Having a BLAST: Analyzing Gene Sequence Data with BlastQuest

William G. Farmerie[1], Joachim Hammer[2], Li Liu[1], and Markus Schneider[2]

University of Florida
Gainesville, FL 32611, U.S.A.

Abstract

An essential problem for the biologist is the processing and evaluation of BLAST query results. We advocate the deployment of database technology and describe a user-driven tool, called *BlastQuest*. BlastQuest provides interactive, Web-enabled query, analysis, and visualization facilities beyond what is possible by current BLAST interfaces. Specifically, the BLAST results are extracted, structured, and stored persistently in a relational database to support a series of built-in analysis operations that can be used to select, filter, and order data from multiple BLAST results efficiently and without referring to the original result files. In addition, users have the option to interact with BLAST results through a forms-based interface.

1. Introduction

Biologists are nowadays confronted with two main problems, namely the exponentially growing volume of biological data of high variety, heterogeneity, and semi-structured nature, and the increasing complexity of biological applications and methods afflicted with an inherent lack of biological knowledge. As a result, many and important challenges in biology and genomics are challenges in computing and here especially in advanced information management and algorithmic design.

The currently most widely used and accepted tool for conducting similarity searches on gene sequences is BLAST (Basic Local Alignment Search Tool) [1]. BLAST comprises a set of similarity search programs that employ heuristic algorithms and techniques to detect relationships between gene sequences and 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 impedes its direct 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, filter, group, and order these 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 is being supported by the Interdisciplinary Center for Biotechnology Research (ICBR) at the University of Florida, and is used by campus researchers and their collaborators across the United States.

2. Biological tool requirements

A typical DNA sequencing project involves collections of several hundred to tens of thousands of DNA sequences. Nucleotide sequence homology searches are frequently the first step toward identifying the biological function of unknown nucleotide sequences. Most university-based investigators lack the computational expertise and infrastructure to initiate and manage 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 first having to overcome the *management* of their data.

With this in mind, there has been a clear need to build a centralized system to manage BLAST results. The BlastQuest project was initiated to help with the challenge of managing BLAST results and make this information available in a web-based interface accessible 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 our initial list of functional requirements for the BlastQuest system:

- *A BLAST results viewing tool accessible to research groups 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.

[1] Affiliation: ICBR Molecular Services Division: DNA Sequencing Core, Interdisciplinary Center for Biotechnology Research.
[2] Affiliation: Department of Computer & Information Science & Engineering.

- *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 only the most significant results is initially very informative.
- *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 that might happen by chance. There are no hard limits to the significance of these statistical parameters. The user will choose parameters giving either a more relaxed or restricted view as needed.
- *Selective data grouping on criteria such as GI number, or a defined number of top-scoring results*. 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.
- *A convenient interface for getting queries into and BLAST results out of the system*. The interface must be attractive and logically implemented so users will be able to find and use the tools the system provides.

3. BlastQuest user interface

BlastQuest simplifies large-scale analysis in gene sequencing projects by providing scientists with a means to filter, summarize, sort, group, and search BLAST output data. BlastQuest *extracts* gene data from XML files, which are returned as the result of homology searches from BLAST engines, and stores them in an underlying relational database. This allows the user to benefit from well-known database concepts like transactions, controlled sharing, and query optimization. Finally, BlastQuest also allows users to perform homology searches of their proprietary sequence data against public domain data, such as NCBI databases, etc.

The most frequently used user operations are hard-wired in the user interface and accessible via command buttons. To enable data analysis that is not directly supported, BlastQuest offers a more flexible, forms-based query interface. This interface essentially allows the user to construct complex boolean expressions as selection conditions which may include logical operators and substring search predicates.

In addition, BlastQuest can be linked to the so-called *SMART* (Simple Modular Architecture Research Tool [5]). The integration of BlastQuest output into SMART is in direct response to the desire by scientists for new tools and interfaces capable of accessing and integrating external resources into one system.

Finally, BlastQuest enables to manage BLAST data on a per-project or per-user basis using the *security features* of the underlying DBMS while at the same time allow *controlled sharing* of this data in order to support collaboration. A startup page facilitates the extraction of gene data from original, external BLAST files into a MySQL database. Due to the large volume of data, a simple page-by-page viewing is not helpful to the user but selection mechanisms are needed to find the data of interest. The overall strategy is to apply a sequence of consecutive operations on the data to gradually approach the data of interest. In the following we describe the main user interface features for doing this.

The first feature is to let BlastQuest create a *summary page* for selected sequence segments. Users require this high level summarization of their sequences because the volume of BLAST output data for large-scale sequencing projects is well beyond simple page-by-page viewing. This summary page gives an abbreviated overview of each query sequence with possible function. For each query DNA sequence, only the sequence database match with the best statistical score calculated by BLAST is displayed with a summary of important biological information like gene or protein name, possible biological functions, and, for each matching sequence, the GenBank sequence ID, gene definition, and expect value.

