# An alternative approach for Noise reduction in Protein Multialignments

**NamrataBajpai, ShaliniDasgupta, Minh Khoa Nguyen**
**School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden**
**Correspondence to: namrata@kth.se, shalinid@kth.se, mknguyen@kth.se**

## Abstract

Noise reduction is required to establish good phylogenetic inference in protein multi alignments. TrimAl and Gblocks are well known applications for trimming unnecessary information while preserving important information in biological alignments. The unnecessary information is deemed as noise and an approach for noise removal was proposed and compared with these tools. To achieve the goal, a collaborative project was set up online. TrimAl with heuristic algorithm, Gblocks with default parameters and our proposed noise reduction method were applied to remove noise from the same input data. Phylogenetic trees are then inferred for the noise-reduced data. Results showed that phylogenetic inference is difficult on increasing complexities of the input data and our proposed approach indeed showed better performance in reducing noise compared to TrimAl and Gblocks for the given input data

## Author Summary

Among the different methods to infer phylogenetic trees, one is maximum likelihood. To show a multi alignment and its similar sets, phylogenetic inference is made. It represents the closer relatives. There are many algorithms and tools to infer phylogenetic inference but the choice of tool depends on three main factors which are time, accuracy and a clear interpretation. The raw data do not give a transparent picture of phylogenetic relationship. Hence, it is important to process the data and remove noise or unwanted regions beforehand.
In the present work, a comparison is made about tree recovery of the processed (noise reduced) and unprocessed (noisy) data. This comparison is utilized to decide whether noise removal is really necessary. Further, the result obtained via the proposed algorithm is compared with two prominent tools for noise reduction: TrimAl and Gblocks to determine the efficiency of proposed algorithm.

## Introduction

Noise in protein multi alignments can be defined as regions in homologous proteins that are not conserved, or regions that have evolved so fast that obtaining a correct alignment for these regions has become impossible. Some researchers prefer to remove these noisy regions to obtain meaningful, evolutionarily conserved alignments whereas others keep them in order to prevent loss of important sequence information. In this work, an algorithm for noise reduction was implemented on a given set of aligned protein sequences. The results obtained were compared with TrimAl[1] and Gblocks[2], two automated tools available for trimming ‘bad’ columns from alignments. Phylogenetic trees were inferred for the original data set (input data). Similarly ,the trees for the trimmed alignments obtained from TrimAl, Gblocks and the proposed algorithm were also generated. Statistical comparison was then performed between the original reference trees and the noise reduced trees to evaluate if noise reduction had a significant effect on the alignment quality as well as on the phylogenetic trees generated.

**Material and Methods**

In this study, a method for noise removal from a protein alignment was implemented. Our approach and two applications TrimAl and Gblocks were applied on the same input data to get different noise-reduced data. Phylogenetic trees were then generated for both the input data and the noise-reduced data. These trees were then compared against the reference trees to see how many recovered trees were obtained from each noise-reduced method. The coding and data processing were done in Ubuntu environment and Python. Figure 1 shows the overview of the methodology in this study

