genesubsetselector

The goal of this software tool is to select, among large numbers of gene sequence alignments from phylogenomic data sets, a subset that is characterized by a set of criteria. At least one criterion will be based on single-gene trees that have been built from these same sequence alignments. Therefore, as with several of the decontamination tools, the program will take as input   
(1) a folder with a set of sequence alignments in fasta or ali format and  
(2) a folder with a set of trees in newick format derived from these alignments and with a file name exactly identical to the alignments, except for the extension (.nwk or .tre). [note that some programs such as MrBayes or PAUP produce their out files in Nexus format; Dendropy might be able to read these seamlessly, but especially the output trees from MrBayes are very complex and may not be properly readable. If so, then this is no problem and can be ignored – the trees that must be properly readable by the program are those from RAxML and FastTree].

The user will be able to specify either a threshold or a desired number of trees, and the criterion to be used. This usually means that the program will have to screen first all of the alignments and/or trees, then rank them according to the selected criterion, and then decide which ones will be selected. The selected and the unselected alignments and trees will be copied to new, separate folders; this means the program will produce 4 output folders: (1) selected alignments, (2) selected trees, (3) not selected alignments, (4) not selected trees.

The user should also specify if the sequence is nucleotide or protein (default is nucleotide).

The program should in addition output a simple table (tsv format) with the ranking of the alignments/trees and the scores for the criterion used.

Although it would be tempting to allow a combination of criteria, in practice this would cause a lot of problems. Therefore, the program should be designed in a way that only one of the criteria listed below can be selected at one time. If a user wants to use several criteria, then she/he needs to run the program multiple times.

The user should be able to select among three alternatives regarding the decision for the file selection:  
(1) number(of files): if this option is selected, an integer number needs to be provided. If for example the number 200 is given, then the first 200 alignments/trees in the ranking will be selected.  
(2) percentage: If this option is selected, then a value between 0 and 100 needs to be given. If for instance 50 is given, then the upper 50% percent of alignment/tree files in the ranking are selected  
(3) cutoff: If this option is selected, then a value needs to be given by the user that corresponds to a value of the selection criterion, and all alignments/trees above this value are then selected. For instance, if the selection criterion is the length of the alignment and the user inputs 500, then all alignments with more than 500 positions will be kept. It is obvious that this is a garbage in – garbage out option – if the user inputs a value that is not corresponding to the range of values of the respective criterion, then either all or none of the files will be selected.

What to do if multiple alignments/trees have exactly the same values for the criterion used? For instance, the user decides to rank files by alignment length and to select the 100 longest alignments; however, there are 200 alignments that all have the same (longest) alignment length. In this case, the program should just rank and select the files by input order, but provide the user with a warning that the selection criterion is ambiguous and that the selection choices should be modified, directing the user to check the output file with the ranking order.

The criteria to be used are the following (more could be added later):

1. Support values in the trees. For this criterion, the program first checks all of the trees. It looks at the support values (Dendropy should be able to identify these) which can be bootstrap, Bayesian posterior probability, or others. Because some trees/alignments may contain more or less taxa (leafs) and thus more or less internal nodes, we cannot just sum the support values of all nodes, but we need to calculate an average. So the program, for each tree, extracts the support values for all internal nodes/branches, and calculates the average for that tree. All trees/alignments are then ranked by average support values in decreasing order, i.e., the "best" trees are those with the highest average support values.

2. Alignment length. The program checks all of the alignment files for the length of the alignments. The files are ranked by alignment length in decreasing order, i.e., the "best" alignments are the longest ones.

3. Proportion of missing data.The program checks all of the alignment fles for the proportion of missing data (?Nn- in nucleotide alignments, ?\*X- in protein alignments) relative to the total number of characters in the entire alignment, and ranks them in increasing order (i.e., the "best" alignments have few or no missing data).

4. Number of included taxa. The program checks all of the alignment files for the number of included taxa, and ranks them in decreasing order. The "best" alignments are those with the most taxa present.

Additional criteria that might be added later:

- Presence/absence of particular taxa: Select all alignments/trees in which a certain taxon or a combination of taxa are prpesent/absent

- Amount of parsimony-informative sites

- GC content of nucleotide alignment

- "gappyness": amount of gaps or other missing data that are not at the beginning or end of the sequences

- Perhaps the other criteria used in the previous decontamination scripts.

Basic version of the program:

To be able to finish a basic version of this program soon, the following points would be the priority:

\* implement only the "number of files" option (not percentage, not threshold)

\* implement only the "support value in trees" criterion

Everything else can be left out and added to the script later, although probably most of the points are so trivial that adding all of them at once would not make a difference.