last accessed).

54. BioVelo Advanced Query Page. http://biocyc.org/query.html.

55. Oberhardt MA, Chavali AK, Papin JA. Flux balance analysis: interrogating genome-scale metabolic networks. *Methods Mol Biol* 2009;**500**:61–80.

56. Raman K, Chandra N. Flux balance analysis of biological systems: applications and challenges. *Brief Bioinform* 2009; Advance access published 15 March 2009.

57. Satish Kumar V, Dasika MS, Maranas CD. Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics* 2007;**8**:212.

58. BioCyc Comparative Analysis. http://biocyc.org/comp-genomics?tables=reaction&tables=pathway (24 September 2009, date last accessed).

59. Gat E. Point of view: lisp as an alternative to Java. *Intelligence: New Visions of AI in Practice* 2000;**11**(4): 21–4.

60. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 1988;**28**:31–6.

61. Karp PD, Lowrance JD, Strat TM, Wilkins DE. The Grasper-CL graph management system. *LISP and Symbolic Computation* 1994;**7**:245–82.

62. Karp PD. The design space of frame knowledge representation systems. Technical Report 520, SRI International AI Center, 1992. http://www.ai.sri.com/pubs/files/236.pdf (24 September 2009, date last accessed).

63. Paley SM, Karp PD. Adapting EcoCyc for use on the World Wide Web. *Gene* 1996;**172**:GC43–50.

64. GD Library. http://www.libgd.org/ (24 September 2009, date last accessed).

65. Yahoo User Interface Library. http://developer.yahoo.com/yui/ (24 September 2009, date last accessed).

66. Mueller LA, Zhang P, Rhee SY. AraCyc, a biochemical pathway database for Arabidopsis. *Plant Physiology* 2003;**132**:453–60.

67. AraCyc Database. http://www.arabidopsis.org/biocyc/ (24 September 2009, date last accessed).

68. Romero P, Wagg J, Green ML, *et al*. Computational prediction of human metabolic pathways from the complete human genome. *Genome Biol* 2004;**6**(1):1–17.

69. HumanCyc Database. http://HumanCyc.org/ (24 September 2009, date last accessed).

70. MouseCyc Database. http://mousecyc.jax.org:8000/ (24 September 2009, date last accessed).

71. Seo S, Lewin HA. Reconstruction of metabolic pathways for the cattle genome. *BMC Syst Biol* 2009;**3**:33.

72. BioCyc Database Collection. http://BioCyc.org/ (24 September 2009, date last accessed).

73. Urbanczyk-Wochniak E, Sumner LW. Mediccyc: a biochemical pathway database for medicago truncatula. *Bioinformatics* 2007;**23**(11):1418–23.

74. MediCyc Web Site. http://www.noble.org/mediccyc/ (24 September 2009, date last accessed).

75. RiceCyc Web Site. http://pathway/gramene.org/gramene/ricecyc.shtml (24 September 2009, date last accessed).

76. SolCyc Web Site. http://sgn.cornell.edu/tools/solcyc/ (24 September 2009, date last accessed).

77. DictyCyc Web Site. http://dictybase.org/Dicty_Info/dictycyc_info.html (24 September 2009, date last accessed).

78. YeastCyc Database. http://pathway.yeastgenome.org/ (24 September 2009, date last accessed).

79. CandidaCyc Database. http://pathway.candidagenome.org/ (24 September 2009, date last accessed).

80. LeischCyc Web Site. http://bioinformatics.bio21.unimelb.edu.au/leishcyc.html (24 September 2009, date last accessed).

81. EcoCyc Database. http://EcoCyc.org/ (24 September 2009, date last accessed).

82. ScoCyc Web Site. http://scocyc.streptomyces.org.uk:14980/SCO/ (24 September 2009, date last accessed).

83. ApiCyc. http://apicyc.apidb.org/ (24 September 2009, date last accessed).

84. PathoSystems Resource Integration Center. http://patric.vbi.vt.edu/ (24 September 2009, date last accessed).

85. MicroCyc. https://www.genoscope.cns.fr/agc/mage/wwwpkgdb/MageHome/index.php?webpage=micro (24 September 2009, date last accessed).