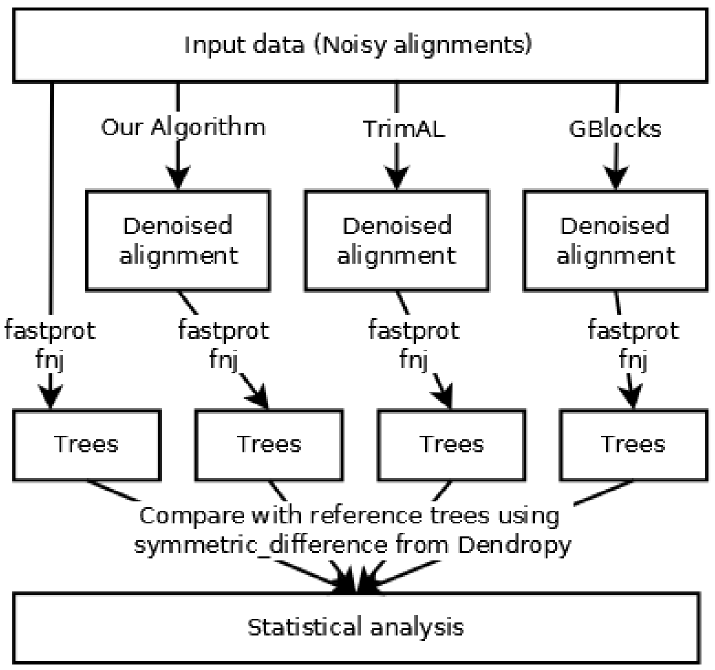


**Figure 1:** Overview of the methodology of the project

**Input data descriptions**

The input data comprise of six categories: asymmetric 0.5, asymmetric 1.0, asymmetric 2.0, symmetric 0.5, symmetric 1.0, and symmetric 2.0. Each category contains one reference tree and 300 alignments. The alignments in each category were aligned using muscle from generated sequences evolving along the reference tree (asymmetric or symmetric) at the mutation rate 0.5, 1.0 or 2.0 (average amount of mutations per site in the sequences) indicated in the category’s name.

**Our approach for Noise removal**

In our algorithm, a column in one protein alignment is considered noisy and hence removed if one of the following conditions is fulfilled

- There are more than 50% indels i.e. letter “-“
- At least 50% of amino acids are unique, i.e. the number of different amino acids is equal or greater than the total number of amino acids in the column
- No amino acid appears more than twice

Python was used for the algorithm implementation

**Comparison of our algorithm with TrimAl and Gblocks**

The input alignments were processed by our algorithm, TrimAl and Gblocks to remove noise. The parameter defined for TrimAl was ‘automated1’ which is the heuristic approach. Gblocks was run on its default value settings.

The noise-reduced alignments were then processed by the applications fastprot and fnj provided by Fastphylo tool [3] to generate phylogenetic trees in newick format. Trees from the original data (noisy alignments) were also generated in the same manner.

Each resulting tree is compared against the reference tree in the same category using symmetric difference algorithm provided by Dendropy [4] to see whether it recovers the reference tree, i.e., whether it is the same as the reference tree.

A statistical analysis was done for quantifying the effectiveness of our algorithm and comparing with TrimAl and Gblocks. Unix script files and Python were combined to automate the data processing.

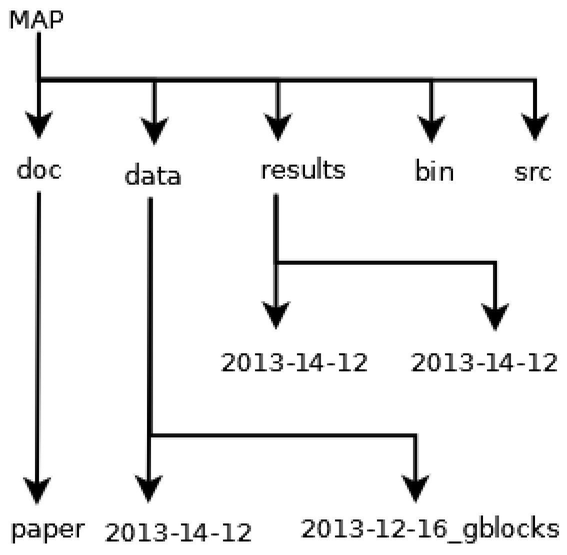


**Figure 2:** The main folder structure of the project

**Organization of the project**

The whole project was organized according to Noble's suggestion[5]. A project is set up on Github website at https://github.com/namrata007/MAPfor collaboration among the members. Figure 2 shows the core folder structure of the project. The root folder's name is MAP. Under MAP are folders: doc, data, results, bin and src. Input data, processed data or experiments reside in MAP/data folder chronologically. The lab notes which contain the elaboration of experiments, observations and summary of results reside in MAP/results folder also in a chronological manner. In this way one can easily find out which lab note corresponds to which experiment based on the date. The lab notes were created using various editors such as Microsoft Word, Libre Writer, etc. MAP/bin folder contains all the scripts, batch files or compiled codes. MAP/doc contains external sources and miscellaneous materials. MAP/src contain source codes

**Results and Discussion**

Noise-reduced data with the proposed algorithm show the greatest number of recovered trees compared with the other algorithms as well as with the original data for both asymmetric and symmetric categories.

