ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Data & Knowledge Engineering 53 (2005) 75–97

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

Biological workflow with BlastQuest

William G. Farmerie [a,*], Joachim Hammer [b], Li Liu [a],
Anuj Sahni [a], Markus Schneider [b]

[a] *ICBR Molecular Services Division: DNA Sequencing Core, Interdisciplinary Center for Biotechnology Research, University of Florida, Gainesville, FL 32611, USA*

[b] *Department of Computer and Information Science and Engineering*

Received 22 June 2004; accepted 22 June 2004
Available online 3 August 2004

Abstract

Besides domain-specific biological problems, biologists are confronted with many computational problems. The large amount of varying, heterogeneous, and semi-structured biological data, the increasing complexity of biological applications, methods, and tools afflicted with uncertainty and missing knowledge, as well as the lacking interoperability of available tools necessitate integrative measures to enable biology workflow. In this paper we address these problems in the context of the processing and evaluation of BLAST query results. We present a new tool, called *BlastQuest*, which relies on database technology and provides sophisticated interactive and Web-enabled query, analysis, and visualization facilities for genomics data. The interface with the Gene Ontology and the KEGG pathway databases decisively foster the biological workflow. Finally, based on our experience with BlastQuest, we briefly sketch a new concept, called *Genomics Algebra*, for solving genomic data management problems from a broader perspective.

© 2004 Elsevier B.V. All rights reserved.

Keywords: BLAST; Gene Ontology; KEGG pathway database; Genomics Algebra; Unifying database

[*] Corresponding author.
E-mail address: wgf@biotech.ufl.edu (W.G. Farmerie).

0169-023X/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2004.06.014

1. Introduction

Besides domain-specific biological problems, biologists are confronted with many computational problems. For example, the exponentially growing volume of heterogeneous, semi-structured biological data that has to be processed and analyzed. Another problem is the increasing complexity of biological applications, methods, and tools afflicted with an inherent lack of biological knowledge as well as intrinsic uncertainty. A third problem is the lacking interoperability of available tools, i.e., biological tools are more or less self-contained and isolated, mostly only visualization-based, and incapable of exchanging data with each other. As a result, many challenges in biology and genomics are now challenges in computing and here especially in information management and algorithm design. This necessitates the development of an appropriate "communication interface" between biologists and computer scientists who each live in their own world but also recognize the chances for jointly solving important problems in common future research.

This paper deals with the three aforementioned problems in the context of BLAST (Basic Local Alignment Search Tool) [1], a common tool for conducting similarity searches. BLAST comprises a set of similarity search algorithms that employ heuristics to detect relationships between gene sequences and that rank the computed "hits" statistically. An essential problem for the biologist is currently the processing and evaluation of BLAST query results, since a BLAST search yields its result exclusively in a textual format (e.g., ASCII, HTML, XML). This format has the benefit of being application-neutral but at the same time prevents efficient analysis.

In this paper, we describe a new powerful tool called *BlastQuest* for managing BLAST results stemming from multiple individual queries. This tool provides the biologist with interactive and Web-enabled query, analysis, and visualization facilities beyond what is possible by current BLAST interfaces. In particular, BLAST results from multiple queries are imported, structured, and stored in a relational database to support a series of built-in analysis operations that can be used to select, browse, filter, group, order, search, and annotate sequence data efficiently and without referring to the original BLAST result files. In addition, users have the option to interact with the data through a forms-based query interface. BlastQuest also establishes connections to the *Gene Ontology* (GO) [10], which is a controlled vocabulary about gene and protein roles in cells, and the *Kyoto Encyclopedia of Genes and Genomes* (KEGG) [16], which is a pathway database and integrates current knowledge on molecular interaction networks in biological processes.

The rest of this paper is organized as follows. Section 2 briefly reviews some important biological concepts and addresses the biological workflow when working with BLAST, Gene Ontology, and the KEGG database. Section 3 emphasizes the need for tools capable of processing BLAST results and identifies their functional requirements. In Section 4, we describe our BlastQuest prototype from the system architecture and implementation perspective. An example session in Section 5 describes the main features and data analysis options of the BlastQuest system and its user interface. Section 6 evaluates the BlastQuest system and reports on our experiences with it. Section 7 discusses related work. Section 8 considers desired improvements to BlastQuest. We conclude the section with a brief description of a new, data model, language, and architecture called *Genomics Algebra* for integrating, processing and querying genomic information. Finally, Section 9 summarizes the paper.

2. Biological workflow

Analysis of genetic sequences is geared toward understanding what the nucleotide sequence means in a biological context. Here a biological context is the possible roles, functions, and processes that the genetic entity participates in or controls. There are many methods to discover the biological functions of genetic entities, either computationally or through empirical experimentation. One of the common computational methods is through gene sequence homology searches. Using the gene homology search method, information discovery is an iterative process cycling through three fundamental steps: (1) use known nucleotide sequence to formulate a database query, (2) store query results, and (3) analyze query results. Analyzing the results of each query leads to formulation of new queries and the cycle is repeated until hopefully all available information related to the gene is in hand. Fig. 1 schematically describes one of the ways gene information may be discovered. In this example, the primary search path begins with an input nucleotide sequence (Sequence A in Fig. 1), which is searched against a nucleotide or protein sequence database using the criteria of sequence similarity or homology. The search results and the input nucleotide sequence are stored in a results database. This search path may be repeated several times, by selecting different nucleotide or protein sequence databases as search targets. Analysis of primary search results may lead to formation of secondary queries against additional biological repositories, which may be iteratively searched on the basis of text or other cross-references determined from the initial search results.

As depicted in Fig. 1, one common path to gene information discovery begins with searches of the NCBI GenBank database [3] using the BLAST search engine. The BLAST search engine takes a query sequence and matches it against the GenBank database returning information used to evaluate the probability that database matches or “hits” are a reasonable homolog for the query sequence. For each hit a brief text summary of known biological properties is produced. Such a summary contains statistical scores (*bit scores* and *e-values*) making real sequence homology matches easier to distinguish from matches that might happen by chance. Other information included in the BLAST result includes a short text string describing the biological properties of the

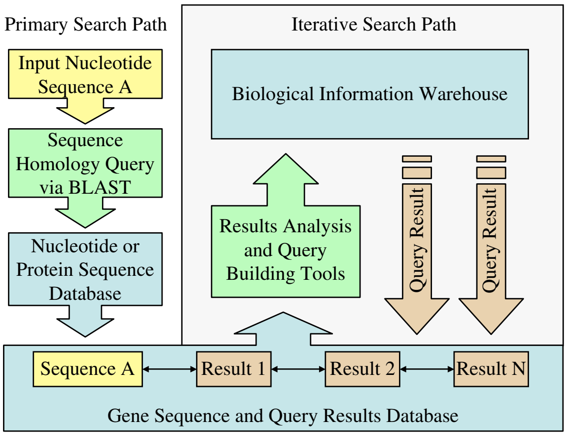

Primary Search Path
Input Nucleotide Sequence A
Sequence Homology Query via BLAST
Nucleotide or Protein Sequence Database
Iterative Search Path
Biological Information Warehouse
Results Analysis and Query Building Tools
Query Result
Query Result
Sequence A
Result 1
Result 2
Result N
Gene Sequence and Query Results Database

Fig. 1. A common gene discovery process.

database match, and several unique identification numbers, the *GI Number* (unique ID for GenBank records) and the *Accession Number*, linking the matched sequence back to the GenBank database and to additional information stored in the full database record. Each query submitted to the BLAST engine returns an output file containing possibly hundreds of matching GenBank database records. Large-scale genomics projects generate literally thousands of BLAST queries and the prospect of manually manipulating, summarizing, and interpreting thousands of output files is a daunting task. The BLAST program outputs search results in one of several formats, including flat text, HTML, and XML. Of these formats, XML is the most desirable for downstream computational processing.

Starting with an unknown nucleotide sequence, an annotated gene sequence database such as GenBank, and a sequence homology search tool such as BLAST, we may be able to infer the identity and assign a probable function to the majority of newly determined DNA sequences. However, BLAST by itself only gives us information about each specific query sequence. Genes and the processes they control work in sequential and interconnected networks or pathways. A collection of Expressed Sequence Tag (EST) sequences represents a global view of the genes expressed in a cell, tissue or organism. Taken individually, each EST sequence tells us about a single role or function. In order to assemble the global context, we must impose a higher order of organization on the individual sequences. This involves establishing biologically relevant relationships between the individual EST sequences. We do this by classifying sequences into related groups using Gene Ontology or KEGG relationships.

Gene Ontology (GO) is a controlled vocabulary about gene and protein roles in cells. In addition, hierarchical structures are imposed upon those defined terms. There are three major categories, namely biological process, molecular function, and cellular component, which are further subdivided into subcategories of various depths. Directed acyclic graphs are formed by GO terms and their associated "is-a" and "part-of" relations. This set of terms and the hierarchical relationship among them gives us the ability to identify related gene groups on various specificity levels. The KEGG pathway database integrates current knowledge on molecular interaction networks in biological processes. Macromolecules participating in the same metabolic or regulatory pathway are represented in graphical and hypertext-based format describing their functional relationship and dependency. Putting EST sequences into pathways reveals the functional contexts in which genes interact with each other so that cells can perform appropriate functions.

3. Biological tool requirements

A typical DNA sequencing project involves collections of several hundred to tens of thousands of DNA sequences. As outlined in Section 2, nucleotide sequence homology searches are often the first step toward identifying the biological function of unknown nucleotide sequences. Most academic investigators lack the computational expertise and infrastructure to conduct BLAST homology searches on the hundreds or thousands of nucleotide sequences generated by their projects. Biological scientists want to *gain insight* from their data without having to struggle with the *management* of their data.

