# A Compressing Method for Genome Sequence Cluster using Sequence Alignment

Kwang Su Jung[1], Nam Hee Yu[1], Seung Jung Shin[2], Keun Ho Ryu[1†]
[1]*Database/Bioinformatics Laboratory, Chungbuk National University, Korea*
[2]*Divison of IT, Hansei Univiersity, Korea*
[1]*{ksjung,nami,khryu}@dblab.chungbuk.ac.kr,* [2]*expersin@hansei.ac.kr*

## Abstract

*After identifying the function of a protein, biologists produce new useful proteins by substituting some residues of the identified protein. These new proteins have high sequence homology (similarity). We define a sequence cluster as a cluster that is constituted of similar sequences. As another example of a Sequence Cluster, we consider a SNP (Single Nucleotide Polymorphism) Cluster. A SNP is a DNA sequence variation occurring when a single nucleotide in the genome (or other shared sequence) differs between members of a species (or between paired chromosomes in an individual). We suggest a new compressing technique for these sequence clusters using a sequence alignment method. We select a representative sequence which has a minimum sequence distance in the cluster by scanning distances of all sequences. The distances are obtained by calculating a sequence alignment score. The result of this sequence alignment is utilized to author conversion information called an Edit-Script between the two sequences. We only stored representative sequences and Edit-Scripts of each cluster into a database. Member sequences of each cluster can then be easily created using representative sequences and Edit-Scripts.*

## 1. Introduction

When designing and producing a useful protein, biologists use a well-know protein which is utilized as a target and a template protein. After identifying the function of a protein, biologists produce new useful proteins by substituting some residues of the identified protein. In the case of a DNA (Deoxyribonucleic Acid) sequence, biologists select nucleic acid to substitute. From this substitution, they then synthesize a new protein. These new proteins or DNA sequences have high sequence homology (similarity). We define a sequence cluster as a cluster which is constituted of similar sequences. As another example of a sequence cluster, we consider a SNP (Single Nucleotide Polymorphism) Cluster. A SNP is a DNA sequence variation occurring when a single nucleotide in the genome (or other shared sequence) differs between members of a species (or between paired chromosomes in an individual).

We suggest a new compressing technique for these sequence clusters using the Smith-Waterman [13] sequence alignment method. We select a representative sequence which has a minimum sequence distance (the Smith-Waterman alignment score) in a cluster by scanning the distances of all sequences. The distances are obtained by calculating the sequence alignment result. Specific substitution matrices for the DNA sequence and protein sequence are applied to score the alignment. The result of the sequence alignment is utilized to make conversion information named the Edit-Script between the two sequences. We, then, only store representative sequences and Edit-Scripts for each cluster into a database. Member sequences of each cluster are easily created using representative sequences and Edit-Scripts. This work can be adapted to any sequence clusters which have a high sequence similarity.

## 2. Related work

Sequence alignments are aligning the sequences of nucleic acid or protein in order to indicate the relationship among sequences. The homology of sequences is well presented. These sequences are classified as having pair-wise or multiple alignments according to the number of sequences at once. Pair-wise alignment [8, 10, 13, 17, 18] is aligning two sequences at once and in case of more than two

[†] Corresponding Author

978-1-4244-2358-3/08/$20.00 © 2008 IEEE

sequences at once, we have multiple sequence alignment [3, 4, 7, 9, 14, 15, 16], respectively. Also, sequence alignments are categorized as either global or local alignment according to the type of homology. When comparing two sequences such that these sequences are the same type as the DNA sequences or amino acid sequences, if we want to get a maximum homology, then we align two whole sequences. We could say that this type of homology is that of global similarity. This alignment is called global alignment [10, 12]. When predicting 3D structures of unknown protein from a sequence, global alignment is used to select the target template of the protein. On the other hand, if we want to know which part of the sequences is similar when aligning, local homology is adopted and we call this alignment local alignment [1, 2, 11, 13]. It is effective to use local alignment when finding the functional sharing part of sequences.