This symmetric distance between two trees forms the basis of statistical computations done to retrieve the value added by noise reduction in phylogenetic inference.

| Types of trees | Mean Value | Standard Deviation | % of Reference trees recovered |
|---|---|---|---|
| Asymmetric 0.5 | 7.64 | 2.90 | 0.34 |
| Asymmetric 1.0 | 9.43 | 2.94 | 0 |
| Asymmetric 2.0 | 12.34 | 3.47 | 0 |
| Symmetric 0.5 | 4.16 | 2.35 | 7.67 |
| Symmetric 1.0 | 5.014 | 2.54 | 5.67 |
| Symmetric 2.0 | 6.96 | 3.19 | 1.34 |

**Table 1:** Statistical results of Symmetric distances for trees obtained from noisy alignments.

| **Types of trees** | **Mean Value** | **Standard Deviation** | **% of Reference trees recovered** |
|---|---|---|---|
| Asymmetric 0.5 | 7.27 | 2.89 | 0.34 |
| Asymmetric 1.0 | 9.11 | 3.029 | 0 |
| Asymmetric 2.0 | 11.92 | 3.383 | 0 |
| Symmetric 0.5 | 4.08 | 2.30 | 9.67 |
| Symmetric 1.0 | 4.82 | 2.50 | 6.67 |
| Symmetric 2.0 | 6.52 | 3.09 | 4 |

**Table 2:** Statistical results of Symmetric distances for trees obtained from noise reduced alignments using the proposed algorithm.

From table 2, it can be inferred that mean value and standard deviation for noise reduced trees is slightly less than their respective unprocessed trees (Table 1). This indicates a reduction in the total symmetric distances or an increase in the distribution with lower values on noise reduction. Also, the percentage of recovered trees is higher for noise reduced category than that for the unprocessed ones.

The trees from the respective categories were compared based on mean value and the number of recovered trees to draw a concrete conclusion about the above statement

| **Types of trees** | **Mean Value Original Trees** | **Mean Value Noise Reduced Trees** | **% Change in mean values** |
|---|---|---|---|
| Asymmetric 0.5 | 7.64 | 7.27 | -4.80 |
| Asymmetric 1.0 | 9.43 | 9.11 | -3.32 |
| Asymmetric 2.0 | 12.34 | 11.92 | -3.40 |
| Symmetric 0.5 | 4.16 | 4.08 | -1.92 |
| Symmetric 1.0 | 5.014 | 4.82 | -3.86 |
| Symmetric 2.0 | 6.96 | 6.52 | -6.23 |

**Table 3:** Comparison of Mean values of the Original/noisy trees and noise reduced trees using the proposed algorithm.

The percentage change in mean values of the noise reduced trees is negative for all the categories. This justifies that the noise reduced trees have a lower mean value.

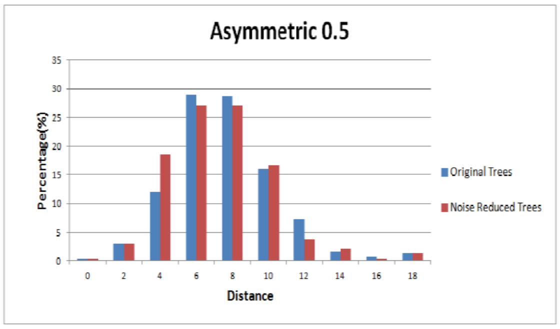
| **Types of trees** | **% of Reference trees Original data** | **% of Reference trees Noise reduced data** | **% Change in reference trees recovered** |
|---|---|---|---|
| Asymmetric 0.5 | 0.34 | 0.34 | 0 |
| Asymmetric 1.0 | 0 | 0 | 0 |
| Asymmetric 2.0 | 0 | 0 | 0 |
| Symmetric 0.5 | 7.67 | 9.67 | 26.08 |
| Symmetric 1.0 | 5.67 | 6.67 | 17.65 |
| Symmetric 2.0 | 1.34 | 4 | 200 |

**Table 4:** Comparison of Standard deviation of the Original/noisy trees and the noise reduced trees using the proposed algorithm:

