Laboratory for Bioinformatics Tools
Assignment 3

Nir Borger

Part 1 – Reading a scientific paper:

- Pseudo-Code:
  1) For each alignment, convert each sequence into an array of indices.
  2) Construct the following graph $G = (V, E)$: V is the set of nodes and each node represents a column in the MSA. E is the set of directed edges between consecutive columns, representing the transition between them. Each edge has the weight of the number of MSA's that the two nodes the edge connects exist in.
  3) Calculate a path score and a path length of each node recursively: current node = n. It has at least one incoming edge $e_i = (x_i, n)$, where x_i is the source of the edge. Denote weight of edge $w(e_i)$, and the edge with highest score $e_s = (x_s, n)$. Note that e_i, e_s are irrelevant for the first node. Finally, denote $s(x_i), l(x_i)$ to be that path score and the length score of node x_i, respectively.
     Use following recursion computations:
     base case: $\begin{cases} s(x_0) = 0 \\ l(x_0) = 0 \end{cases}$
     recursion: $\begin{cases} edge\ score \coloneqq\ e_i = \frac{s(x_i)+w(e_i)}{l(x_i)+1} \\ s(x_i) = s(x_s) + w(e_s) \\ l(x_i) = l(x_s) + 1 \end{cases}$
  4) Create traceback edges $t = (n, x_s); w(t) = w(e_s)$.
  5) Use the optimum path to reconstruct the consensus MSA while each weight of the traceback edges is used to score each column.
- According to the authors, MergeAlign notably outperforms the rest of the methods for sequences with low percentage identity, i.e., 0-10% sequence identity. As seen in Figure 4B in page 5, MergeAlign achieves 3% improvement in accuracy, approximately.
- In order to validate the methods they provided, and compare them with the alternatives, the authors established alternative tests. The first group of tests were performed on benchmark MSAs and gave two scores: a Q-score and an iRMSD score. These scores proved that combining of MSAs using different models of amino acid substitutions can improve MSA accuracy. The second group of tests were two independent tests that didn't use benchmark MSAs. These tests demonstrated that MergeAlign alignments increase not only the resolution of phylogenetic trees, but also increase the topological agreement between phylogenetic trees inferred from individual gene families.
- The evaluation matrices were chosen by assessing MSA performance of a set containing 142 amino acids substitution matrices. Those matrices were used to identify how each matrix performed on benchmark MSAs. The authors used the MAFFT MSA method with FFT-NS-2 strategy to align a randomly chosen set of a thousand benchmark sequences, using each matrix. Past the alignment of the benchmark sequences, for each amino acid substitution matrix, the authors calculated a mean F-score that was obtained

over all MSAs. Then, the matrices were ranked according to their scores and the authors selected the 91-top scoring matrices for generating input alignments for the MergeAlign algorithm.

Part 2 – Constructing phylogenetic trees and assessing statistical significance:

Subsection f:

There is a slight difference between the two trees. While in the globalpw_dist_tree_fig, the *Rhodotorula toruloides NP11* is the outgroup of the tree, in the kmer_dist_tree_fig tree it is inside the tree, and the *Nematostella vectensis* serves as the outgroup. This is the main difference. The second difference is the fast that in globalpw_dist_tree_fig, the *Laodelphax striatella* and *Litopenaeus vannamei* are each an outgroup for a subtree, they are shown together as the same cluster in kmer_dist_tree_fig.
We can account the differences for the way the distance matrices were calculated. The globalpw_dist_tree_fig is calculated with a distance matrix that was based on global alignment, meaning it is based on similarity between species. This similarity is defined as the similarity of the sequences of amino acids, with importance on the order of these amino acids. The k-mer system is based on similarity without considering the order of the similar AA. Therefore, in k-mer calculations, the minimal value is higher and can differ from sequence to sequence, thus creating different distance matrix.

