Comparative Genomics 2018

Practical 5: Gene order analysis

Group number: 6

Group members: Kyle Kimler, Kajetan Juszczak

**Summary**

Another method of understanding phylogenies is through gene-order analysis. This can be very difficult to understand in eukaryotic organisms because of the separation of their chromosomes. However, in prokaryotes it is possible to understand evolutionary relationships of organisms by the organization of their whole genomes. One can also use gene-order analysis on a smaller scale to understand phylogenies of single genes or operons, because promoter/enhancer regions are often conserved. This method can be used in eukaryotic genomes. These smaller alignments can be organized on a large scale to understand larger patterns of phylogeny. In this lab we produce a dotplot to compare synteny of all of the orthologs we clustered in the earlier practicals, coding each matching ortholog in a cluster as a random sequence to reduce computational load and to allow for grouping distantly related orthologs as generally conserved sequences. We align these random sequences to produce a synteny dotplot and build a new phylogenetic tree.

**Key Questions to Answer**

1. We used ortholog groups from Practical 4, since InParanoid output

2. If you use this script, please answer the following questions

1. Can there be ambiguities in clustering, so that one gene appears in several clusters in the cluster file? If so, why is this? What does this script do in that case?

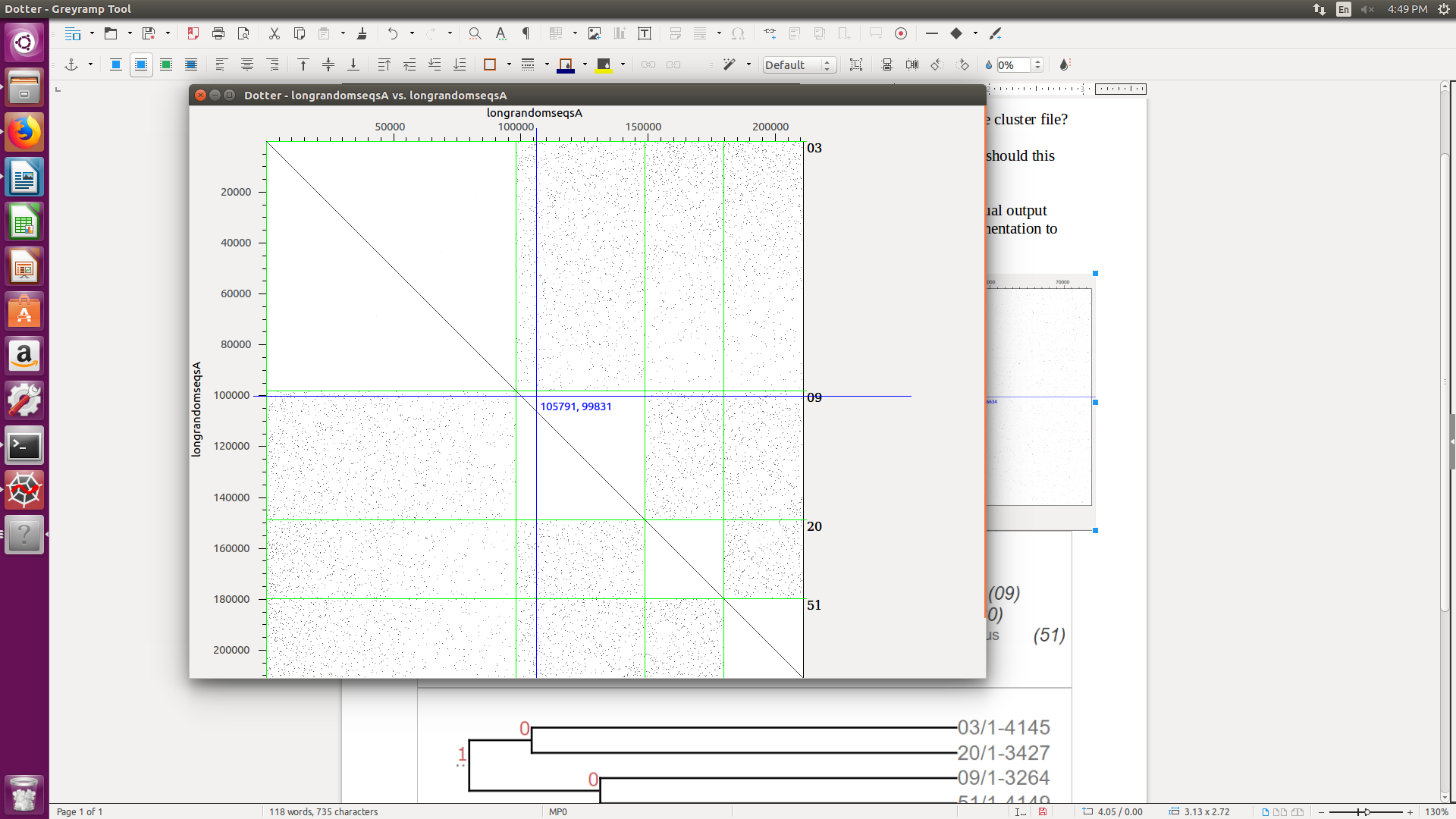
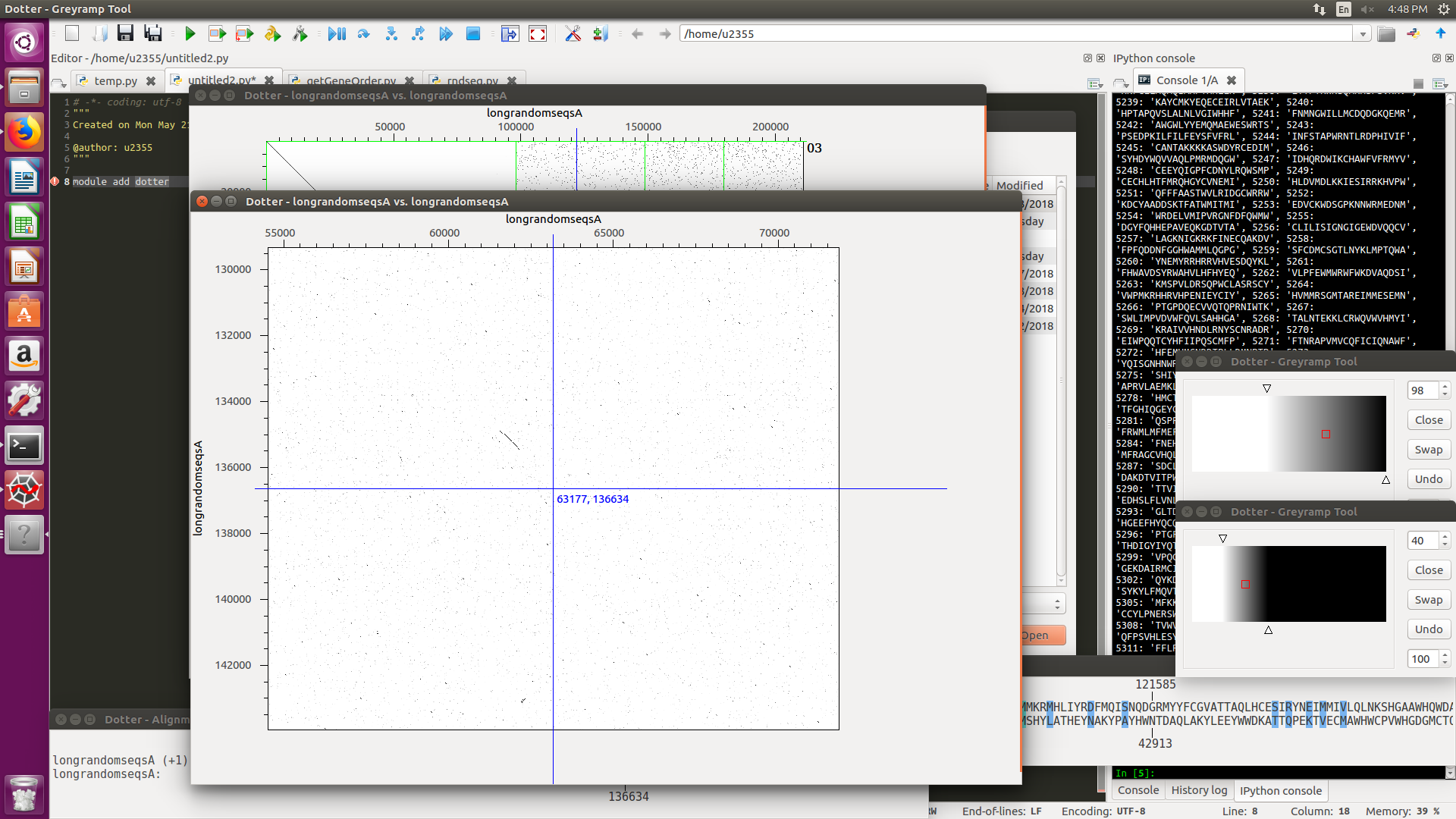
Yes, it is possible for one gene to appear in multiple clusters if it was the best hit for multiple genes in reference genome. This might be caused by reference genome containing multiple similar genes due to e.g. duplication event, while in other species there is only a single version of this gene and therefore the gene can be best hit for multiple reference genome orfs and as a result appear in multiple clusters. In script there is an if statement: “if not partOfCluster.has\_key (aWord):” which prevent from adding same key again.

