BIOINFORMATICS

Vol. 18 no. 7 2002
Pages 996–1003

PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways

E. Demir[1,3], O. Babur[1,3], U. Dogrusoz[2,3], A. Gursoy[2,3], G. Nisanci[2,3], R. Cetin-Atalay[1,3] and M. Ozturk[1,3,*]

[1]Department of Molecular Biology and Genetics, [2]Computer Engineering Department and [3]Center for Bioinformatics, Bilkent University, Ankara 06533, Turkey

Received on October 12, 2001; revised on December 11, 2001; accepted on February 1, 2002

ABSTRACT

Motivation: Availability of the sequences of entire genomes shifts the scientific curiosity towards the identification of function of the genomes in large scale as in genome studies. In the near future, data produced about cellular processes at molecular level will accumulate with an accelerating rate as a result of proteomics studies. In this regard, it is essential to develop tools for storing, integrating, accessing, and analyzing this data effectively.

Results: We define an ontology for a comprehensive representation of cellular events. The ontology presented here enables integration of fragmented or incomplete pathway information and supports manipulation and incorporation of the stored data, as well as multiple levels of abstraction. Based on this ontology, we present the architecture of an integrated environment named PATIKA (Pathway Analysis Tool for Integration and Knowledge Acquisition). PATIKA is composed of a server-side, scalable, object-oriented database and client-side editors to provide an integrated, multi-user environment for visualizing and manipulating network of cellular events. This tool features automated pathway layout, functional computation support, advanced querying and a user-friendly graphical interface.

We expect that PATIKA will be a valuable tool for rapid knowledge acquisition, microarray generated large-scale data interpretation, disease gene identification, and drug development.

Availability: A prototype of PATIKA is available upon request from the authors.

Contact: patika@cs.bilkent.edu.tr

INTRODUCTION

Even prokaryotic genomes, with a few thousand genes, encode for a fascinating array of regulatory proteins creating a network of information, ultimately dictating the cell what to do under existing conditions. Human genome, which contain at least 28 000 genes (Jasny and Kennedy, 2001; Lander and Linton, 2001), is expected to create a much more complex network, composed of hundreds and thousands of different molecules and factors (Arnone and Davidson, 1997; Miklos and Rubin, 1996). Knowing the exact map of this network is very important since it will potentially explain the mechanisms of life processes as well as disease conditions. Such knowledge will also serve as a key for further biomedical applications such as development of new drugs and diagnostic approaches. In this regard, a cell can be considered as an inherently complex multi-body system. In order to make useful deductions about such a system, one needs to consider cellular pathways as an interconnected network rather than separate linear signal routes.

Our knowledge about cellular processes is increasing at a rapidly growing pace, enabling us to understand and predict a cell's behavior much better. Consequently, successful simulations and predictions of cellular events in different scopes have already been reported (Tomita *et al.*, 1999; Schaff and Loew, 1999; McAdams and Arkin, 1997; Regev *et al.*, 2001). Simulations require a comprehensive knowledge of the phenomena that is being modeled. However, most of the data available for cellular processes such as signaling pathways are not suitable for simulation studies as they are often available in incomplete and fragmented forms. One of the most important challenges in bioinformatics is development of tools to represent and integrate this type of knowledge. Such a knowledge base then can act as a blueprint for simulations and other analysis methods.

A conventional approach is based on pathway drawings composed of still images (BPC, 2001; BBID, 2001; BioCarta, 2001; SPAD, 2001). Although easy to create, such drawings are often not reusable and can not be integrated programmatically. Another approach is the development of interaction databases, in an attempt to

*To whom correspondence should be addressed.

cope with rapidly emerging protein–protein and protein–DNA interaction data (Xenarios *et al.*, 2001; BRITE, 2001; Bader *et al.*, 2001). However these approaches deal with intermolecular interactions, but not with cellular processes *per se*.

