N-GRAM INDEXING FOR PROTEIN SEQUENCE DATABASES

Jinsuk Kim[1], Mi-Nyeong Hwang[1], Sul-Ah Ahn[2], and Hyeon S. Son[2,3]*

[1]Information Technology Department, Korea Institute of Science & Technology Information, P.O. Box 122, Yuseong-gu, Daejeon 305-600, Republic of Korea

Email: jinsuk@kisti.re.kr and mnhwang@kisti.re.kr

[2]Center for Computational Biology & Bioinformatics, Korea Institute of Science & Technology Information, P.O. Box 122, Yuseong-gu, Daejeon 305-600, Republic of Korea

Email: snowy@kisti.re.kr

**[3]Bioinformatics Department, School of Public Health, Seoul National University, 28 Yongon-dong, Chongno-gu, Seoul 110-799, Republic of Korea*

Email: hss@kisti.re.kr

ABSTRACT

Though the sequence databases of proteins and DNAs are increasing in size exponentially, still exhaustive sequence search systems are commonly used in conducting biological researches. However, due to the advancement of information technology, many information retrieval algorithms have been developed to search strings in large-scale text databases and are proved to be successful. We propose that these algorithms could also be applied to the biological data. Four n-gram indexing methods (tri-gram, tetra-gram, penta-gram, and hexa-gram) were applied to extract indices from protein sequences of the PIR-NREF database, and their retrieval effectiveness and speed were measured. Penta-gram method showed the best results that its retrieval effectiveness matches for BLASTP and its retrieval speed was about 38 times faster than BLASTP program. Our protein sequence search service is accessible at http://proses.kisti.re.kr.

Keywords: Indexing, Information retrieval, BLAST, Sequence Retrieval, N-gram, Protein sequence

1 INTRODUCTION

For newly discovered proteins or DNA sequences, it is important to find similar or homologous sequences from existing databases, and it often gives some useful clues to their function. Popular biological sequence search systems find answers by computing local alignment scores between a query sequence and every sequence in the target database. Computing time in exhaustive search systems is proportional to

the both lengths of the query and target sequences as well as to the number of sequences in the database. The amount of calculation is obviously huge in such systems; therefore efficiency is crucial in order to get search result within tolerable time limit. As biological databases are rapidly growing in size, this efficiency problem is becoming serious. To avoid the huge computational tasks required by conventional local alignment algorithms, heuristic search algorithms are often employed instead. They attempt to select candidate sequences and to determine final answers by calculating local alignments only for these candidates. Among them, FASTA suggests a *k-tuple* method, where *k* is a small integer and a k-tuple is a sub-string interval, of fixed-length *k*, extracted from the biological sequences. The k-tuple method is a two-phase approach (Lipman & Pearson, 1985); in the first phase, it tries to find identical k-tuple subsequences that occur in both the query and the database sequences, then, it tries to extend these subsequences to get possible local alignments. This k-tuple method is further improved in BLAST (Basic Local Alignment Search Tool) program (Altschul, 1991), which soon became a dominant tool for biological sequence retrieval. Another work, derived from information retrieval researches, extends the first step of FASTA to inverted files of n-gram index (Williams & Zobel, 2002). It introduces a two-phase search process embodied in a research prototype system, *CAFE*. The first phase, called *coarse search*, uses an inverted index to select a subset of sequences that display broad similarity to the query sequence. The second phase, called *fine search*, is a computationally more expensive step, which ranks the resultant sequences from the coarse search in order of local alignments to the query. The *CAFE* system is shown to be superior to BLAST in speed and its retrieval effectiveness is also comparable to that of BLAST (which is a collection of the most commonly used search tools for biological sequences).