The second feature is *user-controlled selection*. Unfortunately, the statistically calculated ranking of matching sequences provided by BLAST does not necessarily correspond to the biological knowledge and experience of the user who may tag a different result as better for expressing the possible function of the query sequence. By manually selecting a specific query result, the user can get additional information such as the percentage of identity, the alignment of the query sequence and the matching sequence, or a detailed display of sequence alignments as a free-text formatted BLAST result to which most BLAST users are accustomed.

The third feature refers to *built-in selection* facilities activated by mouse-clicks and operating on all query sequences and their query results. Examples are the

displays of hits with expect values less than a particular *threshold* by selecting from a pull-down menu (e.g., shown in Figure 1), or restricting the display to the best *n* database matches for each query sequence. This permits the user to reduce the original BLAST result to a manageable size and to remove results of low quality.

The fourth feature comprises *ordering* and *grouping functions*. These help the user to discover relationships among genes or expression patterns. For example, there may be more than one sequence or contig that are derived from different regions of the same mRNA or gene. Grouping on GI number will cluster these related sequences and identify them for further analysis of their relationship. A special feature is grouping sequences on UniGene ID. This is an additional step to identify EST sequences that come from gene orthologs or gene paralogs. Another example is that biologists sometimes want to know which sequences have their functions well resolved by BLAST search, and which have not. By ordering query sequences by the expect values of top scoring BLAST hits, users identify sequences with high-quality hits, sequences with only low-quality hits, or even sequences having no hit. This step rapidly classifies sequences for different types of additional analysis. For example, if the user asks for grouping on GI number or query sequence, related sequences and their BLAST results are grouped together rather than appear randomly or out of context. This is also a proven method to identify EST sequences that come from different regions of the same mRNA, gene orthologs, or gene paralogs.

The fifth feature enables user-defined, forms-based queries because the built-in functionality of BlastQuest is sometimes insufficient for specific analysis tasks. For example, if a user wants to find out which sequences are homologous to genes with reverse transcriptase function, which is not hypothetical but is proved by empirical data, BlastQuest does not have built-in selection facilities for this specific query. To solve this problem, BlastQuest allows the user to interactively and textually construct complex boolean filter expressions which may include logical operators like "AND" and "OR" and substring search predicates like "Contains" or "Not Contains."

The sixth feature to be mentioned is *interoperability* between BlastQuest and other biological information systems. Creating links to other systems to make use of their specific functionality becomes more and more important for the biologist. In BlastQuest, after having examined the query sequences and their probable identities, we wish to derive the protein sequences encoded by the nucleotide sequence. Rather than translate the nucleotide sequence directly, BlastQuest takes the 'best' match, which represents a homologous gene closely related to the unknown query sequence, and retrieves the corresponding protein sequence as translated by BLAST. After grouping search results by query sequence (e.g., the best five statistical matches) the user is presented with the screen shown in the top half of Figure 1.

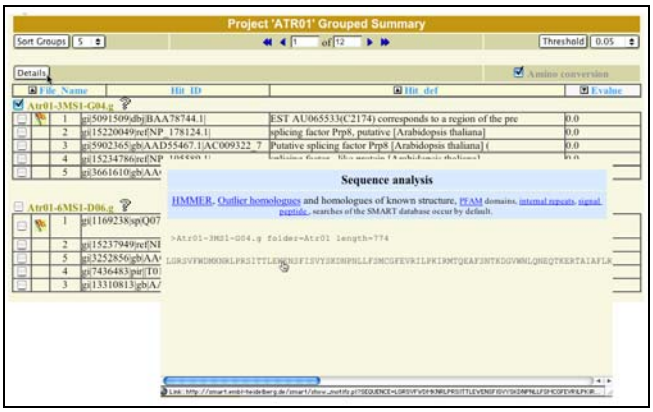

Figure 1: Filtering and grouping BLAST results per project.

Next, the user checks the 'amino conversion' box at the right top of the screen, and the check box adjacent to the query sequence they wish to translate into an amino acid sequence. When the user clicks the 'Details' button, the 'Sequence Analysis' screen shown in the bottom half of Figure 1 appears. The user may submit the derived protein sequence to the SMART protein analysis Web site by simply clicking on the amino acid sequence. Results of the SMART analysis will appear in the browser window.

Figure 2: Internal BLAST search user databases.