Based on this observation, there was a clear need to build a centralized system to manage BLAST results. The BlastQuest project was initiated to manage BLAST results and make this information available to client researchers located anywhere with internet access. It began with several modest goals, foremost the delivery of a Web-based tool for viewing, searching, filtering, and summarizing large numbers of BLAST results files. Our solution began with asking our user community for ideas about the types of analysis they would like to perform. The result of these interviews produced the following list of functional requirements for the BlastQuest system:

- *A BLAST results viewing tool accessible to researchers at remote locations.* Users should have access to their BLAST results from anywhere on the Web including the ability to share results with colleagues in other locations.
- *Selective browsing of BLAST homology search results.* Biologists want a broad overview of the possible biological functions of the many genes sequences represented in their DNA sequence data. The ability to reduce and summarize BLAST data to the most significant results is very useful.
- *Search capability on a variety of criteria, such as text terms on biological properties or gene functions.* As biological scientists identify their most interesting gene sequences, they need a way to focus and retrieve only those search results related to the topic of interest.
- *Selective data filtering on various BLAST statistical criteria such as e-value or bit score.* These statistical parameters help discriminate between real sequence homology matches and matches due to noise. The user will choose parameters giving either a more relaxed or more restricted view as needed.
- *Selective data grouping by GI number or by score.* For example, viewing the three statistically best-scoring results for each query sequence is a convenient way to summarize and browse BLAST results for many query sequences. Grouping query sequences by GI number collects all of the query sequences having sequence homology matches with the same sequences from the database. Two or more query sequences sharing the same database homology match imply the query sequences are related to each other and suggest additional analysis of the relationship is warranted.
- *Privacy constrained sharing of results among the scientists.* DNA sequence data is often proprietary and may constitute intellectual property. Such data should not be made public until properly protected.
- *An intuitive interface for getting queries into and BLAST results out of the system.* The interface must be attractive and implemented in a logical manner so users will be able to find and use the tools the system provides.
- *Identifying global expression patterns.* For a query sequence, its functions can be inferred based on the functional annotations associated with its BLAST hits. If sequences with similar functions are grouped together, it is very helpful to identify global expression patterns. This task can be performed by employing the Gene Ontology.
- *Organizing sequences by orthologs and inferred pathways.* Taken individually, each EST sequence tells us about a single role or function. Putting EST sequences into pathways reveals the functional contexts, relationships, and dependencies in which genes interact with each other so that cells can perform appropriate functions. This information is provided by the KEGG pathway database.

In the next section we show how BlastQuest satisfies these requirements.

4. Architecture and prototype implementation of BlastQuest

BlastQuest simplifies large-scale analysis in gene sequencing projects by providing scientists with an integrated tool suite for gene analysis. Specifically, it allows its users to filter, summarize, sort, group, and search BLAST data and to augment the output with annotations from GO and KEGG. In addition, Blast data can be linked to SMART (Simple Modular Architecture Research Tool). BlastQuest uses XML as its internal data representation language and warehouses all of the data including the outcome of experiments as well as the entire GO vocabulary and KEGG orthologs and their relationships with pathways in a relational database. This lets users benefit from well-known relational concepts like transactions, controlled sharing, and query optimization. In addition, having all of the data and tools (including a BLAST Server) available locally increases performance, provides maximum flexibility to the users, and more than compensates for the overhead that is required to maintain the consistency and freshness of the data in the warehouse.

BlastQuest also supports a rich set of analysis functions (Section 5). The most frequently used ones are directly accessible via command buttons. To enable data analysis that is not directly supported by the built-in user interface operations, BlastQuest offers a more flexible, mask-oriented, *non-SQL* query interface; it has been our experience that biologists object to SQL due to its complexity and low-level abstraction. Our interface essentially allows the user to construct complex boolean expressions as selection conditions which include logical operators and substring search predicates. The underlying query execution is based on parameterized SQL queries, which are instantiated and automatically translated into executable SQL code by the relational database management system (RDBMS). Finally, BlastQuest lets users manage BLAST and related experiment data on a per-project or per-user basis using the *security features* of the RDBMS while at the same time allowing *controlled sharing* of this data in order to support collaboration.

4.1. Architecture of BlastQuest

Fig. 2 depicts a conceptual overview of the system architecture. Starting with the database back-end, the *Operational Data Store* (ODS) is managed using an Oracle v. 9i RDBMS. Its contents are divided into BLAST results generated by the local *BLAST Engine* (left-hand side of Fig. 2) and stored in the ODS for persistence, as well as GO and KEGG annotations that are downloaded and imported from their original sources using the *Loader* module (lower right-hand side portion of Fig. 2). The loader converts the ASCII file representation of the GO and KEGG source files into an appropriate set of load files, which are batch-loaded into the ODS using Oracle's bulkloader.

The ODS supports the concept of user and project information that facilitates the management of BLAST results and the corresponding analysis over time and among multiple collaborators. An additional data source for the BlastQuest system, although not managed by the Oracle server, is the collection of BLAST databases, for example, from NCBI, EBI, and KEGG (collectively referred to as *BLAST DB* on the left-hand side of Fig. 2). BLAST databases are stored as files. The contents of the GO and KEGG databases in Oracle as well as the file-based BLAST databases are periodically refreshed (currently monthly) by downloading updated source files from the original sites and then reloading their contents into the appropriate repositories.

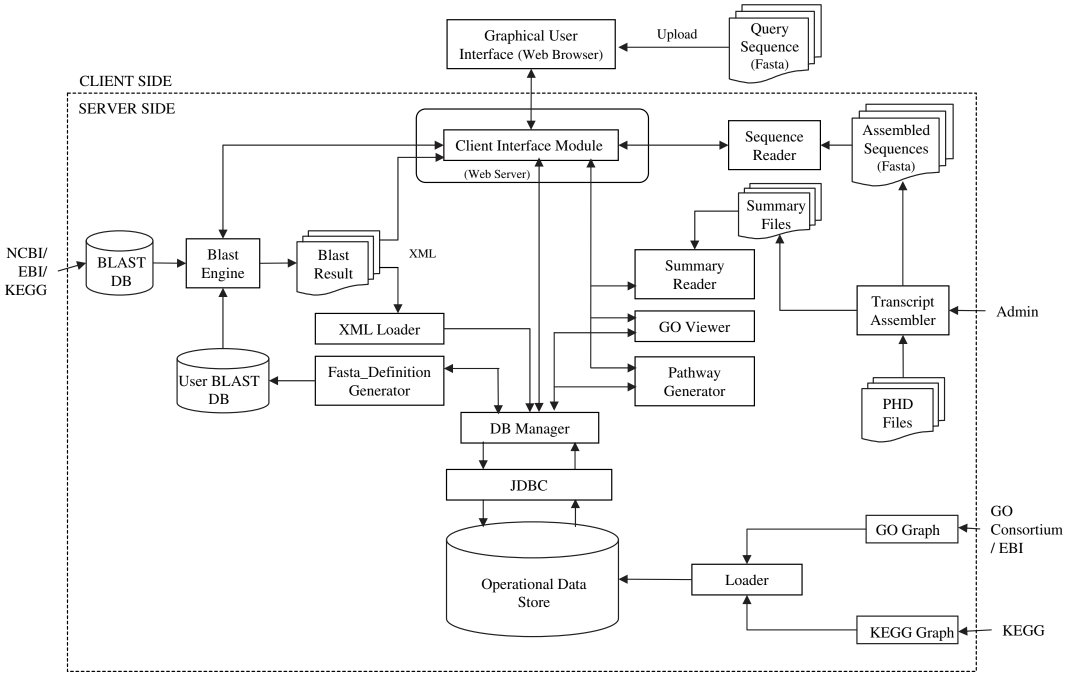


Fig. 2. Conceptual overview of the BlastQuest system architecture.

Using the *Graphical User Interface* (top of Fig. 2) a user submits a set of gene sequences in FASTA format to the *BLAST Engine*. As mentioned above, the BLAST Engine resides on a BlastQuest server and accesses the local copies of publicly available BLAST databases. Submitted gene sequences are stored in the Oracle database. For each gene sequence that produced a match during the BLAST search, the XML-formatted results are returned to the user in the form of HTML pages generated by the *Client Interface Module* residing on a Web server. By making our interface Web-based, users can load and analyze BLAST results in their BlastQuest accounts from anywhere on the Web as long as they have access to a Web browser.

BLAST results are stored persistently in the Operational Data Store using the *XML Loader*. The XML Loader is converts the XML formatted BLAST Result into a set of load files that can be loaded into the Operational Data Store using the *DB Manager* via the JDBC Call-Level API. In addition to creating the SQL load commands for BLAST Results, the DB Manager translates commands from the Graphical User Interface into SQL queries, which are executed by the Oracle RDBMS that manages the Operational Data Store. Analogously, the DB Manager also processes the query results and creates the XML that is converted into HTML output by the Client Interface Module.

One of the advantages of BlastQuest is that is allows users to conduct BLAST searches not only against the BLAST DB containing data from publicly available repositories but also against their own data. The *FASTA_Definition Generator* on the left-hand side of Fig. 2 converts the BLAST hit and sequence data stored in the Operational Data Store into BLAST Data in FASTA format using the *formatdb* application. This represents a second source for the Blast Engine (User BLAST DB).

Assembled sequences are created by the BlastQuest administrator using either Paracel's *TranscriptAssembler*^{TM} (Paracel Inc, Pasadena, CA) or the *phred–phrap–consed system* [5,6,11]. The input to the assembler are PHD [1] files [5,6] containing multiple gene segments which are assembled into consensus (DNA) sequences. TranscriptAssembler^{TM} produces two sets of outputs: (1) The consensus sequences in FASTA format, which are stored in the Operational Data Store using the *Sequence Reader*. These assembled sequences can be made available for subsequent BLAST searches by converting them back into FASTA format using the FASTA_Definition Generator. (2) A set of summary files from which the *Summary Reader* generates the assembly statistics including number of sequences before and after the assembly, number of sequences in the clusters, and number of unique sequences after the assembly. Assembly statistics are viewable through the Graphical User Interface.