Independently, in the field of computer science, “information retrieval” systems have been shown to be effective search tools for large text databases (Salton & McGill, 1983; Witten *et al.*, 1999). “Information retrieval” is a problem of retrieving relevant documents to a query from a collection of documents. This research has been used in many full-text indexing systems and Internet search engines successfully. On the other hand, biological database search tools retrieve a set of sequences, which is similar to a query sequence from the biological sequence databases. Therefore, the concepts and practical approaches of searching similar sequences from biological sequence databases and retrieving relevant documents from text databases can be practically identical, as shown practically in the case of *CAFE* system. Regarding biological sequences as texts written in DNA base or amino acid codes, we expect that one can implement an information retrieval system for biological sequence database if appropriate indexing techniques are used. As a beginning, in this report, we suggest indexing schemes for protein sequences based on n-gram method and present their retrieval speed and effectiveness for the search of the PIR-NREF database.

2 SYSTEMS AND METHODS

2.1 System Environments

We implemented a simple information retrieval system dedicated to protein sequence databases with C++ STL(Standard Template Libraries) and an embedded database system called Berkeley DB (Loverso & Seltzer, 2002). Our system, named as ProSeS – of which name stands for *Pro*tein *Se*quence *S*earch – extracts n-gram indexes from protein sequence data written in FASTA format and stores the indexes into B+ tree provided by Berkeley DB system. Details of sequence searching based on n-gram indexing will be described in the later section **2.4 Sequence Similarity Measure**.

We used a desktop Linux machine with dual CPUs of Pentium-IV 2.4GHz, three giga-bytes of system memory, and an Ultra-160 SCSI hard drive. All the tests were carried out while the machine was under light load.

2.2 Test Data

For biological sequence databases, it is difficult to build proper test collections to measure the retrieval effectiveness. In a research conducted by Williams *et al*. (2002), the PIR super-family database was chosen to be the test database to measure the retrieval effectiveness. The super-family classification in the PIR database is mainly based on multiple alignments, i.e., *global* alignments of sequence sets (Wu *et al.*, 2003). However, since the sequence searches are mostly focused on local similarities (Altschul *et al.*, 1997), it is inappropriate to use the PIR super-family database as test data to measure the retrieval effectiveness. Thus, we chose the PIR-NREF database (Wu *et al.*, 2002) as a test collection and compared our system's performance with that of BLASTP. BLASTP, a protein sequence search program of BLAST tool collection, searches protein sequences that show significant local alignments with respect to a query protein sequence (Altschul *et al.*, 1997). At the time of our test, the PIR-NREF database Release 1.26 was used and the database contains 1.27 million sequence entries with average length of 317 amino acids and 405 million amino acids in total.

2.3 N-gram Indexing

Documents, especially those written in western languages, are made of words or terms that can be separated by blanks. Usually, in information retrieval systems, these terms extracted from the documents are stored in inverted files (Salton & McGill, 1983; Witten *et al.*, 1999). However, biological sequences such as DNA and protein sequences are strings without any blanks. Furthermore, it is difficult to distinguish one meaningful segment to another in a sequence string. Some suitable heuristic indexing methods have been designed for such cases and n-gram token method is one of them. Indexing documents

with n-grams has been successfully applied to several text collections such as Chinese texts (Wilkinson, 1998) and OCR texts (Harding *et al.*, 1997). These texts are also built in a manner that word boundaries are absent or ambiguous. As mentioned above, biological sequences also have no "word" boundaries and therefore n-gram indexing will be an appropriate choice.

N-gram is just another nomenclature of "k-tuple" used in FASTA literature (Lipman & Pearson, 1985) and "w-mer" used in BLAST literature (Altschul *et al.*, 1997). Thus n-grams can be defined as intervals occurring in each sequence, where the intervals are overlapping sub-strings of some fixed-length *n* (Williams & Zobel, 2002). For example, if *n*=4 and regarding a protein sequence of ACEPITCH then the final n-grams are ACEP, CEPI, EPIT, PITC, and ITCH.

