MoBIoS: A Metric-Space DBMS to Support Biological Discovery[1]

Daniel Miranker, Weijia Xu, Rui Mao

2/8/03

{miranker,xwj, rmao}@cs.utexas.edu
Department of Computer Sciences
University Of Texas at Austin
Austin, Texas 78712

Abstract

MoBIoS is a specialized database management system whose storage manager is based on metric-space indexing, and whose query language entails biological data types. When relational database management systems are used to support biological data, important data types are relegated to blob and unstructured text fields. Consequently, even simple, but critical queries are executed by sequentially dumping the data to utilities outside the database.

Metric-space indexing exploits the intrinsic clustering of a dataset without regard to a mapping of the data to a coordinate system. It is clear from an abundance of bioinformatic discoveries that biological data is not random and exhibits interesting structure with respect to clustering. Just as Geographic Information Systems have been enabled by spatial databases, we argue that Biological Information Systems will be enabled by metric-space databases. We show that both biological sequence data and mass-spectrometry protein signatures can be effectively managed in metric-space. Further, concomitant object-relational extensions to SQL allow concise declarative expression of complex proteomic algorithms.

1. Introduction

Important biological data types cannot be effectively mapped to low dimensional coordinate systems on which O(log n) indexing methods rely. This includes biological sequences, mass spectra, protein and ligand structures, phylogenetic and pathway graphs, to name a few. When RDBMSs are used to manage biological data, first class data is often stored in blob and unstructured text fields. Annotations of the data, such as organism name and protein family name, may exist as part of the record. Relational tests against the annotations may be used to filter the database. Ultimately, mining of the biological data types is accomplished by utilities outside the database (e.g. BLAST). Even when object-relational or semi-structured representations are used (e.g. XML and ASN.1), there are few results in building persistent access paths capable of supporting fast retrieval methods [Dub00]. These systems cannot be expected to scale, neither in performance as the volume of data increase, nor in ongoing software development costs as functionality is increased.

The MoBIoS architecture (Molecular Biological Information System) embodies metric-space indexing and object-relational models of complex biological data types [MMS02]. Biological data is far from random. Any review of the literature will reveal that clustering is a primary method underlying bioinformatic discovery. Thus, biological data is not random and very likely exhibits the intrinsic structure necessary for metric-space indexing to succeed [Bri95].

[1] Research funded in part by the Texas Higher Education Coordinating Board

Definition 1: A *metric space* is a set of objects, S, with a (*metric) distance function*, d, such that, given any three objects, x, y, z,

(i) $d(x, y) = 0$ and $d(x, y) = 0$ iff $x = y$. (*Positivity*)

(ii) $d(x, y) = d(y, x)$. (*Symmetry*)

(iii) $d(x, y) + d(y, z) = d(x, z)$. (*Triangle Inequality*)

A primary challenge of the approach is that established biological models of similarity do not form metrics. Most biological similarity functions reward more similar features with greater positive numbers. Metrics require the distance of more similar objects to be closer to zero. The similarity models are often derived from probabilistic methods that are difficult to algebraically transform into metrics. In the case of sequence homology, the biologically interesting query fails to form a metric under any distance model.

We explain the progress we have made in mapping both protein sequence data and proteomic mass-spectrometer data to metric-space models. We show, through an example, that solutions to these problems coupled with extending object-relational database architecture to support similarity queries in metric-space enables concise declarative expression of an otherwise complex bioinformatic algorithm.

2. Sequence Homology

Many papers on metric-space search state categorically that their results are applicable to the comparison of biological sequences [Bri95, CPZ97, CNB01]. But it is only indirectly true. The most important type of biological sequence comparison, ***local sequence alignment*** (also known as the Smith-Waterman alignment [Gus97, SmW81]) does not form a metric.

