Computational Biology Final Project: A Practice on Multiple Sequence Alignment

Yufan Liu

School of Life Sciences, Tsinghua University, Bejing 100084, China

Abstract With the emergency of biological sequencing technology, sequence alignment has become an active research area in the field of bioinformatics. The biological functions of many DNA, RNA, or protein fragments are revealed by sequence alignment. There are mainly two types of sequence alignment are applied, pairwise sequence alignment (PSA) and multiple sequence alignment (MSA). In general, MSA is more advantageous and pervasive in analysis, which considers more sequences and could provide family relationship information. In this project, we used two sets of protein sequence data and performed the sum of pairs score algorithm and Feng-Doolittle algorithm respectively to solve questions, and obtained the result successfully.

1. Introduction

Sequences of DNA, RNA, and protein are the main research object of bioinformatics, and biology function is almost directly determined by sequence according to central dogma. Thus, one can obtain knowledge of structure, function, and interaction network via analysis origin sequences. Specifically, sequence analysis will be helpful for phylogenetic tree reconstruction, hidden Markov modeling, secondary and tertiary structure prediction, functional prediction, and PCR primer design. Among the procedure of sequence analysis, alignment is the most fundamental and important step. For this purpose, numerous algorithms and software were designed for pairwise or multiple sequence alignment, which will be stated later. However, for the same set of sequences, the different algorithm will lead to quite another result, thus the selection of algorithm carefully is essential for analysis and help to obtain meaningful biological function [1]. In this project, we focused on multiple sequence alignment to solve the given questions.

Multiple alignment sequence (MSA) refers to a kind of process of three or more biological sequences, mainly protein, DNA, or RNA. Generally, in MSA, the same amino acid, base or amino acid with lower substitution cost should be arranged in the same column as much as possible, if there's a majority of elements are aligned, which is meant possible homology in evolution, i.e., the same ancestor. MSA has many important utilities, especially in the field of comparative genomics, molecular evolutionary biology. MSA identifies conserve regions and functional motif among multiple sequences or species, and estimate the evolutionary divergence between sequences [2]. Given the role of MSA, biologically realistic alignment methods are needed. Algorithm for MSA is a complex question, basic dynamic programming is not feasible, therefore, most of the algorithms are an approximation, for this reason, most MSA algorithm is based on PSA or started with PSA to reduce the inconvenience of dynamic programming in a higher dimension. Algorithms can be divided into several classes like dynamic programming, heuristic algorithm, stochastic algorithm, divide-and-conquer algorithm, etc. [3], which we'll then discuss in detail later. For DNA or RNA sequences, the score to estimate the similarity between bases is simple: if bases in both sequences are identical, will be assigned a positive value, or negative, in reverse. But cases are different in protein sequences because different amino acids will have different substitution scores. For this reason, a substitution matrix is necessary for protein sequence alignment. The substitution matrices mostly consulted for protein are point accepted mutation (PAM) [4] and blocks substitution matrix (BLOSUM) [5].

Dynamic programming is one of the earliest methods for MSA, like the Needleman-Wunsch algorithm [6], MSA can be solved similarly. There are some slight differences between algorithms of

PSA and MSA, that is PSA need to consider the maximum of value top or left added by a gap penalty and value from diagonal added by match or mismatch score, whereas MSA need to consider values in more sequences, that is to say, MSA in three sequences will consider a 3D matrix. However, because of the large cost of storage and time, normal dynamic programming is impossible for MSA contains more than three sequences. Carrillo and Lipman proposed an optimized algorithm [7], which uses a sum of pairs (SP) score to find the best alignment score in a certain position, that is, every pair of the sequences will be aligned once, and a score is obtained by substitution matrix, then sum all the scores as an SP score, and the alignment with the best score will be set as the best alignment. This method takes all pairwise information into account and the number of sequences that can be aligned is lifted to ten. In our project, question 1 is solved by this algorithm and will be illustrated in detail in Method.