Current sequence alignment algorithms [1, 2, 11] as well as heuristic approaches to scan whole databases are fast but these approaches have low accuracy. The sequence alignment algorithms in an early stage using dynamic programming [10, 11, 12] and have high accuracy. These algorithms are not acceptable for whole database scanning when the database has a large number of entries. Compression of a sequence cluster does not need fast response unlike a database search. The number of sequences in a cluster is significantly fewer than entries in the whole database. The algorithms [10, 13] which have higher accuracy can generate shorter Edit-Scripts. When the sequences in a cluster are globally similar, however during alignment, global alignment yields a number of sequence gaps which makes Edit-Scripts longer because global alignment has a propensity to make the length of two sequences the same with many gaps. The character, hyphen (-), is used to express the gaps. Therefore, the Smith-Waterman algorithm [13] is quite suitable for a sequence cluster compression.

## 3. The proposed method

We suggest a new compression method to reduce sequence cluster size. To achieve this, we first select the representative sequence of a cluster. In this procedure, we calculate the average sequence distance of all member sequences with a Smith-Waterman alignment score. The specific substitution matrices for nucleic acid and amino acid (BLOSUM62 [6]) are utilized when scoring. We create an Edit-Script, ε, from the Smith-Waterman alignment results. The changed information is written as an Edit-Script. Only a representative sequence of a cluster and each Edit-Script of a member sequence are stored into the database. In this section, we explain these procedures in detail.

### 3.1. The representative sequence of a cluster

The purpose for which we select the representative sequence of a cluster is to find which sequence in the cluster makes the shortest Edit-Script. For this, we need to choose the sequence close to the center of the cluster. This work is achieved by calculating each distance, d, in Figure 1. We assume that a sequence cluster consists of genome sequences which have high sequence similarity (homology). Figure 1 shows an example of a sequence cluster, SC = $\{s_0,s_1,s_2,s_3,\ldots,s_n\}$, where $S_n$ denotes the member sequences of the cluster and n denotes the number of sequences in the cluster.

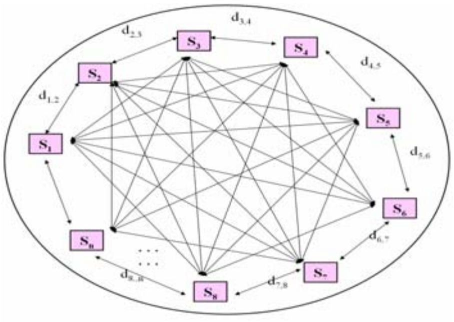


Figure 1. A Sequence Cluster.

We then calculate the average distance of each member sequence $S_n$. Each sequence in the cluster has (n-1) distances to other sequences. The average distance is obtained from a summation of the total distances of one member sequence by the following Equation (1) where $d_{i,j}$ denotes the distance between sequences $S_i$ and $S_j$, and $S_0$ denotes the current member sequence. For simplifying Equation (1), if we substitute "$d_{s0,si}$" to "$d_{si}$", we finally get Equation (2).

$$\frac{\sum_{i=1}^{n-1}(d_{s_0,s_1} + d_{s_0,s_2} + d_{s_0,s_3} + d_{s_0,s_4} + \cdots + d_{s_0,s_n})}{n-1} \quad (1)$$

$$\text{Average Distance of } S_0 = \frac{\sum_{i=1}^{n-1} d_{s_i}}{n-1} \quad (2)$$

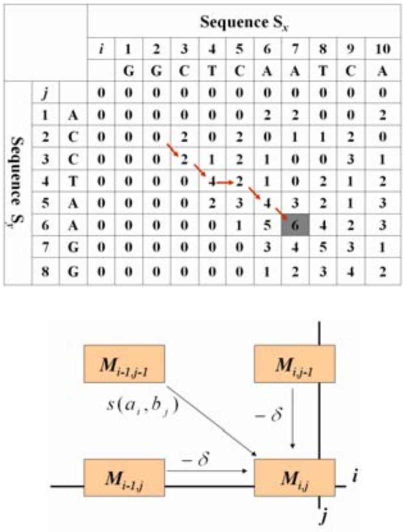
The representative sequence of the cluster means the member sequence Si which has minimum average distance. A detailed method to calculate the distance between sequences is explained in the next section.

