Bioinformatics and Biosystems 2006, Vol. 1, No. 1, pp. 46-50

# Protein Sequence Search based on N-gram Indexing

**Mi-Nyeong Hwang, Jinsuk Kim**

**System Development Team, Knowledge Information Center**
**Korea Institute of Science and Technology Information (KISTI), Korea**

## Abstract

According to the advancement of experimental techniques in molecular biology, genomic and protein sequence databases are increasing in size exponentially, and mean sequence lengths are also increasing. Because the sizes of these databases become larger, it is difficult to search similar sequences in biological databases with significant homologies to a query sequence. In this paper, we present the N-gram indexing method to retrieve similar sequences fast, precisely and comparably. This method regards a protein sequence as a text written in language of 20 amino acid codes, adapts N-gram tokens of fixed-length as its indexing scheme for sequence strings. After such tokens are indexed for all the sequences in the database, sequences can be searched with information retrieval algorithms. Using this new method, we have developed a protein sequence search system named as ProSeS (PROtein Sequence Search). ProSeS is a protein sequence analysis system which provides overall analysis results such as similar sequences with significant homologies, predicted subcellular locations of the query sequence, and major keywords extracted from annotations of similar sequences. We show experimentally that the N-gram indexing approach saves the retrieval time significantly, and that it is as accurate as current popular search tool BLAST.

**Keywords**: homology search, N-gram indexing, sequence retrieval, sequence search tool, ProSeS

## Introduction

Genomic and protein databases assist molecular biologists in understanding the biochemical function, chemical structure, and evolutionary history of organisms. In the early years of sequence searching, only a few specialized centers had access to the necessary computing facilities and programming expertise to perform comparison scans. Today, the optimum choice is again swinging towards databases maintained at a few centers, but now fast networks and windowing workstations allow the user to use software locally and be unaware that the search is being carried out on a computer in another country.

Popular systems for searching databases match queries to answers by comparing a query to each of the sequences in the databases. Efficiency in such exhaustive systems is not satisfactory, since servers must process many queries simultaneously and solution of each query requires comparison to over the huge sequence databases. BLAST (Basic Local Alignment Search Tool) is one of the most commonly used software to search biological sequences based on homology so far Altschul et al. (1990). BLAST uses a heuristic method to find the highest scoring and locally optimal alignments between a query sequence and sequences in the database. The program has been developed by NCBI (National Center for Biotechnology Information) and benefits from technical supports for strong and continuing refinement. Although BLAST also adopts simple indexing scheme to build sequence databases and to choose candidates from the database, it does not fully utilize indexing features of information retrieval. Furthermore BLAST requires powerful computational facilities of the CPU processors for the reason of its origin in dynamic programming. This leads BLAST to a problem, where many simultaneous users through the Internet do not satisfied with search speed.

In this paper, we propose the method of protein sequence search based on N-gram indexing for efficient similarity search. And we show experimentally that query evaluation using our new technique has the necessary properties such as rapidity, accuracy, scalability, and efficiency. ProSeS can be used for the same search tasks as the popular BLAST search

Corresponding Authors: Mi-Nyeong Hwang, Jinsuk Kim, Tel: +82-42-828-5124, Fax: +82-42-828-5179, Email: {mnhwang, jinsuk}@kisti.re.kr
This work was supported in part by the IBM Korea.
Availability: The Protein Sequence Search (ProSeS) service is available at http://proses.kisti.re.kr.

system.

## Related Works

### Indexed Genomic Searching

A general method for reducing search costs is to store an abstraction or index that can be used to assess broad similarity to a query. While the cost is the need to store index, the potential saving is that fetching a limited volume of information should enable identification of a small number of sequences as likely answers.

Interval-based indexing of genomic databases was first proposed by Orcutt and Barker (1994). They proposed their algorithm as a method to identify amino-acid sequences in the PIR database - an international protein sequence database. But no implementation of the approach is given. After that, Barton notes that an implementation, SCAN, was available with the PIR database for use in exact matching in Barton (1996). The SCAN approach was not highly successful and has not been developed in fear of simple measurements of matching intervals and non-overlapping intervals.