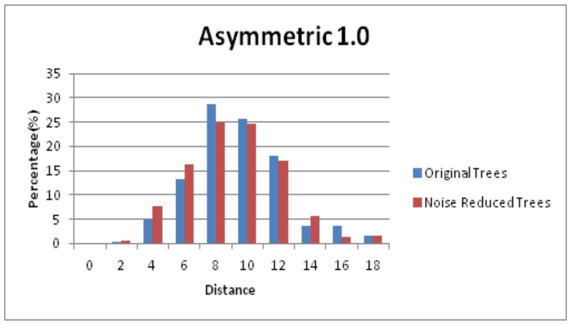
The percentage change in the number of the recovered trees after noise reduction is higher for all symmetric trees. This is justified as the input data from the symmetric categories are recovered better after noise reduction when compared to that from the asymmetric categories. It displays an inefficiency of noise reduction in retrieving phylogenetic inference in the asymmetrical categories. It is difficult to recover these trees by the criteria used.

As mentioned previously, the distribution of trees before and after noise reduction is looked upon to infer the effect of noise reduction for all the categories.

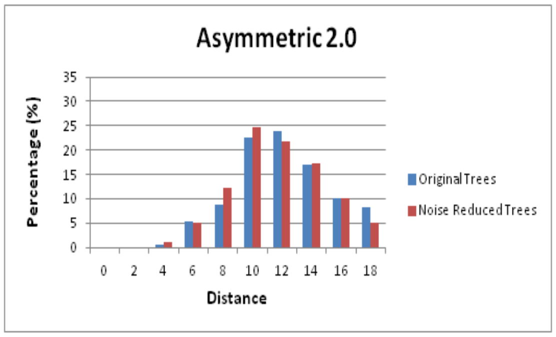
Histogram 1:



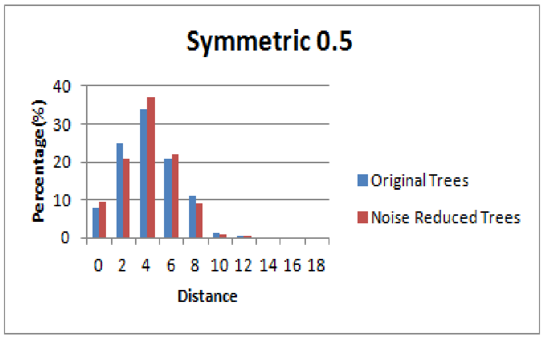
Histogram 2:



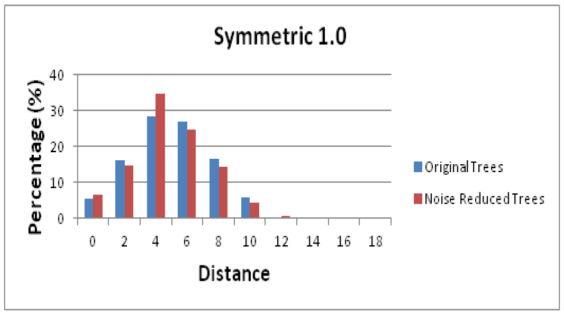
Histogram 3:



Histogram 4:



Histogram 5:



Histogram 6:

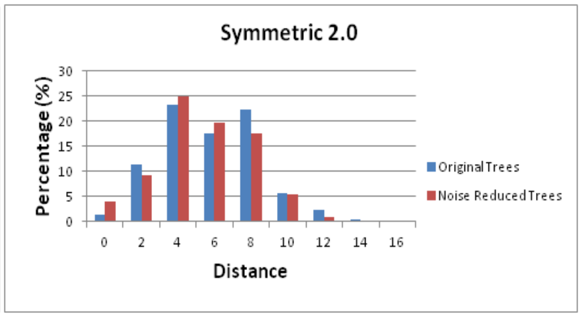


**Figure 3:** The distribution of trees along the distance from the reference trees for the proposed noise-reduced algorithm and the input data (noisy data).

Histogram 1:
An increase in the distribution of trees with lower distance is observed after noise reduction. Although, the number of recovered trees is only one but an increase in the distribution indicates a closer phylogenetic inference after noise reduction.