### 3.2. The sequence distance

The Euclidean distance is widely used when calculating distance between objects in a common cluster. In the case of a genome sequence, applying the Euclidean distance is not acceptable because sequences do not have any absolute position. Thus, sequence alignment methods are employed to calculate the distance between sequences. We use the Smith-Waterman alignment score [13] to get sequence distances. The reason we adopt the Smith-Waterman alignment method is explained in the related work, Section 2. Now, we discuss how the Smith-Waterman alignment method is applied in detail.

$$M_{i,j} = \max\begin{cases} M_{i-1,j-1} + s(a_i,b_j) \\ M_{i-1,j} - \delta \\ M_{i,j-1} - \delta \\ 0 \end{cases}, \quad \begin{array}{l} \mathrm{s(a_i,b_j) = +2, if\ a_i = b_j} \\ \mathrm{s(a_i,b_j) = -1, if\ a_i \neq b_j} \\ \delta = \mathrm{s(a_i,-) = s(-,b_j) = -2} \end{array} \qquad (3)$$

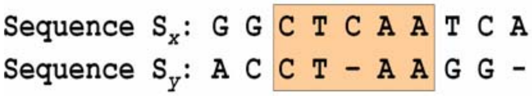
Given sequences, $S_x$={GGCTCAATCA} and $S_y$={ACCTAAGG}, Figure 2(a) shows the alignment matrix with dynamic programming. Each value of the cell, $M_{i,j}$ is obtained from Equation (3). Here, each cell has three candidate values: the score from above, the left and the diagonal. Figure 2(b) denotes these three candidate values. The scores, $s(a_i,b_j)$, for matches and mismatches are +2 for a match and -1 for a mismatch, respectively. The gap penalty score $\delta$, $s(a_i,-)$, and $s(-,b_j)$ are the same, -2, when the score comes from above or the left cell value. If a cell value. $M_{i,j,}$ falls below zero, the score is replaced by zero.



| | | | Sequence $S_x$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | *i* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | | | | G | G | C | T | C | A | A | T | C | A |
| Sequence $S_y$ | *j* | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | A | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 2 |
| | 2 | C | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 1 | 1 | 2 | 0 |
| | 3 | C | 0 | 0 | 0 | 2 | 1 | 2 | 1 | 0 | 0 | 3 | 1 |
| | 4 | T | 0 | 0 | 0 | 0 | 4 | 2 | 1 | 0 | 2 | 1 | 2 |
| | 5 | A | 0 | 0 | 0 | 0 | 2 | 3 | 4 | 3 | 2 | 1 | 3 |
| | 6 | A | 0 | 0 | 0 | 0 | 0 | 1 | 5 | 6 | 4 | 2 | 3 |
| | 7 | G | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 5 | 3 | 1 |
| | 8 | G | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 2 |

Figure 2. The smith-waterman algorithm.

For example, cell $M_{7,6}$ has three candidate values where 4 + 2 = 6 from the diagonal, 5 - 2 = 3 from the left, and 3 - 2 = 1 from above. Among these three values, the value from diagonal (6) is assigned because it is maximum. Every cell value in Figure 2(a) memorizes the direction where the cell value comes from. The directions of $M_{i,j}$ are displayed in Figure 2(a) as arrows. For clarity, we have included only the arrows around the highest-scoring path. These arrows are mandatory when tracing the alignment path. The best local alignment is the one that ends in the matrix element having the highest score ($M_{7,6}$=6). Arrows from the left or above denote gaps (insertion and deletion) which are represented by a hyphen (-) in Figure 3. The highlighting areas, $S_x \supset$ {CTCAA} and $S_y \supset$ {CT-AA}, are locally similar. Calculating the score for this alignment is shown in the next section.

