# Indexing Protein Sequences in Metric Space

Weijia Xu[1], Daniel P. Miranker[1], Rui Mao[1], Shu Wang[2]
[1] Department of Computer Sciences
{xwj, miranker, rmao }@cs.utexas.edu
[2] Department of Electrical and Computer Engineering
swang5@ece.utexas.edu
University of Texas at Austin
Oct 31. 03

## Abstract

The hyper-exponential growth of biological sequence data and complex queries demand new approaches of managing sequence databases where sequence data is preprocessed off-line and organized in data structures such that on-line queries can be executed quickly. Due to the complications of computing biological similarity based on local alignments, such index structures are typically constructed on q-grams (substrings of length q) and embody a three-way trade-off between speed, accuracy and scalability.

The development of a biologically effective distance metric on amino-acid substitution, mPAM, permits this approach to be extended beyond direct nucleotide comparison to codon similarity and protein sequences. We consider the storage and retrieval of protein q-grams using a metric-space index structure, the MVP-tree. Using a controlled sequence homology benchmark, we evaluate the trade-off between sensitivity and selectivity as a function of speed and length of the q-grams. We conclude that the system only slightly penetrates the curse of dimensionality and can be expected to offer scalable performance. We will discuss our experimental results to show that the protein sequences can be indexed in metric space with accuracy and scalability.

# 1. Introduction

The literature on biological sequence analysis speaks to two kinds of alignments, global alignment and local alignment, but only local alignment is of practical interest to biologists. A further challenge is that in most applications biologically interesting results must entail a model of sequence evolution. Distance models based on simple edit distance and Hamming distance form metric distance functions (*metrics*) among global alignments but are not effective evolutionary models. BLAST variants form a defacto-standard for computing local alignments. However, a BLAST search comprises a linear scan of a database of sequences. These issues, together with the growing volume of data, are the driving forces in the development of new methods of biosequence database management that comprise initializing a database of biosequences, off-line, in order to speed up the execution of on-line queries. This problem forms a rich research environment to investigate, with trade-offs between accuracy, scalability and speed.

**Definition 1 Global Alignment Problem:** Given an alphabet, $\mathcal{A}$, and a similarity substitution matrix, $\mathcal{M}$, corresponding to an evolutionary model, the global alignment problem for two sequences $s=\{s_1s_2s_3\ldots s_m|s_i\in\mathcal{A}\}$, $t=\{t_1t_2t_3\ldots t_n|t_i\in\mathcal{A}\}$ is to find strings $\alpha$, $\beta$, which are obtained from s and t respectively by inserting spaces either into or at the ends of s and t, and whose score computed using $\mathcal{M}$ is at a maximum (for similarity measure) or minimum (for distance measure) over all pairs of such strings obtained from s and t.

**Definition 2 Local Alignment Problem:** Given an alphabet $\mathcal{A}$ with a similarity substitution matrix $\mathcal{M}$, corresponding to an evolutionary model, the local alignment problem for two sequences $s = \{s_1s_2s_3\ldots s_m|s_i\in\mathcal{A}\}$, $t = \{t_1t_2t_3\ldots t_n|t_i\in\mathcal{A}\}$ is to find substrings $\alpha$ and $\beta$ of s and t, respectively, whose similarity (optimal global alignment) value is maximum over all pairs of substrings from s and t. (Gusfield, 1997)

In this work we investigate the use of metric-space index methods to accelerate protein-sequence retrieval. Since the similarity of nucleotide sequences is often determined by considering the similarity of putative proteins encoded by such genetic sequences, the results are important for genomic databases as well.

**Definition 3 Metric Space** Metric space is a set of objects with a binary distance function, d, satisfying the following conditions for every three objects x, y & z (Chavez et al., 2001):

i. $d(x,y) \geq 0$ and $d(x,y) = 0$ iff $x = y$; (Positivity)
ii. $d(x,y) = d(y,x)$; (Symmetry)
iii. $d(x,z) + d(y,z) \geq d(x,y)$. (Triangle Inequality)

The merit of metric-space indexing is that the triangle inequality may be leveraged to safely rule out the similarity of the query object with a large number of objects by computing a single distance between a query object and a well chosen data object. Metric-space indexing exploits the intrinsic clustering of a dataset and can prune a search space without regard to a mapping of the data to a coordinate system (Chavez et al., 2001). It is clear from an abundance of bioinformatics discoveries that biological data is not random and exhibits interesting structure with respect to clustering, a necessary properties for metric space indexing (Linial et al., 1999; Brin 1994).

A primary challenge of this approach is that established biological models of similarity, PAM and BLOSSUM, do not form metrics (Dayhoff et al., 1978; Henikoff and

Henikoff, 1992). Most biological similarity models are derived from probabilistic methods that reward more similar features with greater positive numbers and are difficult to algebraically transform into metrics. Even given a metric for global-alignment, local alignment still does not form metric. Consider the optimal local alignment among three sequences, R, S and T. The ordered set of subsequences, representing an optimal local alignment of sequences S and T could be completely disjoint from those for R and T. If so, we cannot make any statement concerning the relatedness of R and S without comparing R with S.

One approach for finding useful optimal local alignments, likely inspired by BLAST, is to divide the sequences in a sequence database into substrings of length q, called q-grams, and similarly divide the query sequence into q-grams. A global alignment is resolved among the q-grams. Those results were then used to determine a local alignment. In BLAST, matching q-grams (hot-spots) between the query and database are found through a sequential scan of the database. For sequences with a sufficient number of matching q-grams, those q-grams can be chained together to form a complete local alignment (Gusfield, 1997). By dividing the database into q-grams, which BLAST does not do, the database may be structured using an index.

Some systems, such as the SST and BLAT systems, have made progress by structuring nucleotide databases in this manner (Giladi et. al., 2002; Kent, 2002). However, these approaches use Hamming distance and simple edit distance, respectively. Initial success was achieved by targeting the sequence assembly problem where evolutionary criteria are unimportant. Subsequently these systems are being effectively applied to genomic analysis problems whose data is limited to sequences from evolutionarily close organisms (Rouchka et al., 2002).

In our previous work, we revisited the mathematics used to derive the PAM family of amino acid substitution matrices (Point accepted mutation model). Starting from the original raw data, in-lieu of computing the frequency of substitutions, we computed the expected time between substitutions. The resulting weight matrix, mPAM, forms an evolutionary metric on the amino acid alphabet. Using the Smith-Waterman algorithm for computing local-alignments, we validated that the mPAM matrix produces biologically effective results (Xu and Miranker, 2003).