Since the limitation of dynamic programming, heuristic algorithms are then proposed, and most of the prevalent algorithms are heuristic. Star alignment is the earliest heuristic method for MSA. First, select a sequence c as the center of the star, in general, a star is the most similar to all of the rest of the sequences. For each sequence x_1, ..., x_k that index $i \neq c$, perform a pairwise sequence alignment method like Needleman-Wunsch global alignment, at last, aggregate alignment with the principle "Once a gap, always a gap". This is an important principle in heuristic algorithms that guarantees a relatively good alignment. The most widely used heuristic method with the best effect is the progressive alignment algorithm, which was first proposed by Hogeweg and Hesper [8] and then improved by Feng and Doolittle [9]. We used the latter one to solve question 2, which will be illustrated in detail in Method. The Feng and Doolittle algorithm calculates all pairwise alignments and score them, and uses scores to generate a distance matrix, then builds a tree called 'guide tree' based on the distance matrix, finally add all the sequences by pairwise alignment by the order determined by the tree. In practice, the alignment result is directly determined by the guide tree, thus this algorithm cannot find the best alignment but a better one. However, on the other hand, the iterative process in the method by realigning the sequences overcomes the drawback described below and greatly reduces memory usage and running time.

Other algorithm applied in MSA includes stochastic algorithm, which is useful for complex, poorly defined optimization problems, like a genetic algorithm, simulated annealing algorithm, and particle swarm algorithm, and dived-and-conquer algorithm, etc. Besides, the hidden Markov model (HMM) is also a popular model for MSA, it's a probabilistic model that takes all sequences into account, and a profiled HMM is proposed for MSA shows great power in database searching. However, an aligned set of sequences need to be prepared in advance for HMM training, thus it is not flexible for de novo multiple sequence alignment.

Based on the algorithms, a lot of software were developed, such as MULTAL, T-coffee, KAlign, and CLUSTAL based on progressive alignment; ProbCons based on probabilistic models like HMM; SAGA based on genetic algorithm, etc. In conclusion, MSA is an NP-complete problem, so it is impossible to find an exact global optimum. Besides, several hyper-parameters are needed for alignment algorithms. With the increase of computing power in past few years, novel methods like artificial neural networks are now possible to solve MSA problems effectively.

2. Definition of the problem

The algorithms we used in our project are mainly the Needleman-Wunsch algorithm, Feng-Doolittle algorithm, and unweighted pair group method with arithmetic mean (UPGMA).

The Needleman-Wunsch is an algorithm that is used to find the optimal global alignment between two sequences. The Needleman-Wunsch matrix for prefix alignment of a and b is defined as:

$$(D_{ij})_{0 \leq i \leq |a| \ and \ 0 \leq j \leq |b|}$$

with $D_{ij} := min\{w(u^*, v^*) | (u^*, v^*)\}$ is an alignment of prefixes$(a_1, \ldots, a_n, b_1, \ldots b_n)$. Note that w is the cost of aligning.

The Feng-Doolittle is used to align 3 or more sequences together, the problem is defined as, let $a^1, \ldots, a^n$ be N sequences. A multiple sequence alignment of sequences above is a matrix:

$$A = (A_{i,j})_1 \leq i \leq N, 1 \leq j \leq K,$$

where the rows correspond to the sequences, and columns correspond to the MSA columns.

The UPGMA is a clustering method that is used to create phylogenetic trees. The definition of the problem is: a phylogenetic tree is an undirected, connected, acyclic binary graph:

$$G = (V, E), V \; are \; nodes \; or \; vertices \; and \; E \subseteq V \times V \; are \; edges,$$

where the rows correspond to the sequences, and the columns correspond to the MSA columns.

3. Method

We used a sum of pairs (SP) score algorithm to solve question 1 and the Feng-Doolittle algorithm to solve question 2.