It appears that our knowledge about metabolic pathways are much more detailed and structured. As a result, databases mainly focusing on the metabolic parts of an organism are more extensive compared to their signaling counterparts (Ogata *et al.*, 1999; Karp *et al.*, 2000; WIT, 2001; BRITE, 2001). In all of these databases, the enzymes are classified according to the Enzyme Commission list of enzymes (EC numbers). Although these databases have strong ontology and a very extensive knowledge base, their scope is strictly limited to metabolic pathways and they often lack dynamic visualization and advanced querying.

Signaling pathway databases take the challenge of modeling more complex signaling networks (Wingender and Chen, 2001; Takai-Igarashi and Kaminuma, 1999; Bader *et al.*, 2001). As the complexity of the phenomena increases, efforts focus on decreasing modeling time and increasing regularity of the model. As a result these databases have strong ontology. Nevertheless they still lack important features like automated integration and visualization. Such weaknesses have been addressed by alternative approaches such as BioJake (Salamonsen *et al.*, 1999) which provides facilities for drawing cellular pathways, storing them in a database and performing queries. Unfortunately it lacks automated integration and layout and underlying ontology is not very clear.

Here we describe a new ontology and subsequently the design of a computational tool, to provide researchers an integrated collaborative environment for modeling networks of cellular processes through integration of information on individual pathways.

SYSTEM AND METHODS

An ontology for cellular processes

First step towards a computational tool is a formal ontology of cellular processes. Such an ontology must be able to describe most of the biological phenomena, while maintaining comprehensibility. Incomplete information as well as varying levels of abstraction in the available information must be supported. It must also be suitable for visualization, complex queries, and functional computations. Finally, the ontology should be flexible enough to let easy integration of new data and modification of existing knowledge.

In a cell, decision mechanisms are governed by molecules. A molecule starts its life cycle by either being synthesized from its precursors or transported into the cell, then it goes through a series of transitions. Examples

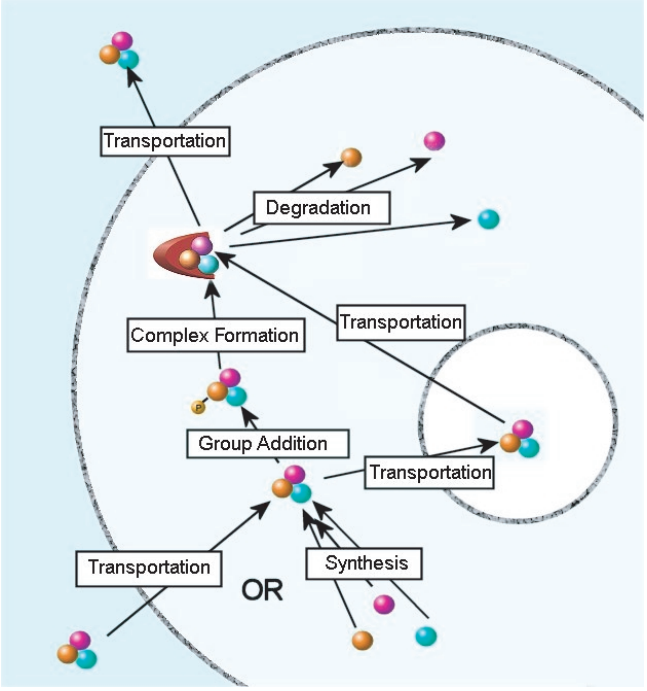

Fig. 1. Life cycle of a molecule.

of such transitions include group addition and removals, isomerizations, complex formations and transportations. Each transition changes the information carried by the molecule, transforming it into a new state like phosphorylated state of a protein or a certain splice form of RNA. A molecule's life cycle ends either by being degraded or transported out of the cell. A molecule may go through a certain set of possible transitions under a specified physiological condition and a totally different one under another condition. In fact this is the very mechanism of cellular regulation (Figure 1).

