Lab report: Reducing noise in protein multialignments

Ha Anh Tuan Nguyen
euSYSBIO student, KTH
hatng@kth.se

Hoang Son Nguyen
euSYSBIO student, KTH
hsnguyen@kth.se

***Abstract*—Using multiple sequence alignment is the most common method to build a phylogenetic tree. However, sometimes the alignment may contains some noisy columns that could reduce the accuracy of the constructed tree. For this reason, the very first step in constructing the tree is removing noise from the alignment. In this project, we implement a very simple noise reducing method to measure the efficiency of this method. As the result, many distances between constructed trees and reference trees in the given test data have been lowered.**

I. Introduction

Building phylogenetic tree or evolutionary tree is a well-known problem in bioinformatics in recent year. Phylogenetic tree is a branching tree that has organisms as its leaves and some joints between these leaves. Using the phylogenetic tree, scientists could determine the relationships between various biological species. For instance, two species that share the same root in phylogenetic tree are believed to have same ancestor.

One of the most popular methods to construct a phylogenetic tree is using the pairwise distances between sequences after the alignment procedure. The alignment procedure is performed by multiple sequence alignment (MSA). MSA is a sequence alignment of three or more DNA, RNA, or protein sequences. Between aligned sequences, there are three possible mutations at each column, they are: insertion, deletion, and substitution. However, as the insertion and deletion cannot be classified clearly, scientists often call these two mutations as indel mutation. Finding the optimal result for MSA is impossible because it is a np-complete problem. For this reason, many algorithms have been introduced to solve MSA problem. For example, One of the most common algorithms is progressive construction which continuously combines two alignments (at the beginning, a single sequence is pretended to be an alignment). This algorithm is used by Clustal[1] and T-Coffee[2]. Another well-known method is MUSCLE[3]. MUSCLE uses a improvement of progressive algorithm that update the distance of two alignments iteratively, therefore, it is classified as an iterative refinement method. Furthermore, even Hidden Markov Model could also be used to solve MSA problem as it has been used by ProbCons[4].

Other than the inability to obtain the optimal results, MSA algorithms sometimes may produce some noisy columns in the results. For instance, there could be some column that has mostly indel mutations. As the consequence, those noisy columns could affect the tree construction procedures and that lead to the differences between the constructed tree and the expected tree or the reference tree.

In this project, we would like to experiment a very simple method to remove the noisy columns in MSA. After that, the distance between reference tree and constructed tree will be calculated to verify the significance of those removals.

II. Methods

A. Reducing noisy column

A noisy column is determined by three following characteristics:

- There are more than 50% indel
- At least 50% of amino acids are unique
- No amino acid appears more than twice

If any column has one out of those three characteristics, it is considered as a noisy column.

To implement this definition, we created a very simple python script that takes a Fasta alignment file as the input. After that, the program will verify every columns in the input alignment. If a noisy column is detected, it will be removed by our program. Finally, the program writes the refined alignment as a single file that has Fasta format.

B. Phylogeny reconstruction

Each of the multi-alignment is then used to formulate a corresponding phylogenetic tree. This kind of tree reflects the diversities of protein sequences based on their alignments. In this project, only simple non-weighted trees are created thanks to two modules `fastprot` and `fnj` of FastPhylo package.

Firstly, `fastprot` is invoked to calculate the distance matrix of protein sequences by taking the multi-alignment fasta file as the input. This module utilized a fast algorithm of distance estimation [5]. After having the distance matrix, `fnj`, which stands for fast neighbor joining [6], is used to reconstruct the phylogeny. The tree is produced in Newick tree format as the last result of this step.

C. Evaluating the reconstructed trees

To evaluate the efficiency of the trimming process, the phylogenies from the previous step are needed to be measured. In our work, the symmetric distance is used to estimate the difference between two phylogenetic trees. This metric is the sum of the number of splits found in one of the trees but not the other and fortunately it is available in Dendropy package of Python.

D. Storing data in SQLite file

We use SQLite to store the data of distances between constructed trees and reference trees. The schema of the distance table is:

```
CREATE TABLE Distances (
        type TEXT,
        MSA_id TEXT,
        original_distance float,
        reduced_distance float
)
```

Where "type" is the symmetry or asymmetry property of the given MSA, "MSA_id" is the id of MSA that is created according to its file name, "original_distance" and "reduced_distance" are the distances between constructed tree and reference tree before and after noisy columns are removed respectively.

III. Experimental results

A reduced dataset from TrimAI [7] are used in experimental process. There are different subsets of data, each based on either a symmetric or asymmetric reference tree. From a reference tree, 300 alignments are created by evolving sequences along the reference tree with a fixed parameter indicating the average amount of mutations per site in the sequences. There are three possible values of this parameter: $0.5, 1.0, 2.0$, results in six subsets of multi-alignments.

For every alignments of the dataset, another noise-reduced version is created by the trimming method above. Then two phylogenies are reconstructed consequently and by comparing these trees with the given reference tree, we could understand the usefulness of our work.