3.1 Question 1

For question1, the most important metric for alignment result is SP score, here, SP score is defined as:

$$SP = \sum_i S(m_i) = \sum_i \sum_{k<j} s(m_i^k, m_i^j),$$

where i is the column index of the sequence, s is the score function, which can be defined in the substitution matrix. In our project, the score is simply divided as exact match and mismatch, and scores are 3, -1, respectively, and a score of a gap is -2, and we need to calculate the maximum SP score. Here we denote scores as:

$$s(a, -) = s(-, a) = -2$$
$$s(a, a) = 3$$
$$s(a, b) = -1,$$

where a and b are bases in the protein sequence.

First, we initialized the boundary case. Because we need to maximize SP score, thus for a certain column, the minimal possible value is with a mismatch and two gaps, that is, -5. Then we used the Needleman-Wunsch algorithm to solve the dynamic programming question and record all the intermediate processes. In detail, there are seven cases in dynamic programming, we use 1 to 7 to record the alignment status in a list, which is denoted as:

In Case 7, all three sequences move forward with no gaps:

$$opt[i, j, k] = opt[i-1, j-1, k-1] + score(s_1[i-1], s_2[j-1]) + score(s_1[i-1], s_3[k-1]) + score(s_2[j-1], s_3[k-1])$$

In Case 6, 5, 4, two sequences move forward and the left one will add a gap:

$$opt[i, j, k] = opt[i-1, j-1, k] + score(s_1[i-1], s_2[j-1]) + gap * 2$$
$$opt[i, j, k] = opt[i-1, j, k-1] + score(s_1[i-1], s_3[k-1]) + gap * 2$$
$$opt[i, j, k] = opt[i, j-1, k-1] + score(s_2[j-1], s_3[k-1]) + gap * 2$$

In Case 3, 2, 1, only one sequence move forward and the rest of them will add a gap:

$$opt[i,j,k] = opt[i-1,j,k] + gap * 2$$
$$opt[i,j,k] = opt[i,j-1,k] + gap * 2$$
$$opt[i,j,k] = opt[i,j,k-1] + gap * 2,$$

where, opt is a list to record the best alignment score, and gap is the gap penalty, here we set -2. What has to be aware of is, add an item here, we used SP score to consider all the cases included in one column, such as, in case 6, 5, 4, we need to calculate the score of exact match and match with a gap, so the gap penalty was added twice. Cases are the same in case 3, 2, 1, but the score of two gaps are not allowed, so we set it to zero, which is not shown in the formula described above.

We then did a traceback according to the status list, and get a new list record reflecting the best alignment score, according to this list, the optimal alignment of sequences can be obtained. The code and results are saved in the appendix file and will be discussed below.

3.2 Question2

We used the Feng-Doolittle algorithm to align the given 44 protein sequences.

First of all, for every pair of sequences, we computed the alignment score using a pairwise alignment algorithm. For convenience, we used the Needleman-Wunsch algorithm to do so, and the gap penalty was set to -1. Considering the biological reality, the affine gap penalty is more suitable, thus one can use the Gotoh algorithm [10] instead. After pairwise alignment, a distance matrix of size 44*44 is generated by a given distance formula:

$$D = -logS_{eff} = -log\frac{S_{obs} - S_{rand}}{S_{max} - S_{rand}},$$

where S is the score generated by pairwise sequence alignment, and, $S_{max} = (S(a,a) + S(b,b))/2$ provides the best alignment score of sequence a and b, S_{obs} is the alignment score of sequence a and b, that is $S(a,b)$. S_{rand} is a random sequence alignment score which is calculated by the formula:

$$S_{rand} = \frac{1}{L_{a,b}}(\sum_{x\in\sum}\sum_{y\in\sum} s\,(x,y)\cdot N_a(x)\cdot N_b(y)) + N_g\cdot\alpha,$$

where, $\frac{1}{L_{a,b}}$ is the number of columns in the alignment of sequences a and b, $s(x,y)$ is the similarity score of bases, here we used PAM250 substitution matrix, $N_a(x)$ and $N_b(y)$ are number characters x and y in sequences a and b, respectively, N_g is the number of gaps in both sequences, and α is the gap penalty, here we set to -1. Besides, the values in trackback matrix will record where does the value come from, for convenience, we set all possible value from the top, left and diagonal direction to diagonal; set all possible value from the top and left to left.