Table 1. Features of four N-grams

Symbol	Length	Alphabet #	Term #
N3-A20	3	20	8,000 (20^3)
N4-A20	4	20	160,000 (20^4)
N5-A20	5	20	3,200,000 (20^5)
N6-A18	6	18	34,012,224 (18^6)

We chose *n* to be 3, 4, 5, and 6, of which characteristics are shown in Table 1, where the 'Length' indicates the fixed-length of an interval sub-string, 'Alphabet #' means the number of distinguishable amino acid codes used for indexing, and 'Term #' is the maximum number of unique terms that can be occurred in a protein sequence database.

A protein sequence is a string of combined 20 amino acid codes[1]. For tri-gram(*N3-A20*), tetra-gram(*N4-A20*), and penta-gram(*N5-A20*), the 20-character alphabet is used to segment n-grams from sequences. However, for hexa-gram(*N6-18*), an 18-character alphabet is used. If the 20-character alphabet is applied, theoretical term space will contain 64 million unique words, which could not be handled in our system memory. In order to resolve this problem, we merged two pairs of amino acids to two amino acid codes – (V, I) to I, and (F, Y) to Y – that show the highest score in BLOSUM62 scoring matrix (Altschul, 1991), resulting the 18-amino acid codes and term space of about 34 million words, which then could be processed in memory.

[1] In fact, the number of amino acids is 21. However, the total frequency of an amino acid (selenocysteine) in the PIR-NREF database is extremely low so that it can be ignored. There are three additional wild card characters such as B, Z, and X (with low frequencies) and they are also disregarded in this work.

N-grams are stored into an inverted index implemented on the top of Berkeley DB system. Each entry of this inverted file is composed of a searchable key and its postings lists. For example, consider the following entry in the inverted file.

"ACEP" 36(2), 127(3), 1074(1), ...

This means that the tetra-gram ACEP occurs twice in the 36th sequence, three times in the 127th, once in the 1,074th, and so on. To reduce disk operations in using an index for retrieval, the postings lists are compressed with a general compression algorithm, *gzip*.

2.4 Sequence Similarity Measure

There are several models that calculate the similarities between the query and target documents in the field of information retrieval. Among them, the vector space model is one of the best studied and the most widely used schemes (Witten *et al.*, 1999). In this model, the query and the target documents are represented as vectors of unique terms, each of which stands for a dimension in a high-dimensional space. The similarity between the query and a document is calculated by the inner product of their representative vectors divided by a normalization factor related to the document length. The similarity measure (*Sim(q,d)*) between a query sequence *q* and target sequence *d* is defined to be

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

with:

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

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

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

where $f_{q,t}$ is the frequency of n-gram token *t* in query sequence *q*; $f_{d,t}$ is the frequency of n-gram token *t* in target sequence *d*; *N* is the total number of sequences in the database; f_t is the number of sequences where the n-gram token *t* occurs more than or equal to once; $w_{q,t}$ means the weight of token *t* in the query sequence *q*; $w_{d,t}$ means the weight of token *t* in the target sequence *d*; and W_d represents the length of the target sequence *d*.

2.5 Evaluation

From the PIR-NREF database, 100 protein sequences were randomly selected as test query sequences resulting average length of 316 amino acids and length range of 51 to 1,609 amino acids. Detailed information about these test-queries and their BLAST result against the PIR-NREF database can be obtained at our web site. We ran BLAST with each of these sequences and retrieved at most 1,000 sequences that show local homologies with the input sequence. In addition E-value was set to 0.0001 and other parameters were set to default. These results are regarded as the reference test set to measure the retrieval effectiveness of the n-gram indexing methods. Then we ran our system with same test queries and also retrieved at most 10,000 homologous sequences from the PIR-NREF database. To quantify the relative performance of each n-gram indexing method, the measures of recall and precision against BLAST result set were used. Recall and precision are commonly used measures to demonstrate the retrieval effectiveness of information retrieval systems (Salton & McGill, 1983). In this paper, precision(p) is the measure of the fraction of the relevant BLAST answers retrieved at a particular point, that is

$$p = \frac{\text{BLAST sequences retrieved}}{\text{BLAST and non - BLAST sequences retrieved}}$$