1. Can this script handle forward and reverse strandedness in gene order lists? If not, how should this be done?

No, the script does not make a distinction related to the strandedness of a gene. We might separate forward and reverse genes by adding them to separate dictionaries.

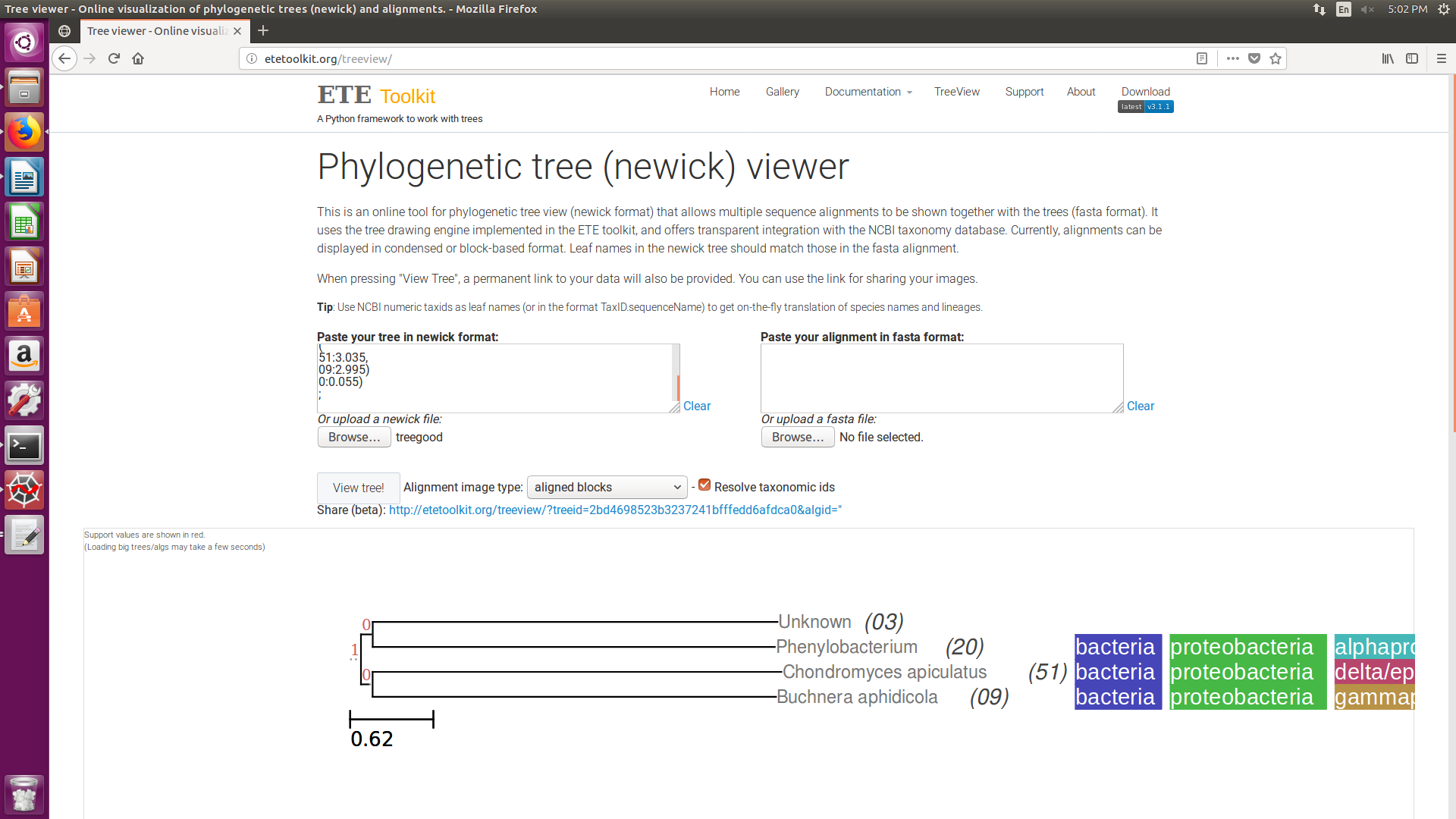
3. Play with the parameters to obtain a visual output that better explains the synteny between these organisms Explain your dotplot visualization.

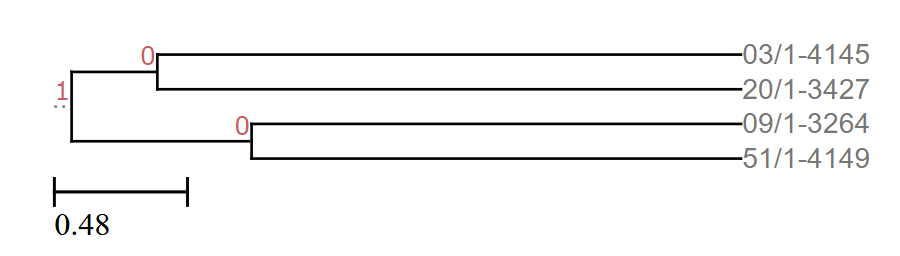
Addining different flags did not make the visual output better. We were only able to zoom it in. The dotplot can be difficult to understand at first glance. At a higher perspective it appears that there is no synteny at all, but when you zoom into a smaller area you can see a couple spots where genes were sequentially conserved, between 03 and 09 in the dotplot we present below. In the overall dotplot presentation, pairwise comparisons are separated into squares as their whole proteome sequences are appended to each other.



*Figure 1. dotplot(right) and its zoomed fragment(left)*

4.Compare the tree obtained in Practical 5 against the one obtained in Practical 4. How do they differ and why?

The trees are topologically the same but differ in distance because the input data is much smaller, as genes are reduced to random sequences. This also made it difficult to see how far apart the later splits were from the original. GRIMM finds the same topology here as our meta-tree from Practical 4 because the orthology clusters are conserved in their coding to random sequences.



*Figure 2. tree obtained in practical 5 (upper) and one obtained in practical 6 (lower)*