Since meaningful analysis requires access to state-of-the-art reference materials, BlastQuest allows its users to augment the results of their BLAST searches with information from KEGG (pathways) and GO (terminology and relationship to other known genes). In the case of GO, the *GO Viewer* on the right-hand side of Fig. 2 identifies the location of each sequence with respect to its place in the GO ontology, which is shown to the user in the form of a browsable graph structure. In the case of KEGG, the *Pathway Generator* produces a summary file and a set of hyperlinked pathway graphs that describes the functional interconnections among genes.

4.2. Implementation

The architecture shown in Fig. 2 has been implemented as a fully functional prototype system that is installed and running on a cluster of high-end PCs in the Interdisciplinary Center for Biotechnology Research (ICBR) at the University of Florida. In the current prototype, the mapping of software to data and compute servers is as follows:

- The Graphical User Interface is implemented as a set of Java scripts which are downloaded and executed inside the client's Web browser.
- The Blast Engine on the left-hand side of Fig. 2 together with the Blast DB files are installed on a four-node, 2.40 GHz dual processor, Intel(R) Xeon(TM) cluster server. Each node has 2 GB of RAM and has access to a 100 GB disk array.
- TranscriptAssembler^{TM} is installed on the second BlastQuest server, which is a single-node, 1.60 GHz dual AMD Athlon Processor with 2 GB of RAM and 80 GB of hard disk space.
- The Client Interface Module, the Operational Data Store, and the remaining modules are installed on the third BlastQuest server, which has a single-node, 2.2 GHZ AMD Athlon Processor with 1 GB of RAM and 80 GB of hard disk space.

The reason we installed TranscriptAssembler^{TM} on its own server is that the assembly process (i.e., cleaning, clustering, and assembly of the EST sequences) is very compute-intensive and should be done in parallel with the remaining BlastQuest processing for better performance.

[1] A PHD file is the output file from a base-calling program "Phred" for DNA sequence traces. It contains bases and their quality values.

As far as the implementation is concerned, the Interface Module is implemented as a series of Java Server pages (JSPs) that execute inside a Tomcat server. The Operational Data Store is managed by an Oracle 9i RDBMS. The remaining modules including TranscriptAssembler^{TM} are implemented as Java classes. The various modules and data servers are connected through Remote Method Invocation (RMI) procedures which are also written in Java.

The division of data into three broad categories, namely BLAST results, GO, and KEGG annotations inside the Operational Data Store is also reflected in its database schema, which is depicted using the Entity–Relationship (E/R) model in Fig. 3. Although the actual database uses the relational model, we illustrate the schema using E/R notation, which is easier to understand. In the diagram in Fig. 3, rectangles represent sets of entities or major concepts, ovals describe their characteristics (attributes), and diamonds connecting entities (via edges) represent relationships among entities. The letter 'N' ('1') above an edge represents a cardinality constraint and indicates that the corresponding entity may participate multiple times (once) in the relationship. Attributes that are used to uniquely identify individual entities are underlined.

Starting from the upper left-hand corner, user and project information is stored in the entities User and Project respectively. Each project has access to zero or more submitted gene sequences which are organized in the SEQUENCE entity set. Since every sequence from the SEQUENCE entity can be BLASTed against multiple databases, e.g., non-redundant (NR),

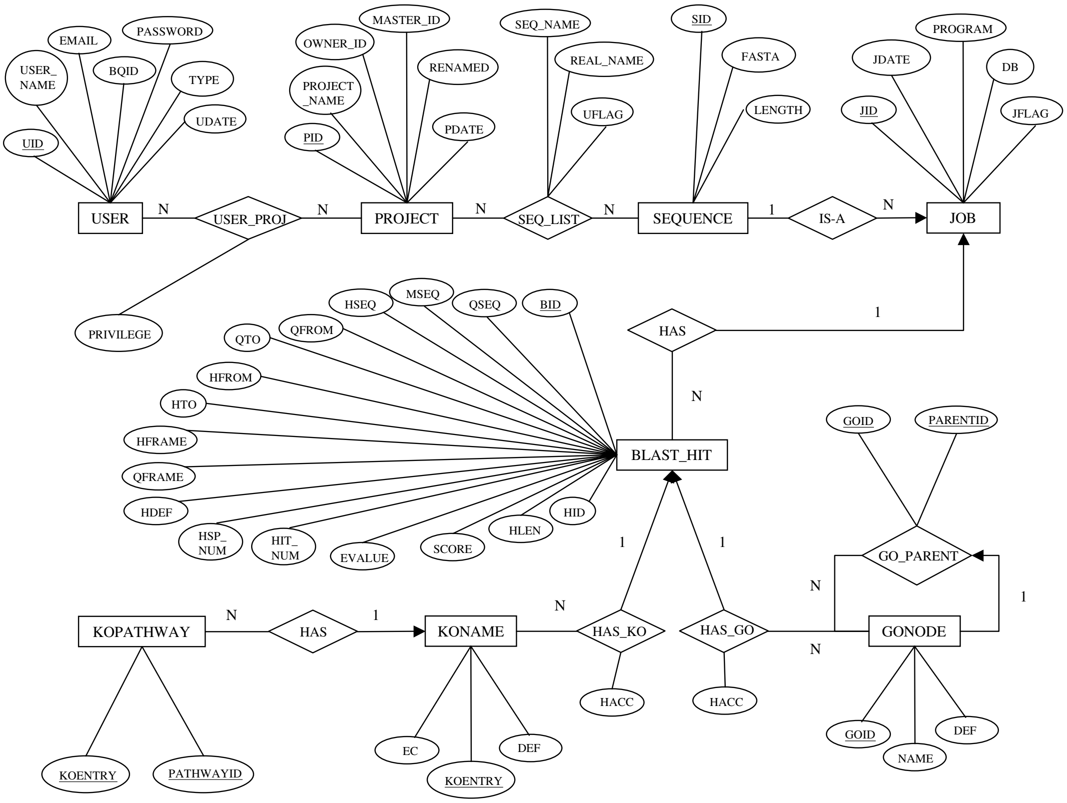


Fig. 3. Entity–relationship representation of the BlastQuest database schema.

nucleotide (NT), SwissProt (SP), etc. using different programs e.g., BLASTX, BLASTN, TBLASTX, the JOB entity is used to maintain the relationships between a sequence and the repository and program that are used to compute the matches. BLAST results are represented by the BLAST_HIT entity containing detailed hit information, such as hit definition, expect value, bit score and so forth.

As mentioned above, BlastQuest allows its users to augment the results of their BLAST searches with information from KEGG and GO. Internally, the GO graph is represented as a set of GO nodes (GONODE entity set) with links to the corresponding parent(s) which are also GO-NODES. The parent link is captured in the GO_PARENT relationship type. The KEGG orthologs are stored as a set of pathways (KOPATHWAY entity set) with their corresponding KONAMES. The relationships between BLAST_HITS and GO terms and KEGG pathways are represented by the HAS_GO and HAS_KO relationship types respectively.

5. An example session with BlastQuest

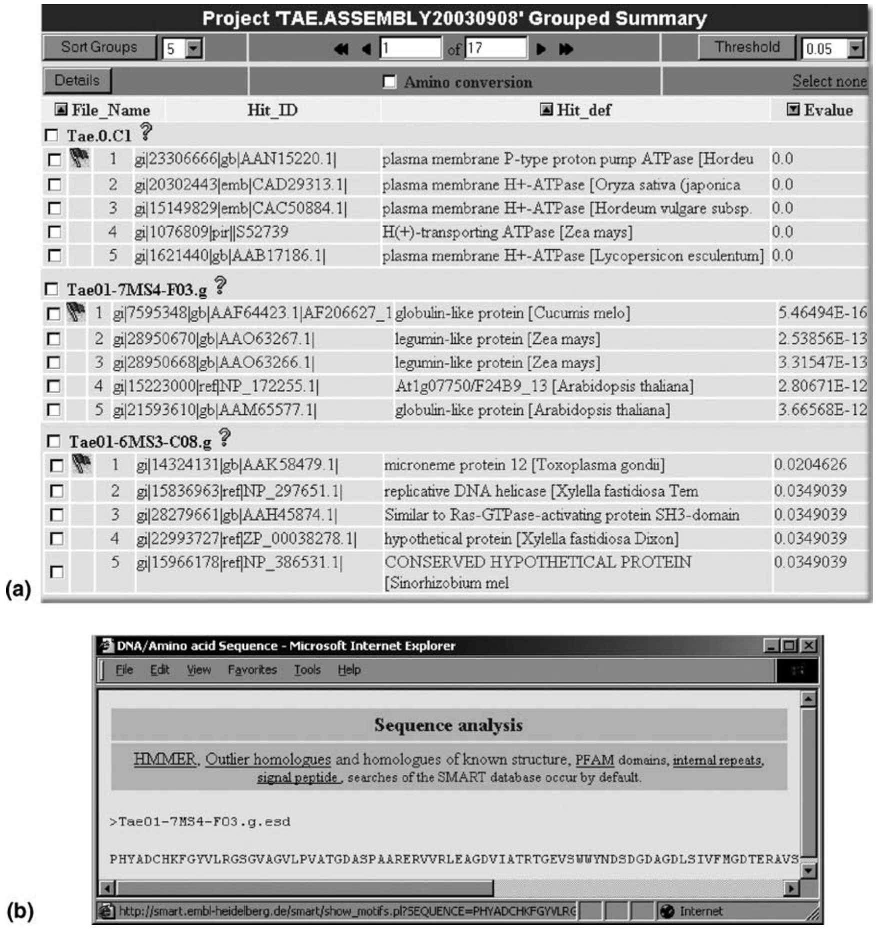
The Web-based Graphical User Interface of BlastQuest has two modes: the *administrator mode* for creating and setting up projects, and a *user mode* for data analysis. In administrator mode, there are several pages (not shown) facilitating the extraction of data from original, external BLAST files into the operational data store, the construction of a GeneOntology graph with user sequences, the generation of a page describing the orthologs and pathway information for user sequences, the creation of a summary file about sequence assembly statistics, and other administration activities. After the administrator has finished the loading of data, the project's owner or guests will be able to view and analyze their data in user mode.

