bweedop/Splinter

Multiple Sequence Alignment While Assessing Saturation Across Sequence Data

K. Bodie Weedop and William D. Pearse

UtahStateUniversity
DEPARTMENT OF BIOLOGY

INTRODUCTION

Constructing and analyzing phylogenetic trees is central to biological disciplines such as evolutionary and systematic biology. Accurate phylogenetic inference improves the estimation of evolutionary relationships, rates of molecular evolution, and Operational Taxonomic Units (OTUs). Careful alignment of sequence data is critical prior to any phylogenetic reconstruction, and there are many different multiple sequence alignment programs that are currently used (reviewed in Edgar & Batzoglou 2006). However, difficulty persists when using alignments to accurately determine actual genetic divergences. A major, yet under-explored, problem is saturation: the repetition of base substitutions at a single site within a sequence. Saturation causes issues because numerous substitutions in sequences within an alignment can erroneously underestimate divergence. Here, we present an algorithm, Splinter, that identifies and accounts for saturation during DNA sequence alignment.

THE SPLINTER ALGORITHM

We implemented a novel algorithm to detect sequence saturation and then alleviate it by forming sub-groups of sequences, aligning those groups, and then merging those groups in a master alignment (See Figure 1). Using BioPython (Cock et al. 2009), we aligned sequences using MAFFT (Katoh & Stanley 2013). Sequences are initially hierarchically clustered, and tested for saturation using the method presented by Smith et al. (2009). The method applies dispersion statistics using the euclidean distance between the raw sequence distances and corrected sequence distances to assess dispersion. The corrected distances correspond to the Jukes and Cantor model. The distance between the uncorrected and corrected sequence distances are then used to calculate the median absolute deviation (MAD):

$$MAD = 1.4826 \times Med\ (|\ x_i - Med\ (x)|)$$

A MAD value greater than ~0.01 for a collection of sequences identifies that saturation is present. If saturation is detected, sequences within a cluster are separated. Sequences within the newly formed cluster are aligned to a consensus sequence and tested for saturation anew. This continues until saturation is not present in any one of the sequence clusters.

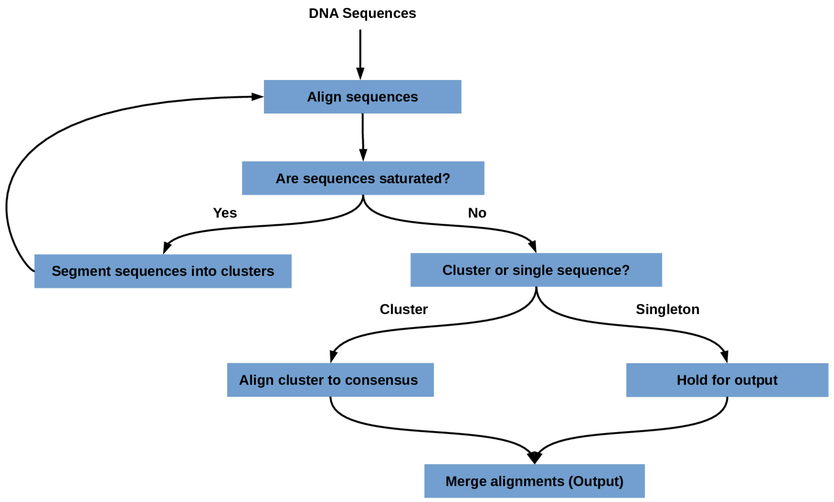

Figure 1. Schematic diagram of Splinter algorithm.

PERFORMANCE ASSESSMENT

We assessed Splinter's performance to align known sequences simulated (using DAWG; Cartwright 2005) along known phylogenies (simulated under a pure-birth Yule process; Harmon et al. 2008). We assessed Splinter by three criteria: execution speed, phylogenetic accuracy, and accuracy of its clustering algorithm. All results are reported with reference to MAFFT.

- Speed: Splinter does more than, and as such is slower than, MAFFT. Splinter shows a nearly linear increase in execution time with species number (Fig 2A.). All execution times were divided by the greatest execution time to show relative difficulty. Splinter appears to have most relative difficulty as the sequence count rises to 200 (Fig. 2B).

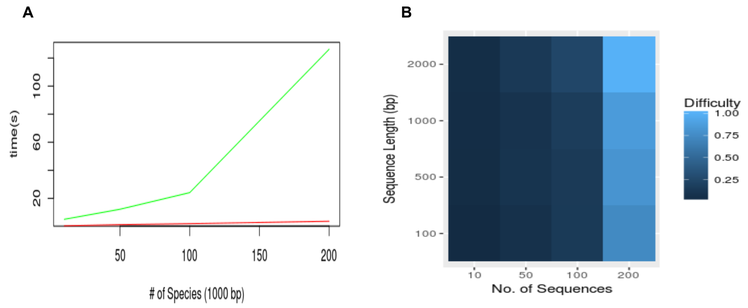

Figure 2. A) Total elapsed computing time comparison between MAFFT and Splinter. MAFFT is identified by the red line. Splinter is identified by the green line. B) Relative difficulty of a given sequence dataset for Splinter.