2.1 Local Alignment

Edit distance as it commonly appears in the literature on similarity search in metric spaces is known in biology as ***global sequence alignment*** (or Needleman-Wunsch alignment [NeW70, Gus97]); that is a single score for the entire query pattern. However biologists need to know if short interspersed (gapped) substrings from the query string match substrings in the database. This problem is known as local sequence alignment. A local alignment query asks, given a query sequence, S, a database of sequences, T and a similarity matrix corresponding to an evolutionary model, return all subsequences of T that are sufficiently similar to a subsequence of S. The similarity matrix provides weights for individual substitutions between any nucleotides pair or amino acids pair. We further note that a subseqence, s, of a sequence S, is any ordered subset of the elements of S. Thus, there is an intrinsic exponential in the size of a local alignment problem.

Briefly stated, a metric-space index cannot be used directly for local alignment for two reasons:

1. Local alignment produces a set of answers. A metric distance function must return a single value for each pair of arguments.
2. The important evolutionary models PAM and BLOSUM are formulated as log-odds matrices. These models violate the metric properties if used to weight an edit distance, even for a global alignment. Efforts that use non-evolutionary similarity metrics have met with algorithmic success but are of only limited applicability [GWW02, Ken02].

2.2 Computing Local Alignment from Global Alignment

One of Myers' foundational papers in bioinformatics analyzes an algorithm that addresses problem 1 above [Mye94]. In the following, we abstract that algorithm into a general framework for computing local alignments from global alignments. Let the database comprise a sequence, A, of length N. Let the query be a string, W, of length P, and a similarity threshold D.

Off-line:

1) Divide a sequence database into small substrings of fixed length, T.
2) Build an index to support constant-time, O(1), look-up of exact matches.

On-line query:

1) Divide the query string W into overlapping substrings, W_i, of length T, $i = 0..\left\lceil \frac{W}{T} \right\rceil$. (See [Mye94, GWW02] for details of the boundary conditions.)
2) For each W_i, generate all strings that fall within the similarity threshold D. Use the index to determine if each generated string is in the database. If not, discard it.
3) Chain the valid strings together to form solutions to the full query.

The three on-line steps are not each executed to completion in sequence. Rather, since the total discrepancy in similarity cannot exceed D, Myers details how these steps may be integrated into a search procedure that, as it descends the search tree, considering W_i, at level i, reduces the threshold D based on the accumulated discrepancy to that point.

Those already familiar with the BLAST program for sequence search may recognize its similarity with this framework. In lieu of building a static index, BLAST builds an index of exact matches with the query string at query time, and then uses chaining to extend the hot spots. Nevertheless, the performance analysis of Myer's algorithm is often seen as an explanation of the performance of BLAST.

Our approach is, in lieu of building, off-line, an index for exact matches, we divide the database into substrings of length T, and build a metric space index based on global alignment. In the on-line phase of the algorithm, we replace the "generate and test" process of step 2 with a direct O(log n) retrieval from the index of all the sequences in the database that are within similarity D.

2.3 A Metric Accepted Point Mutation Model

"*[Nothing in biology makes sense except in the light of evolution*" Dobzhansky (1963) [Dob63]

Contrary to what one might expect from high school biology or the popular press, changes in a cell's DNA are very common. In humans, there are changes virtually every time a cell duplicates. A marvel of life is that we live in the face of this dynamic string. The resulting computational challenge is that when comparing two biological sequences the significance of each individual substitution of an amino acid or nucleotide must be carefully weighted. Some substitutions have little biochemical impact and thus are common. Other substitutions, if made in sensitive places, may assure the death of the cell.

The only generally accepted weighting schemes, the PAM and BLOSUM matrices, are similarity matrices based on log-odds probabilities. Log-odds scores are often negative values that can yield negative global alignment scores. This violates positivity. Methods used to directly scale or normalize the PAM matrices still result in the violation of the triangle inequality.

There have been previous efforts to exploit metric-space indexing for biological sequence look-up. Giladi et.al. reported scalable speeds starting at 25 times the speed of BLAST on a system using Hamming distance[GWW02]. Aberer and Chen suggested a bounding relationship between global alignment and local alignment and the use of M-trees [ChA97]. Kent developed an exact matching approach based on hashing n-gram representations [Ken02]. In each case, biologically meaningful results were limited to applications to sequence assembly or queries between very close sequences (measured in evolutionary terms). A more recent paper shows effective look-up times in main-memory for string-compression distance [SMT03]. The approach has a biological basis, but still faces validation.