A very intuitive and widely accepted representation for cellular processes is in the form of *directed graphs* where *nodes* correspond to molecules and *edges* correspond to interactions between them. This representation is limited in the sense that it only uses one node per molecule, not facilitating the representation of different states of the molecules. Sometimes so-called active and inactive states are represented separately, which is an incomplete solution since a molecule has often more than two states. Consequently some prior knowledge is often required to correctly interpret these graphs. These representations also use hyper-edges extensively, in order to represent interactions involving more than two molecules. This poses another restraint for visualization and analysis. Although these problems may not have been significant so far due to relatively less complexity of the data, they

surely will become so as the average number of molecules per pathway increases.

We propose to represent each state and transition as a separate node in the graph (Figure 2). A *state* can be a macromolecule, a small molecule, a complex or even physical phenomena such as light. A *transition* represents group additions or removals, complex formations and disassociations as well as transportations and other cellular events. This model is very similar to the chemical equations of the form

$$A \xrightarrow{C} B$$

where A is a *substrate*, B is a *product* and C is an *effector*. This analogy is very useful since most of the biological reactions are essentially chemical reactions. Other non-chemical phenomena like transportation can be described with this ontology as well (Figure 2).

Changes in the subcellular location of a molecule are regarded as a change in its information context. In order to reflect these changes each state is associated with exactly one *compartment* such as cell membrane, nucleus or mitochondria.

A *pathway graph* $G = (V, E)$ is defined by a finite set V of states and transitions and a finite set E of interactions between these states and transitions (e.g. inhibition). A pathway is an abstraction of a certain biological phenomena and is the uppermost abstraction in our biological hierarchy. Its context can change from a single molecule–molecule interaction to a complete set of all the interactions in a cell. A pathway may contain other pathways, and in turn may be a subset of another pathway. This nesting mechanism provides multiple levels of abstraction as well as facilitating the representation of incomplete information (Figure 2). Nesting is essential for analyzing cellular mechanisms since the complete network of cellular events is clearly beyond human perception. To be more precise, in our ontology, a *pathway* is defined by a compound graph $CG = (G, T)$ where G is a pathway graph $G = (V, E^G)$ as defined above and $T = (V, E^{\mathrm{T}}, r)$ is a tree rooted at r, sharing the same set of nodes with G. Essentially G defines an *interaction graph* and the tree T is a *decomposition tree*, defining desired multiple levels of abstraction. Compound graphs have been used for similar purposes in the past (Fukuda and Takagi, 2001).

An important consideration in the design of such an ontology is ease of integration of new data. Since the entire pathway network is considerably large, a user will normally work on a relatively small subgraph of this network. Once the modifications are made, they need to be integrated to the database such that the pathway network remains intact with valid topology. PATIKA ontology uses an atomic approach of defining consistency, checking only the immediate neighborhood of each object. That is, only the associated interactions (incident edges) of a state or transition (node) are used to decide whether its topology is valid. Thus, when merging pathways one needs to check only those objects that are modified along with their immediate neighborhood. Moreover PATIKA enables integration of incomplete information via nesting (Figure 3).

Our ontology has similarities with Petri-net approaches developed to represent biochemical networks (Reddy *et al.*, 1993; Hofestädt and Thelen, 1998) in the way that Petri-net modeling has places (states) and transitions as nodes in the interaction graph. The Petri-net approaches have been developed mostly for metabolic pathways, and the emphasis is on the simulation of pathways. However, one of the strengths of PATIKA ontology is the ability to represent incomplete information and the above Petri-net approaches do not address this important issue.

We believe that this ontology is flexible enough to encompass most of the biological phenomena, while maintaining a well-defined and comprehensible structure.

An integrated environment

Considering the current pace of the biological research, it is very important to distribute new knowledge quickly. Past experience hints that the most efficient method for this is public databases where users may submit recently discovered information. Public databases also offer a single organized source reducing the time spent on literature search. Since the nature of the data is highly interrelated, this database should be necessarily curated.

Given the fact that the data is fairly complex a user probably would not be able to analyze all of the available information at once. Thus it is crucial to provide facilities for querying and navigating the database. A user should be able to perform queries on the database, retrieve the relevant subset of the database, view the query results pictorially, and analyze it. It is also essential to provide users with means to submit new information to the database, or modify an existing entry. These tasks require a potent editor which is tightly integrated to the database. Furthermore such a tool should provide support for functional computation (e.g. simulations) and large scale data analysis.