It follows from Sellers theorem, that if the mPAM weighting/substitution matrix is used to compute the global alignment of amino acid sequences, then the global alignment forms a metric (Sellers 1974). Consequently, amino-acid q-grams may now be organized, off-line, in metric-space index structures such that evolutionary criteria can be used to quickly determine q-gram matches. Following we describe the construction and performance of a tree based index structure, the MVP-tree (Bozkaya and Ozsoyglu, 1997), for matching q-grams of protein sequences using global alignment as a distance function parameterized by mPAM. This improves on similar work where only exact or near exact matching q-grams can be found. The index supports range searches returning all q-grams in a neighborhood of radius r centered on the query q-gram. The q-grams returned from range queries can be used to compute local alignments. The trade-off among speed, selectivity and sensitivity is directly affected by the length of the q-gram, q, and searching radius, r. By sensitivity, we mean the ability to identify all true positive hits on a query sequence. By selectivity, we mean the ability to filter out false positive hits during initial q-grams searching. Low selectivity will directly increase the workload for a

successive chaining algorithm, while low sensitivity will decrease the accuracy of results. In general, longer q-grams increase selectivity but imply larger radius searches. Larger radius searches increase sensitivity but decrease selectivity. We have conducted a set of experiments for empirical analysis to determine optimal parameter selection. Our results show that the protein sequences can be effectively indexed in metric space with comparable accuracy to basic BLASTp and scalable performance.

## 2. **Related Work**

Recently several efforts have focused on building scalable offline sequence database index structures to support faster online search. The ED-tree, an index structure designed for homology searches on a DNA sequence database, shows 6 times the speed of BLASTn (Tan et al., 2003). The SST algorithm partitioned each sequence into overlapping q-grams and mapped those q-grams to a vector space. Similarity was measured as the Hamming distance between vectors. A tree-structure index was built by k-means clustering and vector quantization to achieve O(nlogm) scalability, where n is the length of the query sequence and m is the length or size of the database (Giladi et.al., 2002). Giladi et. al. report an SST execution speed 27 times the execution speed of BLAST2 for sequence assembly on databases of 120,000 nucleotides and estimate a 200 fold speed-up over BLAST2 on mega-base-pair databases. BLAT claimed a speed 40 times faster than WU_BLASTX, based on simple edit distance supported by hashing to achieve O(n) scalability using O(m) memory (Kent, 2002).

The primary goal of all of these approaches is to improve the speed of search for matching q-grams. Whether stated directly, as in the ED-tree, or indirectly, as in SST and BLAT, these methods can also be used in heuristic algorithms to deduce a local alignment from the returned q-grams. Tree based index structures can offer a scalable, O(log(N)) performance for q-gram searching. The popular homology sequence search tool, BLAST, begins with a linear scan of the database, O(N), to construct the *hot-spot* index of exact matches, followed by a heuristic extending algorithm to approximate the local alignment result. The expected time complexity of BLAST comprises three parts: aW for generating W q-grams within neighborhood of the q-gram in query, bN for scanning database with N residues for exactly matches, and $cNW/20^{w}$ for extending hits. (Altschul et al., 1990).

Although SST and BLAT can be applied to protein sequences, the actual feasibility is severely restricted. First, the alphabet size of peptides is much larger than the alphabet size of nucleotide sequences. Second, both algorithms determine exact or near-exact fragment matches. Finally, important popular evolutionary models for peptides, PAM and BLOSUM, are formulated as log-odds matrices, which violate all of the metric properties.

The doubling time of the sequence content of GenBank has shrunk from 18 months to 15 months and its rate of growth continues to accelerate (Benson et al., 2002). Hence, the volume of biological sequence data is growing faster than Moore's law (Patterson and Hennessy, 1996) and it has now reached a rate of growth that ensures a widening gulf between computer capacity and biological computing requirements. In addition to the scale of data, the workload of queries is also increasing dramatically. For example, one needs to run millions of searches to compare two genomes. The consequent degradation in the performance and increasing demands require investigation into search methods that organize the database off-line in order to speed on-line search.

## 3. Algorithms and Implementation

Our homology search algorithm modifies a general framework for computing local-alignments by matching q-grams first proposed and analyzed by Myers, contemporaneously with the development of BLAST (Wu et al., 1990; Myers, 1994). We will refer to the precise algorithm as Myers94. Our algorithm is as follows:

Build Index Structure (off-line):

1) Divide the sequences to be indexed into a set of overlapping substrings of length q, with step size 1.
2) Build an index structure D using weighted Hamming distance to support fast online range query.

Homology Search Query (on-line):

1) Divide query string W into a set of overlapping substrings, $\mathcal{F}=\{w_i|\ i=0..|W|-q\}$, of length q with step size 1.
2) For each $w_i$ in $\mathcal{F}$, run range query $Q(w_i, r)$ against database D to find a set of matching q-grams, $R_i=\{f_{i,j} \mid d(f_{i,j}, w_i)<=r,\ f_{i,j}\in D\ w_i\in \mathcal{F}\}$, where d is the distance function.
3) Using a greedy heuristic algorithm to extend and chain all fragments in $R_0 U R_1 U \ldots U R_{W-T}$ to deduce the result of homology search based on local alignment for query W.

The primary change to Myers94 that we investigate in this paper is that, by virtue of the metric-space index for each query fragment, we are able to look-up, on-line, all matching q-grams in the database within a neighborhood of radius r. In Myers94 the off-line index is constructed to support exact matching of fragments. In the on-line phase for each q-gram of the query sequence, Myers94 generates every possible q-gram within similarity distance r. For each generated q-gram the index is used to determine which q-grams in the database, if any, match the query q-grams. In other words, we replace Myers' generate and test method with a more powerful index that directly finds matching fragments. We note that BLAST uses the similar neighborhood generating method as Myers94. In BLAST, since similarity is used to characterize the neighborhood, the triangle inequality is absent and a direct index-lookup for q-grams cannot be done using metric-indexing methods. Thus, our entire approach was predicated on achieving the mPAM result. Also, the average-case complexity analysis of Myers94 is dominated by the number of matches found in the index, not the number of fragments in the generated neighborhood. Thus, this improvement, by itself, does not let us improve upon the formal algorithmic analysis. Last but not least, the algorithmic framework illustrated above includes several other simplifications on Myers94. These simplifications, detailed later, represent performance enhancing extensions that we leave for future work.