86. TBestDB: Taxonomically Broad EST Database. http://tbestdb.bcm.umontreal.ca/searches/welcome.php (24 September 2009, date last accessed).

87. Pathema. http://pathways.jcvi.org/ (24 September 2009, date last accessed).

88. Pathway Tools PGDB Registry. http://BioCyc.org/registry.html (24 September 2009, date last accessed).

89. Kanehisa M, Goto S, Hattori M, *et al*. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res* 2006;**34**:D354–57.

90. Okuda S, Yamada T, Hamajima M, *et al*. KEGG atlas mapping for global analysis of metabolic pathways. *Nucleic Acids Res* 2008;**36**:W423–26.

91. Maltsev N, Glass E, Sulakhe D, *et al*. Puma2-grid-based high-throughput analysis of genomes and metabolic pathways. *Nucleic Acids Res* 2006;**34**:D369–72.

92. van Helden J, Naim A, Lemer C, *et al*. From molecular activities and processes to biological function. *Brief Bioinformatics* 2001;**2**:81–93.

93. Salomonis N, Hanspers K, Zambon AC, *et al*. GenMAPP 2: new features and resources for pathway analysis. *BMC Bioinformatics* 2007;**8**:217.

94. van Iersel MP, Kelder T, Pico AR, *et al*. Presenting and exploring biological pathways with Pathvisio. *BMC Bioinformatics* 2008;**9**:399.

95. Krishnamurthy L, Nadeau J, Ozsoyoglu G, *et al*. Pathways database system: an integrated system for biological pathways. *Bioinformatics* 2003;**19**(8):930–7.

96. Ozsoyoglu ZM, Ozsoyoglu G, Nadeau J. Genomic pathways database and biological data management. *Anim Genet* 2006;**37**(Suppl. 1):41–47.

97. Hu Z, Ng DM, Yamada T, *et al*. Visant 3.0: New modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Res* 2007;**35**:W625–32.

98. Joshi-Tope G, Gillespie M, Vastrik I, *et al*. Re-actome: a knowledgebase of biological pathways. *Nucleic Acids Res* 2005;**33**:D428–32.

99. Vastrik I, D'Eustachio P, Schmidt E, *et al*. Reactome: a knowledge base of biologic pathways and processes. *Genome Biol* 2007;**8**(3):R39.

100. Matthews L, Gopinath G, Gillespie M, *et al*. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res* 2009;**37**(Database issue): D619–22.

101. Shannon P, Markiel A, Ozier O, *et al*. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 2003;**13**(11):2498–504.

102. Stein LD, Mungall C, Shu S, *et al*. The generic genome browser: a building block for a model organism system database. *Genome Res* 2002;**12**(10):1599–610.

103. Donlin MJ. Using the generic genome browser (gbrowse). *Curr Protoc Bioinformatics* 2007;Chapter 9:Unit 9.9.

104. Markowitz VM, Korzeniewski F, Palaniappan K, *et al*. The integrated microbial genomes (IMG) system. *Nucleic Acids Res* 2006;**34**:D344–8.

105. Entrez Genome. http://www.ncbi.nlm.nih.gov/genome?db=genome (24 September 2009, date last accessed).

106. Peterson JD, Umayam LA, Dickinson T, *et al*. The comprehensive microbial resource. *Nucleic Acids Res* 2001;**29**(1): 123–5.

107. Hinrichs AS, Karolchik D, Baertsch R, *et al*. The UCSC Genome Browser Database: update 2006. *Nucleic Acids Res* 2006;**34**(Database issue):D590–8.

108. Hubbard TJ, Aken BL, Ayling S, *et al*. Ensembl 2009. *Nucleic Acids Res* 2009;**37**:D690–97.

109. Kazakov AE, Cipriano MJ, Novichkov PS, *et al*. RegTransBase — a database of regulatory sequences and interactions in a wide range of prokaryotic genomes. *Nucleic Acids Res* 2007;**35**(Database issue):D407–12.