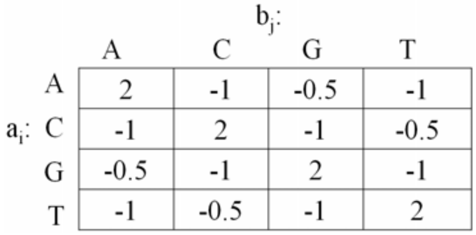


Sequence $S_x$: G G C T C A A T C A
Sequence $S_y$: A C C T - A A G G -

Figure 3. The result of smith-waterman algorithm

### 3.3. The soring matrix

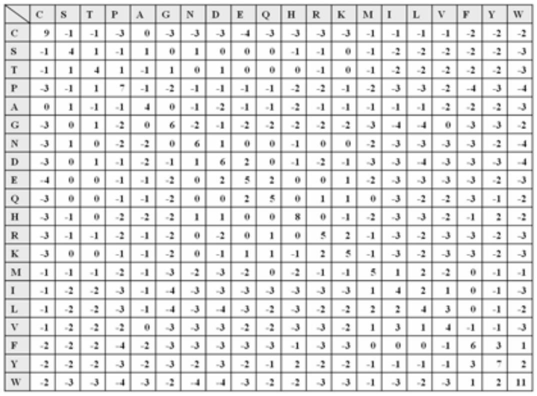
We have used alignments of nucleic acids as examples. One very important application of alignment is protein alignment. The scoring matrix for amino acids is much more complicated than that of nucleic acids (DNA : 4 characters, Protein : 20 Characters). Figure 4 indicates the scoring matrix for a DNA sequence. This scoring matrix contains the assumption that aligning A with G is not just as bad as aligning A with T because studies of mutations in a homologous gene indicate that transition mutations (A→G, G→A, C→T, or T→C) occur approximately twice as frequently as do transversions (A→T, T→A, A→C, G→T). Thus, the alignment score for the Figure 3 alignment is 0.5 (-0.5-1+2+2-2+2+2-1-1-2) whereas the score for the match is +2. The score for the gap is -2 and the score for the mismatch is -1 or -0.5 shown in Figure 4. In our proposed method, this score is called the sequence distance.



| | | $b_j$: | | | |
|---|---|---|---|---|---|
| | | A | C | G | T |
| | A | 2 | -1 | -0.5 | -1 |
| $a_i$: | C | -1 | 2 | -1 | -0.5 |
| | G | -0.5 | -1 | 2 | -1 |
| | T | -1 | -0.5 | -1 | 2 |

Figure 4. The scoring matrix for DNA sequence

Figure 5 denotes the BLOSUM62 substitution matrix for amino acids which is commonly used. BLOSUM (BLOcks SUbstitution Matrices) matrices [6] are based on aligned protein sequence blocks without assumptions about mutation rates. Different levels of the BLOSUM matrix can be created by differentially weighting the degree of similarity between sequences. For example, a BLOSUM62 matrix is calculated from protein blocks such that if two sequences are more than 62% identical, then the contribution of these sequences is weighted to sum to one. In this way, the contributions of multiple entries of closely related sequences are reduced. We applied BLOSUM 35, 45, 62, and 80 matrices in selecting a representative sequence, but the representative sequence was not changed according to the types of BLOSUM matrix.