Any such tool clearly needs to be Internet-based, facilitating easy retrieval of the required information for analysis as well as sharing of the information. It should also facilitate plug-in of any new components. It is expected that development of such databases and tools will be one of the major research subjects in bioinformatics for the near future (Vidal, 2001; Endy and Brent, 2001).

PATIKA is an integrated software environment designed to provide researchers with a complete solution for modeling and analyzing cellular processes. The PATIKA

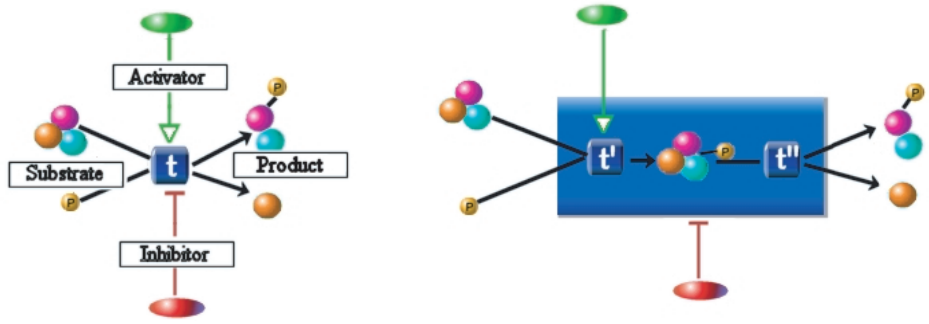

Fig. 2. (left) A basic, robust ontology for cellular processes. A molecule may have any number of states to depict changes in its information context. A transition, represented as a distinct type of node, provides a convenient mean for conveying event-specific information like reaction constants or the complex regulatory behavior. Each transition has a number of associated states, which may be products, substrates or effectors of the transition. These relations are represented by different edge types. **(right)** The model also supports abstraction and representation of incomplete information through nesting.