We have reworked the mathematics of the accepted-point mutation model using Dayhoff et.al.'s published data used to derive the PAM250 matrix and used it to derive a matrix with metric distance properties, mPAM250 [MMS02[2],DSO78]. An intuitive explanation of our approach is very simple. The original PAM calculations were concerned with ascertaining, from experimental data, the frequency in which individual pairs of amino acids substituted for each other. From the frequency we used standard queuing theory models to compute an expected time between substitutions. Events with relatively high frequency are expected to occur in relatively shorter time.

Using a benchmark query set developed and distributed by Altschul, we compared, for accuracy, the answer sets of mPAM250 using a Smith-Waterman alignment against BLASTs answer sets using the PAM250 matrix [AAS01]. We have not yet implemented the chaining component of algorithm. Given a correct selection of boundary conditions a chaining algorithm can meet the accuracy of the Smith-Waterman algorithm, (although execution time may change with different boundary conditions). For each of the 103 queries in the benchmark we computed ROC_{50} scores for each implementation. We compared the scores by subtracting our score from corresponding score computed using the Smith-Waterman algorithm. Figure 1 is a histogram showing the number of queries for each value. The histogram reveals competitive performance.

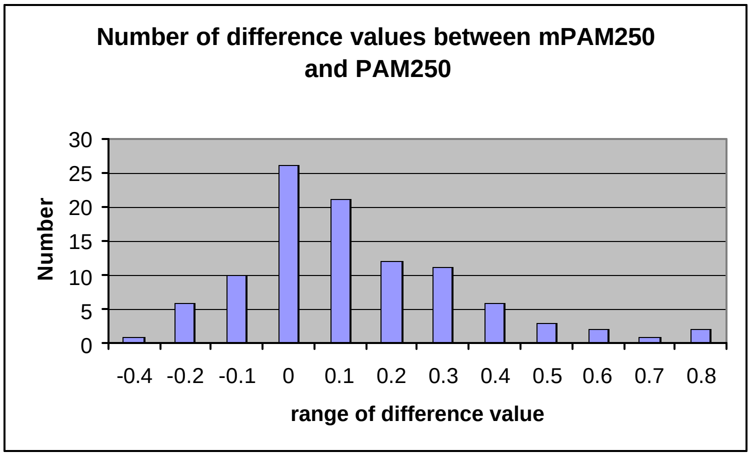


Figure 1 The difference between ROC_{50} using mPAMS and PAM250 The label of x scale shows the range of difference. For example, the bar of 0.2 shows the number of queries on which the PAM250 has better ROC_{50} than mPAMS within 0.1~0.2. The negative label means mPAMS has better performance than PAM250. The bar of 0 shows that there are 26 queries that mPAMS and PAM250 has same ROC_{50} value.

3. Spectral Look-Up in Metric-Space

A key data type in proteomics is spectra. Detailed below, protein identification experiments comprise a similarity search between a database of computed spectra, and the measured spectrum of an unknown protein produced by a mass-spectrometer. Every mass-spectrometer has an operating range and resolution. Thus, the number and location of measurable peaks is quantized, and each possible peak can be mapped to a dimension in the vector space. In this way, spectra can be expressed as binary valued high dimensional vectors, and typical measures of spectral similarity can be computed on these vectors. For example, the inner product between two such

[2] The matrix values published in the poster abstract are different than the matrix used in the validation results reported here. A full report concerning this other derivation is in preperation. New matrix values are available upon request.

vectors computes the shared peaks count, a popular measure of similarity in proteomics [PMD01].

We have tested whether spectra would cluster in a manner that would benefit from metric space indexing. We used Protein Prospector [CBB99] to assemble a database of computed mass-spectrometer spectra for the yeast proteome following trypsin digestion**.** We then computed the cosine distance between all pairs of spectra, of which the corresponding angle serves as metric approximation of shared-peak count. The results, plotted in Figure 2, indicate that the spectra are clearly not uniformly distributed in the space of spectra and show several local maximal much closer to zero degrees than expected by chance, evidence of clustering [Bri95]. These results indicate that mass spectra can be effectively organized in a metric space model.

