Algorithm for Protein Sequence Analysis

Krishna Bikram Shah
Dept. of Computer Science and Engineering
Sikkim Manipal Institute of Technology
Majitar, Rangpoo, E. Sikkim – 737136, India

Samarjeet Borah
Dept. of Computer Science and Engineering
Sikkim Manipal Institute of Technology
Majitar, Rangpo, E. Sikkim – 7371136, India

ABSTRACT
When comparing two sequences to identify similarities and differences between them, generally, a measure of how similar they are is also desirable. Similar sequences probably have the same ancestor, share the same structure, and have a similar biological function. A typical approach to solve this problem is to find a good and reasonable alignment between the two sequences. Then, given an appropriate scoring scheme, their similarity can be computed. This study describes the concepts, tools and implementation of selected algorithms being used for sequence analysis.

Keywords
Genome, DNA, adenosine (A), cytosine (C), guanine (G) and thymine (T).

1. INTRODUCTION
Genome is the complete set of DNA molecules inside any cell of a living organism that is passed from one generation to its offspring. It is recognized as what makes two living beings biologically similar or distinct. Similar sequences probably have the same ancestor, share the same structure, and have a similar biological function.

The DNA (deoxyribonucleic acid) [1] is basically a double chain of simpler molecules called nucleotides, tied together in a helical structure popularly known as the double helix (Figure 1). The two chains, called strands, are complementary in such a way that is possible to infer one strand from the other. The nucleotides are distinguished by a nitrogen base that can be of four kinds: adenosine (A), cytosine (C), guanine (G) and thymine (T). Adenosine always bonds to thymine (A-T) whereas cytosine always bonds to guanine (C-G), forming base pairs.

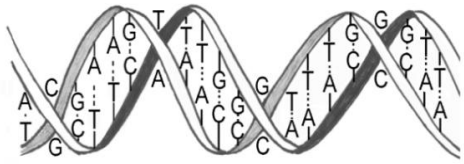

Figure: 1 Double Helix Structure of DNA [2]

Base pairs are the most common unit for measuring the length of a DNA. A DNA can be specified uniquely by listing its sequence of nucleotides, or base pairs. Therefore, for practical purposes, the DNA is abstracted as a l*ong* text over a four-letter alphabet, each representing a different nucleotide: A, C, G and T.

Table 1. Amino Acids Commonly Found in Proteins [3].

One-letter	Three-letter	Name
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic Acid
E	Glu	Glutamic Acid
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
U	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine

Proteins are responsible for what a living being is and does in a physical sense. They are the molecules that accomplish most of the functions of a living cell, determining its shape and structure. Again, a protein is a linear sequence of simpler molecules called amino acids. Twenty different amino acids are commonly found in proteins, and they are identified by a letter of the alphabet or a three-letter code (Table 1). Like the DNA, proteins are conveniently represented as a string of letters expressing its sequence of amino acids. There is an intimate relation between DNA and proteins sequences. To produce a protein, the cell "reads" a sequence of three nucleotides from the DNA string, called a codon, to generate each of its amino acids.

Table 2. The genetic code that maps the RNA to amino acids [3]

First	Second Position				Third
Position	***U***	***C***	***A***	***G***	***Position***
	Phe	Ser	Tyr	Cys	**U**
	Phe	Ser	Tyr	Cys	**C**
	Leu	Ser	Stop	Stop	**A**
U	Leu	Ser	Stop	Trp	**G**
	Leu	Pro	His	Arg	**U**
	Leu	Pro	His	Arg	**C**
	Leu	Pro	Gin	Arg	**A**
C	Leu	Pro	Gin	Arg	**G**
	Lle	Thr	Asn	Ser	**U**
	Lle	Thr	Asn	Ser	**C**
	Lle	Thr	Lys	Arg	**A**
A	Met(Start)	Thr	Lys	Arg	**G**
	Val	Ala	Asp	Gly	**U**
	Val	Ala	Asp	Gly	**C**
	Val	Ala	Glu	Gly	**A**
G	Val	Ala	Glu	Gly	**G**

