ODM BLAST: Sequence Homology Search in the RDBMS

Susie Stephens
Oracle Corporation
10 Van de Graaff Dr.
Burlington, MA 01803
susie.stephens@oracle.com

Jake Y. Chen *
Indiana University
School of Informatics
Indianapolis, IN 46202
jakechen@iupui.edu

Shiby Thomas
Oracle Corporation
10 Van de Graaff Dr.
Burlington, MA 01803
shiby.thomas@oracle.com

Abstract

Performing sequence homology searches against DNA or protein sequence databases is an essential bioinformatics task. Past research efforts have been primarily concerned with the development of sensitive and fast sequence homology search algorithms outside of the relational database management system (RDBMS). Oracle Data Mining (ODM) BLAST enables BLAST to be performed in a RDBMS. ODM BLAST relieves the burden of moving data out of the RDBMS, eliminates the need to parse data files, and allows BLAST results to be integrated with existing RDBMS data. Oracle has simplified BLAST searches to a single SQL statement. ODM BLAST shifts algorithm development from bioinformaticians to the RDBMS provider.

1 Introduction

Sequence homology searching is an essential bioinformatics task. Tools such as BLAST [AGM$^+$90] and FASTA [Pea00] can be used to search a query sequence against a target database of sequences. If matched sequences are found, users can further examine sequence similarity to determine the identity of the query sequence, or characterize its functions by homology. With the rapid accumulation of genomic sequences, sequence homology searching has become a daily routine for genome annotation, comparative genomics, and evolutionary biology studies.

High-throughput biological data from sequencing machines, microarrays and protein mass spectrometers present new challenges for sequence homology searching. Web-based sequence homology search tools are popular; however, users can only perform searches one at a time. To perform the large-scale batch searches that are now required, software developers have had to build stand-alone sequence homology search servers. Frequently the software needed to perform such tasks has been written for individual groups, and consequently has poor portability and customizability. This re-invention of software functions across many organizations is an ineffective use of resources. Unless there is a robust strategy to integrate the results of homology searches with the in-house biological data that are managed in relational databases, the interpretation of gigabytes of sequence data becomes intractable.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

*Also at: Purdue University, School of Science, Department of Computer and Information Science, Indianapolis, IN 46202.

Past research efforts have been primarily concerned with the development of sequence homology search algorithms outside of the RDBMS [Ken02, Sim99]. Additionally, a relational database operator, *Similar_Join*, was developed to make an abstraction of batch sequence homology searches [CC03], and DiscoveryLink relied upon application and data source wrappers to make results from tools such as BLAST available to SQL analysts [HSK+01].

ODM BLAST enables data analysts to use SQL to invoke BLAST functions in Oracle Database 10*g*. This work is built on the idea of extending the capability of a general-purpose RDBMS into the biology domain. With ODM BLAST being integrated into the RDBMS, data can remain in the RDBMS for analysis, which has performance and data management advantages. Once data have been entered into the database, no more parsing of the data is required, regardless of the group that is accessing the data. A strong RDBMS environment provides security, auditing, and high availability of data. With ODM BLAST, it becomes feasible to ask questions including "Retrieve similar sequences, where the sequence was entered into Genbank after 2002, and the sequence is from E. coli". Batch and automated SQL queries also become simpler, for example, "Query all human sequences against all yeast sequences".

2 Implementation

Oracle implemented BLASTN, BLASTP, BLASTX, TBLASTN, and TBLASTX inside Oracle Database 10*g*. BLAST_MATCH can be invoked to retrieve the sequence identifier and similarity results; and BLAST_ALIGN can be invoked to retrieve the sequence identifier, similarity results and full alignment information. Detailed overview of ODM BLAST is available at [ora04]:

The ODM BLAST implementation takes advantage of the Oracle table function feature. This feature is part of the Oracle RDBMS extensibility framework, which allows developers to write code that is invoked using SQL queries. A table function returns its results as virtual tables, which can be manipulated like other relational tables. This implementation allows ODM BLAST to be invoked either by *ad hoc* SQL queries or by embedding the functionality into applications.