110. Matys V, Kel-Margoulis OV, Fricke E, *et al*. Transfac and its module transcompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res* 2006;**34**(Database issue): D108–10.

111. Baumbach J. Coryneregnet 4.0 — a reference database for corynebacterial gene regulatory networks. *BMC Bioinformatics* 2007;**8**:429.

112. Klein J, Leupold S, Munch R, *et al*. ProdoNet: identification and visualization of prokaryotic gene regulatory and metabolic networks. *Nucleic Acids Res* 2008;**36**(Web Server issue):W460–4.

113. Sierro N, Makita Y, de Hoon M, Nakai K. DBTBS: a database of transcriptional regulation in bacillus subtilis containing upstream intergenic conservation information. *Nucleic Acids Res* 2008;**36**(Database issue):D93–6.

114. Aurrecoechea C, Brestelli J, Brunk BP, *et al*. PlasmoDB: a functional genomic database for malaria parasites. *Nucleic Acids Res* 2009;**37**(Database issue):D539–43.

115. Gama-Castro S, Jimnez-Jacinto V, Peralta-Gil M, *et al*. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res* 2008;**36**:D120–4.

116. Smedley D, Haider S, Ballester B, *et al*. BioMart — biological queries made easy. *BMC Genomics* 2009;**10**(22):22–33.

117. Birkland A, Yona G. Biozon: a system for unification, management and analysis of heterogeneous biological data. *BMC Bioinformatics* 2006;**7**(70):70–93.

118. Kono N, Arakawa K, Tomita M. Megu: pathway mapping web-service based on KEGG and SVG. *In Silico Biol* 2006; **6**(6):621–5.

119. Dahlquist KD, Salomonis N, Vranizan K, *et al*. Gen-MAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat Genet* 2002;**31**:19–20.

120. Holford M, Li N, Nadkarni P, Zhao H. VitaPad: visualization tools for the analysis of pathway data. *Bioinformatics* 2004;**15**:1596–602.

121. Chung HJ, Kim M, Park CH, *et al*. ArrayXPath: mapping and visualizing microarray gene-expression data with integrated biological pathway resources using scalable vector graphics. *Nucleic Acids Res* 2004;**32**:W460–4.

122. Reed JL, Vo TD, Schilling CH, Palsson BO. An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). *Genome Biol* 2003;**4**(9):R54.

123. Feist AM, Henry CS, Reed JL, *et al*. A genome-scale metabolic reconstruction for *Escherichia coli* K-12 mg1655 that accounts for 1260 ORFs and thermodynamic information. *Mol Syst Biol* 2007;**3**:121–38.

124. Singh S, Malik BK, Sharma DK. Choke point analysis of metabolic pathways in e. histolytica: a computational approach for drug target identification. *Bioinformation* 2007; **2**(2):68–72.

125. Rahman SA, Schomburg D. Observing local and global properties of metabolic pathways: 'load points' and 'choke points' in the metabolic networks. *Bioinformatics* 2006; **22**(14):1767–74.

126. Kim TY, Kim HU, Lee SY. Metabolite-centric approaches for the discovery of antibacterials using genome-scale metabolic networks. *Metab Eng* 2009, in press.

127. Jeong H, Tombor B, Albert R, *et al*. The large-scale organization of metabolic networks. *Nature* 2000;**407**:651–4.

128. Ma HW, Kumar B, Ditges U, *et al*. An extended transcriptional regulatory network of *Escherichia coli* and analysis of its hierarchical structure and network motifs. *Nucleic Acids Res* 2004;**32**(22):6643–49.

129. Segre D, Zucker J, Katz J, *et al*. From annotated genomes to metabolic flux models and kinetic parameter fitting. *OMICS* 2003;**7**(3):301–16.

130. Mavrovouniotis ML. Computer-aided design of biochemical pathways. PhD thesis, Massachusetts Institute of Technology, 1989.

131. McShan DC, Rao S, Shah I. Pathminer: predicting metabolic pathways by heuristic search. *Bioinformatics* 2003; **19**(13):1692–8.