For instance, the triplet AAG, when found in a DNA strand, instructs the cell to generate the lysine amino acid. The correspondence between codons and amino acids is known as the genetic code (Table 2). Surprisingly, not all parts of a DNA molecule encode genes; some segments are called "junk DNA" because they have no known function. DNA can act not only as a template for making copies of itself but also as a blueprint for a molecule called ribonucleic acid (RNA). The process by which DNA is transcribed into RNA is called transcription. RNA is structurally similar to DNA. It is a polymeric molecule made up of individual chemical unit, but the chemical backbone that holds these units together is slightly different from the backbone of DNA, allowing RNA to exist in a single standard form as well as in a double helix. These single-stranded molecules still form base pairs between different parts of the chain, causing RNA to fold into 3D structure. The individual chemical units of RNA are designated A, C, G and U (uracil, which takes the place of thymin).

The genome provides a template for the synthesis of a variety of RNA molecules, the three main types of RNA are messenger RNA, transfer RNA and ribosomal RNA. Messenger RNA (mRNA) molecules are RNA transcripts of genes. They carry information from the genome to the ribosome, the cell's protein synthesis apparatus. Transfer RNA (tRNA) molecules (figure 2) are untranslated RNA molecules that transport amino acid, the building blocks of proteins, to the ribosome. Finally ribosomal RNA (rRNA) molecules are the untranslated RNA components of ribosomes, which are complexes of protein and RNA. rRNA are involved in anchoring the mRNA molecule and catalyzing some steps in the translation process. Some viruses also use RNA instead of DNA as their genetic material.

2. SEQUENCE ALIGNMENT

Alignment of two sequences [3] which may be of different sizes is to write one on top of the other, and "break" them into smaller segments by inserting spaces in one or the other so that identical subsequences are eventually aligned in a one-to-one

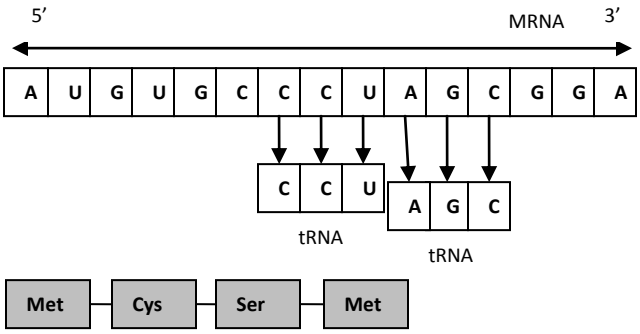

Figure: 2 Schematic of RNA being translated into

correspondence naturally. The spaces are not inserted in both sequences at the same position. At the end of process, the sequences end up with the same size [5]. The following example illustrates an alignment between the sequences

A="ACAAGACAGCGT" and B="AGAACAAGGCGT".

A = ACAAGACAG-CGT

| | | | | | | | | | | |

B = AGAACA-AGGCGT

Figure 3 Alignment of two sequences.

The objective is to match identical subsequences as far as possible. In the example, nine matches are highlighted with vertical bars. If the sequences are not identical, mismatches are likely to occur as different letters are aligned together. Two mismatches can be identified in the example: a "C" of A aligned with a "G" of B, and a "G" of A aligned with a "C" of B. The insertion of spaces produced gaps in the sequences. They are important to allow a good alignment between the last three characters of both sequences.

An alignment can be seen as a way of transforming one sequence into the other. A mismatch is regarded as a substitution of characters. A gap in the first sequence is considered an insertion of a character from the second sequence into the first one, whereas a gap in the second sequence is considered a deletion of a character of the first sequence. In the previous example, A can be converted into B in four steps: 1) substitute the first "C" for a "G"; 2) substitute the first "G" for a "C"; 3) delete the second "C"; and 4) insert a "G" before the last three characters.

Once the alignment is produced, a score can be assigned to each pair of aligned letters, called aligned pair, according to a chosen scoring scheme. We usually reward matches and penalize mismatches and gaps. The overall score of the alignment can then be computed by adding up the score of each pair of letters. For instance, using a scoring scheme that gives a +1 value to matches and −1 to mismatches and gaps, the alignment of Figure 2 scores $9 \cdot (1) + 2 \cdot (-1) + 2 \cdot (-1) = 5$.