In contrast, recall(r) is the fraction of the relevant BLAST answers against total BLAST answers at a particular retrieval point, that is

$$r = \frac{\text{BLAST sequences retrieved}}{\text{Total number of BLAST sequences}}$$

The relative retrieval effectiveness of four n-gram methods will be shown in mean recall-precision plots, with precisions measured at 11 points of recall, *i.e.* 0.0, 0.1, 0.2, 0.3, ..., 1.0, for all query sequences and averaged at each point(Figure 1). Also values of 11-pt average recall-precision measures, averaged precision over 11 recall points, are given for clarity to compare the relative effectiveness.

3 RESULTS AND DISCUSSION

3.1 Retrieval Effectiveness

Figure 1 shows mean recall-precision plots for four indexing types used in this experiment. Results are shown for the PIR-NREF database with 100 sequence query test set and their relevant sequences judged

by BLAST tool as described in the previous section **2.5 Evaluation**. The results show that hexa-gram(*N6-A18*) and penta-gram(*N5-20*) worked best among four indexing types. However, the retrieval effectiveness of tri-gram(*N3-A20*) was very poor, while that of tetra-gram(*N4-20*) was in between. The 11-point average recall-precision values for N3-A20, N4-A20, N5-A20, and N6-A18 n-grams were calculated to be 0.1376, 0.5038, 0.6342, and 0.6337, respectively. Compared with the 11-point average recall-precision within the range of 0.25 to 0.40 for general text retrieval (Voorhees, 2002; Oard & Gey, 2002), the value of around 0.63 for N5-A20 and N6-A18 is significantly high enough to draw an inference that searching only with penta-gram or hexa-gram index can match for BLAST.

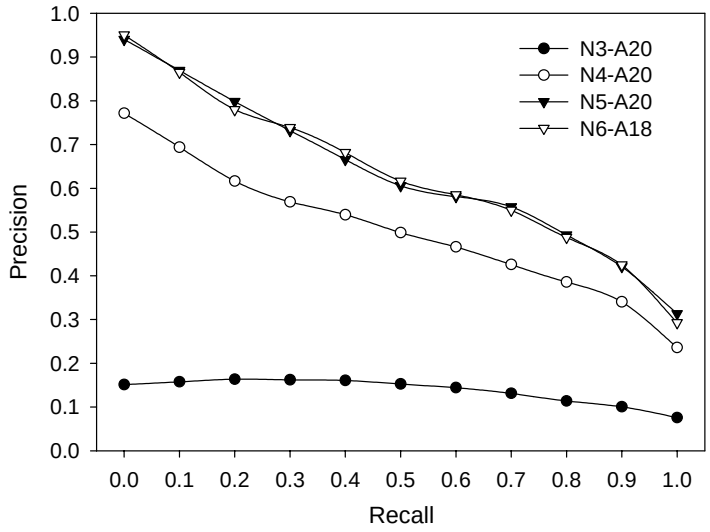

Figure 1. Mean 11-point recall-precision plots for four n-gram index methods. We used 100 protein sequences chosen randomly from the PIR-NREF database as test query set. Precision was measured at particular recall points from 0.0 to 1.0 at the interval of 0.1 against BLAST result sequences. Mean precisions for 100 query sequences at 11 recall points are shown.

Figure 1 shows a trend that as the interval length of n-gram grows, the retrieval effectiveness also increases, except for the case of hexa-gram(N6-A18). Note that N6-A18 indexing scheme used the 18-character alphabet while the other three schemes used the 20-character alphabet set. If we could have implemented an indexing by hexa-gram with the 20-character alphabet, we might have seen some increase in the retrieval effectiveness. However, the increment may be expected to be relatively small, from the fact that the difference of 11-point average recall-precision between penta-gram and tetra-gram is 2.8 times smaller than that of tetra-gram and tri-gram(*see* Figure 1 and Table 2). It is possibly because as the length of n-gram grows, each interval's information content also increases, but the longer the interval, the more chances to miss potential alignments in small areas less than the interval length.