Due to the large volume of data, simple page-by-page viewing is not helpful to the user; instead, selection mechanisms are needed to find the data of interest. The user interface has two aspects: (1) apply a sequence of consecutive operations to gradually arrive at the data of interest; (2) generate a global view of the data on a project level in order to discover expression patterns. In the remainder of this section, we will present a sample session to describe the main user interface features for doing both. The data and screens are from an ongoing EST sequencing project for dwarf wheat [14].

The first feature is a *summary page* for all or selected query sequences. For each query sequence, the top BLAST hit (i.e., the sequence database match with the best statistical score) is by default regarded as the best matching sequence. It is displayed with a summary of important biological information, which contains, for each matching sequence, the sequence ID (e.g. GI number from GenBank), the gene definition, and the expect value. [2] This view is similar to Fig. 4a except that only the top hit is displayed.

The second feature is *user-controlled selection*. It is well known that the statistically calculated ranking of matching sequences provided by BLAST does not necessarily correspond to the biological knowledge and experience of the user. BlastQuest keeps up to 20 of the top scoring BLAST hits for each query sequence. In addition, the user may choose to tag a statistically less significant

[2] An *expect value* (*e-value*) expresses the probability that a hit occurred by chance. The lower the e-value, the more significant the hit is. The lowest e-value is 0.



File_Name		Hit_ID	Hit_def	Evalue
□ Tae.0.C1 ?				
□	1	gi|23306666|gb|AAN15220.1|	plasma membrane P-type proton pump ATPase [Hordeu	0.0
□	2	gi|20302443|emb|CAD29313.1|	plasma membrane H+-ATPase [Oryza sativa (japonica	0.0
□	3	gi|15149829|emb|CAC50884.1|	plasma membrane H+-ATPase [Hordeum vulgare subsp.	0.0
□	4	gi|1076809|pir||S52739	H(+)-transporting ATPase [Zea mays]	0.0
□	5	gi|1621440|gb|AAB17186.1|	plasma membrane H+-ATPase [Lycopersicon esculentum]	0.0
□ Tae01-7MS4-F03.g ?				
□	1	gi|7595348|gb|AAF64423.1|AF206627_1	globulin-like protein [Cucumis melo]	5.46494E-16
□	2	gi|28950670|gb|AAO63267.1|	legumin-like protein [Zea mays]	2.53856E-13
□	3	gi|28950668|gb|AAO63266.1|	legumin-like protein [Zea mays]	3.31547E-13
□	4	gi|15223000|ref|NP_172255.1|	At1g07750/F24B9_13 [Arabidopsis thaliana]	2.80671E-12
□	5	gi|21593610|gb|AAM65577.1|	globulin-like protein [Arabidopsis thaliana]	3.66568E-12
□ Tae01-6MS3-C08.g ?				
□	1	gi|14324131|gb|AAK58479.1|	microneme protein 12 [Toxoplasma gondii]	0.0204626
□	2	gi|15836963|ref|NP_297651.1|	replicative DNA helicase [Xylella fastidiosa Tem	0.0349039
□	3	gi|28279661|gb|AAH45874.1|	Similar to Ras-GTPase-activating protein SH3-domain	0.0349039
□	4	gi|22993727|ref|ZP_00038278.1|	hypothetical protein [Xylella fastidiosa Dixon]	0.0349039
□	5	gi|15966178|ref|NP_386531.1|	CONSERVED HYPOTHETICAL PROTEIN [Sinorhizobium mel	0.0349039

Sequence analysis

HMMER, Outlier homologues and homologues of known structure, PFAM domains, internal repeats, signal peptide, searches of the SMART database occur by default.

```
>Tae01-7MS4-F03.g.esd

PHYADCHKFGYVLRGSGVAGVLPVATGDASPAARERVVRLEAGDVIATRTGEVSWWYNDSDGDAGDLSIVFMGDTERAVS
```

http://smart.embl-heidelberg.de/smart/show_motifs.pl?SEQUENCE=PHYADCHKFGYVLRG

Fig. 4. Consistency checking and filtering BLAST results and interoperability with other systems.

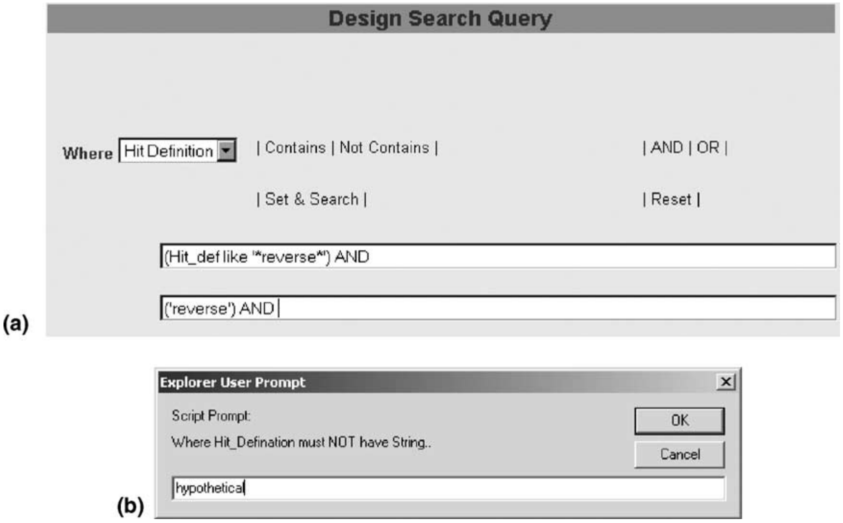
hit as better for expressing possible functions of the query sequence. By manually selecting a specific BLAST hit, the user gets additional information such as the percentage of identity or the translational reading frame. The available detailed display of sequence alignments is identical to the free-text formatted BLAST result to which most BLAST users are accustomed.

The third feature is a function that checks for *consistency*. Frequently, BLAST returns multiple hits for a query sequence. The consistency among these hits reveals more information about the query sequence's real identity than a single hit does. To do so, users ask BlastQuest to display the top *n* hits for every query sequence by clicking the "Sort Groups" button shown in Fig. 4a. Fig. 4 also demonstrates three different scenarios. Sequence "Tae.0.C1" represents the first scenario where all or most of the hits identify the same genes (e.g., they could be from different organisms or different tissues). In such case, the identity of the query sequence will be assigned as the consensus name. The second scenario is when the hits are not the same genes but genes with similar functions. The query sequence and the hits may share some common segments, which BlastQuest can help to identify. It is demonstrated with the sequence "Tae01-7MS4-F03.g" that hits two potential genes—globulin or legumin. Checking the detailed sequence alignments by clicking the "Details" button shows that all the hits match the same region of Tae01-7MS4-F03.g. By checking the "Amino conversion" option and clicking the "Details" button again, Tae01-7MS4-F03.g is translated into an amino acid sequence, which is automatically submitted to the SMART database [17] for domain search, as shown in Fig. 4b. A SMART search reveals that the common region among globulin, legumin and Tae01-7MS4-F03.g corresponds to a conservative domain "Cupin" that is present in plant seed storage proteins and germins. Even though the real identity of Tae01-7MS4-F03.g is not clear yet, we can deduce its functional identity as a gene encoding plant seed storage proteins or germins. Tae01-6MS3-C08.g is an example showing the third scenario where BLAST hits are very divergent, revealing little about the query sequence's identity. Further analysis needs to be done to discover the real identity of the query sequence, which might be a new gene, a non-protein-coding RNA, or a gene not included in the BLAST sequence database.

The fourth feature is related to *filtering and ordering* facilities, which can be activated by mouse-click and which operate on all query sequences and their BLAST results. Examples are the display of hits with expect values less than a particular *threshold* selected from a pull-down menu, or sorting hits based on expect values. Together, the consistency checking and filtering functions provide researchers the ability to reduce the original BLAST results to a manageable size, and to categorize sequences based on the qualities of BLAST hits. In general, researchers like to apply different analysis processes to sequences with high quality BLAST hits and sequences with low quality BLAST hits. Furthermore, consistency checking, filtering, and adjustments of analysis processes are highly dependent on the particular research focus, project status, and prior knowledge of query sequences, which require these functions to be both flexible and easy to use.

The fifth feature enables *user-defined, mask-oriented, non-SQL queries*. This feature refers to the problem that the built-in functionality of BlastQuest is sometimes insufficient for specific analysis tasks while many biologists are not familiar with SQL queries. BlastQuest provides a special Web page which allows the user to click on particular buttons, to manually insert text, and in this way to interactively and textually construct complex Boolean expressions which may include logical operators as well as substring search predicates. For example, when a user clicks on "Contains" or "Not Contains" in Fig. 5a, a pop-up window shown in Fig. 5b is displayed, where the user types in a text string. A search field (e.g., "Hit Definition" in our example) to which the Boolean expression is compared can be selected by a drop-down menu. Fig. 5a shows two textual representations of the same Boolean expression under construction. The first representation is a test mode translating the "natural language" condition into an SQL query. The second expresses the same condition in "plain English." The construction of the Boolean expression and hence of the query is completed by clicking the "Set & Search" button. BlastQuest assembles the SQL query, sends it to the operational data store, receives the results and displays them. In the example in Fig. 5, the user is specifying a query requesting hits that contain the word "reverse", but not "hypothetical."

