Applications of Dynamic Programming in Sequence Alignment

Anuj Gupta[1, *, +], **Rahul Alapati**[1, +], **Vineet Nayak**[1, +], **Sritika Chakladar**[1, +] **and Austin Ream**[2, +]

[1] Graduate Student, Computer Science and Software Engineering, Auburn University, 36830, USA [2] Undergraduate Student, Computer Science and Software Engineering, Auburn University, 36830, USA

[*] azg0076@auburn.edu

[+] these authors contributed equally to this work

ABSTRACT

Protein Sequence Alignment is a basic operation mostly used in protein sequence analysis. Protein sequence alignment is an optimization problem and the use of dynamic programming, a careful brute force approach is quite helpful in finding the optimal pairwise sequence alignment. This report lays out the implementation details of two popular dynamic programming algorithms namely: Needleman-Wunsch and Smith-Waterman. The Needleman-Wunsch algorithm is widely used for optimal global alignment, particularly when the quality of the global alignment is of the utmost importance. The Smith-Waterman algorithm performs the local sequence alignment, that is for determining similar regions between two strings of protein sequences. The resulting optimal local and global alignments are then compared with the alignments produced by the BLAST algorithm.

1. Introduction

Sequence comparison is motivated by the fact that all living organisms are related by evolution. That implies that the genes of species that are closely related should have higher similarity than those which are distantly related. Two protein sequences are homologous if they have a common ancestor. Homologous sequences can be inferred from the similarity of two sequences. A measure of likeliness between two sequences is percentage identity. We calculate the percentage identity of two sequences by counting the exactly matching characters when the sequence is aligned and divide by the length of the longer of the two compared sequences.

Sequence Alignment is a way of arranging the sequences of protein to

identify regions of similarity that may be a consequence of functional, structural or evolutionary relationships between the sequences. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns.

Very short or very similar sequences can be aligned by hand. However, most interesting problems require the alignment of lengthy, highly variable or extremely numerous sequences that cannot be aligned solely by human effort. Instead, human knowledge is applied in constructing algorithms to produce high-quality sequence alignments.

Computational approaches to sequence alignment generally fall into two categories: global alignments and local alignments. Calculating a global alignment is a form of global optimization that "forces" the alignment to span the entire length of all query sequences. By contrast, local alignments identify regions of similarity within long sequences that are often widely divergent overall. Local alignments are often preferable but can be more difficult to calculate because of the additional challenge of identifying the regions of similarity.

If we try to apply general brute force to optimally align two sequences of length n, then we end up enumerating exponentially large number of alignments, which is humanly impossible. Hence, we implemented dynamic programming algorithms to solve optimal pairwise sequence problem.

Dynamic Programming is an algorithmic technique for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of these subproblems just once, and storing their solutions. The next time the same subproblem occurs, instead of recomputing its solution, one simply looks up the previously computed solution, thereby saving computation time at the expense of a modest expenditure in storage space. Dynamic programming algorithms are often used for optimization.

2. Methods

The techniques of dynamic programming can be applied to produce global alignments using the Needleman-Wunsch algorithm, and local alignments using the Smith-Waterman algorithm. Protein alignments use a substitution matrix to assign scores to amino-acid matches or mismatches, and gap penalty for matching an amino acid in one sequence to a gap in the other. Here, we have used BLOSSUM62 (BLOcks SUbstitution Matrix) to score the matches and mismatches.

2.1. Needleman-Wunsch Algorithm:

This algorithm was developed by Saul B. Needleman and Christian D. Wunsch and published in 1970. The algorithm essentially divides a large problem into a series of smaller problems and uses the solutions to smaller problems to reconstruct a solution to the larger problem.

Firstly, we prepared a (n+1) * (m+1) DP Table with one sequence along the top and one along the left side. In this DP Table, we can arrive at each cell in one of the following three ways:

- From the cell above, which corresponds to aligning the character to the left with a space.
- From the cell to the left, which corresponds to aligning the character above with a space.
- From the cell diagonally to the above-left, which corresponds to aligning the characters to the left and above (which might or might not match).

We fill up all the cells in the DP Table using the following general recurrences:

$$V(i, j) = \text{MAX} \begin{cases} \delta(S1(i), S2(j)) + V(i-1, j-1) \\ \delta(-, S2(j)) + V(i, j-1) \\ \delta(S1(i), -) + V(i-1, j) \end{cases}$$

The values in the above recurrences are calculated using the BLOSSUM Matrix.