Altschul *et al.* have implemented a similar approach to SCAN, BLAST, which uses a table of all the sequence intervals in the database (see Altschul et al. (1990)). Because of limitations similar to those in SCAN, this approach was slower than exhaustively searching the database those days.

The most recent indexed scheme is the Rapid Access Motif database (RAMdb) system for finding short patterns, and motifs in genomic databases (see *Fondrat et al.* (1995)). In the approach of Fondrat and Dessen, each genomic sequence is indexed by its constituent overlapping intervals in a hash table. For each interval in the collection, a neighboring list of sequence numbers and offsets is stored, which allows rapid location of any motif matching a query motif.

The FLASH (Fast Look-Up Algorithm for String Homology) search tool redundantly indexes genomic data based on a probabilistic scheme in Califano *et al.*(1993). For each interval of length n, the FLASH search structure stores, in a hash-table, all possible orderly contiguous and non-continuous subsequences of length m that begin with the first base in the interval, where $m < n$. For example, for a nucleotide sequence ACCTGATT the index terms for the first $n = 5$ bases, where $m = 3$, would be ACC, ACT, ACG, ACT, ACG, and ATG; each of the permuted strings begins with the base A, the first base in the interval length of $n = 5$. Then the hash table has each permuted m-length subsequences, the sequences that contain the permuted subsequences, and the offsets within each sequence of the permuted subsequence. Califano and Rigoutsos found that FLASH was about ten times faster than BLAST for a small test collection and was superior in accurately and sensitively determining homologies in sequence database searching. However, the redundant index, which is stored in a hash-table structure and is uncompressed, is gigantic. Califano and Rigoutsos report that, for a nucleotide collection of around 100Mb, the index requires 18 GB on the disk, around 180 times the collection size.

Williams and Zobel proposed a two-component partitioned search process embodied in a research prototype system, CAFE (see Williams *et al.* (2002)). The first component of their approach, a coarse search, uses an inverted index to select a subset of sequences that display broad similarity to the query sequence. The second component, a computationally more expensive fine search mechanism, ranks the resultant sequences from the coarse search in the order of relevance to the query sequence.

## Methods

### N-gram based indexing

To achieve efficient and effective retrieval from sequence databases, we propose some modifications to the general method used for text strings in information retrieval researches.

It is relatively clear to consider that texts written in natural languages are comprised of understandable words. Thus, in information retrieval, generally documents are segmented into morphemes or words, and then indexed to inverted files. Protein sequences also can be regarded as texts written in language of 20 amino acid codes. Since, however, it is not clear how to extract meaningful words from the simple string of amino acid codes, ProSeS adopts N-gram tokens of fixed length as its indexing scheme for sequence strings. For example, if the fixed length, or cutting interval is 4, which is known as tetragram, sequence 'ACDEFLERR' is segmented into 'ACDE', 'CDEF', 'DEFL', 'EFLE', 'FLER', and 'LERR'. If the fixed length is 5, pentagram, the sequence is segmented into 'ACDEF', 'CDEFL', 'DEFLE', 'EFLER', and 'FLERR'. After such tokens are indexed for all the sequences in the database, sequences can be searched with information retrieval algorithms.

### Retrieval based on Vector Space Model

The vector space model has been investigated in depth for

information retrieval (see Raghavan *et al.* (1986)). ProSeS uses this retrieval method for searching similar sequences. The similarity measure (*Sim(q,d)*) between query sequence *q* and target sequence *d* is defined as follows:

$$Sim(q,d) = \frac{1}{W_d} \cdot \sum_{t \in q \wedge d} w_{d,t} w_{q,t}$$

with:

$$W_d = \log(1 + \sum_{t \in d} f_{d,t})$$

$$w_{d,t} = \log(f_{d,t} + 1) \bullet \log(\frac{N}{f_t} + 1)$$

$$w_{q,t} = \log(f_{q,t} + 1) \bullet \log(\frac{N}{f_t} + 1)$$

where *fs,t* is the frequency of N-gram token t in sequence *s*; N is the total number of sequences in the data collection; *ft* is the number of sequence where token *t* occurs more than or equal to once; *ws,t* is the weight of term t in the query or target sequence *s*; *Wd* is the normalization factor for the length of target sequence d.