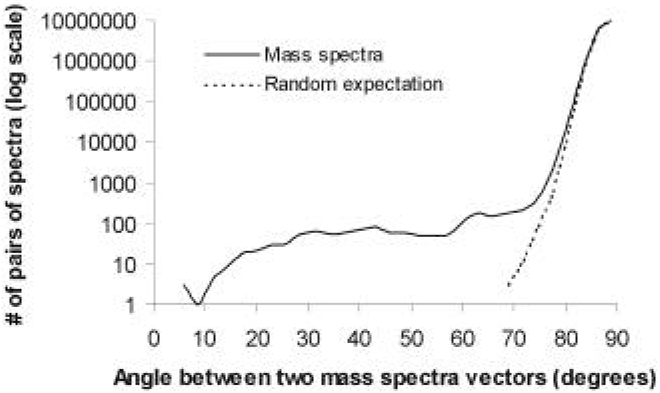
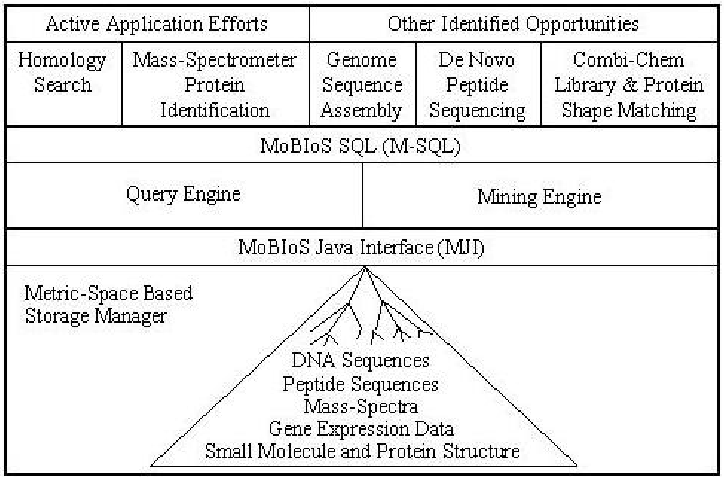


Figure 2. Mass spectra of peptides from the known yeast proteins show considerably more clustering than expected by chance, indicating that spectra can be effectively organized in a metric-space model.



Active Application Efforts
Other Identified Opportunities
Homology Search
Mass-Spectrometer Protein Identification
Genome Sequence Assembly
De Novo Peptide Sequencing
Combi-Chem Library & Protein Shape Matching
MoBIoS SQL (M-SQL)
Query Engine
Mining Engine
MoBIoS Java Interface (MJI)
Metric-Space Based Storage Manager
DNA Sequences
Peptide Sequences
Mass-Spectra
Gene Expression Data
Small Molecule and Protein Structure

Figure 3. Architecture of the MoBIoS Platform

4 The MoBIoS Architecture

4.1 The MoBIoS Storage Manager

Figure 3 presents a schematic of the MoBIoS platform. The system is layered in a manner prototypical of all database management systems. If an existing object-relational database allows the addition of metric-space indexes everything discussed in this paper could be built in that platform. This is still an open research question for this project. Although the problem of conducting nearest neighbor and range queries in metric-spaces has received considerable attention, inexplicably, among the metric-space studies, Ciaccia *et al.*'s M-tree effort stands out as the *single* investigation of metric-space partitioning techniques into an external index structure [CNB01,CPZ97].

> ***"among all metric index structures the M-tree is the only one which is optimized for large secondary-memory based data sets. All others are main memory index structures supporting rather small data sets."[BBK01]***

Although the M-tree implementation was based on GiST [HNP95], examination of the source-code distribution reveals a number of C related pointer tricks that violate encapsulation of the reusable components.