The base cases are:

$$V(i, 0) = \sum_{0 \le k \le i} \delta(S1(k), -)$$

$$V(0, j) = \sum_{0 \le k \le j} \delta(-, S2(k))$$

Now after the DP table is ready, we identify the (n+1) * (m+1) element and traceback until the base case, while generating the optimal alignment for the two sequences.

2.2. Smith-Waterman Algorithm:

The algorithm was first proposed by Temple F. Smith and Michael S. Waterman in 1981.Like the Needleman-Wunsch algorithm, of which it is a variation, Smith-Waterman is a dynamic programming algorithm. The main difference to the Needleman-Wunsch algorithm is that negative scoring matrix cells are set to zero, which renders the local alignments visible. Traceback procedure starts at the highest scoring matrix cell and proceeds until a cell with score zero is encountered, yielding the highest scoring local alignment.

The general recurrences used to fill the DP Table are:

$$V(i, j) = MAX \begin{cases} 0 \\ \delta(S1(i), S2(j)) + V(i-1, j-1) \\ \delta(-, S2(i)) + V(i, j-1) \\ \delta(S1(i), -) + V(i-1, j) \end{cases}$$

The base cases are:

$$V(i, 0) = V(0, j) = 0$$

3. Results

1. **Human Hemoglobin Sequence:**

```
MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNA
VAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
YR
```

Mouse Hemoglobin Sequence:

```
MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHGKKVADALASA
AGHLDDLPGALSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTPAVHASLDKFLASVSTVLTSK
YR
```

Global Alignment Output:

```
Score: 648
MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNA
||||**|||||||||||||*|**|||||||||||*||||||||||||||||||||||||||||||||*||**|||||||*|
MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHGKKVADALASAAGHLDDLPGA


LSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR
|||||||||||||||||||||||||||||||*|||||||||||||||||||||||||||||
LSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTPAVHASLDKFLASVSTVLTSKYR
```

Note: '|' indicates a positive match, i.e. amino acids whose matches have positive scores in BLOSUM62 and '*' indicates a non-positive score in BLOSUM62.

BLAST Output:

unnamed protein product

Sequence ID: Query_149925 **Length:** 142 **Number of Matches:** 1

Range 1: 1 to 142 Graphics ▼ Next Match ▲ Previous Match

NW Score	Identities	Positives	Gaps
648	122/142(86%)	131/142(92%)	0/142(0%)

```
Query  1    MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHG  60
            MVLS  DK+N+KAAWGK+G H  EYGAEALERMF SFPTTKTYFPHFD+SHGSAQVKGHG
Sbjct  1    MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHG  60

Query  61   KKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTP  120
            KKVADAL +A  H+DD+P ALSALSDLHAHKLRVDPVNFKLLSHCLLVTLA+H PA+FTP
Sbjct  61   KKVADALASAAGHLDDLPGALSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTP  120

Query  121  AVHASLDKFLASVSTVLTSKYR  142
            AVHASLDKFLASVSTVLTSKYR
Sbjct  121  AVHASLDKFLASVSTVLTSKYR  142
```

In case of human & mouse sequences, we observed an exact match with the BLAST in terms of global alignment as well as the score. The sequences have an identity of 86%.

Local Alignment:

```
Score: 648

MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNA
||||**|||||||||||||*|**||||||||||*|||||||||||||||||||||||||||||||||*||**|||||||*|
MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHGKKVADALASAAGHLDDLPGA


LSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR
||||||||||||||||||||||||||||||||*|||||||||||||||||||||||||||||
LSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTPAVHASLDKFLASVSTVLTSKYR
```

BLAST Output:

unnamed protein product

Sequence ID: Query_220933 **Length:** 142 **Number of Matches:** 1

Range 1: 1 to 142 Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Method	Identities	Positives	Gaps
253 bits(645)	4e-93	Compositional matrix adjust.	122/142(86%)	131/142(92%)	0/142(0%)

```
Query  1    MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHG  60
            MVLS  DK+N+KAAWGK+G H  EYGAEALERMF SFPTTKTYFPHFD+SHGSAQVKGHG
Sbjct  1    MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHG  60

Query  61   KKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTP  120
            KKVADAL +A  H+DD+P ALSALSDLHAHKLRVDPVNFKLLSHCLLVTLA+H PA+FTP
Sbjct  61   KKVADALASAAGHLDDLPGALSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTP  120

Query  121  AVHASLDKFLASVSTVLTSKYR  142
            AVHASLDKFLASVSTVLTSKYR
Sbjct  121  AVHASLDKFLASVSTVLTSKYR  142
```

In case of human & mouse sequences we observed an exact match with the BLAST in terms of local alignment. The sequences have an identity of 86%.