Histogram 2:
The number of recovered trees is zero due to higher mutation frequency and the asymmetric nature of tree. However, an increase in the percentage of trees with lower distance is seen after noise reduction that gives a closer phylogenetic inference.

Histogram 3:
The number of recovered trees is zero due to much higher mutation frequency and the asymmetric nature. Also, not much change is observed in the tree distribution after noise reduction. This incurs that it is difficult to retrieve the asymmetric trees with high mutation frequency. Thus, there is a flaw even in noise reduction technique to obtain phylogenetic inference among the trees that have more complexities.

Histogram 4:
The higher number of trees is recovered i.e. with zero distance. It is due to a low mutation frequency and the symmetric nature of the trees. Also, a large number of noise reduced trees is observed with lower distance as compared to the original trees. Hence, symmetrical trees are retrieved efficiently by noise reduction and phylogenetic inferences can be made.

Histogram 5:
The percentage of recovered trees and distribution of trees with lower distance is higher than the asymmetrical trees but lower than the symmetrical trees with lower mutation frequency. This indicates that an increase in the mutation frequency tends to lower the percentage of the recovered trees.

Histogram 6:
The above stated hypothesis about mutation frequency is strengthened by the result obtained for this case. A decrease in the percentage of recovered trees and the distribution of tree with lower distance is observed when compared to the symmetrical trees with lower mutation frequencies.

To test the efficiency of the developed algorithm in comparison with the existing ones, the behavior of the noise reduced dataset was also compared with Gblocks and TrimAl performance. Comparison is again made on Symmetric difference calculation.

| **Types of trees** | **Mean Value** | **Standard Deviation** | **% of Reference trees recovered** |
|---|---|---|---|
| Asymmetric 0.5 | 7.55 | 2.89 | 0.3 |
| Asymmetric 1.0 | 9.27 | 2.92 | 0 |
| Asymmetric 2.0 | 10.85 | 3.23 | 0 |
| Symmetric 0.5 | 4.03 | 2.30 | 7.7 |
| Symmetric 1.0 | 4.87 | 2.45 | 6.3 |
| Symmetric 2.0 | 5.64 | 2.91 | 4.0 |

**Table 5:** Statistical results for Symmetric distances of trees obtained using TrimAl.

| **Types of trees** | **Mean Value TrimAl Trees** | **Mean Value Noise Reduced Trees** | **% Change in mean values** |
|---|---|---|---|
| Asymmetric 0.5 | 7.55 | 7.27 | -3.7 |
| Asymmetric 1.0 | 9.27 | 9.11 | -1.7 |
| Asymmetric 2.0 | 10.85 | 11.92 | 9.9 |
| Symmetric 0.5 | 4.03 | 4.08 | 1.2 |
| Symmetric 1.0 | 4.87 | 4.82 | -1.0 |
| Symmetric 2.0 | 5.64 | 6.53 | 15.7 |

**Table 6:** Comparison of Mean Values for Symmetric distances between TrimAl trees and noise reduced trees (from the proposed algorithm).

Comparison of the mean symmetric distance values indicates that the performance of TrimAl and that of the proposed algorithm were similar. However, since more negative percentage changes are obtained, it can be stated that the proposed noise reduction algorithm worked slightly better than TrimAl.

It is also observed in general that the mean and standard deviation values are higher for the asymmetric categories in comparison to the symmetric categories. Moreover, an increased trend is observed for the categories with higher mutation frequency. Thus, Symmetric 0.5 dataset exhibits smallest mean and standard deviation values and Asymmetric 2.0 the highest.

| Types of trees | % of Reference trees TrimAl data | % of Reference trees noise reduced data | % Change in reference trees recovered |
|---|---|---|---|
| Asymmetric 0.5 | 0.3 | 0.3 | 0 |
| Asymmetric 1.0 | 0 | 0 | 0 |
| Asymmetric 2.0 | 0 | 0 | 0 |
| Symmetric 0.5 | 7.7 | 9.67 | 25.6 |
| Symmetric 1.0 | 6.3 | 6.67 | 5.8 |
| Symmetric 2.0 | 4.0 | 4.0 | 4.0 |

