

- extensible and flexible software platform for Bayesian evolutionary analysis
- <https://www.beast2.org/>



Beast2

Bayesian evolutionary analysis by sampling trees

BEAST 2: A Software Platform for Bayesian Evolutionary Analysis



Remco Bouckaert^{1*}, Joseph Heled¹, Denise Kühnert^{1,2}, Tim Vaughan^{1,3}, Chieh-Hsi Wu¹, Dong Xie¹, Marc A. Suchard^{4,5}, Andrew Rambaut⁶, Alexei J. Drummond^{1,7*}

- Sequence files as input
- Model-based (complex parameterization)
- can deal with different clock/tree/sequence evolution models
- Consider gene trees uncertainty

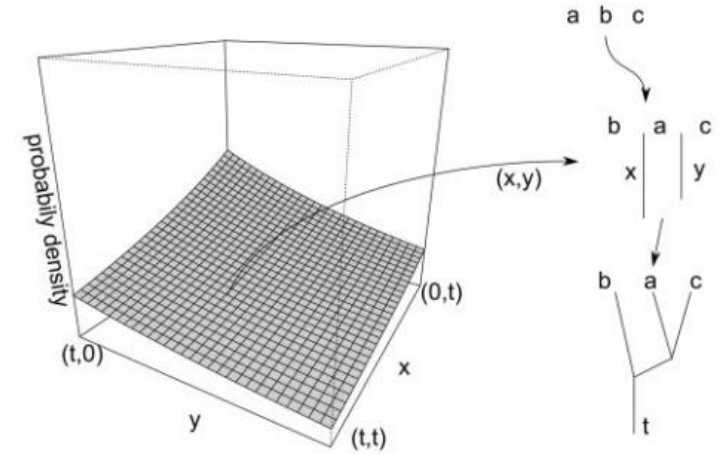


Fig. 1. Sampling trees from the usual birth-death density

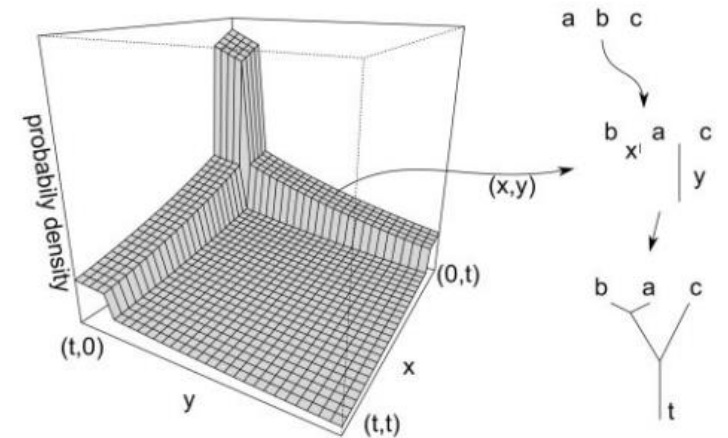


Fig. 2. Sampling trees from the mixture density

- Prior density on node heights including a spike near zero
- ω : prior information on the number of clusters (number of species expected)
- ε : small value (of node height) within which two minimal clusters (samples/populations) will be considered as a single