The similarity of two sequences can be defined as the best score among all possible alignments between them.

3. SEQUENCE ANALYSIS ALGORITHMS

To find an alignment of two sequences is to maximum number of matches while minimizing the number of gaps and mismatches [6].

3.1 The Needleman – Wunsch Algorithm

This is one of the first commonly used approaches in sequence alignment. In this approach uses an alignment scoring system which assigns a penalty to gaps linearly according to their length [7]. For any alignment its score S can be calculated by

$$S = x - \sum W_k - Z_k$$

The score is calculated by counting the total number of matches x and subtracting from this value the product of the number of k-length gaps (Z_k) and an arbitrary gap penalty W_k for a gap of length of k. The linear gap penalties $W_k$can be defined as $W_k = a + bk$ where a and b are arbitrary penalties for gap opening and gap extension. An optimal alignment between two sequences is hence between two sequences is hence calculated by finding the alignment between two sequences which maximizes the score

$$S = \max\,(x - \sum W_k Z_k)$$

The easiest way to represent all possible alignments between two arbitrary protein and nucleotide sequence is as a path graph. A dynamic programming method lends them perfectly to this type of optimization. From a dynamic programming perspective each state in the graph can be scored as follows, using Needleman-Wunsch scoring scheme

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + S_{i,j} \\ M_{i,j-1} - w \\ M_{i-1,j} + w \end{cases}$$

Each node in the path graph $M_{i,j}$ is assigned a score based on the scoring of its incoming states which are connected in the path graph. The value $M_{i-1,j-1}$ represent a diagonal match or mismatch state while the $M_{i,j-1}$ and $M_{i-1,j-1}$ values represent the insertion of gaps in either sequence. All paths are not equally desirable hence we add the value $S_{i,j}$ to match-mismatch paths, and the value is a score for a match between the amino acids or nucleotides represented by i and j. Conversely paths which introduce a gap are penalized by subtracting the value w. Pseudo code for the dynamic programming implementation of the Needleman-Wunsch algorithm is as follows.

Algorithm Needleman-Wunsch

1. *begin*
2. $M_{0,0} \leftarrow 0$
3. *for* $j \leftarrow 1$ *to* N *do*
4. $M_{i,j} \leftarrow M_{i,j-1} + \sigma(\bar{b}_j)$
5. *end*
6. *for* $i \leftarrow 1$ *to* M *do*
7. $M_{i,0} \leftarrow M_{i,j-1} + \sigma(\underline{a}_i)$
8. *for* $j \leftarrow 1$ *to* N *do*
9. $M_{i,j} \leftarrow \max\left[M_{i-1,j-1} + \sigma\binom{a_i}{b_j}, M_{i,j-1} + \sigma(\underline{a}_i), M_{i-1,j} + \sigma(\bar{b}_j)\right]$
10. *end*
11. *end*
12. *write* "Global similarity Score is" $M_{M,N}$
13. *end*

3.2 The Smith – Waterman Algorithm

For local alignment detection we need to consider alignment that can begin and end at any of the MN position in the alignment matrix, where M and N represent the length of each of the sequence being aligned [8]. Since every possible position in the matrix can be the start of an alignment, scores in the matrix are never allowed to fall below zero. Because every point in the matrix can also be viewed as the end of an alignment, we also need to store the highest score from each step. The added complexity of the local alignment method is in fact very simple to implement. Pseudo code for the dynamic programming implementation of the Smith-Waterman algorithm is as follows.