**Table 7:** Comparison of the percentage of Reference Trees recovered for TrimAl trees and noise reduced trees (from the proposed algorithm).

| Types of trees | Mean Value | Standard Deviation | % of Reference trees recovered |
|---|---|---|---|
| Asymmetric 0.5 | 8.35 | 3.46 | 0.3 |
| Asymmetric 1.0 | 9.6 | 3.46 | 0 |
| Asymmetric 2.0 | 11.99 | 3.78 | 0 |
| Symmetric 0.5 | 5 | 2.87 | 7.3 |
| Symmetric 1.0 | 5.59 | 2.91 | 4.3 |
| Symmetric 2.0 | 7.1. | 3.37 | 3.0 |

**Table 8:** Statistical results for the Symmetric distances of trees obtained from Gblocks.

| Types of trees | Mean Value Gblocks Trees | Mean Value Noise Reduced Trees | % Change in mean values |
|---|---|---|---|
| Asymmetric 0.5 | 8.35 | 7.27 | -12.9 |
| Asymmetric 1.0 | 9.6 | 9.11 | -5.1 |
| Asymmetric 2.0 | 11.99 | 11.92 | -0.6 |
| Symmetric 0.5 | 5 | 4.08 | -18.4 |
| Symmetric 1.0 | 5.59 | 4.82 | -13.7 |
| Symmetric 2.0 | 7.17 | 6.53 | -8.9 |

**Table 9:** Comparison of Mean Values for Symmetric distances between Gblocks trees and noise reduced trees (from the proposed algorithm).

Comparison based on mean values indicates that the proposed algorithm performed better than Gblocks.

| Types of trees | % of Reference trees Gblocks data | % of Reference trees noise reduced data | % Change in reference trees recovered |
|---|---|---|---|
| Asymmetric 0.5 | 0.3 | 0.3 | 0 |
| Asymmetric 1.0 | 0 | 0 | 0 |
| Asymmetric 2.0 | 0 | 0 | 0 |
| Symmetric 0.5 | 7.3 | 9.67 | 32.5 |
| Symmetric 1.0 | 4.3 | 6.67 | 55.1 |
| Symmetric 2.0 | 3.0 | 4.0 | 33.3 |

**Table 10:** Comparison of the percentage of Reference Trees recovered for Gblocks trees and noise reduced trees (from proposed algorithm).

The behavior of the noise reduction algorithm in Gblocks is found to be very similar to that in TrimAl. The reference trees are easily recovered by the algorithm for symmetric inputs, whereas no performance improvement for the asymmetric cases.

Also, from the generated histogram, Gblocks shows a similar behavior to the proposed algorithm and TrimAl. However, a minor variation in the distribution of trees is observed.

It is evident that the proposed noise reduction algorithm has a higher efficiency in recovering the reference trees in comparison to TrimAl and Gblocks for the symmetric categories. However, no significant improvement in performance is observed for the proposed algorithm for the asymmetric categories.

The percentage of recovered trees for each symmetric distance is also calculated. Comparison is performed among our proposed algorithm, Gblocks and TrimAl. For the asymmetric category, the percentage of reference trees recovered among the three methods is found to be comparable. The reference tree is recovered only in the category Asymmetric 0.5. This indicates that the increase in the mutation frequency poses an increase difficulty in recovering the reference trees from the asymmetric categories. However, for the symmetric categories, the proposed noise reduction algorithm shows a higher rate to recover the reference trees than Gblocks and TrimAl.

A general overview obtained from all histograms states that the percentage of the number of recovered trees reduces considerably for the proposed noise reduction method. Moreover, the proposed method tends to generate more trees with smaller distance values. This can be considered as an advantage since it indicates that the proposed method tends to produce trees with higher similarity to the reference trees