Since we focus on the searching phase in this paper, we only detail our algorithms used for searching fragments. Our implementation for chaining is essentially equivalent to Gusfield's chaining and thus will not be detailed (Gusfield, 1997).

### 3.1 MVP-tree Index Structure

Algorithm designers have leveraged the triangle inequality to produce entire classes of data structures to speed up metric-space search (Chavez et al., 2001). The "metric tree" and "generalized-hyperplane tree" (GHT) were proposed as tree structures for continuous distance functions in 1991 (Uhlmann, 1991). Later the GHT was extended

to GHT in m dimensions (Brin 1995). A more detailed effort on the "metric tree," detailed vantage point trees (VPT), followed (Yianilos, 1993). There are several derivatives of VPT, such as MVPT (Bozkaya and Ozsoyglu, 1997, 1999), and VPF with time complexity $O(n^{1-p}\log n)$ (Yianiols, 1999). Another algorithm, SAT, or spatial approximation tree, has approximately $O(n^{1-\Theta(1/\log\log n)})$ at query time (Navarro, 1999).

In preliminary work we implemented three methods of a metric-space index, one for each of the three major categories of metric-space index structure: vantage-point, generalized-hyperplane and bounding spheres. We determined that for biological sequence data the vantage-point method was superior.

| | **child$_1$** | **...** | **child$_n$** |
|---|---|---|---|
| **VP$_1$** | $d_{1,1}$ [min,max] | … | $d_{1,n}$ [min,max] |
| … | … | … | … |
| **VP$_n$** | $d_{n,1}$ [min,max] | … | $d_{n,n}$ [min,max] |

(a) Internal node

| | **child$_1$** | **...** | **child$_n$** |
|---|---|---|---|
| **VP$_1$** | $d_{1,1}$ | … | $d_{1,n}$ |
| … | … | … | … |
| **VP$_n$** | $d_{n,1}$ | … | $d_{n,n}$ |

(b) Leaf node

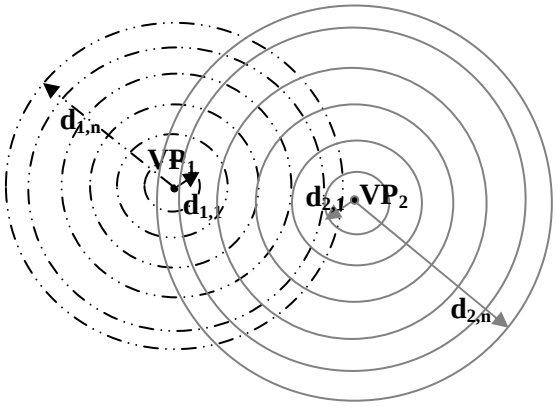


**Figure 1 Vantage point tree structure**

Although our implementation remains in main-memory we organized our node structures and initialization algorithm in a manner consistent with disk-mapped storage structures in anticipation of integrating this into MoBIoS, (Molecular Biological Information System), a next-generation database management system exploring the application of metric-space indexing for life-science data in general (Miranker et al., 2003). In our implementation, the MVP-tree is built recursively from the root. For each level, we first select m points as vantage points using the farthest first traversal algorithm (Hochbaum and Shmoys, 1985). For each vantage point, we select n – 1 distances, split points which partition the data into n evenly sized partitions. Hence, the fanout for each level is $n^m$. This process is continued until the partitions are sized such that a leaf node fits into a disk page. The typical node structure of a multi-vantage-point tree is shown in Figure1.

Our node structure is implemented differently from the original MVP node structures proposed by Bozkaya et al. First, we set the size of each leaf node to be the same as the disk page size, so each node access will take exactly one disk I/O operation if it is not already buffered in memory. Second, for each data object in a leaf node we do not store a list of distances between itself and each vantage point in its parent. Such a list can save the distance calculations when searching a leaf node, but also requires a larger amount of storage space. Since we fix our leaf node size as a disk page, the extra storage required could increase the depth of the tree and, in anticipation of disk-based implementation, increase disk reads.

The precise form of the bounding predicates in an MVP-tree is: $P([V_i, r_{min,\,i}, r_{max,\,i}])$. Specifically, the predicate requires that, for each vantage point $V_i$, the distance from $V_i$ to any data object in the sub-tree of the node is within the range $[r_{min,\,i}, r_{max,\,i}]$. We refer the reader to Bozkaya and Ozsoyoglu for detail of the search process (Bozkaya and Ozsoyglu, 1997,1999).

### 3.2 Distance Function

The key result of this section is that in the context of matching short overlapping q-grams, weighted Hamming distance yields the same matches as global alignment. Thus, we can replace an $O(n^2)$ calculation for an O(n) calculation. Although the use of an index structure substantially reduces the total number of distance calculations, distance calculations remain as the performance bottleneck.

**Definition 4 Substitution Cost Function** The substitution cost function, M(x,y), where x and y are symbols from an alphabet, $\mathcal{A}$, returns a nonnegative real number modeling the cost of substituting the sequence element x with the element y. Substitution weights are usually encoded in a substitution weight matrix; Where '_' denotes the gap, M(x,_) or M(_, x) returns gap penalty g.

**Definition 5 Global Alignment Distance Function** Given substitution cost function M(x,y), and two sequences A: $a_1a_2\ldots a_n$ and B: $b_1b_2\ldots b_n$, where x, y, $a_i$ and $b_i$ are drawn from an alphabet $\mathcal{A}$, the global alignment distance function GD(A, B) is defined as

$GD(A,B)=G_{n+1,n+1}$

where G is an n+1 x n+1 matrix and

$G_{0,i}= g*i$ for i=0…n+1,

$G_{j,0}= g*j$ for j=0…n+1, and

$G_{i,j} = \text{Min}( G_{i-1,j-1}+M(a_i, b_i), \quad \text{for } i,j \neq 0$

$G_{i-1,j}+M(a_i, \_),$

$G_{i,j-1}+M(\_, b_i))$

The global alignment distance calculation is adapted from the Needleman-Wunsch global alignment algorithm, which was proposed for similarity measure. The same algorithm can also be used for distance measure with minor variation. The time complexity for global alignment distance is $O(n^2)$ (Needleman and Wunsch, 1970).