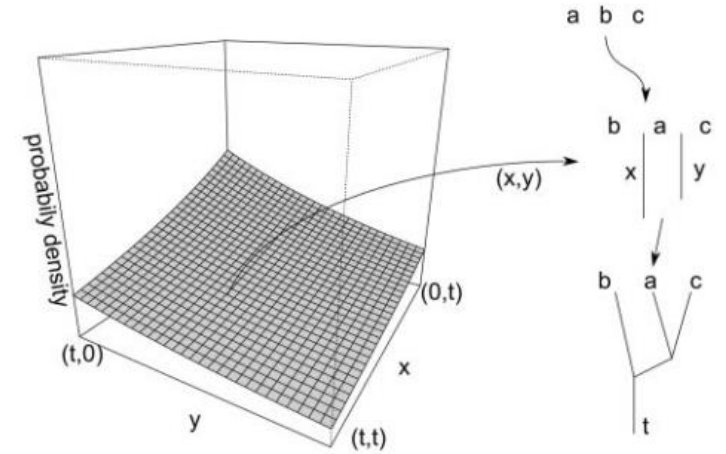


Fig. 1. Sampling trees from the usual birth-death density

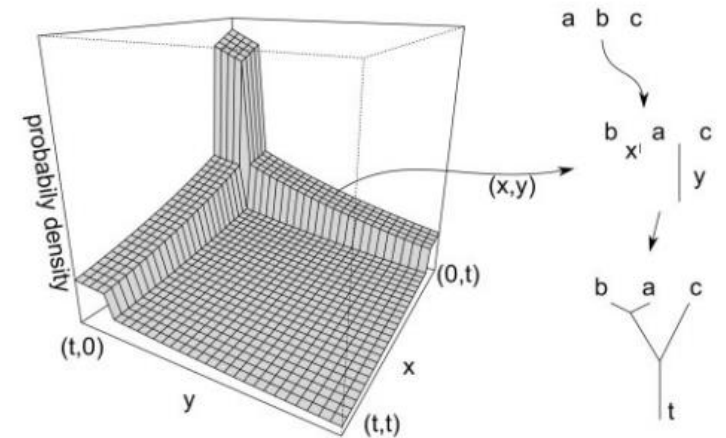
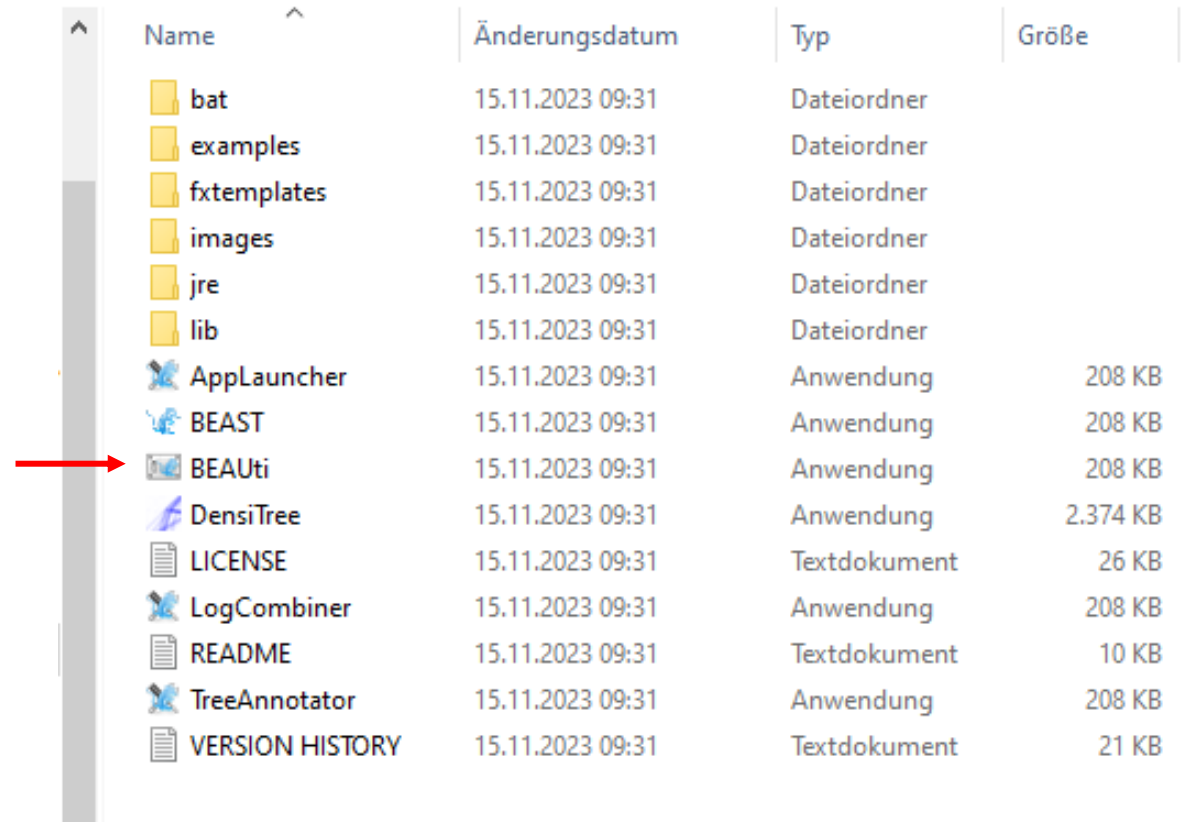


Fig. 2. Sampling trees from the mixture density

- Navigate to the software folder
- A few java scripts for different tasks (no need of installation!)

