OMICS A Journal of Integrative Biology
Volume 7, Number 1, 2003

The Role of Declarative Querying in Bioinformatics

JIGNESH M. PATEL

THE RECENT PUBLICATION of a draft of the entire human genome (McPherson et al., 2001; Venter et al., 2001) has served to fuel an already explosive area of research in bioinformatics that is involved in deriving meaningful knowledge from proteins and DNA sequences (Alberts et al., 2002). Even with the full human genome sequence now in hand, scientists still face the challenges of determining exact gene locations and functions, observing interactions between proteins in complex molecular machines, and learning the structure and function of proteins, just to name a few. The progress of this scientific research is closely connected to the research in the database community in that analyzing large volumes of biological data sets involves being able to maintain and query large databases (Moussouni et al., 1999; Davidson, 2002). Database management systems (DBMSs) could help support life sciences applications, in a number of different ways. A partial list of tasks that such applications require is: querying large structured databases (such as sequence and graph databases), querying semi-structured (such as published manuscripts), managing data replication, querying distributed data sources, and managing parallelism in high-throughput bioinformatics. Unfortunately, current DBMSs have largely ignored supporting life sciences applications, and consequently, the life sciences researches have been forced to write tools and scripts to perform these tasks.

An interesting parallel can be drawn between the state of data management tools in life sciences, and the state of data management tools for business applications, such as a banking application, about three decades ago. Prior to the advent of the relational data model, business data was managed and queried using customized programs/scripts that were developed for each application. Reusing programs, and the algorithms for querying the data, involved rewriting application program and logic, which was very time consuming and expensive. In addition, the querying programs were closely tied to the format that was used to represent the data. Any change in the format of the data representation often would break the querying programs. Furthermore, writing complex queries, such as querying over multiple data sets or posing complex analytical queries, was a daunting task. One of the critical contributions of the relational data model (Codd, 1970) was the introduction of a declarative querying paradigm for business data management, instead of the previously used procedural paradigm. In a declarative querying paradigm, the user expresses the query in a high-level language, like SQL, and the DBMS determines the best strategy for evaluating the query. In addition, the DBMS only presents to the user a logical view of the data against which queries are posed. The physical representation of the data, either on disk or in-memory, can be very different from the logical view. For example, in a relational database management system (RDBMS), indices may be created, and the user doesn't have to query against the index. The user still queries against logical relations, and the system automatically determines if it is faster to use the indices to answer a query. The user is thus insulated from worrying about various details such as physical organization of data on disk, the exact location of the data, tuning the representation for better performance, and choosing the best plan for evaluating a query. This declarative querying paradigm has been a huge success for relational DBMSs, and today commercial RDBMSs manage terabytes of data, and allow very complex querying on these databases. Database management systems can provide similar benefits to the life sciences community, just as it did three decades ago to the business data management community. Many of the data sets that are used in life sciences are growing at an astonishing rate (such as sequence data at NCBI's GenBank (NCBI, 2002)), and the queries

Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan.

that are required to process these data sets to extract biologically meaningful information are also increasing in complexity. Database management systems can and should play a significant role in managing these data sets.

To better support life sciences applications, DBMSs must carefully consider the data types and the operations on the data types that are required to support these applications. One of the commonly used data type in these applications is a sequence. A sequence is often used to represent a protein's primary structure, or genes. Querying on sequences is often approximate, typically using an edit-distance matrix, such as PAM or BLOSUM matrices. Sequences have a well defined structure, and one would expect that the currently generation of commercial object-relational DBMSs, which are very good at dealing with structured data, would have support for this data type. Unfortunately, this is not the case, and such data sets are managed and queried using customized programs that run outside the DBMS. Imagine a research group that queries on such data sets frequently, and has a local copy of the sequence data set and programs to query this data set. Since new sequences are continually added to the *master* copy, keeping the data set up-to-date is now a largely manual process. If the program needs to build any auxiliary structures to process this data, such as an index (Kent, 2002), then this task requires customized solutions, which are cumbersome to manage. On the other hand, this is a form of replication that DBMSs are good at dealing with, and often can be managed transparently without requiring any user intervention. As another example, consider a scenario in which the user wants to query a number of different distributed sequence data sets with the same query, and then combine these results using some scoring criteria, perhaps with some post-processing on the sequence data. This task again requires customized programs, and a declarative querying interface for this task can not only make it easier to pose such queries, but can also be executed more efficiently. Improvements in efficiencies can be achieved since the DBMS can examine a number of alterative plans for evaluating the query, and can then choose the cheapest plan. Similar observations hold for other data types, such as graphs, manuscripts, and micro-array data, and DBMSs can play a significant role in managing such data sets.

In summary, three decades ago DBMSs brought significant benefits to the management of business data, and today most business data management applications are developed using a DBMS. DBMSs can provide similar benefits in managing data sets that are used in life sciences experiments. Of particular interest is the provision of a declarative querying paradigm that allows the scientist to focus on the question that they want to pose to the database system, rather than the methods that they need to combine to extract an answer to that question. A declarative querying paradigm not only eases the task of posing queries that are currently used, but also opens up new opportunities for extracting deeper information from the database by allowing the scientist to pose more complex queries. As part of the Periscope project at the University of Michigan, we are examining various research issues in designing query languages, data storage and indexing methods, query processing algorithms (Hammel and Patel, 2002), and query optimization techniques that are required to support a declarative querying paradigm for life sciences applications.

ACKNOWLEDGMENTS

This research has been supported in part by NSF under grant IIS-0093059 and by a gift donation from Eli Lilly.

REFERENCES

ALBERTS, B., JOHNSON, A., LEWIS, J., et al. (2002). *Molecular Biology of the Cell,* 4th ed. (Garland Publishing, New York).

CODD, E.F. (1970). A relational model of data for large shared data banks. CACM **13,** 377–387.

DAVIDSON, S.B. (2002). Tale of two cultures: are there database research issues in bioinformatics? SSDBM 3.

HAMMEL, L., and PATEL, J.M. (2002). Searching on the secondary structure of protein sequences. VLDB 634–645.

KENT, W.J. (2002). BLAT: the BLAST-like alignment tool. Genome Research **12,** 656–664.

McPHERSON, J.D., MARRA, M., HILLIER, L., et al. (2001). A physical map of the human genome. Nature **409,** 934–941.

MOUSSOUNI, F., PATON, N.W., HAYES, A., et al. (1999). Database challenges for genome information in the post sequencing phase. Presented at the DEXA'99, Florence, Italy.

NCBI. (2002). GenBank statistics. Available: www.ncbi.nlm.nih.gov/Genbank/genbankstats.html. NCBI.

VENTER, J.C., ADAMS, M.D., MYERS, E.W., et al. (2001). The sequence of the human genome. Science **291,** 1304–1351.

Address reprint requests to:
Dr. Jignesh M. Patel
Department of Electrical Engineering and Computer Science
University of Michigan
2239 EECS Building
1301 Beal Avenue
Ann Arbor, MI 48109-2122

E-mail: jignesh@eecs.umich.edu