Sellers showed that if the substitution cost function forms a metric, then the optimal global alignment distance over sequences drawn from the same alphabet is also a metric (Sellers, 1974). A degenerate case of this is where the substitution cost function amounts to the identity matrix. Then optimal global alignment becomes equivalent to the simple edit distance function, which is well known to be a metric. Since our substitution cost function, defined using mPAM, satisfies metric distance properties, the computed global alignment distances also form a metric.

**Definition 6 Weighted Hamming Distance** Given substitution cost function M(x,y), and two sequences A: $a_1a_2\ldots a_n$ and B: $b_1b_2\ldots b_n$, where x, y, $a_i$ and $b_i$ are drawn from an alphabet $\mathcal{A}$, the weighted Hamming distance function SD(A, B) is defined as

$$SD(A,B)=\sum_{i=1}^{n} M(a_i,b_i)$$

**Definition 7 Range Query** Given a set of objects O and a distance function d(a,b), range query $Q_d(q,r)$ returns a set of objects $\{o \in O \mid d(o,q) \leq r\}$.

If, under certain conditions, the global alignment distance is always the same as the weighted Hamming Distance, the global alignment distance can then be replaced by a weighted Hamming Distance computation for the same range query. Due to frequent distance computations during a range query, weighted Hamming distance computation can save a significant amount of query time, since weighted Hamming distance only requires O(n) computation time.

**Lemma 1** Given a substitution cost function M, a set of q-grams O, and a radius r, for all queries $q \in O$, if $r \leq 2g-1$ then $Q_{SD}(q,r)=Q_{GD}(q,r)$, where g is the gap penalty from substitution cost function used in both SD and GD.

**Proof:**

From the definition of the global alignment problem, it is trivial to show that the following two properties hold:

P1. $GD(x,y) \leq SD(x,y)$ for any x, y

P2. $GD(x,y) = SD(x,y)$, if there is no space insertion to form optimal global alignment.

a) Assume there is a q-gram $o \in O$, such that $o \in Q_{SD}(q,r)$ and $o \notin Q_{GD}(q,r)$.

By range query definition, $SD(o, q) \leq r$. P1 implies that $GD(o,q) \leq SD(o, q) \leq r$. So o must be in $Q_{GD}(q,r)$, which contradicts the assumption that $o \notin Q_{GD}(q,r)$. Hence, if $o \in Q_{SD}(q,r)$, $Q_{SD}(q,r) \subseteq Q_{GD}(q,r)$.

b) Assume there is a q-gram $p \in O$, such that $p \in Q_{GD}(q,r)$ and $p \notin Q_{SD}(q,r)$ when $r \leq 2g-1$,

The assumption indicates that $Q_{SD}(q,r) > r > 2g-1 \geq Q_{GD}(q,r)$. Since $2g-1 \geq Q_{GD}(q,r)$, there muse be no space inserted into either p or q to form the optimal global alignment. Otherwise $GD(p,q) \geq 2g$. From P2, it follows that $GD(p,q) = SD(p,q)$ and $p \in Q_{SD}(q,r)$, which contradicts the assumption $p \notin Q_{SD}(q,r)$. Hence, if $p \in Q_{GD}(q,r)$, then $p \in Q_{SD}(q,r)$ i.e. $Q_{GD}(q,r) \subseteq Q_{SD}(q,r)$ when $r \leq 2g-1$.

Therefore, from a) and b), $Q_{SD}(q,r)=Q_{GD}(q,r)$ when $r \leq 2g-1$. □

Based on many biological models, the gap penalty, which corresponds to the cost of insertion or deletion, is bigger than any mismatch score, which corresponds to the cost of substitution (Gusfield, 1997). So radius of 2g-1 is quite a large radius for a short q-gram. In the course of our experiments we observed there was no need to perform any range queries with a radius bigger than 2g-1. We promptly moved to weighted Hamming distance and witnessed precisely a factor of q improvement in execution times.

# 4. Experiment Results

The primary variables of our empirical analysis are the length of the q-gram and the radius of the search. We show that an MVP-tree provides scalable search. With the goal of producing an effective system, we assess how the choice of q-gram length and neighborhood radius impacts raw speed and accuracy.

## 4.1 Workload and Methodology

To assess sensitivity, we use an accuracy benchmark suite curated and furnished by NCBI (ftp.ncbi.nlm.nih.gov/pub/impala/blastest). The data set contains 6433 yeast protein sequences (about 2,892,155 residues). The query set contains 103 sequences whose true positive hits have been identified by human experts and whose curation is continually refined (Schaffer et al., 2001). The benchmark suite was downloaded in August 2002. The tests were conducted using Java 1.4 for Linux (SUSE 8.0; dual AMDXP 1800+ processors with 2GB memory).

For each query sequence s of length k**,** we divided s into a set of q-grams, $\{f_i | i=1..k-q+1\}$, referred to as query fragments. We collected the results from the range query $Q_{SD}(f_i,r)$ for all i and used a chaining algorithm to form final answers (Joseph et al., 1992). To control trade-off issues, as they may be influenced by the heuristics of the chaining algorithm, we exhaustively considered all of the returned q-grams. Since this paper focuses on the parameterization of the searching stage, we will not discuss the chaining algorithm, which is also a very important and complex problem in homology search. All of the timing results, unless otherwise noted, are time for the q-gram look up and do not include time spent on the chaining algorithm.

We use receiver-operating characteristic (ROC) scores to measure the accuracy of the search result (Gribskov and Robinson, 1996). For each query, the $ROC_{50}$ value is computed by comparing the result list with the list of true positive hits. The $ROC_{50}$ value has been computed as follows:

$$ROC_n = \frac{1}{nT} \sum_{i=1}^{n} t_i \qquad (1)$$

where $t_i$ is the number of true positive hits ranked ahead of the ith false positive, and T is the total number of true positives.