The seventh and final major feature is the capability to perform BLAST searches against the users' *own* sequence database. This allows the user to query their own sequence data with a specific nucleotide or protein sequence. If a user obtains an interesting sequence from other resources, internal BLAST search helps to find out whether s/he owns similar sequences. In this case, the corresponding clone is identified and retrieved from the users clone bank where it may be used for further experiments. In the example shown in Figure 2, the user pasted the query sequence into the top text area. The interface also allows input of a sequence file location for uploading. From drop-down menus, the user may choose one of several BLAST programs and different local target databases that s/he owns or has a "guest" privilege for.

BlastQuest also provides choices for choosing a homology matrix via a drop-down menu. After the user clicks the "BLAST" button, the query sequence is submitted with selected parameters. For individual blast query, the result will be displayed in HTML format. If the user has "owner" privilege, s/he can choose to either parse and store this BLAST output persistently into the MySQL database or delete it when the session ends. For batch queries, BLAST results will be parsed and automatically stored in the MySQL database for later analysis.

All operations described here can be combined to analyze data generated in a larger project. For example, one may use BlastQuest to retrieve hits with expect value lower than 0.05, followed by grouping on gene ID, and only display the top five matching hits per GI number (as illustrated in Figure 1).

4. Architectural overview of BlastQuest

Figure 3 depicts a conceptual overview of the 3-tiered BlastQuest system architecture. Tier 1 contains the database backend, which is implemented using the MySQL[3] RDBMS. The database backend stores and manages BLAST and PHRAP (Phragment Assembly Program) [4] results, which are represented as XML and ACE[4] (ArChivE) documents and whose structure has been mapped into the relations `Query`, `Assembly`, `Hit`, and `Query_Hit` shown later in Figure 4.

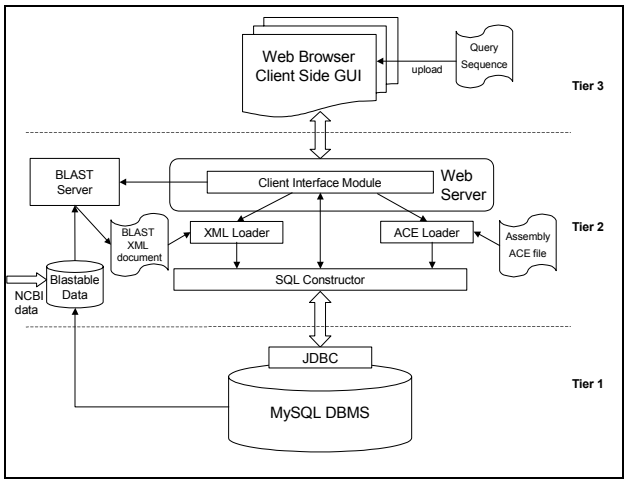

Figure 3: Conceptual overview of the BlastQuest system architecture.

For each query sequence submitted to the BLAST server (shown in the upper right-hand corner of Fig. 3), the relation `Hit` stores detailed hit information, such as hit definition, expect value, bit score, pairwise alignments and so forth. For queries, which do not produce a match in the homology search, the fields are marked as NULL. From a biological point of view, sequences with no homologous sequence match often lead to new genes and are analysed in a different manner (outside of BlastQuest). In addition, the homology search criteria for each BLAST search, such as the BLAST program name, database name, matrix, and date, are stored in `Query_Hit` table. These parameters are important to users because for the same query sequence, BLAST generates different results based on different criteria. For example, BLASTN results and BLASTX results may indicate different functions for the same query sequence. In addition, the same BLAST search on different days may generate different hits since BlastQuest's BLAST server is regularly updated with the latest version of the NCBI data files. The MySQL database also stores information about how related gene segments are assembled into single consensus DNA sequences by PHRAP, which is external to BlastQuest and invoked before the DNA sequence results are submitted to BLAST. PHRAP outputs its results in an ACE file, which is mapped into the relation `Assembly`. If the user considers the results of the BLAST search interesting, s/he may want to extract the physical clones from which the specific query sequences are generated or assembled. This is possible by joining the `Assembly` and `Query` tables via the "qid" foreign key to retrieve all segments and corresponding clone names that are clustered into a specific query sequence.

User (uid, user_name, password, email)

Project (pid, project_name, status)

User_Proj (uid[User], pid[Project], privilege)

Underlined attributes denote the unique identifiers (primary keys) for each relation. Attributes with a superscript are foreign keys (superscript denotes the referenced relation).

Query (qid, pid[Project], query_name, query_sequence, aid)

Assembly (aid, qid[Project], genus_name, lib_number, master_pl_id, sequence_pl_id, well_row, well_column, type_flag)

Hit (hid, hit_num, hit_id, hit_acc, hit_def, hit_len, hsp_num, hsp_bit_score, hsp_score, hsp_evalue, hsp_positive, hsp_identity, hsp_density, hsp_hit_frame, hsp_query_frame, hsp_align_len, hsp_query_fram, hsp_query_to, hsp_hit_fram, hsp_hit_to, hsp_pattern_from, hsp_pattern_to, hsp_gaps, hsp_qseq, hsp_hseq, hsp_midline)

Query_Hit (qid[Query], hid[Hit], program, database, matrix, date)

Figure 4: Relational schema of the BlastQuest MySQL database.