For sequence analysis, BLAST searching is only the initial step. In addition to predicting the identity of query sequences on a gene-by-gene basis, more information can be obtained by viewing the BLAST results collection. In order to assemble the global context, we must impose a higher order of organization on the individual sequences. This is illustrated in the next three features.



(Hit_def like '*reverse*') AND

('reverse') AND

Script Prompt:

Where Hit_Defination must NOT have String..

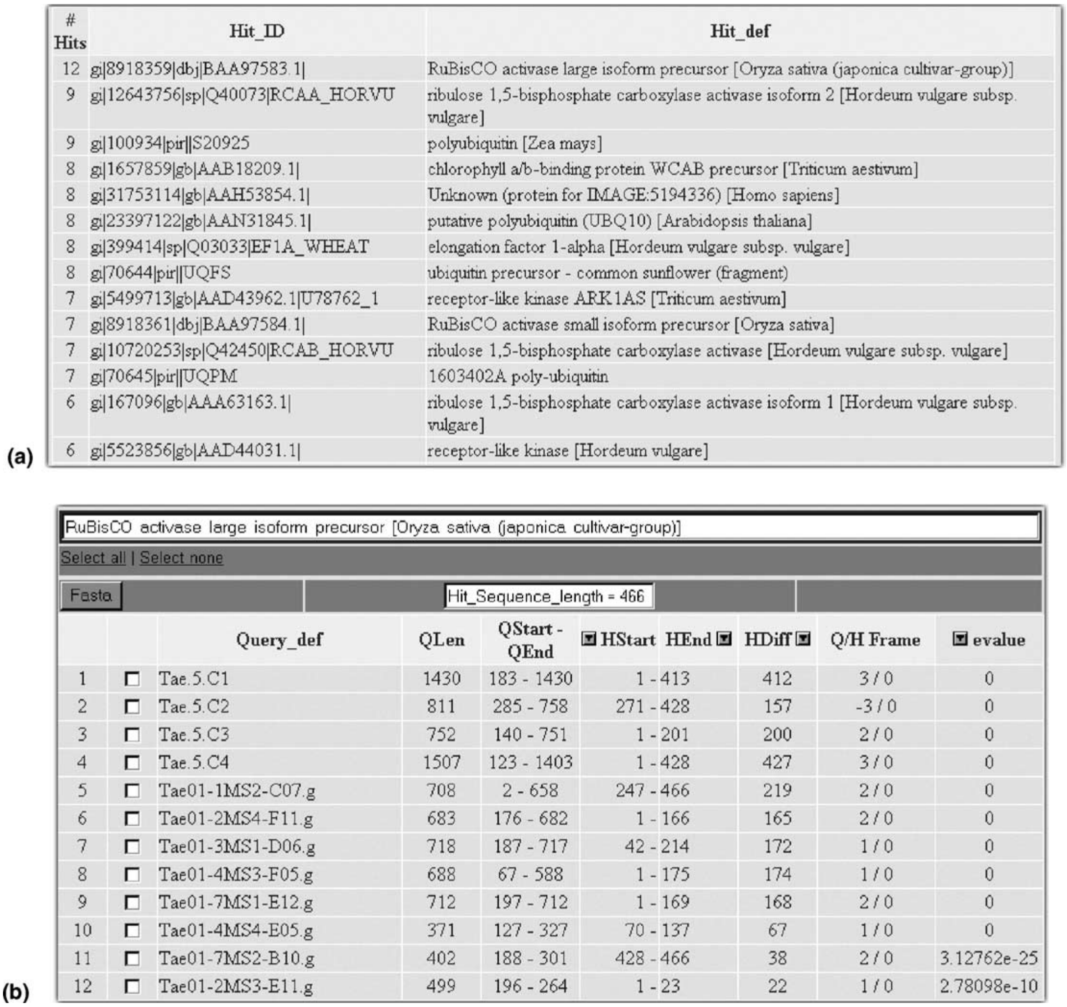
hypothetical

Fig. 5. User-defined, mask-oriented, non-SQL query.

The sixth feature comprises *grouping* functions. For example, if the user asks for grouping on GI number or BLAST hit definition, related sequences and their BLAST results are grouped together rather than appear randomly or out of context. Fig. 6a shows how to group BLAST results on GI number for all sequences in the project. Clicking on a specific hit leads to a page displaying all query sequences that match this database sequence. Fig. 6b shows 12 query sequences that encode RuBisCo genes with matching regions specified. This is an established method to identify EST sequences that come from different regions of the same mRNA, gene orthologs, or gene paralogs. [3] Grouping also reveals possible transcript variants.

The seventh feature is to *categorize sequences by inferred functions based on Gene Ontology*. For a query sequence, its functions can be inferred based on the functional annotations associated with its BLAST hits. On a project basis, if sequences with similar functions are grouped together, it is helpful for biologists to clarify the relationships among query sequences and to identify global expression patterns. Thanks to Gene Ontology (GO) [10], a controlled vocabulary about gene and protein roles in cells, we are able to impose a clearly defined hierarchical order on our sequences. As shown in Fig. 7a, GO has three major categories, namely biological processes, molecular functions, and cellular components, which are further subdivided into subcategories of various depths. Directed acyclic graphs are formed by GO terms and their associated "is-a" and "part-of" relations. Because GO annotations are available for sequences in the SwissProt database [2], such GO terms can be assigned to query sequences if BLAST search against the SwissProt database returns good statistical matches. Fig. 7b shows the "physiological processes" branch in the GO graph. There are two numbers following each GO term branch. The first number tells how many unique sequences are annotated with this term; and the second number tells how many annotations or

[3] Gene *orthologs* are genes that are derived by divergent evolution, such as the α-hemoglobin gene from human and from mouse. Gene *paralogs* are genes that are duplications, such as α-hemoglobin and β-hemoglobin.



# Hits	Hit_ID	Hit_def
12	gi|8918359|dbj|BAA97583.1|	RuBisCO activase large isoform precursor [Oryza sativa (japonica cultivar-group)]
9	gi|12643756|sp|Q40073|RCAA_HORVU	ribulose 1,5-bisphosphate carboxylase activase isoform 2 [Hordeum vulgare subsp. vulgare]
9	gi|100934|pir||S20925	polyubiquitin [Zea mays]
8	gi|1657859|gb|AAB18209.1|	chlorophyll a/b-binding protein WCAB precursor [Triticum aestivum]
8	gi|31753114|gb|AAH53854.1|	Unknown (protein for IMAGE:5194336) [Homo sapiens]
8	gi|23397122|gb|AAN31845.1|	putative polyubiquitin (UBQ10) [Arabidopsis thaliana]
8	gi|399414|sp|Q03033|EF1A_WHEAT	elongation factor 1-alpha [Hordeum vulgare subsp. vulgare]
8	gi|70644|pir||UQFS	ubiquitin precursor - common sunflower (fragment)
7	gi|5499713|gb|AAD43962.1|U78762_1	receptor-like kinase ARK1AS [Triticum aestivum]
7	gi|8918361|dbj|BAA97584.1|	RuBisCO activase small isoform precursor [Oryza sativa]
7	gi|10720253|sp|Q42450|RCAB_HORVU	ribulose 1,5-bisphosphate carboxylase activase [Hordeum vulgare subsp. vulgare]
7	gi|70645|pir||UQPM	1603402A poly-ubiquitin
6	gi|167096|gb|AAA63163.1|	ribulose 1,5-bisphosphate carboxylase activase isoform 1 [Hordeum vulgare subsp. vulgare]
6	gi|5523856|gb|AAD44031.1|	receptor-like kinase [Hordeum vulgare]

RuBisCO activase large isoform precursor [Oryza sativa (japonica cultivar-group)]

Select all | Select none

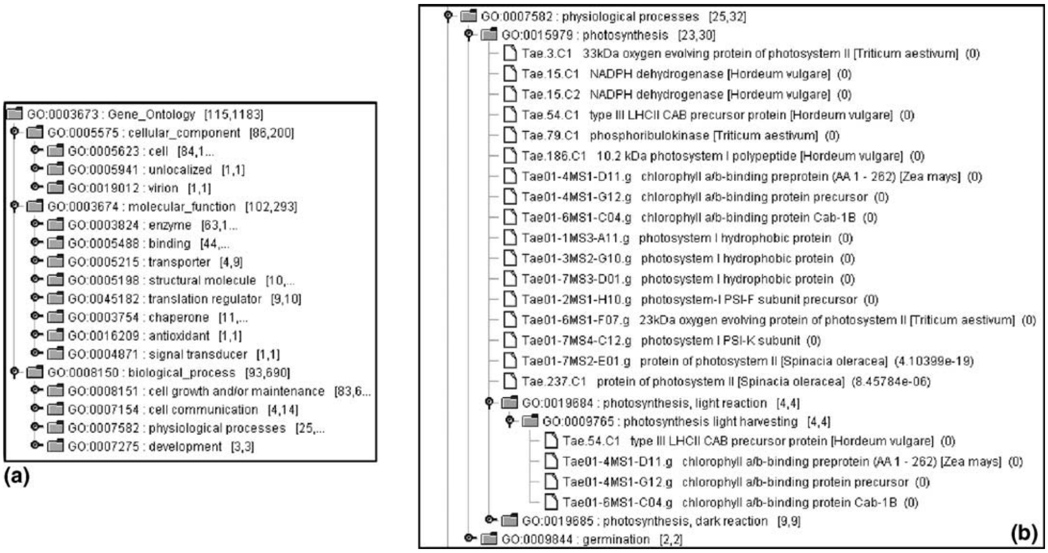
Fasta Hit_Sequence_length = 466

		Query_def	QLen	QStart - QEnd	▣ HStart	HEnd ▣	HDiff ▣	Q/H Frame	▣ evalue
1	□	Tae.5.C1	1430	183 - 1430	1 - 413		412	3 / 0	0
2	□	Tae.5.C2	811	285 - 758	271 - 428		157	-3 / 0	0
3	□	Tae.5.C3	752	140 - 751	1 - 201		200	2 / 0	0
4	□	Tae.5.C4	1507	123 - 1403	1 - 428		427	3 / 0	0
5	□	Tae01-1MS2-C07.g	708	2 - 658	247 - 466		219	2 / 0	0
6	□	Tae01-2MS4-F11.g	683	176 - 682	1 - 166		165	2 / 0	0
7	□	Tae01-3MS1-D06.g	718	187 - 717	42 - 214		172	1 / 0	0
8	□	Tae01-4MS3-F05.g	688	67 - 588	1 - 175		174	1 / 0	0
9	□	Tae01-7MS1-E12.g	712	197 - 712	1 - 169		168	2 / 0	0
10	□	Tae01-4MS4-E05.g	371	127 - 327	70 - 137		67	1 / 0	0
11	□	Tae01-7MS2-B10.g	402	188 - 301	428 - 466		38	2 / 0	3.12762e-25
12	□	Tae01-2MS3-E11.g	499	196 - 264	1 - 23		22	1 / 0	2.78098e-10