## 4.2 Choice of Fragment Length and Search Radius

We tested the effect of various parameters and the hypothesis that there is an optimal setting which best trades off between accuracy and speed. There are many parameters whose values have impact on the performance of algorithms. Among those, the length of the q-gram and search radius are the two most important parameters, since they impact the number of q-gram searches and results independent of data structure. For a given fragment length, a larger radius results in higher sensitivity and lower selectivity. As we will quantify, better sensitivity comes at the expense of speed.

| Fragment length | Search Radius | Average % true Positive Hits Returned | Average % fragments returned | Returned TP/ Returned fragment |
|---|---|---|---|---|
| 3 | 0 | 99.30% | 3.53% | 28.13388423 |
| 4 | 2 | 100.00% | 8.85% | 11.30505334 |
| 5 | 3 | 97.91% | 2.10% | 46.53131343 |
| 5 | 4 | 100.00% | 14.87% | 6.726649907 |
| 6 | 5 | 99.40% | 4.25% | 23.36353288 |
| 7 | 6 | 99.70% | 6.78% | 14.7048246 |
| 8 | 8 | 99.80% | 9.40% | 10.6207366 |
| 9 | 11 | 99.70% | 11.92% | 8.36498572 |

**Table-1 Accuracy and selectivity comparison for various fragment lengths and search radii**

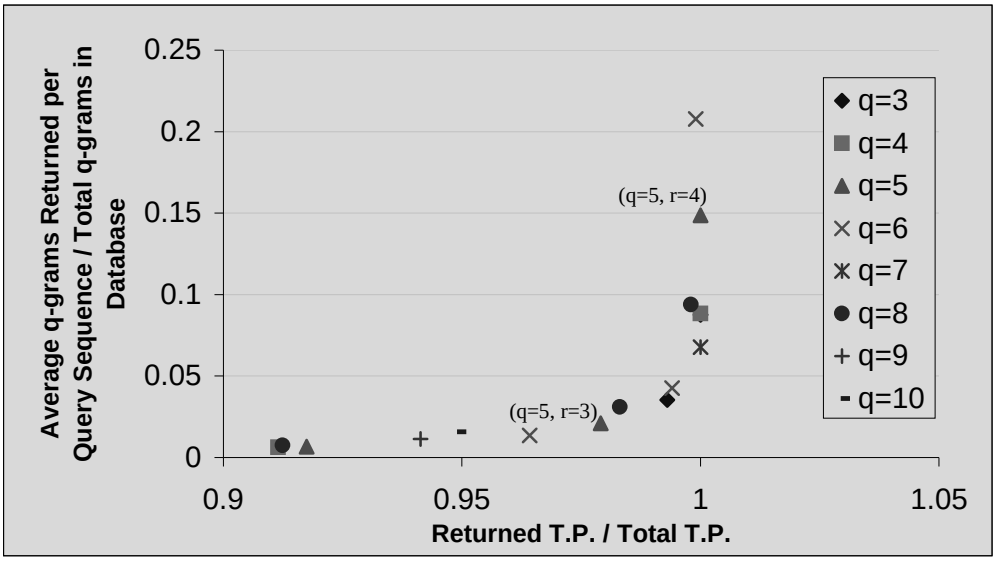


**Figure-2 Range searches with various radius were run on databases of total true positive hits set from benchmark with various q-gram lengths. The accuracy, percentage true positive hits returned, and selectivity, percentage of q-gram returned, are compared for different radius, q-gram length combination.**

In order to study the trade-off between selectivity and sensitivity, we projected the original benchmark database down to those sequences that are a true positive hit for at least one the test queries. Call the new database T. We built a set of q-gram databases T with different q-gram lengths. The entire query set was executed for each fragment length and a range of query radii. We computed the average percentage of the total q-grams in the database returned per q-gram query as selectivity, and the percentage of true positive hits within those results for each query as sensitivity. Table-1 details selected points of interest. The full results are plotted in Figure-2. The goal was to find the minimum radius for each different q-gram length that is necessary to identify all of the true positive hits for all queries. The ideal search result for each query would be such that the result contains at least one q-gram from every expected answer sequence and contains only q-grams from the answer set of that query.

Based on the ratio of percent returned TP and percent returned fragments, we determined to use a radius 3 search on databases with 5-gram as our default setting. The parameters of the MVP trees were 2 vantage points per node, 2 partitions per vantage points and a maximum of 100 q-grams per leaf node. We further tested those parameters on the complete yeast data set included in the benchmark

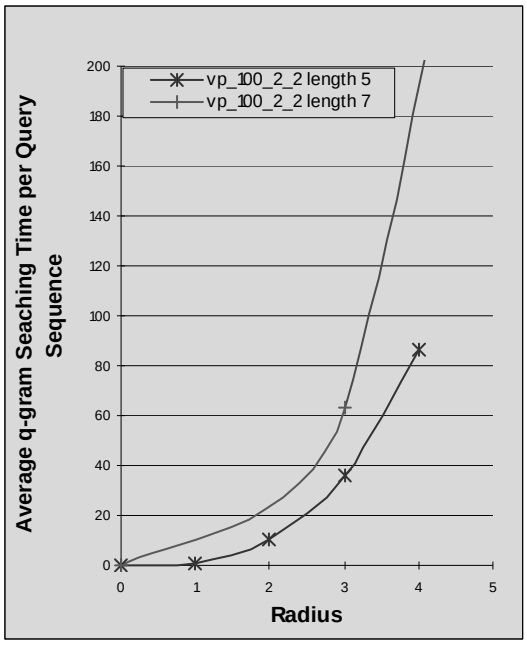


**Figure-3a Average searching time per query sequence vs. radius for yeast data set.**

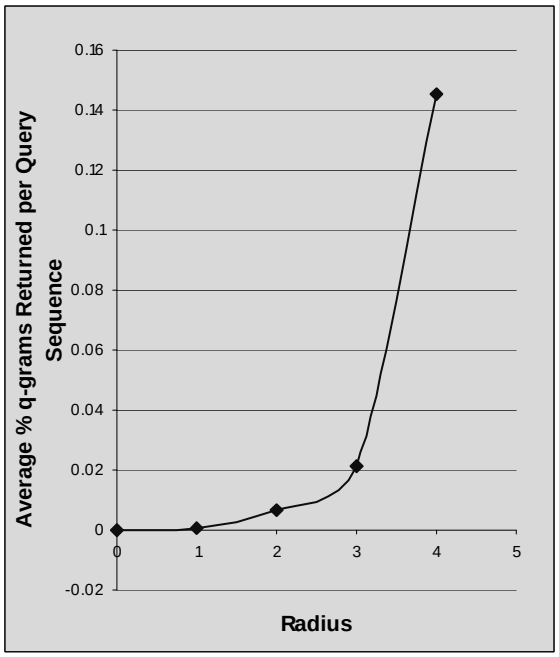


**Figure-3b Average percentage of fragments returned vs. radius for yeast data set**

Figure-3a plots the average search time for each query sequence as a function of radius. We see search time increases quickly for large search radii. For 5-gram, the searching time required for radius 4 is more than twice the amount of time used for radius 3. This phenomenon is known as "the curse of dimensionality" (Bellman 1961). It is arguable that our challenge in this approach is to determine if there is a combination of parameters that produces biologically effective results without encroaching too far beyond the knee of the curse so as to swamp the computation time. Figure-3b further illustrates this problem. While the result size for a radius 3 search is only about twice the result size from a radius 2 search, there are 6 times more results from a radius 4 search than a radius 3 search.