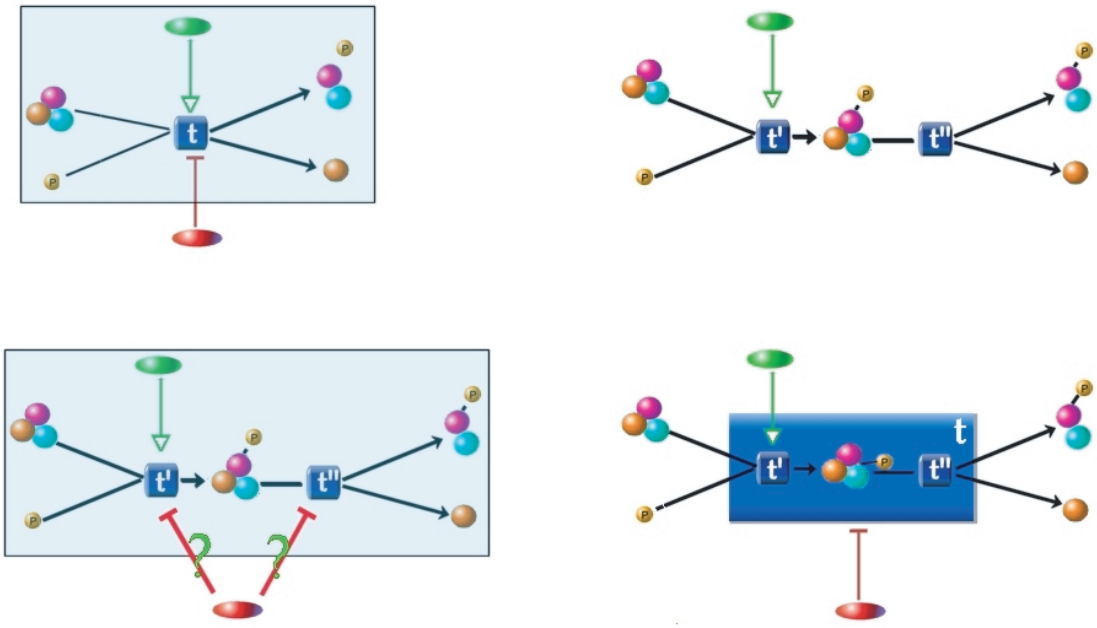

Fig. 3. (upper left) A very simple pathway resides in the database on which a user makes a query and retrieves the framed region. Note that the user might not be aware of the inhibitor. **(upper right)** The user then changes the query result, replacing t with two new transitions, t' and t'', and a new state, and assigning interactions accordingly. **(lower left)** When the modified pathway is submitted to the database, an orphan interaction is generated (labeled with '?'). Assignment of this edge is not a simple task as even the user may not know the exact mechanism. **(lower right)** A solution to this problem is to convert t to a summary transition and define an inhibitor edge as a meta-edge indicating that our knowledge is incomplete.

project aims to build a public database of all cellular processes using a regular, simple yet comprehensive model.

PATIKA uses a client–server architecture for achieving this task[†]. The server acts as a common database and manages queries and submissions as well as users. The PATIKA client, on the other hand, communicates with the server through the proxy and provides a user-friendly interface for querying, retrieving, and manipulating pathway information from the database.

A PATIKA session is started when a user executes the client application. Once logged in, the user can connect to

[†] http://www.patika.org/research/paper/arch.eps

Table 1. User types and associated access levels in PATIKA

User type	*Typical role*	*Access level*
Regular	Medical doctors, Students	Read
Research	Graduate students, Researchers	Read & submit
Expert	Experts in particular domain	Read & write

the server and query the database to construct the desired pathway. Pathways are created on the fly, and drawn automatically. A user can analyze and manipulate the data or may add new information. Finally the resulting pathway may be stored locally or submitted to the database.

Building such a database requires a collaborative effort; so, it is very critical to allow users to submit new data. However in order to preserve the integrity of the database, submissions are required to be processed before they are incorporated to the database. PATIKA has three different types of users with different levels of access (Table 1).

PATIKA *server.* The PATIKA server houses the PATIKA database and provides mechanisms for retrieving and submitting information[‡].

PATIKA represents all the interactions in a particular organism's cell as a graph (network) of states, transitions, and pathways. Since the data is a graph, and most computations will be done on this graph, PATIKA uses an object-oriented database system to store the data exactly as it is described in the ontology for efficient retrieval and querying (as opposed to relational database systems which support tabular data better). PATIKA implementation is not tied to a particular object-oriented database system, the database manager component of the server hides the details of database operations from the rest of the PATIKA software. Specifically, PATIKA stores the following information:

- *ID:* Every object in the database has a unique ID.
- *Name:* Each state and pathway in the database has a name.
- *Author:* Every state, transition and interaction has an author.
- *Type:* Each state and transition keeps information regarding its type.
- *Connectivity:* Every state and transition in this graph knows the states or transitions it is related to.
- *Data:* Each state is associated with a bioentity that keeps a set of links to the data that is accumulated about this molecule as publications, sequences, physical information, origin and associated phenotypes. Bioentities provide a gateway for other public sources of information related to this molecule. PATIKA links its model to other major databases such as SWISS-PROT (Bairoch and Apweiler, 2000), GeneBank (Karsch-Mizrachi *et al.*, 2000), GeneCards (Rebhan *et al.*, 1997).

Although PATIKA currently aims for building a model for humans, it is possible to use PATIKA for other organisms as well. Each organism would obviously need to be stored in a separate database.

PATIKA *Editor.* The client side editor provides a means for analyzing and manipulating pathways visually. It is particularly designed for minimizing modeling time for the users and to allow them to effectively construct and analyze pathways[§].

PATIKA provides most of the functionalities of a graph drawing tool such as zooming, scrolling, selection, dragging, event-handling, and persistent storage.