Second, the resulting distances were used to create a phylogenetic tree. Here, we used the UPGMA proposed by Sokal and Michener [11]. The obtained phylogenetic tree was denoted as Newick format, and we used the score from leaf to root, to create the multiple sequence alignment progressively using the Needleman-Wunsch algorithm. Specially, all gaps marked as '-' will be replaced as a neutral 'X'. The newly joined sequence will be aligned with all aligned 'group' or sequences to obtain an optimal alignment score. The code and resulting files are saved in appendix files and will be discussed below.

4. Results and discussion

We applied the methods above to solve our problems. For question 1, we obtained an optimal alignment using an SP score. The given sequences are hemoglobin from cow and innkeeper worm, and flavohemoglobin from a kind of bacteria. It's can be found that hemoglobin has higher alignment performance, whereas flavohemoglobin is not. Therefore, hemoglobin shows homology to some extent. The situation is more obvious in question 2. In the resulting alignment, we can observe that there are some regions of high identity in sequences, shown in Fig. 1. According to the protein name, we can find out, that all of them performing the job of transporting oxygen, thus the regions of the sequences may play an important role or the key motif of oxygen-transporting.

Fig. 1 Alignment region with high similarity

We also plot the phylogenetic tree in Fig. 2, the leaves are noted as an order of the sequences in the origin file. The sequences with smaller distance values, like ID2 and ID3 in Fig. 2, are mostly isoforms or sequences with a close biological relationship. Other results including distance matrix, sequence alignment of questions 1 and 2, guide tree in Newick format, and plot are wrapped in a file and described in Appendix. What's more, the methods we used are greatly dependent on hyper-parameter. For an instance, things will be different if we use an affine gap penalty or BOLOSUM62 substitution matrix, rather than PAM250 to perform the Feng-Doolittle algorithm. We get this conclusion from a test on http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Feng-Doolittle by using some random sequences, for this purpose.

