Analysis and Improvements on Multiple Sequence Alignment Algorithms

Beichen Zhang, Haitao Lin

Motivation: *Multiple sequence alignment is an important technique in biology. We want to analyze some algorithms about it and come up with some ideas to improve them. We implement two progressive algorithms, CLUSTAL and TCOFFEE, to analyse their accuracy and time complexity. Based on the two algorithms, we also come up with two improvements and implement them, which are refinement and quick TCOFFEE.*
Results: *CLUSTAL is better on time and TCOFFEE is better on accuracy. On test cases, the new algorithms theocratically have good improvements in time complexity and accuracy.*
Availability: *We implement the algorithms by python and the source code are freely available for download at github.com/xiaolinAndy/genomics.*
Contact: *bz2309@clumbia.edu, hl3054@columbia.edu*

Introduction

A multiple sequence alignment (MSA) is a sequence alignment of three or more biological sequences, generally protein, DNA, or RNA. Dynamic programming technique is a method to identify the globally optimal alignment solution, while it's proved to be a NP problem. Therefore, we usually make a tradeoff between accuracy and time and try to change the balance for different kinds of alignments. In the project, we implement CLUSTAL and TCOFFEE algorithms and make some improvements on them.

In CLUSTAL, we first construct a distance matrix by pairwise dynamic programming, then construct a guide tree by NJ algorithm. Finally, along the tree, we can align them from leaves to root and get the multiple alignment.

For TCOFFEE, it's similar to CLUSTAL and consists of constructing distance matrix, building tree and aligning sequences, but in TCOFFEE we have to compute extensions for each pair and use them in progressive alignment step.

Improvement

1. Refinement

Our first improvement on the algorithms is to introduce an additional step called refinement. This idea comes from the algorithm called MUSCLE, which is another algorithm using little time to get a rather good alignment. The refinement follows the steps as below: First we need to finish the normal algorithm and have a multiple alignment. Then we choose an edge in the phylogeny tree and cut if off to get two subtrees. Since each of the subtree has its own alignments, we try to re-align these two sets of sequences to form a new multiple alignment. By testing it on SP score, we choose to save this change or not according to having gain or not in the new alignment. At last, we go back to the first step and try another edge, keeping the same procedures. Usually after several epochs of refinement, we can obtain a much better alignment than the previous one. In the traditional progressive alignment methods, the alignments come at first usually are not global optimized but local optimized, and it cannot be fixed later until the end. By using refinement, this problem is solved in some extent since the biologic meaning of refinement can be considered as fixing the bad alignments appearing in the front of the progressive alignment step.

2. Quick TCOFFEE

As we know, extension in TCOFFEE can avoid local optimum while it takes too much time, and weight in CLUSTAL always doesn't work well. In order to reduce the time complexity, we come up with an idea to improve TCOFFEE, which is called quick TCOFFEE.

In this algorithm, instead of calculating the extension, we build the phylogeny tree first, then we use the phylogeny tree to cluster the sequences and split them into some sets in which the sequences are similar. For each set, we will calculate extension only by the sequences in the set and align them by their extension. Out of the sets, we will use dynamic programming to align all the sets to the root node and get the final alignment. In TCOFFEE, extension takes $O(n^3l^2)$, but in this algorithm, suppose we get k sets by clustering

and the k^{th} set has n_k sequences, its time complexity is

$$\begin{aligned} & n_1^3 + n_2^3 + ... + n_k^3 \\ = & \sum_{i=1}^{k} n_i^3 \\ \approx & O(n^2 l^2) \end{aligned}$$

It's a great improvement in time complexity and theoretically, this algorithm works better when the species are far from each other in the phylogeny tree.

Analysis

We run both algorithms on many different kinds of data in order to make comprehensive analysis.

First we make comparisons on two classic algorithms: CLUSTAL and TCOFFEE. After running them on both short and long data, we conclude that TCOFFEE always need more time than CLUSTAL and the time difference becomes bigger as the number of sequences goes up.

Meanwhile, the longer running time can bring us better alignment than CLUSTAL in the most cases. Only when testing them by simulated data, the advantage of TCOFFEE is not much presented.

As for our improvements, we can receive a lot of information from Figure 1. Part a and b show the refinement performance on the two algorithms. There is an obvious, continuous decrease of SP score in CLUSTAL as the number of refinement epochs rises up. This trend stops at about 30 rounds. However, things are totally different in CLUSTAL as its score stops to decrease at about only 5 rounds and the decrease range is also far smaller than CLUSTAL. Besides, the time that CLUSTAL need to refine is also less than TCOFFEE, which suggests that it is a good way to use refinement on CLUSTAL but not good on TCOFFEE. In our speculation, we think that the main reason leading to this result is the function of refinement and extension in TCOFFEE are similar.

Part c and d reveals the performance of our quick TCOFFEE algorithm. From part c, the time that extension holds for near half of the time when the number of sequences is very big, such as 150 in the chart. However the extension part in quick TCOFFEE only occupies a very low level of time compared with other steps, which leads to more than 50 percent of decrease on time costs. Besides, the loss of accuracy on quick TCOFFEE is acceptable according to part d. From the comparison among CLUSTAL, TCOFFEE and quick TCOFFEE, we can conclude that quick TCOFFEE always present a middle result. It is much better than CLUSTAL and a little bit worse than TCOFFEE. But by considering the time it saves than TCOFFEE, its performance is still pretty considerable.

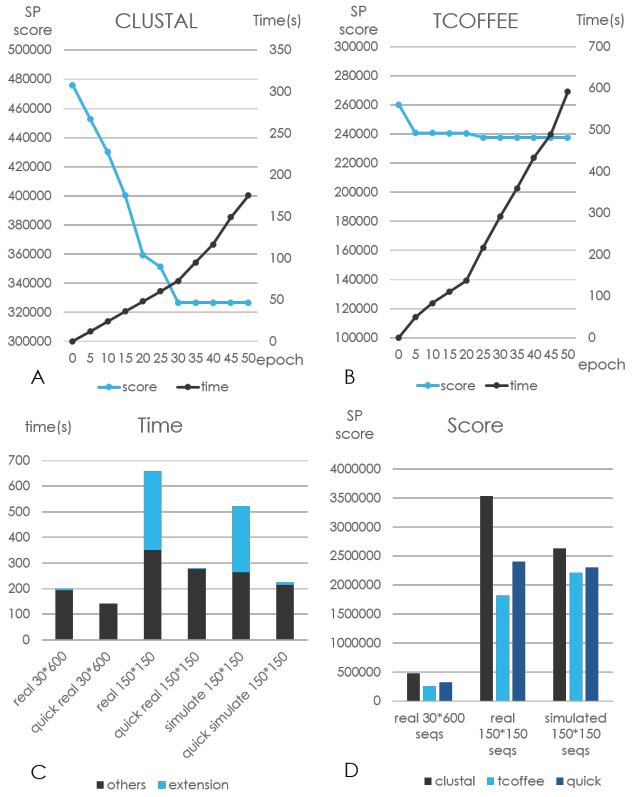

Figure 1: *The comparisons between classic algorithms and improved ones*

Conclusion

According to the result of algorithm and our analysis, we conclude that our improvements on the two algorithms take effects. The refinement step has great improvement on CLUSTAL algorithm with some worthy time costs. The quick TCOFFEE algorithm we involved can solve the time problem when we need to align a huge number of sequences with a little less accuracy than TCOFFEE but way much better than CLUSTAL.