Once a pathway is constructed either manually and/or through a query on the database, the user can:

- manipulate this pathway through operations such as add new state or remove an existing transition,
- edit its contents such as the description of a state or transition,
- change the graphical view of a pathway component,
- define a new pathway from a part (i.e. subgraph) of an existing pathway,
- submit the pathway to the database to commit the changes to the database so they can be shared with others. PATIKA uses a three-level check mechanism to ensure that a new submission can be safely integrated into the pathway network. In the first level it is checked whether the submission complies with the PATIKA model, while the second level checks for database objects that are not present in the submission but are in the immediate neighborhood of the modified objects (Figure 3). Finally an expert user (Table 1) verifies whether the submission is biologically accurate. First two checks are performed programmatically and a proper feedback is sent back to the user for any corrections that might be required. Once these inconsistencies are fixed, the server stores the submission in a separate database and alerts the related experts. As soon as the submission is validated by an expert, it is merged to the PATIKA database;

[‡] http://www.patika.org/research/paper/archSrv.eps

[§] http://www.patika.org/research/paper/archCli.eps

- save the pathway along with its geometric information on disk for later use.

Independent of the user access level, all of these functions are essential since PATIKA is also a workbench where biologists may evaluate their ideas. Adding and removing entities or interactions may require updates of the condition for presence (activation) of states (transitions).

Each pathway document in the editor has an associated cell model, which defines how the cell and its compartments should be drawn on the screen[¶]. The default cell model is the one for humans. However, new cell models for other organisms may be easily defined and integrated into the editor.

Automated layout. PATIKA provides specialized algorithms for the layout of cellular pathways to produce aesthetically pleasing drawings of pathways that are easy to comprehend. Previous work on automated layout of pathway drawings focused on metabolic pathways. Karp and Paley (1994) takes the topology of the graph into account and decides which algorithm to use based on the graph being circular, branched, linear or complex. Becker and Rojas (2001) presents an algorithm based on the work of Karp and Baley, supplementing it with a spring embedder layout algorithm to put the components that are individually laid out together.

Signaling pathways have a more complex topological structure than the metabolic ones, making the decomposition of the graph into components of such regular structure impossible. Furthermore, each state in a signaling pathway belongs to a specific compartment in the cell and further constrains the layout. Thus, we use a layout algorithm based on the spring embedder algorithm (Eades, 1984). The heuristic attempts to align the states of a transition such that geometric distance between products and between substrates are reduced while geometric distances between products and substrates are increased. Coupled with constraints applied for compartments associated with each state, this simple yet general approach proves to be valuable especially for pathways that are generated on the fly[‖]. Even though the theoretical time complexity of this algorithm is cubic in the number of nodes of the pathway to be laid out, it is quite fast in practice with the help of the constraints.

Querying database. Since the projected network of pathways for an organism is vast, searching tools are essential for deducing useful information from this data. The editor has an integrated querying tool that allows the query of the PATIKA database. The editor sends the query in the form of a string and the result is returned as a pathway that is created dynamically by the server based on the query. As the pathway is received by the editor, it is automatically laid out to assign drawing information for a comprehensible display[**].

Pathways resulting from a query may be further extended by follow-up queries or by manual editing.

Queries not only provide a means for accessing the needed entries in the database but also discovering relations that were not noticed before. Since the case we are dealing with is quite unique we need sophisticated query operations along with basic field searches. Examples include:

- search for pathways containing certain states or transitions;
- find states that a certain transition takes or produces or is affected by;
- search for paths between two states or transitions;
- find the shortest path between two states or transitions;
- search for states and/or transitions in a certain neighborhood of a state or transition.

Users can combine queries with basic Boolean operations (AND, OR, NOT) to produce more advanced ones.

The query processing is designed to handle multiple queries running on the database simultaneously. However it is essential that these algorithms are efficient; the theoretical computational complexity of currently implemented query algorithms range from linear to cubic.

IMPLEMENTATION

PATIKA has been designed and is being implemented using an object-oriented approach in pure Java. Other than standard Java libraries and Swing, PATIKA uses Tom Sawyer Software's Graph Editor Toolkit for Java (Dogrusoz *et al.*, 2002) for the basic graph modeling and editing and ObjectStore (Lamb *et al.*, 1991) for database management.