Fig. 6. Grouping BLAST results on a project basis.

associations are constructed under this level. The second number is always equal to or greater than the first number because one sequence may be annotated with more than one GO term and one GO term may have more than one position in the GO graph. As we can see, there are 25 unique sequences in the project that participate in ''physiological processes''. Specifically, there are 23 sequences in ''photosynthesis'' and two sequences in ''germination''. Among those 23 ''photosynthesis'' sequences, some of them are associated with more specific processes, namely ''light reaction'' or ''dark reaction''. In fact, there are other physiological processes defined by GO. However, BlastQuest only displays terms that are associated with at least one sequence. For each sequence it displays a sequence name, possible a gene identity and the expect value.

The eighth feature is to *organize sequences by orthologs and inferred pathways*. The KEGG databases [16] provide information about orthologs and the metabolic and regulatory pathways they are in. BLAST makes the transfer of such information between query sequences and matching



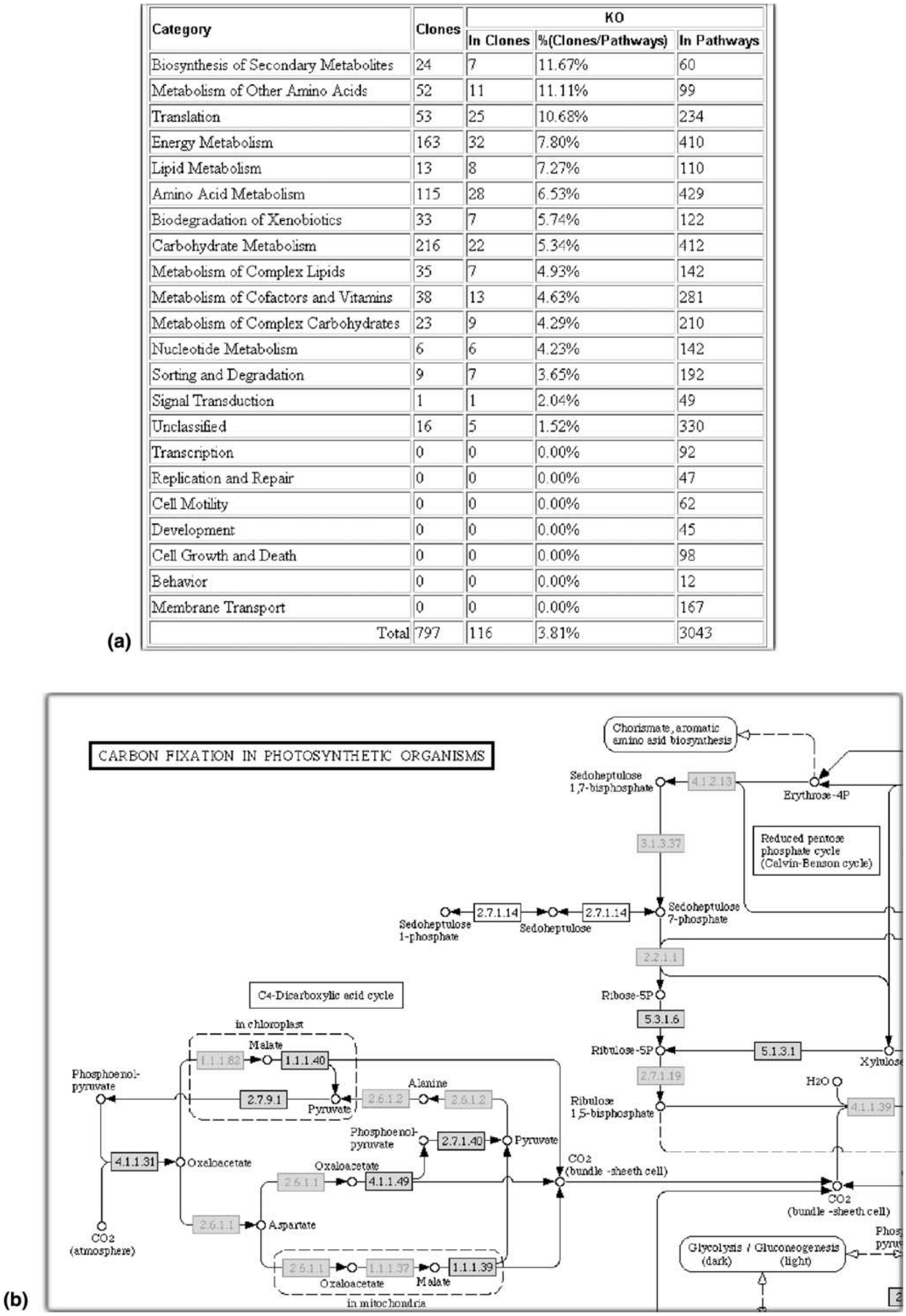
```
GO:0003673 : Gene_Ontology  [115,1183]
  GO:0005575 : cellular_component  [86,200]
    GO:0005623 : cell  [84,1...
    GO:0005941 : unlocalized  [1,1]
    GO:0019012 : virion  [1,1]
  GO:0003674 : molecular_function  [102,293]
    GO:0003824 : enzyme  [63,1...
    GO:0005488 : binding  [44,...
    GO:0005215 : transporter  [4,9]
    GO:0005198 : structural molecule  [10,...
    GO:0045182 : translation regulator  [9,10]
    GO:0003754 : chaperone  [11,...
    GO:0016209 : antioxidant  [1,1]
    GO:0004871 : signal transducer  [1,1]
  GO:0008150 : biological_process  [93,690]
    GO:0008151 : cell growth and/or maintenance  [83,6...
    GO:0007154 : cell communication  [4,14]
    GO:0007582 : physiological processes  [25,...
    GO:0007275 : development  [3,3]
```

```
GO:0007582 : physiological processes  [25,32]
  GO:0015979 : photosynthesis  [23,30]
    Tae.3.C1  33kDa oxygen evolving protein of photosystem II [Triticum aestivum]  (0)
    Tae.15.C1  NADPH dehydrogenase [Hordeum vulgare]  (0)
    Tae.15.C2  NADPH dehydrogenase [Hordeum vulgare]  (0)
    Tae.54.C1  type III LHCII CAB precursor protein [Hordeum vulgare]  (0)
    Tae.79.C1  phosphoribulokinase [Triticum aestivum]  (0)
    Tae.186.C1  10.2 kDa photosystem I polypeptide [Hordeum vulgare]  (0)
    Tae01-4MS1-D11.g  chlorophyll a/b-binding preprotein (AA 1 - 262) [Zea mays]  (0)
    Tae01-4MS1-G12.g  chlorophyll a/b-binding protein precursor  (0)
    Tae01-6MS1-C04.g  chlorophyll a/b-binding protein Cab-1B  (0)
    Tae01-1MS3-A11.g  photosystem I hydrophobic protein  (0)
    Tae01-3MS2-G10.g  photosystem I hydrophobic protein  (0)
    Tae01-7MS3-D01.g  photosystem I hydrophobic protein  (0)
    Tae01-2MS1-H10.g  photosystem-I PSI-F subunit precursor  (0)
    Tae01-6MS1-F07.g  23kDa oxygen evolving protein of photosystem II [Triticum aestivum]  (0)
    Tae01-7MS4-C12.g  photosystem I PSI-K subunit  (0)
    Tae01-7MS2-E01.g  protein of photosystem II [Spinacia oleracea]  (4.10399e-19)
    Tae.237.C1  protein of photosystem II [Spinacia oleracea]  (8.45784e-06)
    GO:0019684 : photosynthesis, light reaction  [4,4]
      GO:0009765 : photosynthesis light harvesting  [4,4]
        Tae.54.C1  type III LHCII CAB precursor protein [Hordeum vulgare]  (0)
        Tae01-4MS1-D11.g  chlorophyll a/b-binding preprotein (AA 1 - 262) [Zea mays]  (0)
        Tae01-4MS1-G12.g  chlorophyll a/b-binding protein precursor  (0)
        Tae01-6MS1-C04.g  chlorophyll a/b-binding protein Cab-1B  (0)
    GO:0019685 : photosynthesis, dark reaction  [9,9]
  GO:0009844 : germination  [2,2]
```

Fig. 7. Function categorization based on Gene Ontology.

sequences possible. Furthermore, BlastQuest organizes such data on a project basis to give users a general impression on how those pathways are populated with their sequences. Fig. 8a is a summary table. The first column lists all 22 major categories of KEGG pathways. The second column tells the number of sequences that are found in pathways for a specific category. The third column tells how many orthologs these sequences belong to. The fifth column gives the number of total KEGG orthologs in all pathways for that category. The fourth column is simply the percentage of orthologs in user sequences and KEGG pathways. As shown in Fig. 8a, pathways involved in biosynthesis of secondary metabolites are the most populated pathways in this project. By looking at the last row of the summary table, users get a general impression on how many gene orthologs have been discovered in the project. Fig. 8b shows a pathway map, in which orthologs identified in user sequences are highlighted. Organizing a set of sequences or genes into pathway maps reveals the functional contexts in which genes interact with each other so that cells can perform appropriate functions. It helps biologists to discover the relationships among those genes in biochemical reactions, regulatory paths, and other biological processes that have been modeled in pathway maps. Furthermore, comparing the pathway maps among several sets of sequences or genes (e.g. genes from different tissues, species, etc.) facilitates the global functional comparisons.

