Dynamic Programming for Protein-by-Protein Matching

ZHOU Zhi-min, CHEN Zhong-wen

Department of Computer Science Science
Zhejiang Water Conservancy And Hydropower College,Hangzhou, China
E-mail: zhouzhm@zjwchc.com, chenzw@zjwchc.com

Abstract. Aligning distantly related protein sequences is a long-standing problem in bioinformatics and a key for successful protein structure prediction. A fast and valid algorithm can benefit the whole process of biology research. We first introduce an algorithm that given a certain evaluation function, will calculate the optimal alignment. Then we give a heuristic approach of the algorithm.

Keywords: algorithm, bioinformatics, protein sequence alignment

1 Introduction

1.1 Background

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.[1]

How to produce high-quality alignment of lengthy and extremely numerous sequences is the major difficulty facing bioinformatics researchers. Before fast algorithms such as BLAST and FASTA were developed, doing database searches for the protein or nucleic sequences was very time consuming by using a full alignment procedure like Smith-Waterman.[2] Various algorithms and programs are developed to solve these problems. This technology is also very useful in aligning other sequences in for example, natural language and financial data.

1.2 Our Result

Data set used in the illustration isCj1293 and HP0840 enzymes from the genome database of IMG (img,jgi.de.gov).

By applying multiple algorithms, we demonstrate a regular procedure of pairwise protein sequence alignment, and then compare their performances.

Before the introduction of any algorithms, we first want to make two concepts clear. The first one is Second one is pairwise and multiple alignment. The idea is easy, a pairwise alignment substitute a sequence to another while a multiple alignment to a number of sequences.

We compare 2 pairwise methods (dynamic programming and local search) and implement a multiple alignment task based on them.

Then we give a demo of showing how to query the whole genome of shewanella, a marine bacterium whose O-linked glycosylation pathway is not known, for protein similar to PseB.

2 Dynamic programming

Both global alignments (Needleman-Wunsch algorithm) and local alignment (Smith-Waterman algorithm) can be produced by applying dynamic programming.

Dynamic programming can be time consuming. In typical usage, protein alignments use a substitution matrix to assign scores to amino-acid matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other.

Figure1 depicts a Needleman-Wunsch alignment of the words "PELICAN" and "COELACANTH."[3]

1) *Initialization*

Construct the matrix. The number of rows and columns are the length of two sequences plus one.

2) *matrix initial configuration*

Initialize the matrix, with zero on the upper left corner, and count from upper left to lower right, apply the gap penalty one by one. And arrows are added to indicate the alignment direction.

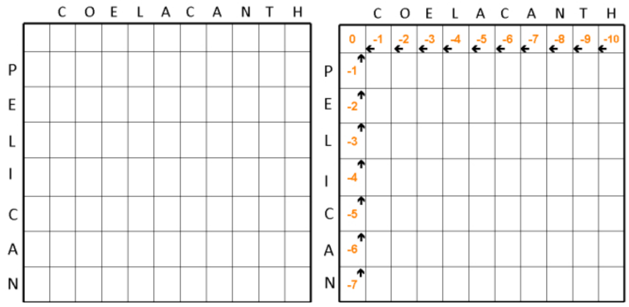


Fig. 1. Initialization of the substitution matrix and the filling of axis

3) Induction

Algorithm aligns the letters sequentially, if two letters match add one point, otherwise minus one point. The new score is calculated three times, with the square to the left, above and upper left. The minimal score is chosen as the final alignment.

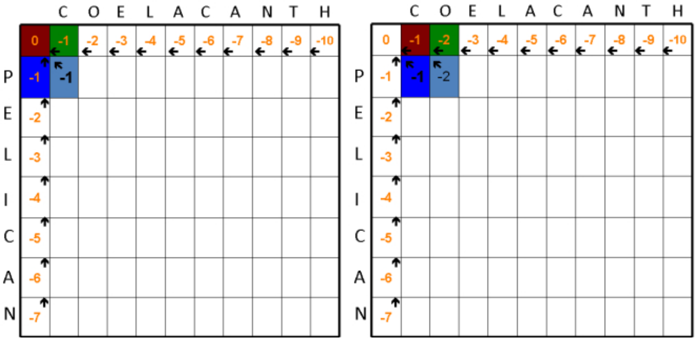
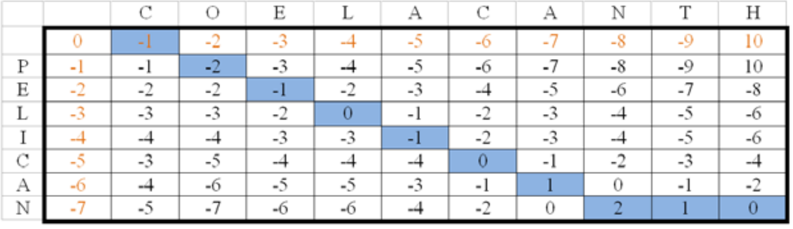


Fig. 2. Induction or filling of the alignment matrix.

4) Traceback

Once the matrix is completed, the optimal alignment is found from the lower right back to the beginning.



