1) Data import: log files

In principle users should be able to run everything using only output files from phylogenetic analyses. That is tree files (.t files used previously in the examples) and log files (.p) with several parameter estimates. For simplicity, I have renamed the input tree and log files here to be: Tree\_1p.t and Tree\_3p.t (tree files with one or three clock partitions) and Log(4runs)10k\_1p.p (combined log files). I updated the vignette to reflect this and I am providing them in attachment. With that, we should have a single folder for all example files to be used as input, which makes user interaction easier.

One thing to consider, however, is that we usually get several log files is the output of such analysis. The single log files used here (e.g., Log(4runs)10k\_1p.p) were produced using a separate program, in which we can also reduce those giant file sizes (with hundreds of thousands of rows), to something more manageable, such as 10k rows. We do that by discarding some of the first few thousands of rows as burn-in because they have poor estimates (usually 25%), and with the remaining data, skipping every X number of rows that the user can determine.

Ideally, however, it would be better to produce this single/combined log file within the package. There are some basic R commands to merge and filter down those files (which can be read as tables) that I am providing in the Edits.R file. However, I could not find a way to skip an x% of the first rows for each file as burn-in. Perhaps could you help with that and combine that into a import\_log function?

2) Data import: melted log files

One of the questions you brought up was how users could obtain the AllRuns\_COMB\_Melted.csv file that a few of the functions need. This file is simply the melted version of the combined log files (Allruns). You obtained that in the vignette using the following:

AllRuns <- read.table("Log(4runs)10k\_1p.p", header = TRUE)

# Reshape Allruns data from wide to long

AllRuns\_Melted <- FBD\_reshape(AllRuns)

Important Note: there, however, a single but important difference between the AllRuns\_COMB\_Melted.csv and the AllRuns\_Melted object. The absence of the variable “analysis”. However, this is not a problem. This variable was something very specific to my study that we do not need to include in the functions of the package. We can ignore it. If users have something equivalent they wish to use, it’s a bit unpredictable what that could be and how they would name it. They would have to format their own log files in csv format accordingly to include such additional variables, just like I did for my study.

I could still run nearly all functions without a problem even in the absence of the “analysis” variable in the data (AllRuns\_Melted). However, I could not run the functions f3 and FBD\_tests2, so we may have to adapt or remove these. Also, the function FBD\_summary still runs, but it outputs 0 values for all parameters under analysis types “tip” and “tipnode”. So basically we should adapt that function to ignore analyses types.

3) Data Import: Nexus files

It would also be better if users could import directly the data matrix from Nexus files (the typical formatting for such files), instead of a .csv file. This is easy to accomplish with the read.nexus.data from the package *ape.* I have added those commands in the Edits.R and the new vignette file PackageTest3.r.

I also updated the get\_gower\_dist (get\_gower\_dist2.r) to remove the transpose function. Reading nexus data directly into R with ape removes the necessity of transposing the table.

4) t-SNE dimensions:#

I would suggest the addition of another argument where users can determine how many t-SNE dimensions they would like to plot (only first 2 or also 3). The reason is that in my experience important insights may come from comparing tSNE\_Dim1 to tSNE\_Dim2, or tSNE\_Dim2 to tSNE\_Dim3.

5) SelectionStrength

No need to use file RateByClockrMelted\_means.csv

Just reshape customized RateTable\_Means (imported as RateTable\_Means\_Clades) with new function clock\_reshape (see Clock\_reshape.r).

6) importing tree files and get\_clockrate\_table function

6a) I noticed you incorporated the drop.tip(tree, "Dummyextant") option in the get\_clockrate\_table function. That’s great! However, most trees will not include these dummy extant tips. This was a trick I did to be able to estimate evolutionary trees with only fossil species (tips) in my study. The function works perfectly well even in trees without “Dummyextant” tips, same principle it’s not a problem. However, perhaps it would be useful to include an argument where users can indicate if they have this a Dummyextant tip. The reason is that they may want, depending on the context, to show the full tree including the Dummyextant tip. It also makes very explicit that they can import trees with this option. We would have Dummyextant = FALSE as the default option.

6b) the tree files can be of two kinds. With the .tre and .t extension. So I am including examples with both when importing the tree.

6c) within the tree file, the rate values are usually depicted as the following parameter: rateTK02Brlens{1}\_median. However, in the tree examples I have provided I forgot to mention I had changed that to: rateTK02Brlens1\_median (removing the curly brackets), because the curly brackets were creating problems when I ran some functions. I simply did a search and replace inside the tree file before importing into R. So, such evolutionary trees (when estimated by Mr. Bayes), will always be uploaded with the curly brackets. I have added here such raw tree file without any of my changes so you can check that out (Tree\_3p(raw).t).

Again, the function get\_clockrate\_table runs fine, with the only difference being that the rate variable names in the output table include the curly brackets with them: rate{1}, rate{2}, etc. however, the introduction of the curly brackets in the tables creates problems for subsequent functions to run (clockrate\_reg\_plot). So I suggest adapting the get\_clockrate\_table to remove the curly brackets or for us to ask users to remove them.

7) Finally, I have added some suggested steps to the vignette or minor comments that you can find by searching for comments under “##T” .

NAMES:

Package: EvoPhylo

Function names:

F1 -> get\_pwt\_rates

F2 -> plot\_treerates\_sgn

F3 -> remove?