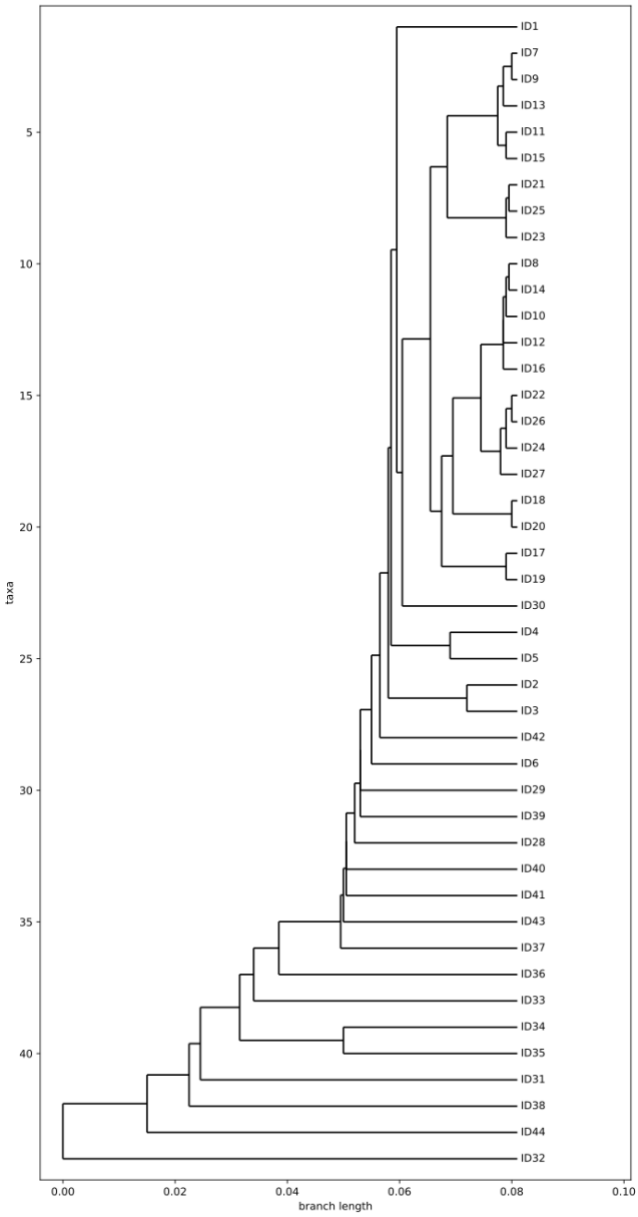

Fig. 2 Phylogenetic tree parsed by distance matrix

5. Conclusion

In this project, we focused on multiple sequence alignment methods and used a sum of pairs score to solve question 1, Feng-Doolittle algorithm to solve question 2, respectively. Then we analyze the alignment output and distinguish the sequences that are homogeneous from other sequences. Our results show that MSA is indeed a powerful method for reveal conserve regions and functional motifs, It is of great significance to sever as a fundamental part of bioinformatics analysis.

Acknowledgment First I'll appreciate the wonderful lecture given by Professor Deng, which inspires me to explore the biological function of sequences using computational methods. What's more, thanks to our team members Jingyi Wang and Yongzheng Jia, whose efforts make our project finished. They proposed a lot of advice for our project and we talk about our program and complete our code together at last.

Appendix Results of question 1 and question 2 are on a zipped field named 'computational_biology_final_lyf' and denoted as 'q1' and 'q2'. In q1, 'Score.py' is code in python and 'question1align.txt' is the multiple sequence alignment result. In q2, the suffix of the file with '.py' is code in python, where 'friend.py' is a class implementing sequences parsing and substitution matrix; 'needlemanwunsch.py' is a class implementing Needlman-Wunsch algorithm; 'uwpgma.py' is a class implementing UPGMA algorithm; 'main.py' is the main function to implement Feng-Doolittle algorithm. Besides, 'distancematrix.xlsx' records distance matrix in a table, 'guidetree.txt' and 'guidetreeplot.pdf' are guide tree in Newick format and the plot respectively, 'sequencealignment.txt' is the multiple sequence alignment of given 44 protein sequences, 'run.log' is the log file when running.

References

[1] B. Chowdhury and G. Garai, A review on multiple sequence alignment from the perspective of genetic algorithm, Genomics. 109 (2017) 419-431.

[2] S. Kumar and A. Filipski, Multiple sequence alignment: in pursuit of homologous DNA positions, Genome Res. 17 (2007) 127-135.

[3] M. Chatzou, C. Magis, J. M. Chang, C. Kemena, G. Bussotti, I. Erb, C. Notredam, Multiple sequence alignment modeling: methods and applications, Brief Bioinformatics. 17 (6) 2016 1009-1023.

[4] M.O. Dayhoff, R.M. Schwartz, B.C. Orcutt, A model of evolutionary change in proteins, in: M.O. Dayhoff (Ed.) Atlas of Prot. Seq. and Struct., vol. 5, National Biomedical Research Foundation, Washington, DC 1978, pp. 345–352.

[5] S. Henikoff, J.G. Henikoff, Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. 89 (1992) 10915–10919.

[6] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48 (1970) 443–453.

[7] H. Carrillo, D. Lipman, The multiple sequence alignment problem in biology, SIAM J. Appl. Math. 48 (1988) 1073–1082.

[8] P. Hogeweg, B. Hesper, The alignment of sets of sequences and the construction of phylogenetic trees: an integrated method, J. Mol. Evol. 20 (1984) 175–186.

[9] D.F. Feng, R.F. Doolittle, Progressive sequence alignment as a prerequisite to correct phylogenetic trees, J. Mol. Evol. 25 (1987) 351–360.

[10] O. Gotoh, Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments, J. Mol. Biol. 264 (1996) 823–838.

[11] R.R. Sokal, C.D. Michener, A statistical method for evaluating systematic relationships, Univ. Kans. Sci. Bull. 28 (1958) 1409–1438.