- Phylogeny Accuracy: Four original neighbor-joined (NJ) trees were constructed. Using those trees, sequence data was simulated and aligned by Splinter. The NJ tree produced by the Splinter alignment was compared to the original NJ tree using a Robinson-Foulds (RF) distance. Splinter is producing alignments that are equivalent to those produced by MAFFT. The mean RF distances for both MAFFT and Splinter alignments were roughly equivalent ($\bar{x}_{MAFFT}$=19.5, $\bar{x}_{Splinter}$=21.5).

- Accuracy of Clusters: Splinter effectively separates a cluster of sequences identified to be saturated by MAD. It can be seen that the sequences are segmented into clusters where saturation is not present. We expect that a group of sequences with a longer branch length would be grouped together by Splinter. However, the sequences within the clusters are not wholly consistent with our expectation of where sequences should be grouped (Fig. 3).

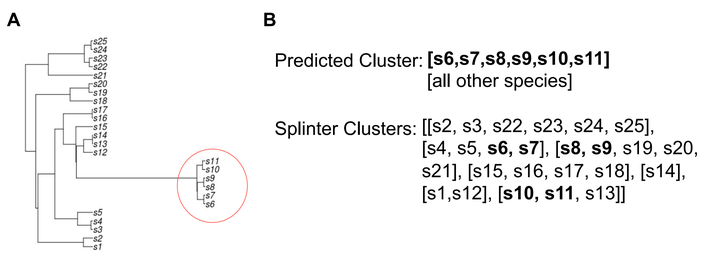

Figure 3. A) Plot of the simulated NJ tree. Sequences encircled in red identify those within expectation. B) Comparison of the expected cluster and clusters given by Splinter.

DISCUSSION

Splinter decreases in speed when aligning more than 200 sequences. However, other biologically accurate alignment programs are simply impractical for more than just 50 sequences (e.g. PRANK; Löytynoja 2014, Pearse et al. 2013). Further, Splinter maintains a nearly linear increase in execution time as the number of species increases. Also, Splinter is performing an alignment, detecting sequence saturation, and grouping sequences which are not saturated. With only a slight reduction of speed, Splinter is producing a sequence alignment that is almost as accurate as MAFFT, while simultaneously considering biological accuracy.

Our comparison of NJ trees produced by both MAFFT and Splinter demonstrates that Splinter is just as effective as MAFFT. The simulated data which we used for analysis is simulated for MAFFT. Even still, Splinter is able to produce an alignment that is just as accurate as MAFFT. Such results demonstrate Splinter is a conservative and safe method for aligning multiple sequence data.

We can see that Splinter is separating sequences into groups which do not contain saturation. The groups that sequences are being partitioned to was not expected. Sequences we expected to be partitioned together are not in the same group consistently but the results are not drastically distant our expectation. However, we are uncertain if the simulations we have performed reflect empirical saturated sequences. With DAWG, it is not possible to simulate sequences where multiple substitutions have occurred at select sites over a specific lineage. Without this capability it is difficult to provide Splinter with saturated sequences and make accurate expectations.

CONCLUSION & FUTURE WORK

- Our results reflect the difficulty of producing a multiple sequence alignment while considering a biological accuracy. Splinter is an effective option for accurate alignment of rapidly evolving gene sequences.
- Improved and more precise sequence simulators should be identified or developed which incorporate saturation. A program of this sort would allow geneticists and phylogeneticists to produce biologically accurate sequences where multiple substitutions have occurred over a specific lineage.

REFERENCES & ACKNOWLEDGEMENTS

Cartwright, R.A., 2005. DNA assembly with gaps (Dawg): simulating sequence evolution. *Bioinformatics*, *21*(Suppl 3), pp.iii31-iii38.
Cock, P.J., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. and De Hoon, M.J., 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25(11), pp.1422-1423.
Edgar, R.C. and Batzoglou, S., 2006. Multiple sequence alignment. Current opinion in structural biology, 16(3), pp.368-373.
Katoh, K. and Standley, D.M., 2013. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular biology and evolution, 30(4), pp.772-780.
Smith, S.A., Beaulieu, J.M. and Donoghue, M.J., 2009. Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. BMC evolutionary biology, 9(1), p.37.
Pearse, W.D. and Purvis, A., 2013. phyloGenerator: an automated phylogeny generation tool for ecologists. *Methods in Ecology and Evolution*, *4*(7), pp.692-698.