

Database management systems are widely used in large-scale bioinformatics pipelines. They provide efficient and stable data repositories for each computational component in the pipeline. However, traditional database systems were not designed with bioinformatics applications in mind, thus, important datatypes and bioinformatics functionalities are required in order to represent the data naturally and integrate biological information seamlessly in these environments. Besides, biological knowledge exhibits diverse and dynamic relationships. Relationships between biological entities might be invalidated or added anytime; therefore, fixed database schema is obviously not enough for biological knowledge representation.

We identified some essential functionalities for bioinformatics databases: 1) support important biological abstractions, such as sequence, range, location, as well as the extensibility required by biological data. 2) support efficient indexing schemes for biological datatypes. 3) integrate database functionality with BLAST. We developed the notion of Bioinformatics Indexing, a conceptual model for representing and managing biological information. We also created BLASTgres, an extension of PostgreSQL to support these requirements.

Bioinformatics Indexing includes two important components: intrinsic indexing – indexable biological datatypes, and extrinsic indexing – joinable BLAST alignment. Intrinsic indexing provides essential biological datatypes to represent biological information in the database while extrinsic indexing relates one biological entity to another.

One of the most essential biological datatypes is location. The location abstraction, a sequence identifier and an integer interval to specify a particular segment in sequences, is used ubiquitously in this field. Biological features are generally attached to a segment of the sequence (i.e. a location), and location is also the basic unit for maps, alignments and other complex relationships. The concept of location might be simple but it can be tricky to handle. Locations can be denoted relative to different reference coordinates and multiple locations can denote the same sequence segments. Without the support of location datatype in bioinformatics database systems, users are required to implement their own codes to handle various location operations. Moreover, different representations of location create difficulties to share and exchange data. Extra efforts are required for data transformation. Locations are handled poorly in traditional databases because they are often unequipped to support location abstraction and queries over intervals. Query optimizers are generally unable to optimize query criteria involving inter-dependent attributes. Implementing complex relationships between locations can be complicated and error-prone. In addition, in order to process the enormous volume of information about locations, support for the location datatype requires indexing -- the ability to create index structures for efficient retrieval of locations. Traditional relational database systems only support a limited set of well-known index structures and these indexing schemes are only used for a limited set of native datatypes. Bioinformatics database systems must provide index schemes for interval and location datatype in order to have efficient data retrieval.

On the other hand, important biological tools, such as BLAST, have to be integrated into the database systems. BLAST identifies local regions of optimal similarity between sequences in sequence databases and a given user's sequence, providing a way to relate one sequence to another. It enables us to infer homologous relationships between sequences and infer functional and evolutionary relationships. One of the major advantages of BLAST is its ability to compare the sequence in question with the most up-to-date databases available. In the post-genomic age, the abundant mass of available biological data has made it possible for researchers to conduct large-scale analyses and make new discoveries in short periods of time. Hence, BLAST is an indispensable component in bioinformatics computing environment and the ability to integrate BLAST results with other biological information is essential.

Even though the current BLAST tool is very powerful, it is usually required an ad hoc approach to utilize its result. Researchers typically have to write codes to integrate biological information with BLAST hit results and a lot of human interventions are needed to annotate the resulting sequences and further processing, such as clustering of BLAST results and filtering out unwanted results. Large-scale bioinformatics research will need scalable, automatic ways to find and annotate sequences of interest. Ideally, interoperation of indexing schemes like BLAST with biological databases should be managed under a unified model.

In deed, the integration of BLAST into a biological database system provides an automatic way to relate various sequences together through similarity relationship. The support of BLAST in database systems not only eases the process of data integration between different heterogeneous bioinformatics data sources, but also provides advanced query abilities that were not provided in traditional BLAST tools. Advanced control over BLAST results can also be achieved through the integration. Since BLAST only supports a very limited number of filtering conditions for the search results, such as accepting only matches with certain range of alignment length or percent identity. Advanced query mechanism found typically in database systems can be utilized to BLAST support. With the BLAST support in database systems, constraints that are more stringent or not supported by BLAST for the results can be set by adding conditions in the WHERE clause. BLAST results can also be viewed and analyzed by sorting or grouping by any attribute.

BLASTgres extends PostgreSQL to support these concepts. We developed more than 30 types of location operations and query optimization information (such as regarding ordering, commutativity or negation) is also provided to permit optimization of important operations like merge-join, hash-join or general theta-join. Location indexing is implemented through GiST (General Index Search Tree) interface. Several query predicates such as overlap, equality, comparison are supported for location indexing. BLAST functionalities are realized through table functions and BLAST results can be joined with other tables like a normal table.

We have used this system for several large-scale analyses such as genome-wide detection of alternative splicing event and cross-genome bioinformatics data mining. A significant performance boost was observed in this implementation. BLASTgres could be a great tool for large-scale bioinformatics data mining research.

BLASTgres rests on PostgreSQL database system, which is supported under major operating systems such as UNIX-like and PC systems. It is freely available over the Internet and can download it from http://www.loni.ucla.edu/Software/Software_Detail.jsp?software_id=15.