Therefore, we suppose that the length of n-gram exceeding 6 may be harmful to the retrieval effectiveness.

Note that mean precision at recall 0.1 for penta-gram is 0.869. This means that ProSeS gives highly similar results to BLAST output at lower recall level. The average size of the test set generated by BLASTP for 100 query sequences is about 335 sequences. At recall 0.1, where about 34 sequences (≈10% of 335 sequences) are retrieved, ProSeS with penta-gram gives about 30 sequences (≈86.9% of 34 sequences at recall 0.1) identical to BLASTP program.

Further analysis of the ProSeS and BLASTP results showed that the number of perfect matches for recall points 0.1, 0.4, 0.7, and 1.0 are 73, 51, 38, and 17, respectively. This means that ProSeS and BLASTP give exactly same results for 73 query sequences at recall 0.1, 51 at recall 0.4, and so on. And 9 query sequences have very poor precision below 0.1 at recall 0.1. For one of them, the 97th query sequence, BLASTP gives 20 sequences from the PIR-NREF database, while ProSeS retrieves only 6 of them in the result pool.

```
>NF01265541 Similar to 5'-nucleotidase, cytosolic III (Fragment) [Xenopus
          laevis]
         Length = 294

 Score =  127 bits (320), Expect = 1e-28
 Identities = 96/288 (33%), Positives = 127/288 (44%), Gaps = 79/288 (27%)

Query: 15  PRALTDKMTLIRDAGPSKFQVF-----------------PTP---------ISEQGDAYY 48
           P  L DK+T I+   G  K Q+                   PT          IS++G
Sbjct: 20  PEGLQDKITRIQRGGQEKLQIISDFDMTLSRFSRNGERCPTCYNIIDNSNIISDEGRK-- 77

Query: 49  DAKRQALYDHYHPLEISPVIPIDEKTKLMEEWWGKTHELLIEGGLTYDAIKKSVANSSIA 108
             K + L+D Y+PLEI P    I+EK  LM  EWW K H+L  E  +   D + + V  S
Sbjct: 78  --KLKCLFDIYYPLEIDPKKSIEEKYPLMVEWWSKAHDLFYEQRIQKDRLAQVVKESQAT 135

Query: 109 FREGVSELFEFLEKKEIPVLIFSAGLADVIEEVTLKSISLLELLSYFCCLYNEYAFVAYS 168
            R+G    F   L ++EIP+ IFSAG+ DV+EE
Sbjct: 136 LRDGYDLFFNSLYQREIPLFIFSAGIGDVLEE---------------------------- 167

Query: 169 HSYQVLRQNLDRTFKNVKIVSNRMVFNDDGQLVSFKGKLIHVLNKNEHALDMAAPLHDRL 228
               ++RQ        N K+VSN M F+D+G L  FKG LIH   NKN   L
Sbjct: 168 ----IIRQ-AGVFHPNTKVVSNYMDFDDNGILTGFKGDLIHTYNKNSSVL---------- 212

Query: 229 GVDIGEEDEENVNMKERRNVLLMGDHLGDLRMSDGLD-YETRISIGFL 275
                ++ E    +   R N+LL+GD LGDL M+DG+   E  I IGFL
Sbjct: 213 -----KDTEYFKEISHRTNILLLGDTLGDLTMADGVSTVENIIKIGFL 255
```

Figure 2. The second entry of BLAST output for the 97th query sequence, "NF00667350 hypothetical protein At2g38680 [Arabidopsis thaliana]". This sequence is not retrieved by ProSeS with the same query.

Figure 2 shows the second BLASTP match for the 97th query sequence, which is not retrieved with ProSeS system using penta-gram indexing. Carefully observing the alignment in Figure 2, one can find that only one identical penta-gram, IFSAG starting at the 129th position of the query, exists in both the

query and the subject sequence. The similarity measure calculated by vector space model in ProSeS should be very small for this sequence since only one index term is matching for both the query and target sequence. Due to this small similarity value, this sequence might be pushed to have worse rank, which resulted in missing this sequence from the ProSeS result set.