Algorithm: Smith-Waterman

1. *begin*
2. $best \leftarrow 0$
3. *for* $j \leftarrow 1$ *to* N *do*
4. $M_{0,j} \leftarrow 0$
5. *end*
6. *for* $i \leftarrow 1$ *to* M *do*
7. $M_{i,0} \leftarrow 0$
8. *for* $j \leftarrow 1$ *to* N *do*
9. $M_{i,j} \leftarrow \max\left[M_{i-1,j-1} + \sigma\binom{a_i}{b_j}, M_{i,j-1} + \sigma(\underline{a}_i), M_{i-1,j} + \sigma(\bar{b}_j)\right]$
10. $best \leftarrow \max(M_{i,j}, best)$
11. *end*
12. *end*
13. *write* "Local similarity Score is" *best*
14. *end*

3.3 The FASTA Algorithm

Although the Needleman-Wunsch and Smith-Waterman algorithms are rigorous methods for detecting optimal local and global alignments between sequences, they are quite computationally intensive [9]. In order to compare a query sequence against a large database of sequence, a significant amount of computation time may be required. These method unlike Needleman-Wunsch and Smith-Waterman method, do not examine the complete space of all possible alignments, but instead use heuristics approach to quickly identify potentially high scoring alignments and hence limit the search space considerably.

The FASTA algorithm can be summarized as follows:

Step1: Identify regions shared by two sequences with the highest density of identities (ktup=1) or pair of identities (ktup=2).
Step2: Rescan the ten regions with the highest density of identities using a substitution matrix. Trim the ends of the region to include only those residue contributing to the highest score. Each region is a partial alignment without gaps.
Step3: If there are several initial regions with score grater than a given cut-off value, check whether the trimmed initial regions can be joined to form an approximate alignment with gaps. Calculate a similarity score that is the sum of the joined initial regions minus a penalty, usually 20 for each gap. The score of the single best initial region found in step 2 is also reported.
Step4: For sequences with scores grater than a threshold, construct an optimal local alignment of the query sequence and the library sequence, considering only those residues that lie in a band centred on the best initial region found in Step 2. For protein searches with ktup=2 a 16 residue band is used by default. A 32 residue band is used with ktup=1.
Step5: After the first 60,000 scores have been calculated, normalize the raw similarity scores using estimates for the statistical parameters of the extreme value distribution. The default strategy regresses the average variance in similarity scores. Z-score are calculated, and the calculation is repeated with library sequences with Z-scores greater than 5.0 and less than -5.0 removed.
Step6: Finally the Smith-Waterman algorithm is used to display the alignment.

3.4 The BLAST

The Basic Local Alignment Search Tool [10] is the most widely used bioinformatics tool ever written. It is an alignment heuristic that determines local alignments between query and a database. It uses an approximation of the Smith-Waterman algorithm. The initial phase of the algorithm is to build a list of words within a query sequence, which represent sequential residue combinations of a specific size, usually 3 for protein sequence. The central idea of the BLAST algorithm is that a statistically significant alignment of two sequences is likely to contain a high-scoring pair of aligned words. A discrete finite automaton is constructed to rapidly detect matches of these words in a database of query sequences. Word matches are then extended into ungapped local alignments, and alignments that score above a specific threshold are extended using slow gapped alignment extensions.

The action of BLAST is summarized as follows:

Step1: For each three amino acids in the query sequence, identify all of the substitutions of each word that have a similarity score grater than a threshold score T.
Step2: Build a discrete finite automation (DFA) to recognize the list of identical and substituted three letter words.
Step3: Use the DFA to identify all matching words in sequences in the database. The refined BLAST algorithm is based upon the observation that a high scoring pair (HSP) of interest is much longer than a single word pair, and may therefore entail multiple hits on the same diagonal and within a relatively short distance of one another. Hence, if two non-overlapping matches are found closer than a threshold distance A, each match is extended both forwards and backwards without allowing gaps. Extended matches above a specific threshold are stored. These matched ungapped alignment segments above the threshold are called high scoring pairs (HSPs). If a high scoring pair exhibits a score above yet another threshold, then gapped extensions are triggered. These extensions start from a seed is chosen by locating an 11 residue alignment with the highest score along the HSP and taking its central residue as the seed. From the seed residue, gapped extensions travel across the path graph originating from the seed residue, and the highest scoring gapped local alignment is stored.
Step4: Report all of the significant alignments detected.