Ultimately, for each of six subsets, there are 300 pairs of distances calculated correspond to 300 pairs of phylogenetic trees created.

As shown in Figure 1, the advance of reduced noise alignments are not too impressive when the two types of distance line are not really separated. There are also the cases when the reduced noise alignments make it more difficult to recover the phylogenies when their distances to the reference tree are greater than ones from the originals. The graph also reflect the difficulies of different datasets. We could see that the alignment inferred from symmetric tree are always easier for reconstruction than ones from remaining topology. The average number of mutation per site of sequences is also critical factor when it is usually inversed proportional to the coresponding numbers of success recoveries (distance equals to zero).

However, by looking at the statistical means on Table I, we could infer that most of the time distances to the reference trees are reduced after applying noise-reduced process. There is only one case when the average distance by doing trimming is greater than doing nothing, however the difference here is not significant.

Furthermore, the histogram of distances over the whole dataset shown in Figure 2 indicates the overall positive results.

TABLE I: Average distances over 300 samples.

Method / Dataset	Original	Noise-reduced
Symmetric 0.5	4.160	4.100
Symmetric 1.0	5.013	4.887
Symmetric 2.0	6.960	6.540
Asymmetric 0.5	7.633	7.587
Asymmetric 1.0	9.427	9.447
Asymmetric 2.0	12.34	11.96

We could say that noise-reduced alignments have higher possibilites in successful reconstruction of phylogenies.

IV. Discussion

The method used in noise trimming is quite simple and not flexible enough for different datasets. There still exists the case when applying this method make the alignments even more difficult for phylogenies reconstruction. One possible solution is to study characteristics of alignments themselves and assign the significant scores to the columns before removing them prematurely.

However, in common, this simple method do help us in reducing the distances in almost of the cases. Also thanks to its simpleness, the implementation is not complicated and its resources requirement is low. Because of that, this method is considered useful for the minor cases and could be used as the very preprocessing step in more complicated problems.

References

[1] R. Chenna, H. Sugawara, T. Koike, R. Lopez, T. J. Gibson, D. G. Higgins, and J. D. Thompson, "Multiple sequence alignment with the clustal series of programs," *Nucleic Acids Res*, vol. 31, pp. 3497–3500, 2003.

[2] C. Notredame, D. G. Higgins, and J. Heringa, "T-coffee: a novel method for fast and accurate multiple sequence alignment," *Journal of Molecular Biology*, vol. 302, no. 1, pp. 205 – 217, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022283600940427

[3] R. C. Edgar, "Muscle: multiple sequence alignment with high accuracy and high throughput," *NUCLEIC ACIDS RES*, vol. 32, pp. 1792–1797, 2004.

[4] C. B. Do, M. S. P. Mahabhashyam, M. Brudno, and S. Batzoglou, "Probcons: Probabilistic consistency-based multiple sequence alignment," *Genome Res*, vol. 15, pp. 330–340, 2005.

[5] I. Elias and J. Lagergren, "Fast computation of distance estimators," *BMC Bioinformatics*, vol. 8, p. 89, 2007.

[6] ——, "Fast neighbor joining," in *Proc. of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, ser. Lecture Notes in Computer Science, vol. 3580. Springer-Verlag, July 2005, pp. 1263–1274.

[7] S. Capella-Gutirrez, J. M. Silla-Martnez, and T. Gabaldn, "trimal: a tool for automated alignment trimming in large-scale phylogenetic analyses," *Bioinformatics*, vol. 25, no. 15, pp. 1972–1973, 2009. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/25/15/1972.abstract