The final feature is to access an internal *BLAST engine*, in which nucleotide sequences derived from the users' DNA sequencing project are formatted into a database that can be searched using the BLAST method. For a sequencing project, a commonly asked question is "Is gene X present in my data?" Querying BlastQuest using PL/SQL might answer this question only if text describing gene X is represented in the BLAST results. However, gene X may not be directly described in the BLAST results even if it is present in the users sequence data collection. For example, gene X may be a newly discovered gene which has not been included in a public sequence database; or gene X is a non-protein-coding sequence, which will not be identified in the BLAST results under certain circumstances. In such cases, if we format the user-derived nucleotide sequence collection



Category	Clones	KO		
		In Clones	%(Clones/Pathways)	In Pathways
Biosynthesis of Secondary Metabolites	24	7	11.67%	60
Metabolism of Other Amino Acids	52	11	11.11%	99
Translation	53	25	10.68%	234
Energy Metabolism	163	32	7.80%	410
Lipid Metabolism	13	8	7.27%	110
Amino Acid Metabolism	115	28	6.53%	429
Biodegradation of Xenobiotics	33	7	5.74%	122
Carbohydrate Metabolism	216	22	5.34%	412
Metabolism of Complex Lipids	35	7	4.93%	142
Metabolism of Cofactors and Vitamins	38	13	4.63%	281
Metabolism of Complex Carbohydrates	23	9	4.29%	210
Nucleotide Metabolism	6	6	4.23%	142
Sorting and Degradation	9	7	3.65%	192
Signal Transduction	1	1	2.04%	49
Unclassified	16	5	1.52%	330
Transcription	0	0	0.00%	92
Replication and Repair	0	0	0.00%	47
Cell Motility	0	0	0.00%	62
Development	0	0	0.00%	45
Cell Growth and Death	0	0	0.00%	98
Behavior	0	0	0.00%	12
Membrane Transport	0	0	0.00%	167
Total	797	116	3.81%	3043

Fig. 8. KEGG orthologs and pathways.

as a BLAST enabled database rather than treat them as query sequences, this question can be easily answered by BLASTing the nucleotide or protein sequence of gene X against this novel user-sequence-derived database. The user interface is the standard NCBI BLAST Web interface. Various customized BLAST databases are constructed and listed in a pull-down menu based on project status and user requests. The BLAST result is displayed in the standard NCBI BLAST HTML format.

6. BlastQuest evaluation

6.1. BlastQuest as a tool for managing information

The BlastQuest project began as a solution for the problem of managing BLAST search results from large-scale DNA sequencing projects. We started with a limited, but specific set of requirements determined by interviewing several biologists involved in large-scale DNA sequencing projects. We believed we could not identify all necessary requirements for the BlastQuest system a priori. For this reason, from the beginning BlastQuest was meant to be a dynamic and extendable platform for testing ways of both representing and delivering project-specific, biological information directly to the biologist's desktop.

At the start, biologists readily adapted to the BlastQuest interface with its look and feel typical of many Web browser tools with pull down menus, check boxes, and hypertext links. We expected that over time the experiences of users would guide development of new BlastQuest functionalities. Given the freedom to explore BLAST results through BlastQuest, biologists appreciated the various grouping and rank sorting features. The ability to perform text searches on the "Hit Definition" field was particularly well liked as was the internal BLAST. While our BlastQuest implementation met the requirements originally set forth, once biologists started using the system we received numerous constructive suggestions.

Biologists rapidly discovered the limitations of BlastQuest functions, particularly as they focused on answering specific biological questions. In most cases delivering the needed information was as simple as writing a suitable SQL query to represent information already residing in the database in a new way. For example, the original BlastQuest implementation did not include the ability to group query sequences based on their having database hits in common, i.e., that two or more query sequences share DNA sequence homology with the same GenBank database sequence record. GI numbers are used to uniquely identify GenBank database records. Grouping BLAST homology results by GI number identifies query sequences having a common homolog, even if each query sequence matches to non-overlapping regions within the homologous sequence.

BLAST sequence homology searches are the primary method for assigning biological function to newly determined genetic sequences. While very powerful, BLAST searches assign biological meaning on a query-by-query basis, but do not establish a biological relationship between an individual query sequence and other sequences in the project. Biologists working with BlastQuest requested we develop a feature for grouping and counting gene sequences that are related by co-participation in common biological processes. The GO and KEGG consortia are two of the groups building classification schemes for biological information. As a result of user requests we added the GO Tree and KEGG pathways database to BlastQuest.

BlastQuest has proven to be a dynamic platform for building and testing a delivery system for biological information. We learned from user experience and implemented user defined changes that extend the functionality of BlastQuest beyond initial requirements. The community of BlastQuest users are also stakeholders in the success of BlastQuest and as the primary beneficiaries remain willing and active participants in this evolving project.

6.2. BlastQuest as focus for interdisciplinary collaboration

Biologists engaged in genomics research are confronted with two main problems, namely an exponential accumulation of heterogeneous, semi-structured biological information, and the increasingly complex applications needed to compile and analyze biological data. As a result, challenges in biology are now also challenges in computer science. Solving these challenges requires interdisciplinary collaboration between biologists and computer scientists. The success of such interdisciplinary groups requires integration of our respective scientific cultures and bridging this gap is the first real challenge for interdisciplinary collaboration. Our BlastQuest implementation did not require development of novel algorithms or new applications. Our initial data management objective was easily met by implementing a standard relational database solution. From the biologists perspective, BlastQuest solved an immediate data management predicament for delivering information and thereby enhanced biological research at our institution. For our interdisciplinary group the benefit of working together and developing BlastQuest is less tangible but perhaps more valuable than BlastQuest itself. It served as a starting point for biologists and computer scientists to begin working together. BlastQuest was a realistic goal and an opportunity to learn about each other in achieving a firm foundation for future collaboration.

7. Related work

Although parsing BLAST results and storing them in a database is not a novel idea, comparable Web-based tools with a similar functionality as BlastQuest are rare. We are aware of several projects, e.g., XS BLAST [15], WebBLAST [8], OCGC BLAST [4], and PEDANT [9], which pursue the same goal of evaluating BLAST query results but fall short in several important aspects.

XS BLAST [15] parses BLAST results in XML-format and stores the information in a relational database. It allows the display of the result in a summary tabular form with links to detailed alignments and provides some simple sorting functions. However, these manipulations can only be applied to BLAST results for a single query, but not a group of query sequences, which is the biggest limitation for XS BLAST.

WebBLAST [8], which is a suite of pipelined Perl programs, is mainly intended for archiving sequencing data and performing basic analysis tasks which are similar to those of BlastQuest. However, WebBLAST tools are purely file-based. Global filtering and grouping operations, or a mechanism for searching all BLAST results on user-supplied text terms are not available because their realization requires database technology.

The OCGC [4] BLAST results manager appears closest to BlastQuest in functionality, allowing selected viewing and data filtering on up to five criteria. However, it does not support the generation of a global view of the data on a project level, which is also the case for XS BLAST and

Table 1
Comparison of functions for BlastQuest and other similar systems

	BlastQuest	XS BLAST	WebBLAST	OCGC BLAST	PEDANT
Summary for one query sequences	+	+	+	+	+
Summary for multiple query sequences	+	–	+	+	–
Filtering and ordering	+	+	+	+	–
Text/String search	+	–	+	+	+
Pattern search	–	–	–	–	+
Internal BLAST search	+	–	–	–	+
Project management	+	–	+	+	–
Grouping	+	–	–	–	–
Protein/Gene function classification	+	–	–	–	+
Ortholog classification	+	–	–	–	–
Organizing into pathways	+	–	–	–	+

WebBLAST. Therefore, OCGC lacks the connection to GO, orthologs and pathways databases. A nice feature is the display of results in three different graphical alignments.

PEDANT [9] is a genome annotation system that stores information on genetic elements, function annotations, and structure annotations. In a general sense, PEDANT covers a much larger range of sequence annotation features than BlastQuest. However, BlastQuest focuses on supporting the analysis of BLAST data to facilitate large-scale EST sequencing projects rather than annotating entire genomes: first, BlastQuest has a user and project based structure, which allows frequent updates according to the status of the ongoing projects. Second, instead of statically displaying assigned annotations for each gene, BlastQuest supports various ordering, filtering, grouping and querying functions to facilitate the iterative gene discovery process. Third, BlastQuest utilizes the wide-accepted GeneOntology terms for function annotation while PEDANT invents its own functional categories. Fourth, BlastQuest helps biologists to discover the interrelationships among genes by putting them in a pathway context, a function which is absent from PEDANT.

In Table 1, we compare some important functions for BlastQuest, XS BLAST, WebBLAST, OCGC BLAST, and PEDANT.