In MoBIoS, we use a general hyper-plane index structure, a similar structure of which is explored in M-tree. Our structure differs with that of M-tree's in two aspects. One aspect is that we designed a new initialization algorithm to build the index from a given dataset, and the other aspect is that we optimize the structure so that the covering radii in index nodes are reduced, which leads to better query performance.

Experimentally, we compare M-tree and our structure by running queries on indices built from Yeast gene sequence segments [MXS03]. Figure 4 shows the average number of distance computations of the queries. From the figure, first we can see that the numbers of distance computations of M-tree queries are much greater than those of MoBIoS index. Especially, for radius 0 (exact matches), M-tree's number is three times of that of MoBIoS index. For radius 10, M-tree's number is two times that of MoBIoS index. We can also see that as the radius increases, the two numbers become closer. Given the curse of dimensionality we would expect that increasing the radius would quickly induce the entire tree to be searched, independent of the initialization. I/O number vs. radius is shown in Figure 5. Similar conclusion can be drawn as from Figure 4. From all the experimental results shown above, we can conclude that the MoBIoS index owns better performance than M-tree.

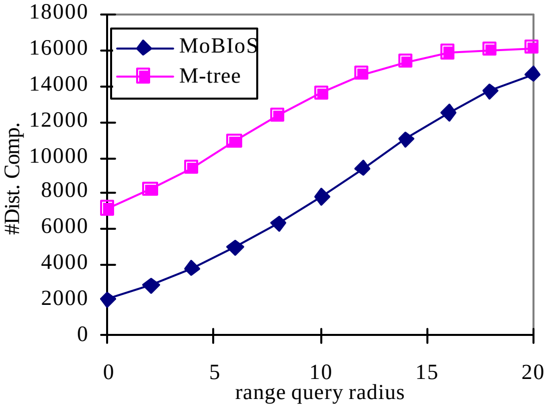


Figure 4. Number of distance computations vs. range query radius

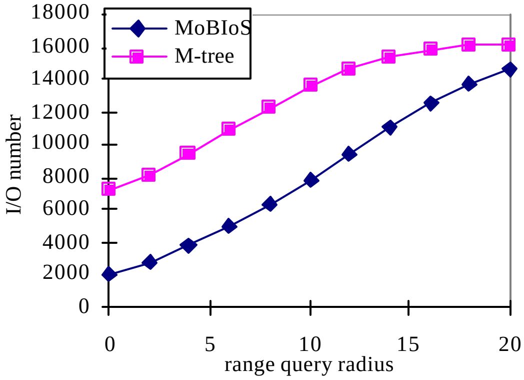


Figure 5. I/O number vs. range query radius

4.2 MoBIoS SQL

MoBIoS SQL (M-SQL) is an SQL-like language under development for MoBIoS. Syntactically M-SQL is consistent with the SQL3 object-relational extensions to SQL.

M-SQL contains built-in abstract data types for DNA, RNA and peptide (amino acid) sequences. The semantics of these data types include subsequence operators and the concept of a local alignment. Mass-spectrum is a built-in data type. A Peak in mass-spectra represents the mass-charge ratio of an individual peptide. A protein is a particular kind of peptide and mass-spectrometers are used to measure the collection of peptides created by fragmenting proteins. The abstract data types preserve these semantics.

The other semantic element for these abstract data types is a concept of similarity between instances of the data types. More precisely, these are, of course, metric-distances. Built into M-SQL is a library of biological distance measures, e.g. mPAM250 and cosine distance, as well as common distance metrics such as the L_k norms.

Where M-SQL may depart from standard object-relational SQL extensions is in the specification of indexes. To specify that the data be stored and accessed per a hierarchical clustering we introduce a reserved word, ‘metrickey” as an argument to create-table. If an index is to be built on a metric, the distance function must also be specified. We introduce the reserve work "metric" as an argument to create index. Figure 6, illustrates a number of create table and index statements that begin a running example.