| | C | S | T | P | A | G | N | D | E | Q | H | R | K | M | I | L | V | F | Y | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 9 | -1 | -1 | -3 | 0 | -3 | -3 | -3 | -4 | -3 | -3 | -3 | -3 | -1 | -1 | -1 | -1 | -2 | -2 | -2 |
| S | -1 | 4 | 1 | -1 | 1 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | 0 | -1 | -2 | -2 | -2 | -2 | -2 | -3 |
| T | -1 | 1 | 4 | 1 | -1 | 1 | 0 | 1 | 0 | 0 | 0 | -1 | 0 | -1 | -2 | -2 | -2 | -2 | -2 | -3 |
| P | -3 | -1 | 1 | 7 | -1 | -2 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -2 | -3 | -3 | -2 | -4 | -3 | -4 |
| A | 0 | 1 | -1 | -1 | 4 | 0 | -1 | -2 | -1 | -1 | -2 | -1 | -1 | -1 | -1 | -1 | -2 | -2 | -2 | -3 |
| G | -3 | 0 | 1 | -2 | 0 | 6 | -2 | -1 | -2 | -2 | -2 | -2 | -2 | -3 | -4 | -4 | 0 | -3 | -3 | -2 |
| N | -3 | 1 | 0 | -2 | -2 | 0 | 6 | 1 | 0 | 0 | -1 | 0 | 0 | -2 | -3 | -3 | -3 | -3 | -2 | -4 |
| D | -3 | 0 | 1 | -1 | -2 | -1 | 1 | 6 | 2 | 0 | -1 | -2 | -1 | -3 | -3 | -4 | -3 | -3 | -3 | -4 |
| E | -4 | 0 | 0 | -1 | -1 | -2 | 0 | 2 | 5 | 2 | 0 | 0 | 1 | -2 | -3 | -3 | -3 | -3 | -2 | -3 |
| Q | -3 | 0 | 0 | -1 | -1 | -2 | 0 | 0 | 2 | 5 | 0 | 1 | 1 | 0 | -3 | -2 | -2 | -3 | -1 | -2 |
| H | -3 | -1 | 0 | -2 | -2 | -2 | 1 | 1 | 0 | 0 | 8 | 0 | -1 | -2 | -3 | -3 | -2 | -1 | 2 | -2 |
| R | -3 | -1 | -1 | -2 | -1 | -2 | 0 | -2 | 0 | 1 | 0 | 5 | 2 | -1 | -3 | -2 | -3 | -3 | -2 | -3 |
| K | -3 | 0 | 0 | -1 | -1 | -2 | 0 | -1 | 1 | 1 | -1 | 2 | 5 | -1 | -3 | -2 | -3 | -3 | -2 | -3 |
| M | -1 | -1 | -1 | -2 | -1 | -3 | -2 | -3 | -2 | 0 | -2 | -1 | -1 | 5 | 1 | 2 | -2 | 0 | -1 | -1 |
| I | -1 | -2 | -2 | -3 | -1 | -4 | -3 | -3 | -3 | -3 | -3 | -3 | -3 | 1 | 4 | 2 | 1 | 0 | -1 | -3 |
| L | -1 | -2 | -2 | -3 | -1 | -4 | -3 | -4 | -3 | -2 | -3 | -2 | -2 | 2 | 2 | 4 | 3 | 0 | -1 | -2 |
| V | -1 | -2 | -2 | -2 | 0 | -3 | -3 | -3 | -2 | -2 | -3 | -3 | -2 | 1 | 3 | 1 | 4 | -1 | -1 | -3 |
| F | -2 | -2 | -2 | -4 | -2 | -3 | -3 | -3 | -3 | -3 | -1 | -3 | -3 | 0 | 0 | 0 | -1 | 6 | 3 | 1 |
| Y | -2 | -2 | -2 | -3 | -2 | -3 | -2 | -3 | -2 | -1 | 2 | -2 | -2 | -1 | -1 | -1 | -1 | 3 | 7 | 2 |
| W | -2 | -3 | -3 | -4 | -3 | -2 | -4 | -4 | -3 | -2 | -2 | -3 | -3 | -1 | -3 | -2 | -3 | 1 | 2 | 11 |

Figure 5. The BLOSUM62 substitution matrix

## 3.4. The Edit-Script, ε

In this section, we will describe how to create an Edit-Script from a given alignment such as Figure 6. The information we extract from an alignment corresponds to the following definitions. *Starting Index* in Figure 6 is an index that starts a matched sequence in sequence $S_x$. The *Ending Index* indicates an index that ends the matched sequence $S_y$. *Prior MatSeq* denotes a prior sequence of the matched sequence in sequence $S_y$. *Posterior MatSeq* describes the posterior sequence of the matched sequence in sequence $S_y$. Finally, *Matched Sequences* mean the locally similar area between Sequence $S_x$ and $S_y$.

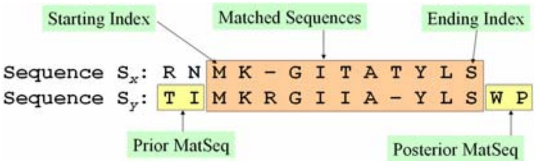

Starting Index
Matched Sequences
Ending Index
Sequence Sx: R N M K - G I T A T Y L S
Sequence Sy: T I M K R G I I A - Y L S W P
Prior MatSeq
Posterior MatSeq