On the other hand, Figure 3 shows the third BLASTP match for the 97th query, which is also retrieved by ProSeS. There are two identical penta-grams in this alignment, IFSAG starting at the 129th position of the query and GDLRM at the 256th position. For ProSeS, this sequence should have better similarity value than the sequence in Figure 2, since this sequence has two index terms in common with the query sequence. Due to this better similarity value, this sequence might rank better than the sequence in Figure 2, leading this sequence into the ProSeS result set.

```
>NF01178199 10 days embryo whole body cDNA, RIKEN full-length enriched
         library, clone:2610024B13 product:HSPC233 (PYRIMIDINE
         5'-NUCLEOTIDASE) (EC 3.1.3.5) (URIDINE 5' MONOPHOSPHATE
         HYDROLASE 1) (SIMILAR TO HYPOTHETICAL PROTEIN) homolog
         [Mus musculus]
        Length = 331

 Score =  124 bits (310), Expect = 2e-27
 Identities = 78/222 (35%), Positives = 118/222 (53%), Gaps = 49/222 (22%)

Query: 55  LYDHYHPLEISPVIPIDEKTKLMEEWWGKTHELLIEGGLTYDAIKKSVANSSIAFREGVS 114
           L + Y+ +E+ PV+ ++EK    M EW+ K+H LLIE G+    +K+ VA+S +   +EG
Sbjct: 122 LKEQYYAIEVDPVLTVEEKFPYMVEWYTKSHGLLIEQGIPKAKLKEIVADSDVMLKEGYE 181

Query: 115 ELFEFLEKKEIPVLIFSAGLADVIEEVTLKSISLLELLSYFCCLYNEYAFVAYSHSYQVL 174
            LF   L++  IPV IFSAG+ DV+EEV  ++                        HS
Sbjct: 182 NLFGKLQQHGIPVFIFSAGIGDVLEEVIRQA--------------------GVYHS---- 217

Query: 175 RQNLDRTFKNVKIVSNRMVFNDDGQLVSFKGKLIHVLNKNEHALDMAAPLHDRLGVDIGE 234
                    NVK+VSN M F+++G L   FKG+LIHV NK++ AL             +
Sbjct: 218 ---------NVKVVSNFMDFDENGVLKGFKGELIHVFNKHDGAL---------------K 253

Query: 235 EDEENVNMKERRNVLLMGDHLGDLRMSDGL-DYETRISIGFL 275
               +   +K+  N++L+GD  GDLRM+DG+  + E  +  IG+L
Sbjct: 254 NTDYFSQLKDNSNIILLGDSQGDLRMADGVANVEHILKIGYL 295
```

Figure 3. The third entry of BLAST output for the 97th query sequence, “NF01178199 10 days embryo whole body cDNA ... [Mus musculus]”. This sequence is also retrieved by ProSeS with the same query.

BLASTP ranks the two sequences in Figure 2 and Figure 3 as the 2nd and the 3rd, respectively. However, ProSeS does not retrieve the sequence in Figure 2 but does retrieve the sequence in Figure 3. From this fact, it is clear that there is some difference between the similarity measure of ProSeS and local alignment of BLASTP. This imposes a need for further research on refinement of the vector space model used in ProSeS. We hope that new similarity measure appropriate for protein sequences will be studied in near future.

Considering the result for the retrieval effectiveness, we suggest that penta-gram is the most suitable indexing scheme for protein sequences. In addition, in the case of penta-gram indexing, search speed for the PIR-NREF database is the fastest among n-gram methods tested in this work, which will be described in the next section.