Secondary indexes are supported, as are user-defined metrics. Users will be provided a standard reflective Java interface so that they may write their own distance functions [Blo01]. The process will be similar to writing stored procedures for client-server databases.

```
Create table protein_sequences (accesion_id int, sequence peptide,
   primary metrickey(sequence, mPAM250);

Create table digested_sequences(accession_id int, fragment peptide,
    enzyme varchar, ms_peak int, primary key(enzyme, accession_id);

Create index fragment_sequence on digested_sequences (fragment)
    metric(mPAM250);

Create table mass_spectra(accession_id int, enzyme varchar,
  spectrum spectrum, primary metrickey(spectrum, cosine_distance);
```

Figure 6: Examples of the M-SQL Data Definition Language Illustrating the Specification of Metric-Space Storage and Secondary Indexes

5 Application of MoBIoS to MS/MS Protein Identification

Protein identification is a central problem in proteomics. Now that entire genomes have been sequenced, the following methodology is often used. An unknown protein is purified and exposed to a protease enzyme that cuts the protein into fragments at a particular sequence of amino acids. By virtue of knowing an entire genome, biologists have deduced representative amino acid sequences of each protein in the corresponding organism. Since the chemical behavior of the protease enzyme is known, it can further be determined how each protein is digested into fragments and the spectrum that that protein would exhibit in a mass-spectrometer computed. The unknown protein can be identified by feeding it into a mass-spectrometer and comparing the output of the mass-spectrometer to the database of computed spectra and finding the closest match.

However, this single analysis is rarely specific enough in the face of mutations and measurement error to uniquely identify the protein. In a tandem mass-spectrometer, (MS/MS), a technician may select a peptide, called a parent peptide or parent peak, for further analysis. The parent peptide goes through a further reaction and data from a second mass-spectrometer can be used to determine the parent's amino acid sequence. This additional information can disambiguate results from the first stage.

Let the experimental inputs to the query be, identification of the enzyme, E, the spectrum produced by the first mass-spectrometer, S, the parent peak, P, and the sequence data for the parent peak, PS, derived from the output of the second mass-spectrometer.

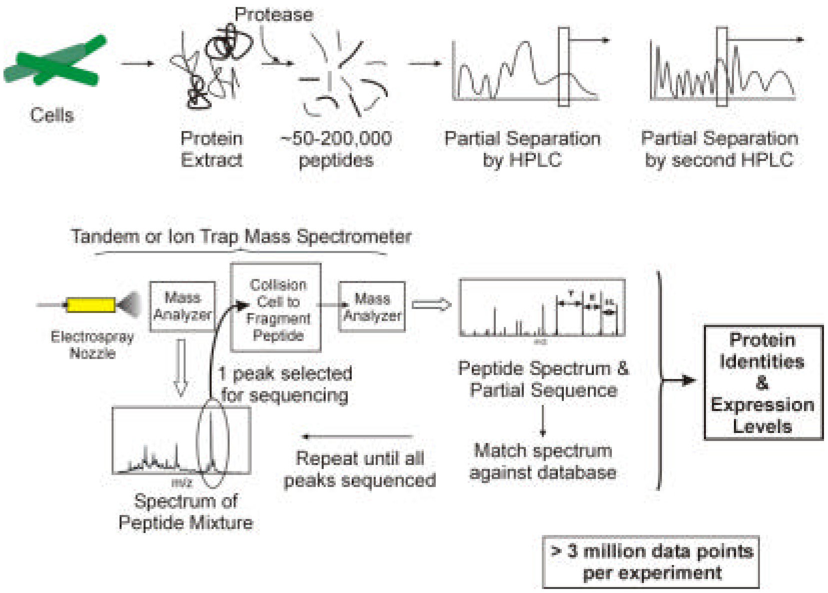

Protease
Cells
Protein Extract
~50-200,000 peptides
Partial Separation by HPLC
Partial Separation by second HPLC
Tandem or Ion Trap Mass Spectrometer
Electrospray Nozzle
Mass Analyzer
Collision Cell to Fragment Peptide
Mass Analyzer
1 peak selected for sequencing
m/z
Spectrum of Peptide Mixture
Repeat until all peaks sequenced
Peptide Spectrum & Partial Sequence
Match spectrum against database
Protein Identities & Expression Levels
> 3 million data points per experiment

Figure 7. Protein Identification through Proteolytic Digestion. In this approach to *proteomics*, a protein is broken into many fragments, called peptides, using a protease enzyme. The protein can then be identified by using a mass-spectrometer to measure the mass-charge ratio of the fragments and comparing the experimental result to a database of precomputed spectra (adapted from [Mar01]).