Table 3. Algorithm for comparing protein sequences

Algorithm	Alignment Type	Scoring Matrix	Gap Penalty	Time Required
Needleman-Wunsch	Global similarity	Arbitrary	Penalty/gap k	$\theta(n^2)$
Smith-Waterman	Global distance	$S_{i,j} < 0.0$	Affine q+rk	$\theta(n^2)$
FASTA	Local similariry	$S_{i,j} < 0.0$	Limited gap size q+rk	$\theta(n^2)/K$
BLASTP	Maximum segment score	$S_{i,j} < 0.0$	Multiple segments	$\theta(n^2)/K$

4. DISCUSSION

A sequence alignment is a way of arranging the primary sequences of DNA, RNA, or protein. Alignment provides a powerful tool to compare related sequences, and the alignment of two residues could reflect a common evolutionary origin, or could represent common structural roles. If two sequences in an alignment share a common ancestor, mismatches can be interpreted as point mutations and gaps as indels introduced in one or both lineages in the time since they diverged from one pairwise sequence alignment methods are used to find the best-matching local or global alignments of two query sequences. Pairwise alignments can only be used between two sequences at a time. For the tool development the code was written in C++. Two protein sequences were given as input to the program It was checked whether the entered sequences were correct or not, that is the correct sequence should not contain any spaces or letter other than the C, S, T, P, A, G, N, D, E, Q, H, R, K, M, I, L, V, F, Y, W. The length of the two sequences was calculated and the local alignment of two sequences is done. For example for the two sequences

>FOSB_MOUSE Protein fosB

"MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAA
ASQECAGLGEMPGSFVPTVTAITTSQDLQWLVQPTLISSM
AQSQGQPLASQPPAVDPYDMPGTSYSTPGLSAYSTGGAS
GSGGPSTSTTTSGPVSARPARARPRRPREETLTPEEEEKRR
VRRERNKLAAAKCRNRRRELTLPGSTSAKEDGFGWLLPP
PPPPPLPFQSSRDAPPNLTASLFTHSEVQVLGDPFPVVSPS
YTSSFVLTCPEVSAFAGAQRTSGSEQPSDPLNSPSLLAL"
and

>FOSB_HUMAN Protein fosB

MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAA
SQECAGLGEMPGSFVPTVTAITTSQDLQWLVQPTLISSMA
QSQGQPLASQPPVVDPYDMPGTSYSTPGMSGYSSGGASG
SGGPSTSGTTSGPGPARPARARPRRPREETLTPEEEEKRR
VRRERNKLAAAKCRNRRRELTLEFVLVAHKPGCKIPYEE
GPGPGPLAEVRDLPGSAPAKEDGFSWLLPPPPPPPLPFQTS

QDAPPNLTASLFTHSEVQVLGDPFPVVNPSYTSSFVLTCP
EVSAFAGAQRTSGSDQPSDPLNSPSLLAL

The output motif is given by

FQAFGDYDSGSRCSSSSAESQYLSSVDSFGSTAAASQECS
GLGEMGSFVTVTAITTSQDLQWLVQTLISSMAQSQGQLA
SQAVDYDMGTSYSTGLSAYSTGGASGSGGSTSTTTSGVS
ARARARRRREETLTEEEEKRRVRRERNKLAAAKCRNRR
RELT----L----------G------STS-AKEDGFFWLLLFQSSRD
ANLTASLFTJSEVQVLGDFVVSSYTSSFVLTCEVSAFAGA
QRTSGSEQSDLNSSLLAL

and

FQAFGDYDSGSRCSSSSAESQYLSSVDSFGSTAAASQECA
GLGEMGSFVTVTAITTSQDLQWLVQTLISSMAQSWGQLA
SQVVDYDMGTSYSTGMSGYSSGGASGSGGSTSGTTSG_G
ARARARRRREETLTEEEEKRRVRRERNKLAAAKCRNRR
RELTLEFVLVAHKGCKIYEEGGGLAEVRDLGSAAKEDGF
SWLLLFQTSQDANLTASLFTHSEVQVLGDFVVNSYTSSF
VLTCEVSAFAGAQRTSGSDQSDLNSSLLAL