3.2 Space and Speed

Table 2 shows the index sizes in Mbytes, indexing times in hours and average retrieval times in seconds tested for the PIR-NREF database. The 11-point average recall-precision measures are shown again for referring purpose. Reminding that there are about 404,532,594 amino acids in protein sequences of the PIR-NREF database Release 1.26, the smallest and the largest index sizes are 3.7 times and 14.0 times bigger than the size of total sequences, respectively. In the case of penta-gram(N5-A20) which performs best in the sequence retrieval, its index size is 7.6 times bigger than the sequence collection size. Though current storage technology is developing rapidly, but considering the sizes of protein sequence collections are also growing exponentially, these are relatively large storage overheads. By introducing techniques such as index compression for nucleotide databases (Williams & Zobel, 1997) and index *stopping* which discards high-frequency n-grams from the index (Williams & Zobel, 1996), we expect that the index size of ProSeS system can be further reduced to an acceptable level.

The indexing time is proportional to the dimension of term space as shown in Figure 4 (*also refer* to Table 2). This is, the longer unique n-gram tokens in the database, the longer the indexing time. In other words, the indexing time is exponentially proportional to the length of n-gram intervals. Though indexing step requires a few hours, it is acceptable time since indexing is a system administrator's job which is executed only once prior to many users' sequence retrieval processes. Furthermore, by carefully designing the memory structure for inverted lists, rather than using C++ STL libraries, indexing time can be tremendously saved to the level of several to dozens of minutes(*data not shown*).

Table 2. The index size, indexing time, retrieval time, and 11-point average recall-precision for four n-grams and BLAST. The best performing values in retrieval speed and 11-point precision are underlined.

N-gram	Inverted lists (MB)	Indexing time (hr)	Retrieval time (sec)	11-point precision
N3-A20	1,514	16.70	1.91	0.1376
N4-A20	3,125	2.40	3.88	0.5038
N5-A20	3,075	3.97	**1.17**	**0.6342**
N6-A18	5,655	8.94	7.61	0.6337
BLAST	583	0.05	44.10	-

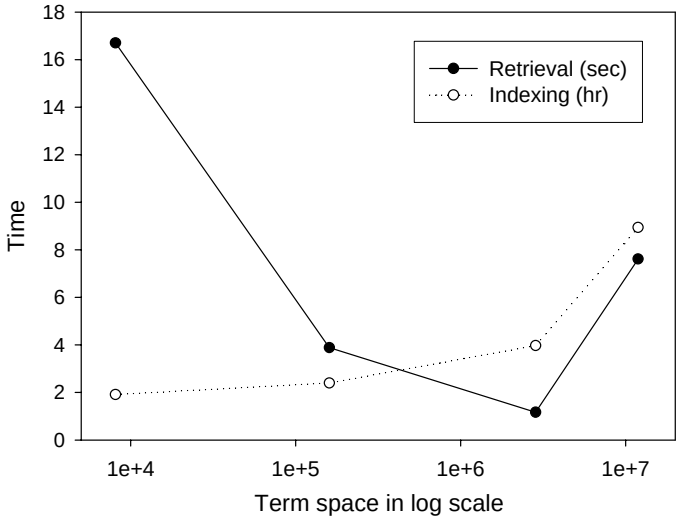

Figure 4. The indexing time and retrieval speed plotted against term space dimension in log scale. The circle marks are for N3-A20, N4-A20, N5-A20, and N6-A18 sequentially from left to right, and their real dimensions of term space are 8.0 **x** 10^3, 1.6 **x** 10^4, 2.8 **x** 10^6, and 1.2 **x** 10^7, respectively. The indexing time is measured in hour to index whole sequences in the PIR-NREF database. The retrieval speed is measured to retrieve 100 query sequences and divided by 100, which is the average search time for each sequence.

On the other hand, the retrieval speed shows an interesting trend. As the interval length of n-gram grows from 3 to 5, the retrieval speed is inversely proportional to the length (Figure 4). However, at the point of 6, the retrieval speed is abruptly increasing. This might be caused by the overhead of index structure to store about 10 times more unique index terms than those of penta-gram. Among four indexing schemes, penta-gram shows the best search speed of average 1.17 seconds per one query sequence (Table 2). Compared with 44.1 seconds in BLASTP, it is about 38 times faster. We again suggest the penta-grams as the most efficient indexing scheme for protein sequences.