### 4.3 Search Strategy

One of our simplifications to Myers94 is that in the on-line portion we have implemented a breadth first algorithm where all search results for all query q-grams are collected and then chained. Myers details a depth-first strategy that integrates search with chaining, such that when a distant q-gram matching the beginning of the sequence is chained into a result, the size of the neighborhood around the search for q-grams matching later in the sequence is shrunk. A depth-first method has significant merit in our framework, since even slight reductions in the search radius may result in substantial reduction in the number of matching q-grams and concomitant execution time. We leave

the depth-first integration of chaining heuristics with search strategy as an open problem subject to future work.

In-lieu of studying depth-first enhancement, we further investigate a biologically motivated refinement to our search strategy, AutoRadiusQuery. When assessing if two biological sequences are homologous biologists consider properties beyond character substitution. The properties include compositional bias (relative frequency of the different acids) and the likelyhood that individual stretches of sequence are chemically active. In BLAST these aspects are embodied in the SEG and DUST filters, e-scores and a user selectable choice of substitution cost functions, i.e. a variety of PAM, BLOSSUM and other matrices (Wootton & Federhen 1993; Altschul and Gish, 1996; Altschul et al., 2001).

In AutoRadiusQuery, the radius of the search is adjusted in anticipation of the search results. In nature, some amino acids are subject to substitution more often than others. As a result, an amino acid with high mutability has a shorter distance to other amino acids. In addition, twenty amino acids are not uniformly distributed. Composition bias of a database and varying mutability of amino acids cause the result size to fluctuate greatly for range queries of different q-grams with the same radius. Due to the choice of small length for q-gram, some simple repeat patterns may occur frequently with no biological significance but considerably increase computation load. AutoRadiusQuery automatically adjusts searching radius based on the predicated size of the result for each fragment. For a given fragment, if the predicated size of the result is smaller than a predetermined lower bound for a default radius, then the actual searching radius will be automatically increased by 1 from the default radius. We compared the accuracy and speed trade-off between fixed radius search and AutoRadiusSeach.

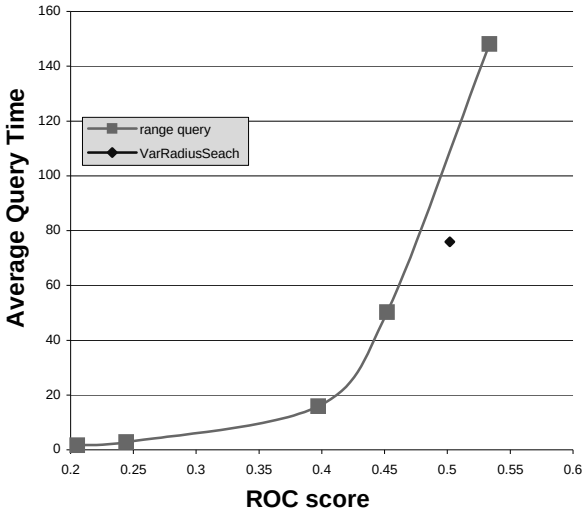


**Figure-4 Comparison of query time and sensitivity (average $ROC_{50}$ score per query) of results; the curve shows results from range queries with different radii. The isolated point was the result of an AutoRadiusQuery.**

| Search Method | Sequential Search with Smith-Waterman local alignment algorithm | | | Indexed Search | | | BLASTP | |
|---|---|---|---|---|---|---|---|---|
| | | | | Radius 3 | Radius 4 | AutoRadius Search | | |
| Matrix name | mPAM | PAM250 | PAM70 | MPAM | | | PAM250 | PAM70 |
| Average $ROC_{50}$ | 0.48 | 0.59 | 0.50 | 0.45 | 0.53 | 0.50 | 0.53 | 0.42 |

**Table-2 Comparison of average of $ROC_{50}$ value for various searches**

Figure4 plots the comparison between fixed radius search and self-adjustable radius search. The $ROC_{50}$ scores were computed after applying the chaining algorithm and the total time also includes the time for chaining. The graph indicates that AutoRadiusSearch produces a better trade-off between accuracy and speed than range search with a fixed radius. Table-2 compared the average $ROC_{50}$ score for each query from our algorithm and the results using other searching algorithm with the same benchmark. Here, we show that using metric space indexing, protein sequence homology search could yield accuracy comparable with BLASTp on same benchmark. Note that the accuracy of results is actually affected by many factors, such as the value of substitution matrices, searching strategies and chaining strategies, etc. We will discuss more on these factors affecting the trade-off between the speed and accuracy in the discussion section.

### 4.4 Scalability Study

Scalability is important in the face of the growth of genomic data. In this section, we tested whether our data structure can scale well as the size of the dataset grows. The data used for the scalability study was downloaded from Genbank in July 2003 (ftp://ftp.ncbi.nih.gov/genbank/genpept.fsa.z). The dataset contains FASTA formatted amino acid translations extracted from GenBank/EMBL/DDBJ records that are annotated with one or more CDS features. A set of databases was built with different subsets of the data that were sequentially taken from the full dataset. The same set of queries from the yeast benchmark was used for all the databases with AutoRadiusSearch.