A previous version of the BlastQuest system has been described in [7]. It depicts a first design of the system architecture and the user interface for the processing and evaluation of BLAST query results. A detailed description of our vision of the Genomics Algebra can be found in [12]. The paper in [13] stresses that the data management challenges in life sciences can only be solved by intensively employing modern database technology, which offers an appropriate framework for biological data handling but which itself also requires several extensions for coping with complex biological data.

8. Planned improvements

It is clear that there continues to be opportunities for additional improvements to BlastQuest. For example, there is need for a representation of the data that is semantically richer than what is currently available including the ability to represent uncertainty and data provenance, for accessing and integrating data from different genomic repositories, and for a high-level query language is easy to learn and supports biological analysis my effectively than SQL does.

Based on the feedback we have received from our users, which illustrates the complexity of the information-related challenges that confront biologists, we are in the process of redesigning BlastQuest from the ground up. For example, providing users with a semantically rich representation of the genomics data as well as support for specialty functions, requires the design of a new data type system and operations, which must be integrated with the underlying database management system for efficient query processing and persistence. Another example, access to multiple genomics repositories, requires the ability to extract, translate, and reconcile heterogeneous data from multiple sources and store the integrated result using a global schema, which has been constructed either from the local schemas of the sources or based on general knowledge of the domain.

As a result, we are proposing a new genomics integration and management system that is based on two fundamental pillars:

- *Genomics Algebra*. This extensible algebra is based on the conceptual design, implementation, and database integration of a new, formal data model, query language, and software tool for representing, storing, retrieving, querying, and manipulating genomic information. It provides a set of high-level genomic data types (GDTs) (e.g., genome, gene, chromosome, protein, nucleotide) together with a comprehensive collection of appropriate genomic operations or functions (e.g., translate, transcribe, decode). Thus, it can be considered a resource for biological computation.
- *Unifying database*. Based on latest database technology, the construction of a unifying and integrating database allows us to manage the semi-structured or, in the best case, structured contents of genomic repositories and to transfer these data into high-level, structured, and object-based GDT values. These values then serve as arguments of Genomics Algebra operations. In its most advanced extension, the Unifying Database will develop into a global database comprising many publicly available genomic repositories.

A detailed description of this proposed approach goes beyond the scope of this paper. However the reader is invited to refer to [12] for an in-depth description of the two pillars and their interactions.

We believe our proposed approach will cause a fundamental change in the way biologists analyze genomic data. No longer will biologists be forced to interact with hundreds of independent data repositories each with their own interface. Instead, biologists will work with a unified database through a single user interface specifically designed for biologists. Our high-level Genomics Algebra allows biologists to pose questions using biological terms, not SQL statements. Managing user data will also become much simpler for biologists, since his/her data can also be stored in the Unifying Database and no longer will she/he have to prepare a custom database for each data collection. Biologists should, and indeed want to invest their time being biologists, not computer scientists.

In addition, we believe the Genomics Algebra approach empowers biologists to perform complex analyses on large-scale collections of data without needing a computer scientist at their side. This is especially beneficial to biologists who work alone or in a small group since it "levels the playing field" permitting any biologist with a good idea to pursue it fully and not be discouraged by the lack of local infrastructure and support personnel.

9. Conclusions

In this paper we have described BlastQuest, a Web-based and interactive tool for importing and persistently storing genomic data from multiple BLAST queries in a relational database, applying DBMS functionality for processing and querying these data, and visualizing them appropriately. In addition BlastQuest supports the ability to connect sequence identities inferred from BLAST results with gene-associated biological functions described through the efforts of the Gene Ontology (GO) Consortium and pathway information from the Kyoto Encyclopedia of Genes and Genomes (KEGG). This type of cross-referencing has shown to be an ideal way to describe the functionality of a newly, discovered gene and helps biologists annotate and catalogue the genes in a way that is universally accepted.

BlastQuest is being supported by the Interdisciplinary Center for Biotechnology Research (ICBR) at the University of Florida and has been successfully employed and tested by scientists on campus and their collaborators around the world for over eighteen months. We are now in the process of developing the next-generation BlastQuest, which addresses the limitations of our existing concept mainly with respect to the need for a more expressive and extensible representation and data model, tools to support the browsing and integration of external repositories, and a richer and more intuitive query language that can be extended with new analytical functions and that can take advantage of the new data model.

References

[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool, Journal of Molecular Biology 215 (1990) 403–410.

[2] A. Bairoch, R. Apweiler, The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000, Nucleic Acids Research 28 (1) (2000) 45–48.

[3] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, D.L. Wheeler, GenBank, Nucleic Acids Research 31 (1) (2003) 23–27.

[4] J. Cuticchia, S. Parameswaran, R. Alexandrova, E. Crowdy, OCGC Blast, 1999. Available form <http://www.ocgc.ca/ocgcblast.htm>.

[5] B. Ewing, P. Green, Base-calling of automated sequencer traces using phred. II Error probabilities, Genome Research 8 (1998) 186–194.

[6] B. Ewing, L. Hillier, M.C. Wendl, P. Green, Base-calling of automated sequencer traces using phred. I Accuracy assessment, Genome Research 8 (1998) 175–185.

[7] W.G. Farmerie, J. Hammer, L. Liu, M. Schneider, Having a BLAST: Analyzing Gene Sequence Data with BlastQuest, in: 1st International Workshop on Biological Data Management, pp. 37–41, 2003.

[8] E.S. Ferlanti, J.F. Ryan, I. Makalowska, A.D. Baxevanis, WebBLAST 2.0: an integrated solution for organizing and analyzing sequence data, Bioinformatics 15 (1999) 422–423.

[9] D. Frishman, K. Albermann, J. Hani, K. Heumann, A. Metanomski, A. Zollner, H.-W. Mewes, Functional and structural genomics using PEDANT, Bioinformatics 17 (1) (2001) 44–57.

[10] Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. Genome Research 11 (2001) 1425–1433.

[11] D. Gordon, C. Abajian, P. Green, Consed: a graphical tool for sequence finishing, Genome Research 8 (1998) 195–202.

[12] J. Hammer, M. Schneider, Genomics Algebra: a new, integrating data model, language, and tool for processing and querying genomic information, in: 1st Biennial Conference on Innovative Data Systems Research, pp. 176–187, 2003.

[13] J. Hammer, M. Schneider, Going back to our database roots for managing genomic data, OMICS—A Journal of Integrative Biology 7 (1) (2003) 117–119.

[14] K. Holland, P. Thimote, L. Liu, E. Almira, W.G. Farmerie, An approach to gene discovery in ALS crops, in: Habitation 2004 Conference on Space Habitation Research and Technology Development, Orlando, FL, January 4–7 2004, Elmsford, pp. 123–145.

[15] J. Moon, E. Lefkowitz, XS BLAST (XML–SQL BLAST). Available from <http://www.poxvirus.org/blast_xml_start.asp>.

[16] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, M. Kanehisa, KEGG: Kyoto encyclopedia of genes and genomes, Nucleic Acids Research 27 (1) (1999) 29–34.

[17] J. Schultz, R.R. Copley, T. Doerks, C.P. Ponting, P. Bork, SMART: a web-based tool for the study of genetically mobile domains, Nucleic Acids Research 28 (1) (2000) 231–234.

William G. Farmerie is an Assistant Director of the Interdisciplinary Center for Biotechnology Research (ICBR) at the University of Florida. He received his B.S. degree in Biological Science from Florida State University in 1973, and his Ph.D. degree in Biomedical Sciences from the University of Tennessee in 1980. Prior to joining the University of Florida, Dr. Farmerie held Research Associate positions at the University of Michigan, and at the University of North Carolina at Chapel Hill. He is the Director of the ICBR large-scale DNA sequencing facility and the ICBR bioinformatics group, which provide genomics-based services to research scientists throughout the University of Florida.

Joachim Hammer is an Associate Professor in the Department of Computer and Information Science and Engineering at the University of Florida. He received his B.S. degree in Computer Science and Applied Mathematics from the University of Rochester in 1988, and his M.S. and Ph.D. degrees in Computer Science from the University of Southern California in 1990 and 1994. Prior to joining the University of Florida, Dr. Hammer was a Research Scientist in the database group at Stanford University. His areas of research are heterogeneous and semi-structured information systems, federated databases, data warehousing, knowledge engineering and data management for biological databases. Dr. Hammer is a member of ACM and IEEE. Since August 2001, he is the elected secretary and treasurer for the Association for Computing Machinery's Special Interest Group Management of Data (SIGMOD).

Li Liu has an M.D. degree from the Peking Union Medical College in China and a M.S. degree in Computer Science from the New Jersey Institute of Technology. Dr. Liu is currently a senior bioinformatician at ICBR of the University of Florida. Her responsibility is to conduct computational analysis of sequence data and microarray data. Prior to joining ICBR, she worked on a biological database integration project at a biotechnology company.

Anuj Sahni received M.S. in Electrical and Computer Engineering from the University of Florida in 2002. He is currently working as a Bioinformatics Software Engineer in the Interdisciplinary Center for Biotechnology Research (ICBR), where he is responsible for designing and implementing high throughput bioinformatics software-tools including BlastQuest.

Markus Schneider received his Diploma degree in Computer Science from the University of Dortmund, Germany, in 1990, and his Ph.D. degree (Dr. rer. nat.) in Computer Science from the University of Hagen, Germany, in 1995. After that, until 2001, he was research assistant (lecturer) at the University of Hagen. Since 2002, he has been an Assistant Professor at the University of Florida, Gainesville, FL, USA. His research interests are spatial, spatio-temporal, fuzzy, and genomics databases. In particular, he focuses on the design and implementation of spatial data types (geo-relational algebra, ROSE algebra, realms), spatio-temporal data types (moving objects), fuzzy spatial objects, spatial and spatio-temporal partitions and networks, and deals with the design and implementation of data structures and geometric algorithms for these topics.