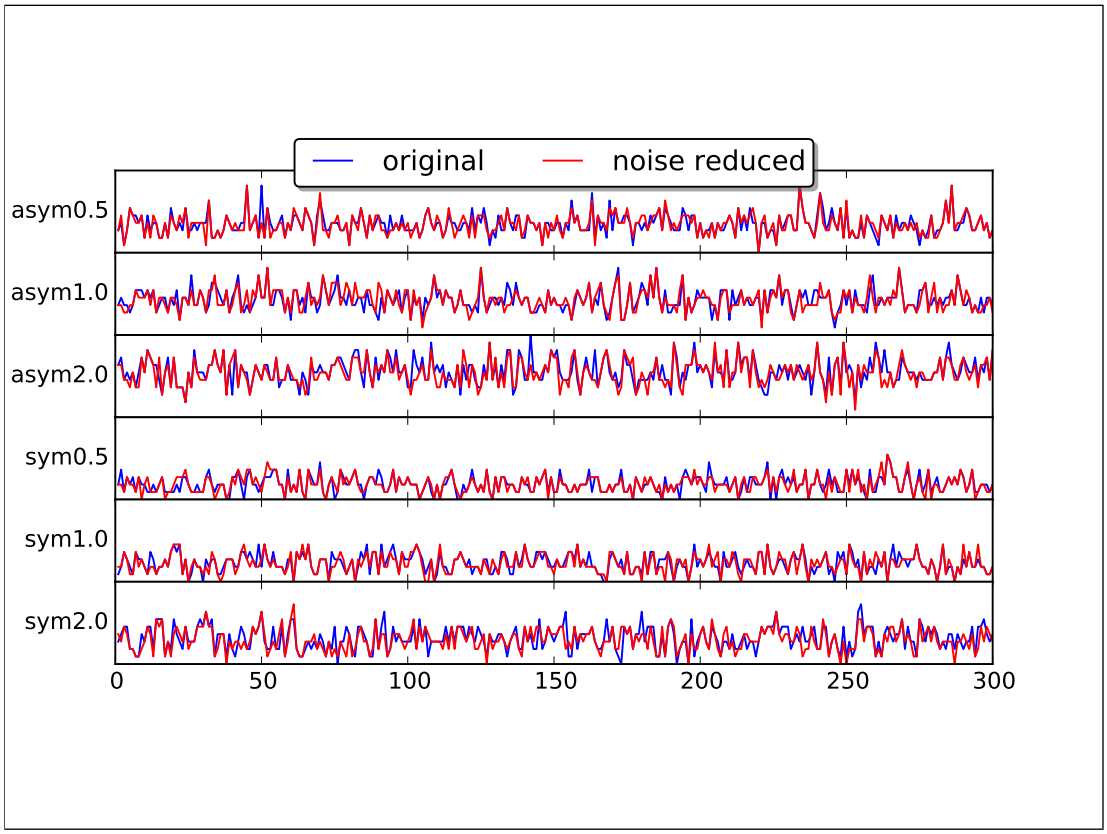
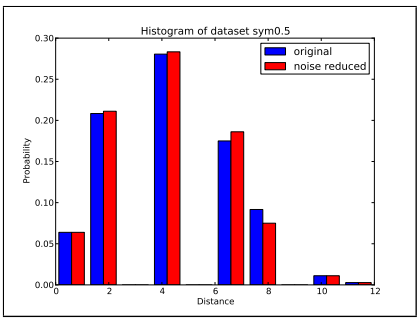
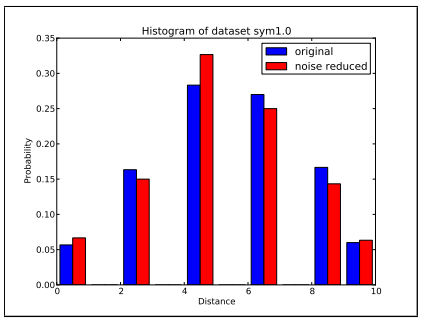


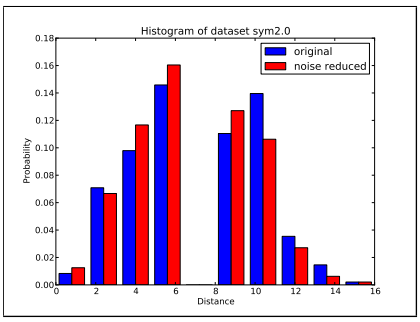
Fig. 1: Line graph showing distances from reference tree to trees recovered by both original and noise reduced alignments.



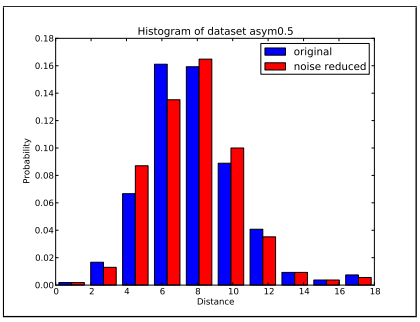
(a) Symmetric 0.5



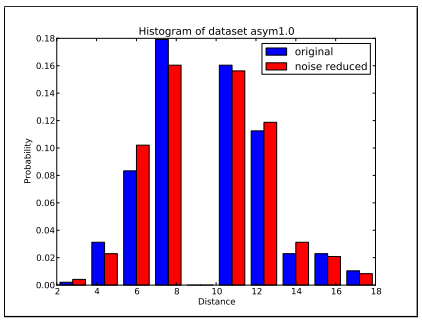
(b) Symmetric 1.0



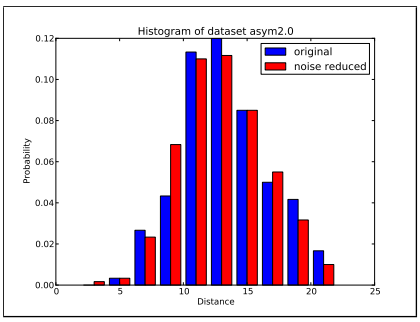
(c) Symmetric 2.0



(d) Asymmetric 0.5



(e) Asymmetric 1.0



(f) Asymmetric 2.0

Fig. 2: Histogram of distances.