Protein identification in this laboratory set-up comprises analyzing both spectral and sequence data. Complete specifications of algorithms that solve this problem are usually very large. Figure 8 illustrates how, given a suitably defined MoBIoS database, Figure 6, protein identification can be expressed as a simple M-SQL query. Logically the query is simply returning the intersection of the data sets that fulfill each of the laboratory inputs per the specified distance tolerances, range1 and range2. The query is possible since both peptide and spectrum are built in abstract data types. Query evaluation is made fast by having specified in the physical schema that the digested sequence table is clustered on disk based on enzyme selection, and has a secondary index based on sequence similarity of the fragments. The mass spectra are clustered on disk per their mutual cosine distances.

```
SELECT  Prot.accesion_id, Prot.sequence
FROM  protein_sequences Prot, digested_sequences DS,mass_spectra MS
WHERE
      MS.enzyme  = DS.enzyme = E and
      Cosine_Distance(S, MS.spectrum, range1) and
      DS.accession_id = MS.accession_id = Prot.accesion_id and
      DS.ms_peak = P and
      MPAM250(PS, DS.sequence, range2)
```

Figure 8: An M-SQL Query Implementing MS/MS Protein Identification

6 Discussion

It is easy to mistake a large biological-data web site as a database management system (DBMS) or to anticipate that the software underlying a biological web site is a DBMS. A DBMS is a body of software that provides an integrated set of services on a collection of data, a database. Many popular biology web sites provide a number of services and their web-masters are usually careful to provide a common look-and-feel to each web page. Thus, an end user witnesses an integrated system. The reality is that each service is provided by an ad-hoc collection of scripted utilities and the underlying administration of the system rarely embodies generalized data management facilities. These systems cannot be expected to scale, neither in performance as the volume of data increase, nor in ongoing software development costs as functionality is increased.

One might argue that the proteomics algorithm illustrated by the query in Figure 8 is a deception; that much of the real work is embedded in look-up as a function of a metric distance property. On the contrary, this is one of MoBIoS' greatest advantages. The built in distance functions and abstract data types will enhance ability of biologists to rapidly generate bioinformatics algorithms and protocols.

We expect secondary indexes based on user-defined metrics to be one of most useful parts of the system. Such indexes will transform MoBIoS from a platform for fast retrieval of biological data, into warehouses for discovery. Users will be able to hypothesize new relationships among warehoused data and to test their hypotheses simply by writing a new distance functions and declaring a new secondary index. In effect, hierarchical clustering, a very important tool for bioinformatic discovery, is a built-in primitive. If a hypothesis proves accurate, the index may remain in the database. Thus, secondary indexes may represent a persistent materialization of new concepts. Since databases normally hide the details of access paths, an open research issue is how to reveal the structure of the index to the user and allow him further associations with data and other indexes.

It is widely understood that the growth of biological data demands that O(log n) indexing structures be developed in order to attain scalable performance of biological databases. Metric-space indexing techniques are often cited as a method to achieve that. In MoBIoS we are integrating metric-space indexing methods and biological data types with traditional database management systems. The result will be a body of software offering an integrated set of services for the management of biological data; a biological information management system.

Reference:

[AAS01] S. Altschul, Alejandro A. Schaffer, L. etal. (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nulceic Acids Research, 2001 vol 29 no14. 2994-3005

[BBK01] Böhm, C., Berchtold, S., Keim, D. A. 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys 33(3):322-373.