The database also maintains information about users and their corresponding gene sequencing projects, which are stored in the three remaining relations, `User`, `Project`, and `User_Proj`. The relation `User_Proj` represents the many-many relationship between scientists and the projects to which they belong. Since all sequence data is organized by project (using the PID foreign key in relation `Query`), BlastQuest provides control over which user has access to which data.

[3] See http://www.mysql.com/.
[4] See http://bozeman.mbt.washington.edu/phrap.docs/phrap.html.

Tier 2 contains the multi-threaded BlastQuest application program, which is divided into five modules: The *BLAST Server*, which is used to conduct BLAST searches against NCBI as well as internal data owned by the users; the *client interface module*, which handles communication with the Web clients in tier 1; the two *loader modules* for extracting and loading data from the XML and ACE input files into the database; and the *SQL constructor* for assembling the queries to be sent to the database. The BLAST Server is downloadable freeware from NCBI. The client interface module is implemented as a series of Java Server pages (JSPs) that execute inside a Tomcat server. The remaining three modules are implemented as Java classes. We briefly highlight the functionality of each module.

BlastQuest maintains a local version of NCBI's "NR" database, which is, updated monthly with new releases and can be searched with a local copy of the BLAST server (labeled "Blastable Data" in Figure 3). In addition to public domain data, this local BLAST database also contains blastable data from each user's proprietary query sequences. The conversion of query sequence data into blastable data is done using the "formatdb" program provided with NCBI's BLAST search engine.

The XML loader parses each BLAST result file into a Document Object Model (DOM) representation using the Xerces Java Parser 1.4.4. The XML loader then extracts the relevant data items needed to populate the `Hit` and `Query_Hit` tables. Specifically, the loader module contains several classes whose data structures correspond to the tables in the database schema. When the loader collects data from an XML file, it populates the appropriate class objects with the extracted values. At the end, the objects are passed to the SQL constructor, which creates the SQL commands to insert the values into the relational database. The ACE loader works in a similar fashion. However, since there was no standard ACE parser available, we created our own. Our event-based parser detects the presence of certain keywords in the ACE input file and extracts the information associated with that keyword. Other, more efficient loading options are possible, for example, by using the bulk loading utilities of the DBMS. However, by making our loader modules part of the Web-based middleware, users can load BLAST results into their BlastQuest accounts from anywhere on the Web.

The *SQL constructor* is the gateway between the database and the middleware. It connects to the MySQL relational database engine via the JDBC driver and manages a pool of connections to the database engine.

Tier-3 is a (thin) *client interface*, which is implemented as dynamic Web pages displayed inside a Web browser. Client-side processing is limited to validation of user input, submitting requests to the BlastQuest application and displaying HTML results.

5. Related research

We are aware of two comparable Web-based tools, WebBLAST 2.0 [3] and OCGC BLAST [2], which pursue the same goal of evaluating BLAST query results but fall short in several important aspects. Both tools are purely file-based, do not offer any kind of database support, and are thus only able to provide the user with a fixed, non-extensible pool of evaluation functions. WebBLAST, 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. Global filtering and grouping operations, or a mechanism for searching all BLAST results on user-supplied text terms are not available. Their realization requires database technology. The OCGC BLAST results manager appears closest to BlastQuest in functionality, allowing restricted selected viewing and data filtering on up to five criteria. A nice feature is the display of results in 3 different graphical alignments.

6. Conclusion

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. BlastQuest is being supported by the Interdisciplinary Center for Biotechnology Research (ICBR) at the University of Florida, and is used by campus researchers and their collaborators across the United States.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403-410, 1990.

[2] J. Cuticchia, S. Parameswaran, R. Alexandrova, and E. Crowdy, "OCGC BLAST," 1999, http://www.ocgc.ca/ocgcblast.htm.

[3] E. S. Ferlanti, J. F. Ryan, I. Makalowska, and A. D. Baxevanis, "WebBLAST 2.0: an integrated solution for organizing and analyzing sequence data," *Bioinformatics*, vol. 15, pp. 422-423, 1999.

[4] P. Green, "PHRAP-sequence-assembly program," 1999, http://www.genome.washington.edu/UWGC/analysistools/Phrap.cfm.

[5] J. Schultz, R. R. Copley, T. Doerks, C. P. Ponting, and P. Bork, "SMART: A Web-based tool for the study of genetically mobile domains," *Nucleic Acids Research*, vol. 28, pp. 231-234, 2000.