		C	O	E	L	A	C	A	N	T	H
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	10
P	-1	-1	-2	-3	-4	-5	-6	-7	-8	-9	10
E	-2	-2	-2	-1	-2	-3	-4	-5	-6	-7	-8
L	-3	-3	-3	-2	0	-1	-2	-3	-4	-5	-6
I	-4	-4	-4	-3	-3	-1	-2	-3	-4	-5	-6
C	-5	-3	-5	-4	-4	-4	0	-1	-2	-3	-4
A	-6	-4	-6	-5	-5	-3	-1	1	0	-1	-2
N	-7	-5	-7	-6	-6	-4	-2	0	2	1	0

Fig. 3. Traceback of the optimal complete alignment.

This algorithm meets the problem of large input scale, which requires memory of O(I2), where I is the input scale.

Dynamic programming can be applied to generate gapped alignments ith insertions and deletions. This algorithm is powerful and is sure to produce the optimal alignment corresponding to a certain scoring function. However, this method also requires lots of computational resources. Since the computational complexity is O(nm), with m, n be the lengths of each sequences, this algorithm could be too slow to be extended to align multiple sequences.

3 Local Search Alignment

As we have shown in the last section, dynamic programming cannot perform well when the protein sequence is long. Researcher Stephen Altschul and colleagues wanted to bypass these challenges and develop a way for databases to be searched quickly on routinely used computers. In order to increase the speed of alignment, the BLAST algorithm was designed to approximate the results of an alignment algorithm created by Smith and Waterman in 1981[5], but to do so without comparing every residue against every other[6]. BLAST is therefore heuristic in nature, meaning it has "smart shortcuts" that allow it to run more quickly[7]. However, in this trade-off for increased speed, the accuracy of the algorithm is slightly decreased.

BLAST is one of the most commonly used tools for local search alignment. The main idea of BLAST is that there are often high-scoring segment pairs (HSP) contained in a statistically significant alignment. BLAST searches for high scoring sequence alignments between the query sequence and sequences in the database using a heuristic approach that approximates the Smith-Waterman algorithm. The exhaustive Smith-Waterman approach is too slow for searching large genomic databases such as GenBank. Therefore, the BLAST algorithm uses a heuristic approach that is less accurate than the Smith-Waterman algorithm but over 50 times faster. The speed and relatively good accuracy of BLAST are among the key technical innovations of the BLAST programs.[8]

4 Progressive Alignment

Another method build upon the dynamic programming is used in multiplealignment. The idea is firstly align the 2 most closest sequences and then adds the next closest one iteratively. This algorithm progresses by dynamicly updating the positions of the indels.[9] If some part of the sequences is overpresented, this algoritm suffers from biased inference caused by the order of mofits.

This heutistic algorithm always gives good results.[10] The most reliable alignments are produced by aligning the most similar pairs of sequences. By this way, a hierarchical tree is constructed and tree analysis in unsupervised learning can be invested to find groups of alignment.

One widely used implementation of profile-based progressive alignment is the CLUSTALW program. CLUSTALW works in much the same way as Feng-Doolittle method except for its carefully tuned use of profile alignment methods.[11]

Algorithm:

- Construct a distance matrix of all N(N-1) pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity score to evolutionary distances using the model of Kimura.
- Construct a guide tree by a neighbor-joining clustering algorithm by Saitou &Nei.
- Progressively align at nodes in order of decreasing similarity, using sequence- sequence, sequence-profile, and profile-profile alignment.

ClustalW is unabashedly ad hoc (designed for this, not generalizable) in its alignment construction and scoring stage.[12]

The result given by ClustalW is shown in the following diagram.

References

1. Mount DM. 2004. Bioinformatics: Sequence and Genome Analysis (2nd ed.). Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY.. ISBN 0-87969-608-7.
2. Korf, I., Yandell, M., &Bedell, J., 2003, BLAST, O'Reilly Media Inc.ISBN: 9780596002992
3. Altschul, S. F., et al. Issues in searching molecular sequence databases. Nature Genetics 6, 119–129 (1994) doi:10.1038/ng0294-119
4. Taubs, G. Sense from sequences: Stephen F. Altschul on bettering BLAST. Science Watch11, 3–4 (2000)
5. Smith, T. F., & Waterman, M.S. Identification of common molecular subsequences. Journal of Molecular Biology 147, 195–197 (1981) doi:10.1016/0022-2836(81)90087-5
6. Altschul, S. F., et al. Basic Local Alignment Search Tool. Journal of Molecular Biology215, 403–410 (1990) doi:10.1016/S0022-2836(05)80360-2
7. Madden, T. The BLAST sequence analysis tool. In NCBI Handbook, ed. J. McEntyre and J. Ostell (National Library of Medicine, Bethesda, MD, 2005)
8. Korf, I., Yandell, M., &Bedell, J. BLAST: An Essential Guide to the Basic Local Alignment Search Tool (O'Reilly, Sebastopol, CA, 2003)
9. Altschul, S. F., et al. Gapped Blast and PSI-Blast: A new generation of protein database search programs. Nucleic Acids Research25, 3389–3402 (1997)
10. Collins, J. F., & Coulson, A. F. Applications of parallel processing algorithms for DNA sequence analysis. Nucleic Acids Research 12, 181–192 (1984)
11. Gish, W., & States, D. J. Identification of protein coding regions by database similarity search. Nature Genetics3, 266–272 (1993) doi:10.1038/ng0393-266 Gotoh, O., &Tagashira, Y. Sequence search on a supercomputer. Nucleic Acids Research14, 57–64 (1986)
12. Aboyoun, P.,2009. Sequence Alignment of Short Read Data using Biostrings. www.bioconductor.org/help/course-materials/.../MatchAlign.pdf