BLAST (Basic Local Alignment Search Tool) also adopts simple indexing scheme to build sequence databases and to choose candidates from the database, but it does not fully utilize indexing features of information retrieval in Altschul (1990). On the other hand, ProSeS' sequence search only depends on the N-gram indexing feature. Due to this fact, ProSeS gives slightly different result (but ignorable) compared with BLAST. But it is about nine times faster than BLASTP and shows less system resource usage.

### Protein Subcellular Localization Predication

It's important to verify protein location(s) in the cell to understand its physiological or structural functions. ProSeS provides a summary list of protein subcellular locations predicted by ProSLP system (see Park et al. (2003)). ProSLP (Protein Subcellular Localization Prediction) performs sequence search by ProSeS and predicts information of subcellular localization by k-NN (k nearest neighbors) algorithm.

### Major Keyword Suggestion

ProSeS suggests major keywords related to the query sequence and their weights. With a simple data mining technique, these keywords are extracted from the names of proteins that show significant homologies or similarities with the query sequence.

## Results

### Test Collection

To make a comparison of retrieval effectiveness between BLAST and ProSeS, we use a subset of the PIR-NREF database (see Wu *et al.* (2002)). The PIR-NREF is a Non-redundant REFerence protein database designed to provide a timely and comprehensive collection of all protein sequence data, keeping pace with the genome sequencing projects and containing source attribution and minimal redundancy. The PIR-NREF current release 1.29, 25-Aug-2003, contains 1,292,569 entries. We select randomly a set of 100 sequences out of 1,292,569 sequences with sequence lengths from 50 to 1000. We ran BLASTP against these 100 sequences and generated local alignments for each test sequences up to 1000 entries. This data set was regarded as relevant answers that retrieval effectiveness was measured.

To quantify the relative performance or retrieval effectiveness of ProSeS, we use the relevance-based measures of recall and precision. Recall and precision are frequently used to demonstrate the retrieval effectiveness of systems, particularly those used for information retrieval. Sequences of resultant query set searched by BLAST with filtering option on/off for each test data assigned collection.

Precision is a measure of the fraction of relevant answers retrieved in the result set at a particular point, that is

$$P = \frac{\#\text{ of relevant items retrieved}}{\text{Total \# of items retrieved}}$$

Recall, in contrast, measures the fraction of the relevant answers retrieved from the relevant answers at a particular point, or

$$R = \frac{\#\text{ of relevant items retrieved}}{\text{Total \# of relevant items in the collection}}$$

To compare the search speed of BLAST and ProSeS, we measured the time to search 100 query sequences from the PIR-NREF database and averaged it.

### Results

Figure 1 shows a mean recall-precision graph for the search results from ProSeS. Results are shown for searching the test collection with our query test set comprised of 100 sequences. It is noted that ProSeS uses an interval length of $n = 5$, pentagram.

The precision of the recall 0.1 is 0.8544(BLAST with no filtering option) and 0.8675(BLAST with filtering option) as shown in Figure 1. These results suggest that ProSeS is almost as effective as BLAST in finding homologies at lower recall levels.

We show in Table 1 the relative speeds of both BLAST with filtering option on/off and ProSeS in searching the PIR-NREF database with our 100-query test set, limiting the maximum number of results as 1000. The parameters are the same as previous experiment. This experiment was carried out on a machine with Intel XEON dual CPU 2.4GHz processors and 3GB RAM which is operated with Linux.