Figure 6. The compositions of Edit-Script

In order to create an Edit-Script we more need three operators which transform the *Matched Sequence* of $S_x$ into the *Matched Sequence* of $S_y$. Focusing on *Matched Sequences*, there exist three kinds of events such as insertion, deletion, and substitution. A specific position on which an event occurs in a Matched Sequence, and changed residue (or nucleic acid) are inputted as parameters of these operators. Now, we can simply express the Edit-Script, ε = { {Prior MatSeq}, Starting Index, {operator1, operator2, operator3, … }, Ending Index, {Posterior MatSeq}. For example, the Edit-Script for Figure 6 is ε = { {TI}, 2, { ins(2,R), sub(5,I), del(7,T) }, 12, {WP} }. With this Edit-Script, we can transform Sequence $S_x$ into Sequence $S_y$. Here, Sequence $S_x$ indicates a representative sequence of the cluster. Sequence $S_y$ means a member sequence of the cluster. In this example shown in Figure 6, the sequence length of the two sequences looks quite short for charity of explanation, but the real length of the DNA sequence reaches to thousands of nucleic acids and hundreds for amino acids respectively.

## 3.5. Creating a member sequence from a representative sequence

The information which is stored into the database is a representative sequence of cluster and Edit-Scripts of each member sequence. When users want to show the member sequence, our system retrieves the representative sequence and Edit-Script of the member sequence upon a user's request. Figure 7 describes this procedure in detail. First, a user gets the representative sequence of the cluster and the Edit-Script of the member sequence where the user wants to create the sequence cluster from database searching. Next, the system extracts the *Matched Sequence* of the representative sequence with starting and ending indices in the Edit-Script. Then, the *Matched Sequence* of the representative sequence is transformed into a matched sequence of the member sequence using Change Operators. Finally, *Prior MatSeq* and *Posterior MatSeq* are added into the *Matched Sequence* of the member sequence. Then the gaps in the member sequence are removed.

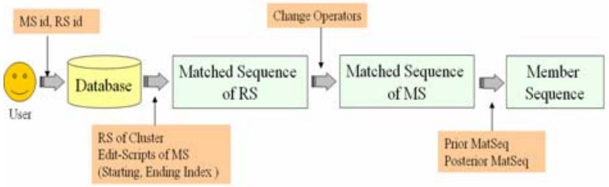


Figure 7. Creating a member sequence from a representative sequence

## 4. Evaluation

In this section, we estimated the compressed size using our method. Estimating compressed size between two DNA sequences was performed in two ways. One way was the compressed size by sequence length such as 2KB, 4KB, and 8KB. Another way was by sequence identity such as 40%, 50%, 60%, 70%, 80%, and 90%.

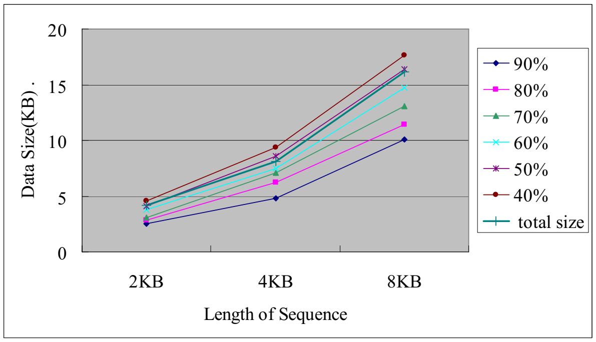


Figure 8. Compressed cluster size by sequence length and similarity.

In Figure 8, the total size (indicate by ─┼─) denotes the total size of two sequences without applying any Edit-Script. The size reduction was discovered when the homology (sequence identity) between sequences was over 50% for 2KB, 55% for 4KB, and 52% for 8KB. The data reduction rate got better when the sequence identity was higher. Generally speaking in this experiment, the size of the applying Edit-Script was reduced when the sequence identity was over 55%. The size was somewhat increased when the sequence identity was lower than 50%. This is because the Edit-Script was longer than the sequence when the sequence identity was lower than 50%. In order to store the information that a nucleic acid was changed, the Edit-Script needed 3 Bytes (type of event, the position, and nucleic acid)

## 5. Conclusion

In this paper, we suggest a new compression method to reduce sequence cluster size. To achieve this reduction, we first select the representative sequence of a cluster. In this procedure, we calculate the average sequence distance of all member sequences using the Smith-Waterman alignment score. The specific substitution matrices for nucleic acid and amino acid (BLOSUM62 [6]) are utilized when scoring. We create an Edit-Script, ε, from the Smith-Waterman alignment result. The changed information is written in Edit-Script. Only a representative sequence of the cluster and each Edit-Script of the member sequence are stored into the database. Through experimental results, the size reduction is achieved when the sequence identity (homology) is over 55%.