Where corresponding matches, mismatches are given by one to one correspondence and '-' represent the gap between the two sequences. Execution of same data on EMBOSS Pairwise Alignment Algorithm [11] gives the following result.

```
#=======================================
# Aligned_sequences: 2
# 1: EMBOSS_001
# 2: EMBOSS_001
# Matrix: EBLOSUM62
# Gap_penalty: 1.0
# Extend_penalty: 0.5
# Length: 309
# Identity:     264/309 (85.4%)
# Similarity:   270/309 (87.4%)
# Gaps:         32/309 (10.4%)
# Score: 1370.5
#=======================================================
1  MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEM   50
   |||||||||||||||||||||||||||||||||||||||||||||||||
1  MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEM   50

51 PGSFVPTVTAITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMP  100
   |||||||||||||||||||||||||||||||||||||||.|||||||
51 PGSFVPTVTAITTSQDLQWLVQPTLISSMAQSQGQPLASQPPVVDPYDMP  100

101 GTSYSTPGLSAYSTGGASGSGGPSTSTTTSGPVS-ARPARARPRRPREET 149
    ||||||||:|.||:||||||||||||.|||||  .  ||||||||||||||
101 GTSYSTPGMSGYSSGGASGSGGPSTSGTTSGP-GPARPARARPRRPREET 149

150 LTPEEEEKRRVRRERNKLAAAKCRNRRRELT------------------- 180
    |||||||||||||||||||||||||||||||
150 LTPEEEEKRRVRRERNKLAAAKCRNRRRELTLEFVLVAHKPGCKIPYEEG 199

181 -----------LPGSTSAKEDGFGWLLPPPPPPPPLPFQSSRDAPPNLTAS 219
              ||||..|||||||.|||||||||||||:|:|||||||||
200 PGPGPLAEVRDLPGSAPAKEDGFSWLLPPPPPPPPLPFQTSQDAPPNLTAS 249

220 LFTHSEVQVLGDPFPVVSPSYTSSFVLTCPEVSAFAGAQRTSGSEQPSDP 269
    |||||||||||||||:||||||||||||||||||||||:|||||
250 LFTHSEVQVLGDPFPVVNPSYTSSFVLTCPEVSAFAGAQRTSGSDQPSDP 299

270 LNSPSLLAL    278
    |||||||||
300 LNSPSLLAL    308
#=======================================================
```

5. CONCLUSION

The Sequence analysis is a very challenging task that needs great understanding. The similarity and alignments concepts cannot be addressed directly with one technique or algorithm. A better performance is achieved by the understanding of different concepts. In practice algorithms are used that run much faster, at the expense of possibly missing some significant hits due to the heuristics employed. The sequence analysis brings very interesting and vital facts between species. We are now able to find genetic similarities and differences in different and diverse creatures, the microarray technology, phylo-genetic tree creation and many other alignment and analysis tools have helped biologist in many aspects. In future the sequence analysis will help biologist to devise genetic therapy and solutions for genetic disorders.

6. REFERENCES

[1] Text available (online): http://en.wikipedia.org/wiki/ sequence_ analysis

[2] Image available (online): http://www.apimba.org.php

[3] Hassan Mathkour and Muneer Ahmad, Genome Sequence Analysis, A Survey, Journal of Computer Science.

[4] Maxime Crochemore, Sequence Alignment Algorithms, Kings College, London

[5] Krishna Bikram Shah and Samarjeet Borah, "Protein Sequence Aaalysis, A Study on Recent Developement", ISBN 978-1-4244-9241-1, IEEE CPF1197F-PRT, Pg. 506-510

[6] Text available (online) http://www.wekipedia.com.

[7] Zhinua Du and Feng Lin, Improvement of the Needleman-Wunsch Algorithm, 2004, Volume 3066/2004, 792-797

[8] Smith TF and Waterman MS, Identification of Common Molecular Subsequences, *J. Mol. Biol.* (1981) 147, 195-197

[9] Text available (online): www.ebi.ac.uk/fasta33

[10] Text available (online): www.ncbi.nlm.nih.gov/BLAST

[11] Result from http://www.ebi.ac.uk/Tools/emboss/align/