Communication between the server and the client is based on servlets and done via http while intra-server communication is handled via Java's RMI (Remote Method Invocation) technology, facilitating the distribution of the server to multiple machines.

DISCUSSION

We have described an improved ontology for representing cellular processes. Based on this ontology, we have developed an integrated software environment named PATIKA for effectively creating, storing, analyzing and sharing information on cellular processes. PATIKA is most similar

[¶] http://www.patika.org/research/paper/p53.bmp
[‖] http://www.patika.org/research/paper/p53.bmp
[**] http://www.patika.org/research/paper/p53Q.bmp

to signaling pathway databases in its structure though it has the unique feature of providing powerful querying mechanisms, support for functional computations, and automated pathway layout as well as flexible pathway visualization and editing framework in a single solution. On the other hand, PATIKA requires a detailed mechanistic definition of the phenomena, increasing modeling times. Considering the application potential of PATIKA we believe that this extra time and effort is worthwhile.

Although PATIKA is still in its development state, it promises quite important benefits for many research fields in life sciences, including but not limited to, rapid knowledge acquisition, microarray data analysis, and drug development. PATIKA with its ability to present information in an integrated and comprehensible manner, provides a much faster alternative to literature search-based knowledge acquisition. PATIKA also appears to be very suitable for reverse engineering microarray generated data. Microarray technology generates gene expression profiles, often for more than 10 000 genes, at an unparalleled detail and speed. However the usefulness of this large-scale raw data is limited, unless it is translated into a network of cellular events as provided by PATIKA.

A third example of potential application fields of our tool would be drug discovery and development. High-throughput approaches in genomics and combinatorial chemistry provide a large number of targets and candidates for drug development. PATIKA could serve as a potent knowledge base for such applications. Being able to perform complex queries on the pathways, researchers could find drug target candidates and predict potential side effects *in silico*.

The ultimate goal is to build a model for a cell as a whole with mechanistic details and to be able to perform functional computations and simulations over this model. Although PATIKA is far from fulfilling such an expectation, its concepts and ontology may be helpful for future efforts in this direction.

ACKNOWLEDGEMENTS

The authors wish to thank C.Evren, S.Onay, B.Ozmen, E.Sahin, and E.Senel for their help with the implementation, and Doron Lancet of Weizmann Institute of Science, Rehovot, Israel, for his helpful suggestions. This work was partially supported by a TUBITAK grant to M.Ozturk.

REFERENCES

Arnone,M.I. and Davidson,E.H. (1997) The hardwiring of development: organization and function of genomic regulatory systems. *Development*, **124**, 1851–1864.

Bader,G.D., Donaldson,I., Wolting,C., Ouellette,B.F., Pawson,T. and Hogue,C.W. (2001) BIND—The Biomolecular Interaction Network Database. *Nucleic Acids Res.*, **29**, 242–245.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Res.*, **28**, 45–48.

BBID (2001) Biological biochemical image database. http://bbid.grc.nia.nih.gov.

Becker,M.Y. and Rojas,I. (2001) A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, **17**, 461–467.

BioCarta (2001) Charting pathways of life. http://www.biocarta.com.

BPC (2001) Biochemical pathways chart. http://biochem.boehringer-mannheim.com.

BRITE (2001) Biomolecular relations in information transmission and expression. http://www.genome.ad.jp/brite/.

Dogrusoz,U., Feng,Q., Madden,B., Doorley,M. and Frick,A. (2002) Graph visualization toolkits. *IEEE Computer Graphics and Applications*, to appear.

Eades,P. (1984) A heuristic for graph drawing. *Congressus Numerantium*, **42**, 149–160.

Endy,E. and Brent,R. (2001) Modelling cellular behavior. *Nature*, **409**, 391–395.

Fukuda,K. and Takagi,T. (2001) Knowledge representation of signal transduction pathways. *Bioinformatics*, **17**, 829–837.

Hofestädt,R. and Thelen,S. (1998) Qualitative modeling of biochemical networks. *In Silico Biol.*, **1**, 39–53.