[Blo01] Bloch, J. Effective Java, Programming Language Guide. Addison Wesley, 2001

[Bri95] Brin, S. 1995. Near Neighbor Search in Large Metric Spaces. In Proc. 21st. Int. Conf. Very Large Data Bases (VLDB), pp. 574-584.

[CBB99] Clauser K. R., Baker P. R. and Burlingame A. L., Role of accurate mass measurement (+/- 10 ppm) in protein identification strategies employing MS or MS/MS and database searching. Analytical Chemistry, Vol. 71, 14, 2871- (1999)

[ChA97] Chen, W., Aberer, K. 1997. Efficient Querying on Genomic Databases. In Proc. of 8th Int. Work on Database and Expert System Applications.

[CNB01] Chavez, E., Navarro, G., Baeza-Yates, R., Marroquin, J. L. 2001. Searching in metric spaces. ACM Computing Surveys 33(3): 273-321.

[CPZ97] Ciaccia, P., Patella, M., Zezula, P. 1997. M-tree: an efficient access method for similarity search in metric spaces. Proc. 23rd Int. Conf. Very Large Databases (VLDB).

[Dob63] Dobzhansky, T. 1963. Evolution, genetics and man. New York: John Wiley.

[DSO78] Dayhoff, M. O., Schwartz, R., Orcutt, B. C. 1978. Atlas of protein sequence and structure 5, Suppl. 3, ed. M. O. Dayhoff.

[Dub00] Olivier Dubuisson, translated by Philippe Fouquart. ASN.1 - Communication between heterogeneous systems. Morgan Kaufmann Publishers, September 2000.

[GrF98] Grossman, D. A., Frieder, O. 1998. Information Retrieval. Kluwer Academic Publishers.

[Gus97] Gusfield, D. 1997. Algorithms on Strings, Trees, and Sequences. Cambridge University Press.

[GWW02] Giladi, E., Walker, M. G., Wang, J. Z., Volkmuth, W. 2002. SST: An algorithm for finding near-exact sequence matches in time proportional to the logarithm of the database size. Bioinformatics 18 (6-7): 873-879.

[HNP95] J. M. Hellerstein, J. F. Naughton and A. Pfeffer. Generalized Search Trees for Database Systems. Proc. 21st Int'l Conf. on Very Large Data Bases, Zürich, September 1995, 562-573.

[Ken02] Kent, W. J. 2002. BLAT - the BLAST-like alignment tool. Genome Research 12: 656-664.

[Mar01] Marcotte EM. Measuring the dynamics of the proteome. *Genome Res*. 2001 Feb;11(2):191-3.

[MMS02] R. Mao, D. P. Miranker, J N. Sarvela and W. Xu. Clustering Sequences in a Metric Space-the MoBIoS project. Poster of the 10th International Conference on Intelligent Systems for Molecular Biology, August 3-7, 2002, Edmonton, Canada.

[MXS03] R. Mao, W. Xu, N. Singh and D. P. Miranker. An Assessment of a Metric Space Database Index to Support Sequence Homology. In the proceeding of the 3rd IEEE Symposium on Bioinformatics and Bioengineering, March 10-12, 2003, Washington D.C. (to appear)

[Mye94] Myers, E. 1994. A sublinear algorithm for approximate keyword searching. Algorithmica. 12(4/5): 345-374

[NeW70] Needleman S.B., Wunsch C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970 Mar;48(3):443-53.

[PDT00] Pevzner, P.A., Dancík, V., Tang, C. L. 2000. Mutation-Tolerant Protein Identification by Mass Spectrometry. Journal of Computational Biology 7(6): 777-787.

[PMD01] Pevzner P.A., Mulyukov Z, Dancik V, Tang CL. Efficiency of database search for identification of mutated and modified proteins via mass spectrometry. *Genome Res*. 2001 Feb;11(2):290-9

[SMT03] C. Sahinalp, S.J. Macker, M. Tasan, M. Ozsovoglu, Distance Based Indexing for String Proximity Search, toappear ICDE 2003

[SmW81] Smith T.F., Waterman M.S. Identification of common molecular subsequences. J. Mol. Biol. 1981 Mar 25;147(1):195-7.