The ODM BLAST table function accepts a query sequence, a reference cursor that specifies the sequences that the query sequence needs to be searched against, and several other parameters that control the search. The query sequence is passed to the underlying server side programming code as a Character Large OBject (CLOB). The reference cursor, which specifies the target sequences, must contain two attributes: a sequence identifier of the data type VARCHAR and a sequence data string as a CLOB. The native programming code then takes these two input parameters, performs the search, and sends the results as a virtual table to the invoking ODM BLAST table function. Since the server-side process runs BLAST by loading query and target sequences from disk directly to memory, the overhead of copying files to different disk locations is eliminated.

2.1 Description

In Figure 1, a BLASTP_MATCH query was invoked to perform a protein sequence homology search against the target protein database *target_db*. The query sequence in *Block A* and the target database in *Block B* are specified as SQL sub-queries. The query sequence in *Block A* is specified on the fly. The query sequence in *Block B* is specified as a cursor for the subset of rows from the table *target_db*. Target sequences beginning with 'NP' are specified [PM01]. The top-level WHERE clause states that the search results must have an E-value of less than 1E-6 ($t.expect < 0.000001$). Finally, in the top-level SQL query FROM clause, *target_db* (table *g*) and BLASTP results (table t) were *JOIN*ed by *g.refseq_id = t.t_seq_id*. This enables search results to be integrated with annotation data in the RDBMS.

Writing SQL based BLAST queries should be simpler for bioinformaticians than writing PERL or Java code. The simple syntax should enable biologists to write basic queries. Queries could be further simplified if query

I. ODM BLAST Query Example

```
SELECT t.t_seq_id, t.score, t.expect, g.function_desc, g.locus_symbol, g.map_location
  FROM target_db g,
       TABLE                                                    -- BEGIN BLASTP
          (BLASTP_MATCH
              ((SELECT 'AKRELKKLKIEARSVERYKKLRIGMENKIMQLQRKVDEQNKDYKCLVEKLTNLEGIYNSETEKL
                        RSDLERLQLSEEEAKVATGRVLSLQEEIAKLRKDLEQTRSEKKCIEEHADRYKQET
                 FROM DUAL),                                                  Block A
               CURSOR (SELECT x.refseq_id, x.sequence_string
                         FROM target_db x
                        WHERE x.refseq_id LIKE 'NP%')
              )                                                   Block B
          ) t                                                   -- END BLASTP
 WHERE t.expect < 0.000001 AND g.refseq_id = t.t_seq_id
/
```

II. ODM BLAST Query Result Table

NP_000250	593	1.8956E-62	Class 5A myosin; motor protein that may be involved in organelle transport	MYO5A	15Q21

Figure 1: Oracle 10*g* BLAST Query and Result.

protein sequences are stored in the relational table *query_db*, and if *Block A* in the example above is replaced with the following SQL sub-query:

(**SELECT** sequence_string
FROM query_db
WHERE sequence_id = 100
).

3 Conclusions

ODM BLAST enables BLAST queries to be performed in Oracle Database 10*g*. ODM BLAST removes the overhead of moving sequence data out of the RDBMS, relieves the need to parse data files, and enables BLAST results to be integrated with existing relational data. Challenging queries become tractable with ODM BLAST. Batch BLAST queries can now be easily performed and ODM BLAST can span diverse data sets and incorporate annotations stored in relational databases. As large-scale integrative biology gains popularity, we expect ODM BLAST to become an essential database toolkit for bioinformaticians with challenging integrated sequence homology oriented queries.

References

[AGM+90] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[CC03] Jake Yue Chen and John V. Carlis. Similar_Join: Extending DBMS with a Bio-specific Operator. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 109–114, 2003.

[HSK+01] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice, and W. C. Swope. DiscoveryLink: A System for Integrated Access to Life Sciences Data Sources. *IBM Systems Journal*, 40:489–511, 2001.

[Ken02] W. James Kent. BLAT: The BLAST-like Alignment Tool. *Genome Research*, 12(4):656–664, 2002.

[ora04] Oracle Data Mining Application Developer's Guide 10g Release 1. http://otn.oracle.com/industries/life_sciences/index.html, 2004.

[Pea00] W. R. Pearson. Flexible Sequence Similarity Searching with the FASTA3 Program Package. *Methods Mol. Biol.*, pages 185–219, 2000.

[PM01] K. D. Pruitt and D. R. Maglott. RefSeq and LocusLink: NCBI Gene-centered Resources. *Nucleic Acids Res.*, 29:137–140, 2001.

[Sim99] P. Simakov. Sequence Server Samurai. *Science*, 285:1226–1227, 1999.