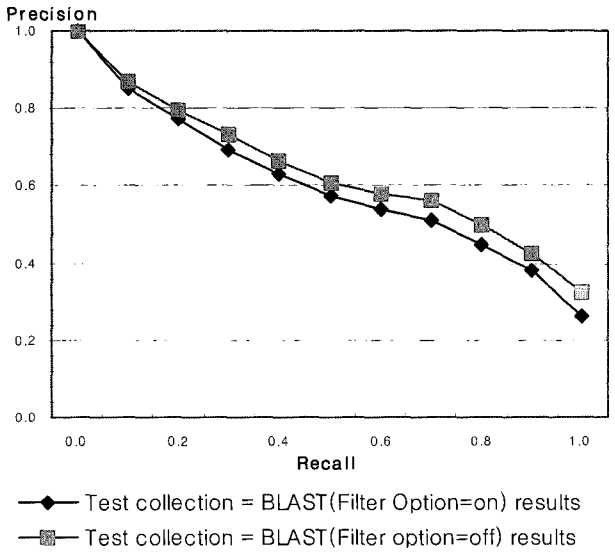


**Figure 1.** Mean recall-precision for search results from ProSeS. The reference data (test collection) is a set of search results of BLAST for 100 test queries. We parameterized BLAST with filtering option on/off, and E-value (expection value) = 0.0001.

Searching by ProSeS can be over nine times faster than by BLAST with filtering option and twelve times faster than with BLAST with no filtering option.

Although the results are shown in another paper in Park *et al.*(2003), N-gram indexing was successfully applied to protein subcellular localization, too. And a user survey showed that keyword suggestion based on data mining for N-gram indexed database was also meaningful.

**Table 1.** Mean elapsed time for searching a protein sequence query on the BLAST and ProSeS when the maximum number of result sequences is limited to up to 1000. The times for searching 100 test queries were measured and their averages are provided.

| System | BLAST (filtering) | BLAST (No filtering) | ProSeS |
|---|---|---|---|
| time(sec) | 19.8 | 28.1 | 2.2 |

## Discussion

In this paper, we have revealed that our N-gram based retrieval system is much faster than the well-known exhaustive search system, BLAST. In addition, its retrieval effectiveness can match for that of BLAST. This satisfiable application of indexing to protein sequences shows that indexing becomes preferable to exhaustive search in case of sufficiently large data sets.

In contrast to previous search techniques, ProSeS is fast with low memory requirement and low overhead of processor but depends on large indexes. We are confident that indexed systems such as ProSeS will be the practical option for searching through the vast quantities of biological data.

## References

[1] Altschul S, Gish W, Miller W, Myers E, Lipman D. (1990) Basic local alignment search tool. *Journal of Molecular biology*, 215: 403-410

[2] Orcutt B, Barker W. (1994) Searching the protein database. *Bulletin of Mathematical Biology*, 46:545-552

[3] Barton G (1996) Protein sequence alignment and database scanning. In M.J.E. Sternberg, editor, Protein Structure Prediction: A Practical Approach. IRL Press at Oxford University Press

[4] Fondrat C, Dessen P (1995) A rapid access motif database (RAMdb) with a search algorithm for the retrieval patterns in nucleic acids or protein databanks. *Computer Applications in the Biosciences*, 11(3):273-279

[5] Califano A, Rigoutsos I (1993) FLASH: A *fast look-up algorithm for string homology.* In International Conference on Intelligent Systems for Molecular Biology, pages 56-64, Bethesda, MD

[6] Raghavan V, Wong S (1986) A critical analysis of vector space model for information retrieval. Journal of

the American Society for Information Science, 37(5), p. 279-87

[7] Williams H , Zobel J (2002) Indexing and Retrieval for Genomic Databases. TKDE, 14(1):63-78

[8] Park HE, Kim JS (2003) ProSLP: a novel predictor for subcellular localization based on N-gram Features. Submiited to Journal of bioinformatics, available at http://proslp.kisti.re.kr.

[9] Wu C, Huang H, Arminski L, Castro-Alvear J, Chen Y, Hu Z, Ledley R, Lewis K, Mewes H, Orcutt B, Suzek B, Tsugita A, Vinayaka C, Yeh L, Zhang J, Barker W, (2002). The Protein Information Resource: an integrated public resource of functional annotation of proteins. *Nucleic Acids Research*, 30, 35-37.