Reducing noise in protein multialignments

Libo Xu

Program: TMLEM

Abstract

Multiple sequence alignment (MSA) is a sequence alignment of three or more biological sequences, which is protein in this project. However, some multialignments include too much noise, i.e., the regions that are not inherited or that have evolved too fast. Hence, I have written a Python program that removes the noise in the test data. To evaluate its performance, I have acquired the symmetric difference between test data (the original alignments) and reference tree and that between filtered data and reference tree. After the analysis of the results, I have concluded that it is not worthwhile to use multialignment noise reduction.

Introduction

Multiple sequence alignment (MSA) is used mainly for inferring homology to research the evolutionary relationships, predicting the secondary and tertiary structure of protein, estimate the total number of proteins folding types, analyzing sequences of genomes, etc. In order to avoid the problems caused by the 'bad' columns, we can make use of tools such as GBlocks or TrimAl to reduce them. In this case, I developed two programs, one for removing the noise and one for evaluating the impact of the first program on phylogeny inference.

Materials

According the project materials, a column is considered noisy if it meets at least one of the three conditions: there are more than 50% indels, at least 50% of amino acids are unique and no amino acid appears more than twice.

Methods

My project consists of two programs, which including three parts. The first program is 'filter.py', which takes the test data as an input and reduces the noise in them and then outputs the alignments without noise into a new directory named 'result'. The other one, 'analyse.py', contains two parts. Firstly, for one alignment file in the test data, the Fastphylo package is applied to infer two trees from its original alignment and the corresponding noise-reduced alignment. Then these trees are respectively compared to the reference tree to generate the symmetric difference by Dendropy package. Next, the difference of these two values is calculated. The input of this program is the sub-directories, such as 'asymmetric_0.5' and these three values of each file in the directory are stored in three lists in order to be easily analyzed in the second part that will be detailed in the result section of this report.

Discussion about Stafford Noble ideas and Controls

During the process of the project, I kept on writing the notebook of my work every work day as the Stafford Noble's ideas said. Also I put the files used in the project into different directories, such as test data in 'data (appbio11)', output in 'result', scripts in

'src' and notebook in 'doc'.

Besides, like the paper mentioned, if the format of input file's content was wrong or the sequence was empty, my program would print a message to standard error and then exit with a non-zero exit status. This is also the controls to establish that my results are correct.

What's more, I didn't use the version control software because this project didn't have many scripts and data files and I would save my scripts to the Google documents every time finish the work of that day.

Results

	asymmetric_0.5	asymmetric_1.0	asymmetric_2.0	symmetric_0.5	symmetric_1.0	symmetric_2.0
frequency of orignal	1	0	0	23	16	4
frequency of reduced	1	0	0	21	20	8
average difference	0.07	0.07	0.39	0.07	0.15	0.37
proportion (%)	15.33	19	31	15.33	20.33	31.67

In the table, the first row is the frequency of reference tree recovered from original alignment and the second row is the frequency of reference tree recovered from noise-reduced alignment. The way to recognize the recovery is that the symmetric difference is zero. The data shows how many zeros are in the 300 values in each sub-directory.

The third row is the average of the difference of two corresponding symmetric difference of each sub-directory.

The forth row is the proportion of the case where the symmetric difference of the tree after noise removal is smaller than that of the original tree, i.e., the reducing of noise is effective.

Conclusion

From the results in the table above, firstly we can see that frequency of recovering reference tree is apparently higher in the symmetric ones than that in asymmetric ones. Secondly, average difference is all positive, which means the symmetric difference of noise-reduced alignments is lower than that of original alignments in average. Hence, we can say that the noise reducing program works in general. Besides, the value of average difference and proportion are increasing as the amount of mutations per sequence position increases, which indicates that the filtering process is more effective when there are more mutations. However, the average difference and the effective proportion are very low, even in the '_2.0' case. Therefore, all in all, it is not worthwhile to use multiple sequence alignment noise reduction.