Also, the local and global alignments for human and mouse sequences are the same. The scores are different from the one in BLAST in case of local alignment.

2. **Sequence 1:**

GTCTATCAC

Sequence 2:

ATCTCGTATGATG

Global Alignment Output:

```
Score: 12
G--TC-TATCAC-
*  || |||*|*
ATCTCGTATGATG
```

BLAST Output:

unnamed protein product

Sequence ID: Query_245625 **Length:** 13 **Number of Matches:** 1

Range 1: 1 to 13 Graphics ▼ Next Match ▲ Previous Match

NW Score	Identities	Positives	Gaps
4	4/13(31%)	4/13(30%)	0/13(0%)

```
Query  1   GTCTATCAC      9
             TCT   A
Sbjct  1   ATCTCGTATGATG  13
```

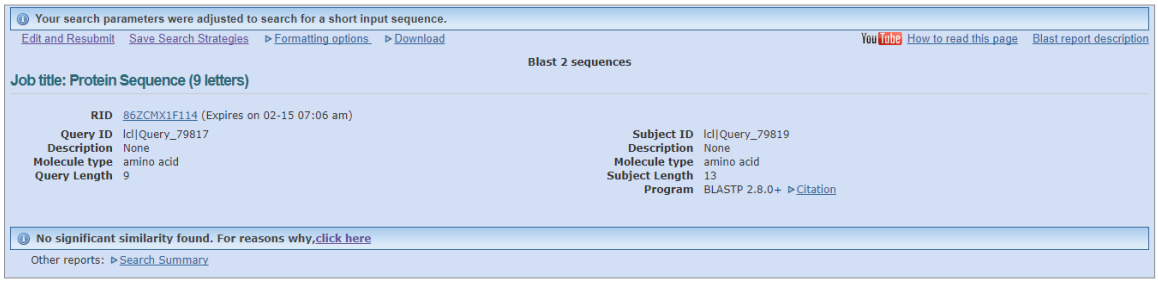
In this case both the global alignment and score are different from BLAST.

Local Alignment Output:

```
Score: 27

GTCTATCAC--------
    |||*|
----ATCTCGTATGATG
```

BLAST Output:



In this case both the local alignment and score are different from BLAST.

4. Discussion

Although the human hemoglobin and mouse hemoglobin alpha subunits come from different species, they have the same function with highly similar sequences. Humans and mice are mammals and they share an ancestor that probably existed 80 million years ago. It is likely that this ancestor had a cytochrome c protein, and the sequences above are the descendants of that primitive sequence, just as species is as a whole. Many of the protein sequences in mice and humans are similar for this reason.

We can observe a difference between our scores and the BLAST scores, because the BLAST score is derived from the raw alignment score, taking the statistical properties of the scoring system into account, whereas we calculate our score based on BLOSSUM62.

Also, in some cases where the lengths of the sequence are small, the BLAST doesn't find any similarities, because BLAST algorithm can align two sequences only if the length of the sequences are greater than a particular threshold.

5. References

1. Steven E. Brenner, Cyrus Chothia and Tim J. P. Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. PNAS 1998 May, 95 (11) 6073-6078.

2. Saul B. Needleman and Christian D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. J. Mol. Biol. (1970) 48, 443453

3. Nur'Aini Abdul Rashid', Rosni Abdullahl, Abdullah Zawawi Haji Talibl, Zalila AH2. Fast Dynamic Programming Based Sequence Alignment Algorithm. IEEE Explore 29th January 2007

4. João Carlos Setubal, PhD and Ruediger Braeuning, PhD. Chapter A05 Similarity Search: Bioinformatics in Tropical Disease Research: A Practical and Case-Study Approach.

5. https://www.ibm.com/developerworks/library/j-seqalign/

6. https://www.ncbi.nlm.nih.gov/books/NBK62051/

7. https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch&PROG_DEF=blastn&BLAST_PROG_DEF=blastn&BLAST_SPEC=GlobalAln&LINK_LOC=BlastHomeLink

8. https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome

9. https://en.wikipedia.org/