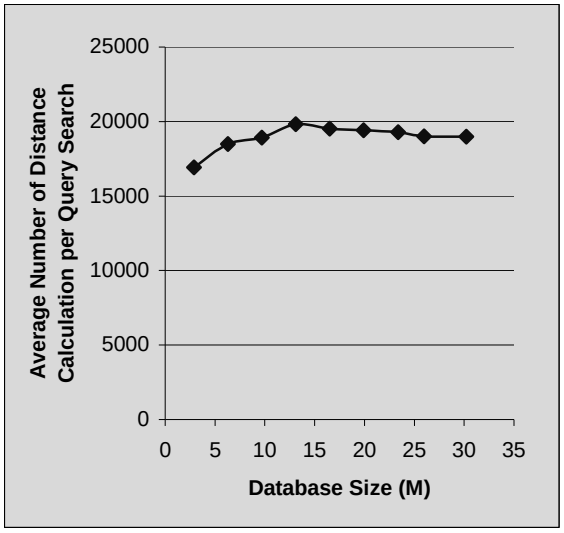


**Figure-5a Average number of distance calculation per query search vs. database size**

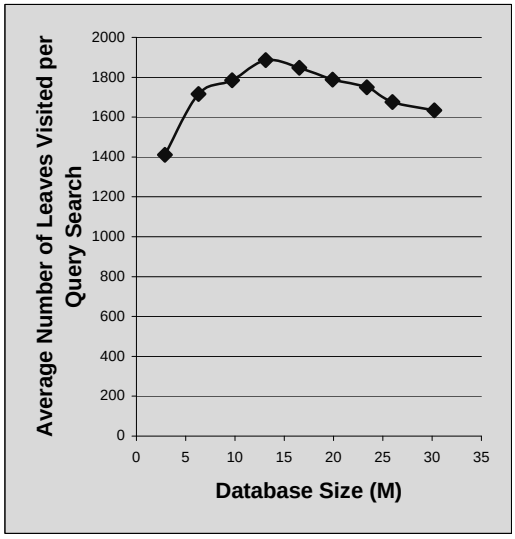


**Figure-5b Average number of leaf nodes visited per query search vs. database size**

The average number of distance calculations and average numbers of leaf nodes visited are plotted in Figure-5. Both figures reveal scalability with the size of the database. It is also interesting to notice that both numbers slightly decreased for a larger data set. We have reason to believe that as the database grows the logical locality of the clusters starts to correspond better to the physical clustering on pages (Mao et al., 2003). The effect is that entire contents of sub-trees could be found and returned in their entirety without further distance calculations, thus reducing the number of distance calculations. Similarly, entire sub-trees can be pruned, reducing the number of leaves visited.

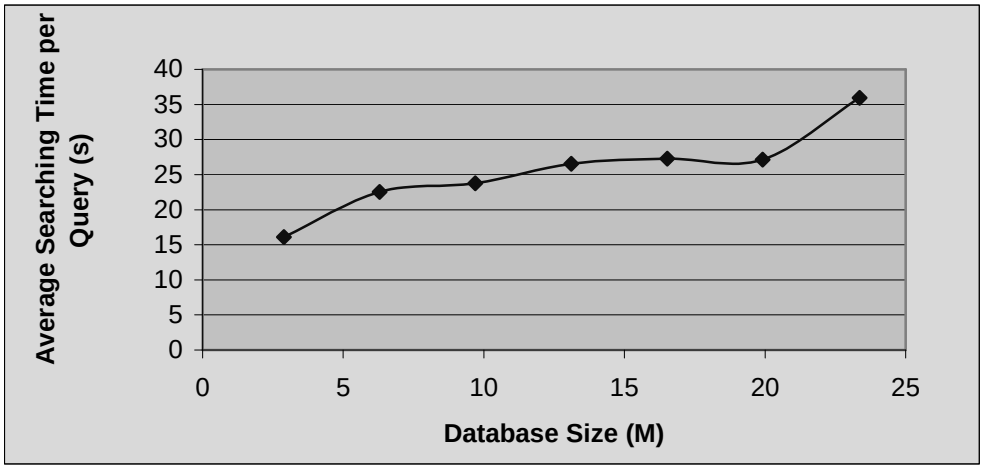


**Figure-6 Average search time vs. database size**

Figure-6 shows average searching time (as wall clock time) for different sizes of data. Despite a slight increase, the results also showed good scalability.

## 5. Discussions and Future Work

We have shown that by using the mPAM substitution matrix and metric-space indexing, protein sequence databases can be managed while taking evolutionary criteria into consideration. In particular, MVP-trees provide scalable q-gram range search performance for peptide sequences and yield accuracy comparable to the popular methods as demonstrated by an established benchmark. These results are also applicable to DNA sequence database when sequence homology must consider the transcription of reading frames to amino acids.

Given this basis, it is fair to ask, “How does this relate to BLAST, and what work remains to create a scalable alternative to BLAST?”. We did not report execution times for our system as it is implemented in Java and our concentration has been on the feasibility of the approach. The most important result of this study is that we have confirmed empirically that the curse of dimensionality affects biological sequence data. Even so, good accuracy can be achieved without encroaching fatally into the unstable region. On that basis we can enumerate a number of open research problems whose resolutions pave the road to new biological sequence management systems.

The choice of substitution matrix is, in general, still an open issue in biology. The mPAM matrix is simply the first biologically effective metric-substitution matrix. Just as the BLOSSUM matrices are sometimes seen as an improvement over PAM matrices, we expect mPAM can be improved upon. Our conjecture is that a better matrix will achieve comparable accuracy at narrower radii with consequent improvements.

A lesson of the AutoRadiusSearch is that good accuracy may be achieved by a context dependent choice of radii. Myers also laid groundwork for varying the radii of the search. Key elements of BLAST include filtering-out low complexity regions and computing a probabilistic significance score (e-value) for each output sequence (Altschul and Gish, 1996; Promponas et al., 2000; Wootton and Federhen 1993). These elements taken together suggest that the ultimate algorithm for scalable sequence retrieval will be composed of a depth-first search strategy where the scope of the search is parameterized by the anticipated significance of the next matching fragments. This amounts to integrating the search and chaining phases. We believe that relying on domain knowledge of the indexed data is a key to avoid the curse of dimensionality.

The choices of using an MVP tree and parameter selections are based on empirical results not yet thoroughly investigated. Based on the results of the scalability test, it is arguable that there is no universally optimal setting. Since the parameter selection directly interferes with clustering, an extension to MVP-trees that allows them to automatically adapt to domain specific clustering is called for (Yianilos, 1999; Navarro, 1999). A data structure called the M-tree attempted to do so, but we found it to be ineffective in this application (Mao et al., 2003). The results also indicate the presence of duplicate entries in large datasets. These contributed to the slight increase in search time while both the average number of distance calculation and the average number of leaves visited stay low. One simple solution is to implement “buckets” in leaf nodes. Each “bucket” represents a q-gram with a unique sequence and stores a list of the q-grams that share the same sequence but are from different locations.

Lastly, it is important to recognize that sequence analysis problems now go far beyond BLAST homology searches. Generating genomic data is no longer difficult for biologists but analyzing and mining the data is. Over 250 genomes from different species have been sequenced. By the end of 2005, over 1000 genomes will have been sequenced. Comparing whole genomes to one another—not merely the genetic sequences, but their annotation as genes and proteins—is proving to be an increasingly powerful means of biological discovery (Marcotte, 2002). O(log(n)) retrieval methods may lead to very fast all against all genome analysis (Wang, et al., 1994), analysis that will soon be impossible with algorithms that use linear scans.