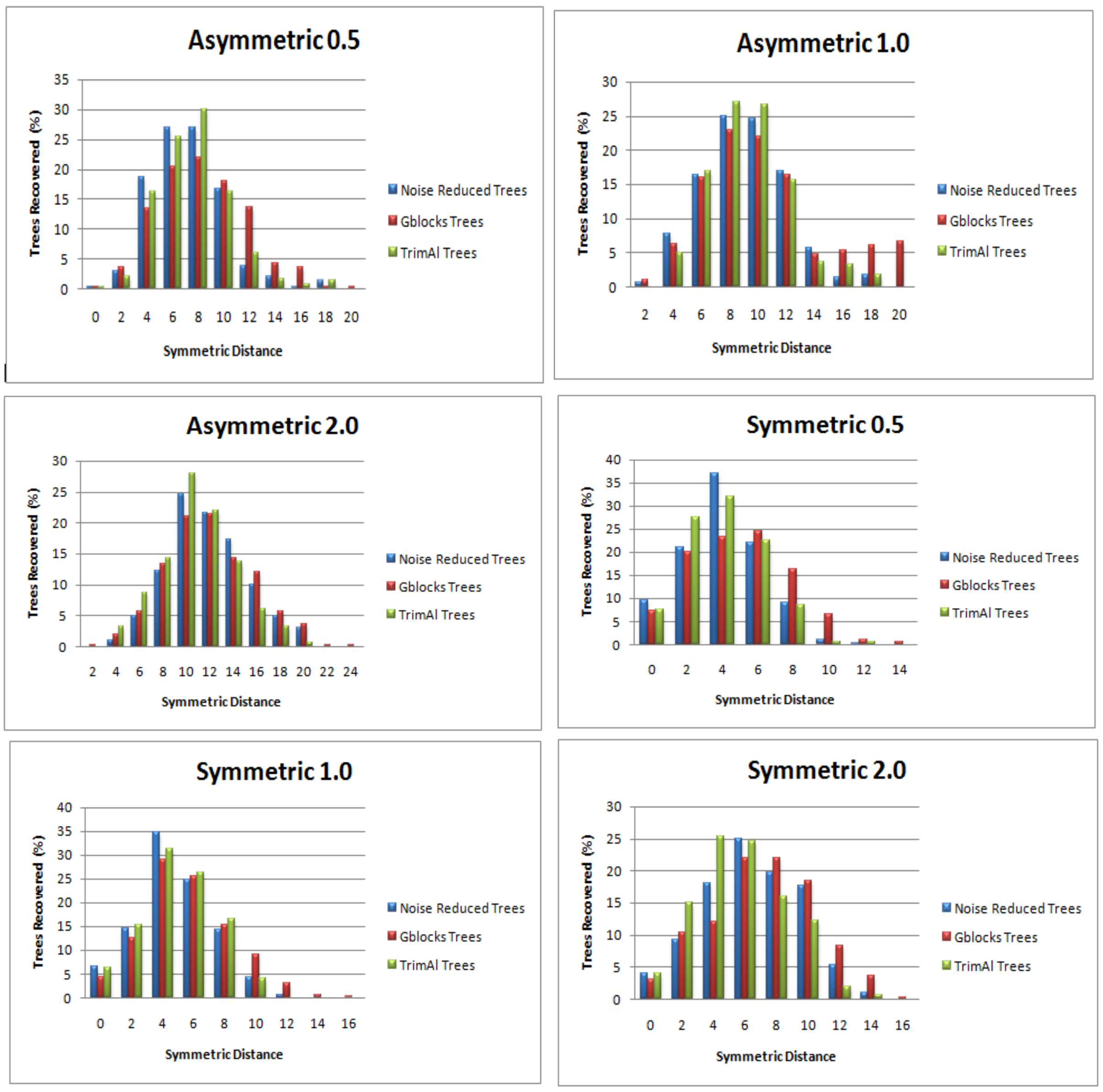


**Figure 4:** Distribution of the trees along the symmetric distance from the reference trees for all methods.

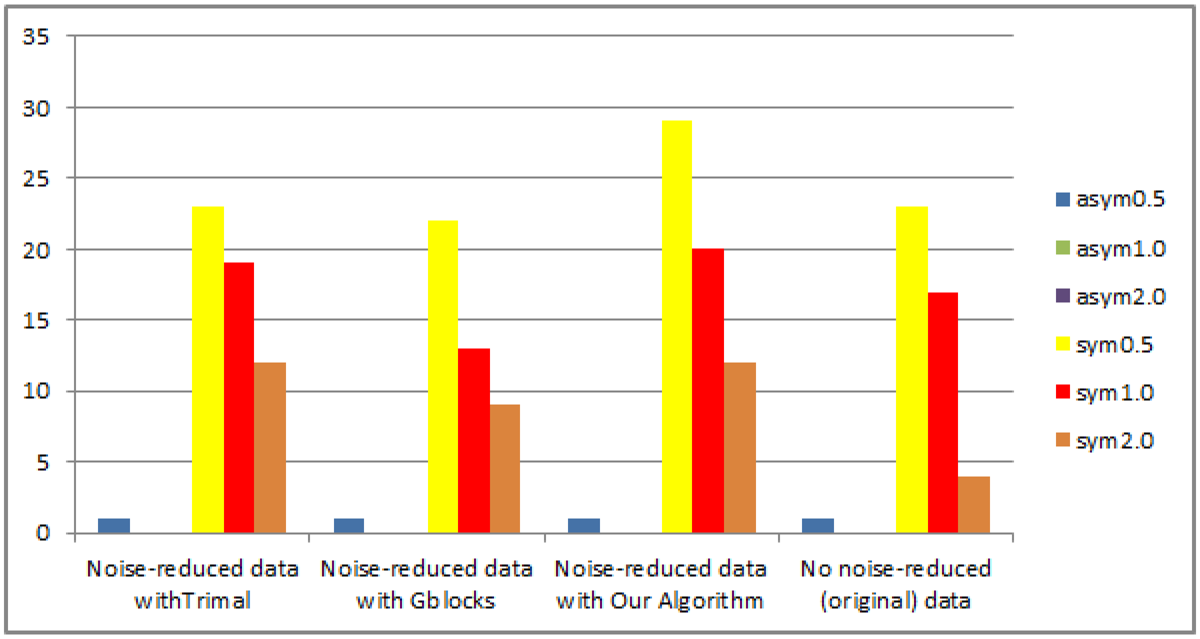


**Figure 5:** The number of recovered reference trees for all the categories for all methods.

Figure 5 shows the number of trees which are exactly the same as the reference trees, i.e. the symmetric distance between each tree and the reference tree is zero, across all categories.
The horizontal axis shows three methods used to reduce noise from the input data (TrimAl, Gblocks, the proposed method) except the last one which is no method, i.e., the raw input data (noisy data).
Each method contains 6 consecutive boxes for 6 categories of input data.
The green and purple columns are not seen across all methods, i.e. the number of recovered trees from the categories Asymmetric1.0 and Asymmetric2.0 are zero whether TrimAl, Gblocks, the proposed method or no noise-removal method is applied.
Figure 5 also shows that regardless of whether a noise reduction algorithm is used or which noise reduction algorithm was used:
- The number of recovered trees obtained for asymmetric categories (blue, green and purple columns) are less than those for symmetric categories (yellow, red and brown columns) and
- Among the symmetric categories (yellow, red and brown columns for mutation rates of 0.5, 1.0 and 2.0 respectively), the number of recovered trees is greater for smaller mutation rate.

Hence, it can be concluded that noise reduction in trees tends to increase the phylogenetic inference of trees. However, the increase in the mutation frequency and asymmetric nature of trees tends to lower the recovery of trees. In other words, phylogenetic inference is difficult on increasing complexities.

## References

1. Capella-Gutierrez S, Silla-Martinez J.M, Gabaldon T (2009). TrimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. Bioinformatics 25: 1972-1973.
2. Castresana, J. (2000). Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Molecular Biology and Evolution 17: 540-552.
3. Khan MA, Elias I, Sjölund E, Nylander K, Guimera RV, Schobesberger R, Schmitzberger P, Lagergren J, Arvestad L (2013). Fastphylo: Fast tools for phylogenetics. BMC Bioinformatics 14: 334
4. Sukumaran, J. and Mark T. Holder (2010). DendroPy: A Python library for phylogenetic computing. Bioinformatics 26: 1569-1571.
5. Noble W.S (2009). A Quick Guide to Organizing Computational Biology Projects. PLOS Computational Biology 5: 1-5