Subsection h:

Counters for each split in trees constructed by alignment approach:

{('Arabidopsis_thaliana', 'Dimocarpus_longan'): 100, (('Arabidopsis_thaliana', 'Bombyx_mori', 'Danio_rerio', 'Dimocarpus_longan', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Laodelphax_striatella', 'Nematostella_vectensis', 'Spodoptera_litura', 'Trichuris_trichiura'), 'Rhodotorula_toruloides_NP11'): 98, (('Arabidopsis_thaliana', 'Danio_rerio', 'Dimocarpus_longan', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Laodelphax_striatella', 'Trichuris_trichiura'), ('Bombyx_mori', 'Nematostella_vectensis', 'Spodoptera_litura')): 66, (('Arabidopsis_thaliana', 'Dimocarpus_longan'), ('Danio_rerio', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Laodelphax_striatella', 'Trichuris_trichiura')): 78, ('Bombyx_mori', 'Spodoptera_litura'): 100, (('Bombyx_mori', 'Spodoptera_litura'), 'Nematostella_vectensis'): 70, ('Danio_rerio', 'Homo_sapiens'): 100, (('Danio_rerio', 'Homo_sapiens'), ('Loa_loa', 'Trichuris_trichiura')): 100, (('Danio_rerio', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Trichuris_trichiura'), 'Laodelphax_striatella'): 93, (('Danio_rerio', 'Homo_sapiens', 'Loa_loa', 'Trichuris_trichiura'), 'Litopenaeus_vannamei'): 93, ('Loa_loa', 'Trichuris_trichiura'): 100}

Time taken: 696 seconds (~11.5 minutes).

Counters for each split in trees constructed by k-mer approach:

{('Arabidopsis_thaliana', 'Dimocarpus_longan'): 98, (('Arabidopsis_thaliana', 'Dimocarpus_longan'), 'Rhodotorula_toruloides_NP11'): 1, ('Bombyx_mori', 'Spodoptera_litura'): 99, (('Rhodotorula_toruloides_NP11', 'Arabidopsis_thaliana', 'Danio_rerio', 'Dimocarpus_longan', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Laodelphax_striatella', 'Trichuris_trichiura'), ('Bombyx_mori', 'Spodoptera_litura')): 45, ('Danio_rerio', 'Homo_sapiens'): 100, (('Danio_rerio', 'Homo_sapiens'), ('Loa_loa', 'Trichuris_trichiura')): 100, (('Rhodotorula_toruloides_NP11', 'Arabidopsis_thaliana', 'Dimocarpus_longan'), ('Danio_rerio', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Laodelphax_striatella', 'Trichuris_trichiura')): 0, (('Danio_rerio', 'Homo_sapiens', 'Loa_loa', 'Trichuris_trichiura'), ('Litopenaeus_vannamei', 'Laodelphax_striatella')): 87, ('Laodelphax_striatella', 'Litopenaeus_vannamei'): 87, ('Loa_loa', 'Trichuris_trichiura'): 100, (('Rhodotorula_toruloides_NP11', 'Arabidopsis_thaliana', 'Bombyx_mori', 'Danio_rerio', 'Dimocarpus_longan', 'Homo_sapiens', 'Loa_loa', 'Litopenaeus_vannamei', 'Laodelphax_striatella', 'Spodoptera_litura', 'Trichuris_trichiura'), 'Nematostella_vectensis'): 60}

Time taken: 253 seconds (~4.25 minutes).

It can be shown that the k-mer approach is much faster but much less accurate – there are much less "hits" for each split than the alignment approach. Thus, we conclude that the alignment approach is more robust, as it has more of the same branches for each new tree when compared to the original tree.

Subsection i:

When using the k-mer based distance matrix in subsection h, we may encounter some potential problems. Generally, the k-mer approach ignores resemblance between sequences when the similarity exists in non-overlapping locations (for example, most of the k-mers will probably sit in repeated regions and may just be discarded as repeats of the same k-mer instead of referring the number of repeats). Moreover, when constructing the $\widetilde{D}$ matrix, we use the new sequences we built by randomly selecting 50% of the columns in the MSA. By doing so we can ultimately miss k-mers that were in the original sequences or create k-mers that didn't exist in the original sequence, thus significantly affecting the resemblance between sequences and yielding overall less significant results. To improve the evaluation of the tree when using k-mers based distance matrices, we change the standard algorithm, presented in subsection h, to account the evolutionary variation (that is caused by the ability of an Assembly and alignment-free method i.e., the k-mer method, to reconstruct it). To do so, instead of selecting 50% of the columns, we will create for each sequence in the MSA their $K(S'_i, k)$ and from the set we will resample with replacement $\frac{1}{k}$ of the total k-mers. From the chosen k-mers we will construct n new sequences and remove all gaps. By using this form to recreate the sequences only 1 out of k of the k-mers is selected to account for the non-independence among k-mers caused by their overlap. Moreover, we don't miss as much k-mers, nor create as much k-mers that didn't exist, like the original algorithm, thus providing more robust results and improving the evaluation of the tree.