4 CONCLUSION

We have shown that protein sequences can be retrieved by indexing protein databases with n-gram methods, and penta-gram showed the best retrieval speed and effectiveness. Though traditional search model for n-gram indexing is still powerful, it is required to fill the gap between local alignment and our search model by further refining our vector space model to be suitable for protein sequences. An additional advantage of n-gram indexed system is its applicability to many other systems that is similar to

or based on information retrieval. For example, ProSeS system provides a function called *keyword suggestion* based on simple data mining method. It also presents a list of predicted *subcellular localization* sites based on text categorization algorithm.

Exhaustive comparison systems are in the face of severe delay in search time. As biological databases grow exponentially, exhaustive systems will be impractical as sequence retrieval tools, in the near future. We hope that index-based sequence retrieval systems such as **ProSeS** will be one of practical alternatives to exhaustive biological database search tools.

5 ACKNOWLEDGMENTS

The first author thanks for GIIS group's helpful conversations on this topic. This work was supported in part by IBM Korea's Research Funds.

6 REFERENCES

Altschul, S.F. (1991) Amino Acid Substitutions Matrices from an Information Theoretic Perspective. *J. Mol. Biol.*, **219**, 555-665

Altschul, S.F., Madden, T.L, Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Research*, **25**, 3389-3402

Harding, S.M., Croft, W.B. and Weir, C. (1997) Probabilistic Retrieval of OCR Degraded Text Using N-Grams. *European Conference on Digital Libraries*, 1997, pp 345-359

Lipman, D.J. and Pearson, W.R. (1985) Rapid and Sensitive Protein Similarity Searches. *Science*, **227**, 1435-1441

Loverso, S. and Seltzer, M. (2002) Tree Houses and Real Houses: Research and Commercial Software, *Proceedings of the 2nd Workshop on Industrial Experiences with Systems Software* (WIESS '02), Berkeley, CA: USENIX Association, December 2002, pp 55-66

Oard, D.W and Gey, F.C. (2002) The TREC 2002 Arabic/English CLIR Track. *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference* (TREC 2002), 2002

Salton, G. and McGill, M.J. (1983) *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983

Voorhees, E.M. (2002) Overview of TREC 2002. *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference* (TREC 2002), 2002

Wilkinson, R. (1998) Chinese Document Retrieval at TREC-6. *NIST Special Publication 500-240: The Sixth Text REtrieval Conference* (TREC 6), pp 25-29

Williams, H.E. and Zobel, J. (1996) Indexing Nucleotide Databases for Fast Query Evaluation. *In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin, Proc. International Conference on Advances in Database Technology* (EDBT), Avignon, France, 275-288

Williams, H.E. and Zobel, J. (1997) Compression of Nucleotide Databases for Fast Searching. *Computer Applications in the Biosciences*, **13**, 549-554

Williams, H.E. and Zobel, J. (2002) Indexing and Retrieval for Genomic Databases. *IEEE Transactions on Knowledge and Data Engineering*, **14**, 63-78

Witten, I.H., Moffat, A., and Bell, T.C. (1999) *Managing Gigabytes: Compressing and Indexing Documents and Images*, The Second Edition, Morgan Kaufmann Publishing, San Francisco, 1999

Wu, C., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z., Ledley, R.S., Lewis, K.C., Mewes, H.-W., Orcutt, B.C., Suzek, B.E., Tsugita, A., Vinayaka, C.R., Yeh, L.-S., Zhang, J. and Barker, W.C. (2002) The Protein Information Resource: An Integrated Public Resource of Functional Annotation of Proteins. *Nucleic Acids Research*, **30**, 35-37

Wu, C.H., Huang, H., Yeh, L.-S.L., and Barker, W. C. (2003) Protein Family Classification and Functional Annotation. *Computational Biology and Chemistry*, **27**, 37-47