Not only sequences in the SNP cluster and results of protein engineering which makes useful protein from well-known protein, but also any sequence clusters which have high sequence similarity are good examples to apply our work to. In going work, we are extending our method to manage versions of sequences using temporal concepts.

## 6. Acknowledgement

This work was supported by a Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (The Regional Research Universities Program/Chungbuk BIT Research-Oriented University Consortium) and the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (R01-2007-000-10926-0)

## 7. References

[1] Altschul, S.F. et. al.: Basic local alignment search tool., J.Mol.Biol., 215, 1990, pp. 403-410.

[2] Altschul, S.F., et. al.: Gapped BLAST and PSI-BLAST:a new generation of protein database search programs. Nucleic Acids Res., 25, 1997, pp. 3389-3402.

[3] Bains, W.: MULTAN : A program to align multiple DNA sequences., Nucl.Acids.Res., 14, 1986, pp. 157-177.

[4] Barton, G.J., and Sternberg, M.J.E.: A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons., J.Mol.Biol., 198 , 1987, pp. 327-337 .

[5] Dayhoff, M.O., Schwartz, R.M., and Orcutt. B.C.: A model of evolutionary change in proteins, Atlas of Protein sequence and Structure, 5(3), 1978, pp. 345-352.

[6] Henikoff, S., and Henikoff, J.G.: Amino acid substitution matrices from protein blocks. Proc.Natl.Acad.Sci.USA, 89, 1992, pp. 10915-10919.

[7] Higgins, D.G., Bleasby, A.J., and Fuchs, R.: CLUSTAL V: Improved software for multiple sequence alignment., Comp.Appl.Biosci., 8, 1992, pp. 189-191.

[8] Lipman, D.J., and Pearson, W.R.: Rapid and sensitive protein similarity searches., Science, 227, 1985, pp. 1435-1441.

[9] Murata, M.J., Richardson, J.S., and Sussman, J.L.: Simultaneous comparison of three protein sequences., Proc.Natl.Acad.Sci.USA, 82, 1985, pp. 3073-3077.

[10] Needleman, S.B., and Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequences of two proteins., J.Mol.Biol., 48, 1970, pp. 443-453.

[11] Pearson, W.R., and Lipman, D.J.: Improved tools for biological sequence comparison. Proc.Natl.Acad.Sci. USA, 85, 1988, pp. 2444-2448.

[12] Sellers, P. H.: An algorithm for the distance between two finite sequences., J.Comb.Th.A, 16, 1974, pp. 253-258.

[13] Smith, T.F., and Waterman, M.S.: Identification of common molecular subsequences. J.Mol.Biol., 147, 1981, pp. 195-197.

[14] Sobel, E., and Martinez, H.M.: A multiple sequence alignment program, Nucl.Acids.Res., 14, 1986, pp. 363-374.

[15] Taylor, W.R.: Identification of protein sequence homology by consensus template alignment. J.Mol.Biol., 188, 1986, pp. 233-258.

[16] Tompson, J.D., Higgins, D.G., and Gibson, T.J.: ClustalW: Improved sensitivity of profile searches through the use of sequence weights and gap excision. Comput. Appilc. Biosci. 10, 1994, pp. 19-29.

[17] Wilbur, W.J., and Lipman, D.J.: Rapid similarity searches of nucleic acid and protein data banks. Proc.Natl. Acad.Sci.USA, 80, 1983, pp. 726-730.

[18] Wilbur, W.J., and Lipman, D.J.: The context dependent comparison of biological sequences. Siam.J.Appl. Math., 44, 1984, pp. 557-567.

[19] Kim, S., Jung K.S., Ryu K.H.: Automatic Orthologous-Protein-Clustering from Multiple Complete-Genomes by the Best Reciprocal BLAST Hits, LNBI, 3916, 2006, pp. 60-70.

[20] Jung, K.S., Kim, S., Ryu, K.H.: A Personalized Biological Data Management System Based on BSML, LNBI, 4115, 2006, pp. 362-371