## Bibliography


Altschul, S.F. & Gish, W. (1996) Local alignment statistics. Methods Enzymol.266, 460-480

Altschul, S.F. Bundschuh, R., Olsen, R. & Hwa, T. (2001) The estimation of statistical parameters for local alignment score distributions. Nucleic Acids Res., 29, 351-361

Altschul, S.F. Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) Basic local alignment search tool. J. Mol. Biol. 215, 403-410

Bellman, R. (1961), Adaptive Control Processes: A Guided Tour, Princeton University Press. U.S.

Benson D.A., Karsch-Mizrachi I., Lipman, D. J., Ostell, J., Rapp, B. A., & Wheeler, D. L. (2002) GenBank. Nucleic Acids Res., 20, 1, 17-20

Bozkaya T. & Ozsoyoglu M. (1997) Distance-based indexing for high-dimensional metric spacs. In Proc. ACM SIGMOD International Conference on Management of Data (1997) 357-368

Bozkaya, T. & Ozsoyoglu, M. (1999) Indexing Large Metric Spaces for Similarity Search Queries. Association for Computing Machinery Transactions on Database System, pages 11--34,

Brin, S. (1995). Near neighbor search in large metric spaces. In Proc. 21st Conference on Very Large Database (VLDB'95), 574-584

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

Dayhoff M.O., Schwartz R. & Orcutt B.C. (1978) Atlas of Protein Sequence and Structure. Vol. 5. Suppl. 3, 345-358

Giladi, E., Walker, G. M., 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), 873-879

Gribskov, M. & Robinson, N. L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. Computers and Chemistry. 20(1), 25-33

Gusfield, D. (1997) Algorithms on Strings, Trees and Sequences Computer Science and Computational Biology. pp.449-454 Press Syndicate of the University of Cambridge, USA

Henikoff, S. & Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. USA 89, 10915-10919

Hochbaum, D. S. & Shmoys, D. B. A best possible heuristic for the k-center problem. Mathematics of Operational Research, 10(2):180-184, 1985.

Joseph, D. Meidanis, J. & Tiwari, P. (1992) Determining DNA sequence similarity using maximum independent set algorithms for interval graphs. In Proc. Of the Third Scand. Workshop on Algorithm Theory. Springer LNCS 621, pages 326-37

Kent, W. J. (2002) BLAT-The BLAST like alignment tool. Genome Res. 12, 656-664

Linial, M., Linial, N., Tishby, N. & Yona, G. (1997) Global self organization of all known protein sequences reveals inherent biological signatures. J. Mol. Bio. 268, 539-556.

Mao, R., Xu, W., Singh, N. & Miranker, D. P.. (2003) 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

Marcotte, E.M. (2002) Predicting protein function and networks on a genome wide scale. Gene Regulations and Metabolism 223-249

Marcotte, E.M. & Date, S.V. (2001) Exploiting Big Biology: Integrating Large-scale Biological Data for Function Inference. Briefings in Bioinformatics 2(4): 363-374 (2001)

Miranker, D. P. Xu, W. & Mao, R. (2003) Architecture and Application of MoBIoS, a Metric-Space DBMS to Support Biological Discovery. 15th International Conference on Scientific and Statistical Database Management. (SSDBM03) 241-244

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

Navarro, G. (1999) Searching in metric spaces by spatial approximation In proc. String Processing and Information Retrieval (SPIRE'99 IEEE) 141-148

Needleman, S.B. & Wunsch C.D. (1970) An efficient method applicable to the search for similarities in the amino acid sequences of two proteins. J. Mol. Biol. 48, 443-453
Patterson, D.A. & Hennessy, J.L. (1996) Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers Inc., San Francisco
Pearson, W. R. & Lipman, D. J. (1988) Improved tools for biological sequence comparison. Proc. Natl Acad. Sci. USA, 85, 2444-2448
Promponas, V. J., Enright, A. J., Tsoka, S., Kreil, D. P., Leroy, C., Hamodrakas, S. J., Sander, C. and Ouzounis, C. A. (2000) CAST: an iterative algorithm for the complexity analysis of sequence tracts *Bioinformatics* 16(10), 915-922.2000
Rouchka, E. C. Gish, W. & States D. J. (2002) Comparison of whole genome assemblies of the human genome. Nucleic Acids Res., 30(22), 5004 – 5014
Schaffer, A. A. Aravin, L. Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V. & Altschul, S.F. (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res., 29(14), 2994-3005
Sellers, P.H. (1974) On the theory and computation of evolutionary distances. J. Appl. Math. (SIAM). 26, 787-793ad. Sci. USA 89, 10915-10919
Smith, T.F. & Waterman, M.S. (1981) Identification of common molecular subsequences. J. Mol. Biol. 147, 195-197
Tan, Z., Cao, X., Ooi, B.C. & Tung, A.K.H. (2003) The ed-tree: an index for large DNA sequence databases In Proc. 15th International Conference on Scientific and Statistical Database Management (SSDBM 2003) 151-160
Uhlmann, J. (1991) Satisying general proximity/similarity queries with metric trees Information Processing Letters 40, 175-179
Jason Tsong-Li Wang, Dennis Shasha (1990) Query Processing for Distance Metrics. In Porc. Of the Very Large Database System Conference, (VLDB'90), 602-613
Wootton. J.C. and Federhen S. (1993) Statistics of local complexity in amino acid sequences and sequence databases Comput. Chem. 17 (1993) 149 --163.
Wu, S. Manber, U. Myers, G. and Miller, W. (1990) An O(NP) Sequence Comparison Algorithm. Information Processing Letters 35(6): 317-323 (1990)
Xu, W. & Miranker, D.P. (2003) A metric model for amino acid substitution, in press Bioinformatics (http://www.cs.utexas.edu/users/mobios/)
Yianilos, P. (1993) Data structures and algorithms for nearest neighbor search in general metric spaces. In Proc. 4th ACM-SIAM Symposium on Discrete Algorithms (SODA'93) 311-321
Yianilos, P. (1999) Excluded middle vantage point forests for nearest neighbor search. In DIMACS Implementation Challenge, ALENEX'99 (Baltimore, MD 99)