Jasny,B.R. and Kennedy,D. The human genome. *Science*, **291**, 1153–2001.

Karp,P.D. and Paley,S. (1994) Automated drawing of metabolic pathways. *Third International Conference on Bioinformatics and Genome Research*. Tallahassee, Florida, pp. 225–238.

Karp,P.D., Riley,M., Saier,M., Paulsen,I.T., Paley,S. and Pellegrini-Toole,A. (2000) EcoCyc: electronic encyclopedia of *E. coli* genes and metabolism. *Nucleic Acids Res.*, **28**, 56 http://ecocyc.pangeasystems.com/.

Karsch-Mizrachi,I., Lipman,D.J., Ostell,J., Rapp,B.A. and Wheeler,D.L. (2000) Genebank. *Nucleic Acids Res.*, **28**, 15–18.

Lamb,C., Landis,G., Orenstein,J. and Weinreb,D. (1991) The ObjectStore database system. *Communications of the ACM*, **34**, 50–63.

Lander,E.S. and Linton,L.M. (2001) Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.

McAdams,H.H. and Arkin,A. (1997) Stochastic mechanisms in gene expression. *Proc. Natl Acad. Sci. USA*, **94**, 814–819.

Miklos,G.L. and Rubin,G.M. (1996) The role of the genome project in determining gene function: insights from model organisms. *Cell*, **86**, 521–529.

Ogata,H., Goto,S., Sato,K., Fujibuchi,W., Bono,H. and Kanehisa,M. (1999) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **27**, 29–34. http://www.genome.ad.jp/kegg/.

Rebhan,M., Chalifa-Caspi,V., Prilusky,J. and Lancet,D. (1997) *Genecards: encyclopedia for genes, proteins and diseases*.

Reddy,V.N., Mavrovouniotis,M.L. and Liebman,M.N. (1993) Petri net representations in metabolic pathways. *1st International Conference on Intelligent Systems for Molecular Biology*. pp. 328–336.

Regev,A., Silverman,W. and Shapiro,E. (2001) Representation and simulation of biochemical processes using the pi-calculus

process algebra. *Pacific Symp. Biocomput.*, **6**, 459–470.

Salamonsen,W.B., Mok,K.Y.C., Kolatkar,P. and Subbiah,S. (1999) BioJAKE: A tool for the creation, visualization and manipulation of metabolic pathways. *Pacific Symposium on Biocomputing*. Hawaii, pp. 392–400.

Schaff,J.C. and Loew,L.M. (1999) The virtual cell. In Altman,R.B., Dunker,A.K., Hunter,L. and Klein,T.E. (eds), *Pacific Symposium on Biocomputing*, volume 4, World Scientific, Singapore, pp. 228–239.

SPAD (2001) Signaling PAthway Database. http://www.grt.kyushu-u.ac.jp/spad/index.html.

Takai-Igarashi,T. and Kaminuma,T. (1999) A pathway finding system for the cell signaling networks database. *In Silico Biol.*, **1**, 129–146.

Tomita,M., Hashimoto,K., Takahashi,K., Shimizu,T.S., Matsuzaki,Y., Miyoshi,F., Saito,K., Tanida,S., Yugi,K., Venter,J.C. and Hutchison,III,C.A. (1999) E-CELL: software environment for whole-cell simulation. *Bioinformatics*, **15**, 72–84.

Vidal,M. (2001) A biological atlas of functional maps. *Cell*, **104**, 333–339.

Wingender,E. and Chen,X. (2001) The TRANSFAC system on gene expression regulation. *Nucleic Acids Res.*, **29**, 281–283.

WIT (2001) What Is There? Interactive metabolic reconstruction on the Web. http://wit.mcs.anl.gov.

Xenarios,I., Fernandez,E., Salwinski,L., Duan,X.J., Thompson,M.J., Marcotte,E.M. and Eisenberg,D. (2001) Dip: the Database of Interacting Proteins: 2001 update. *Nucleic Acids Res.*, **29**, 239–241.