Name	Änderungsdatum	Typ	Größe
bat	15.11.2023 09:31	Dateiordner	
examples	15.11.2023 09:31	Dateiordner	
fxtemplates	15.11.2023 09:31	Dateiordner	
images	15.11.2023 09:31	Dateiordner	
jre	15.11.2023 09:31	Dateiordner	
lib	15.11.2023 09:31	Dateiordner	
AppLauncher	15.11.2023 09:31	Anwendung	208 KB
BEAST	15.11.2023 09:31	Anwendung	208 KB
BEAUti	15.11.2023 09:31	Anwendung	208 KB
DensiTree	15.11.2023 09:31	Anwendung	2.374 KB
LICENSE	15.11.2023 09:31	Textdokument	26 KB
LogCombiner	15.11.2023 09:31	Anwendung	208 KB
README	15.11.2023 09:31	Textdokument	10 KB
TreeAnnotator	15.11.2023 09:31	Anwendung	208 KB
VERSION HISTORY	15.11.2023 09:31	Textdokument	21 KB

- Navigate to the software folder
- A few java executables for different tasks (no need of installation!)
- BEAUti: help producing inputs for the analyses



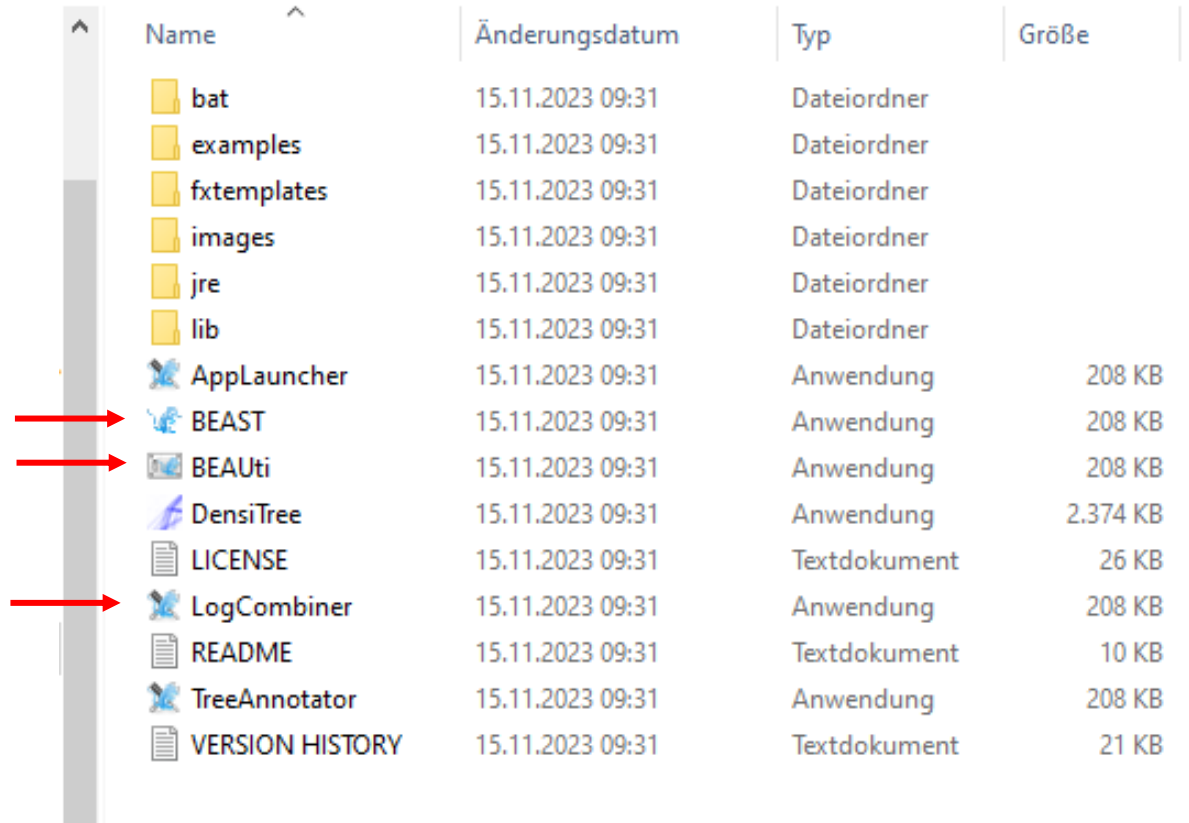
Name	Änderungsdatum	Typ	Größe
bat	15.11.2023 09:31	Dateiordner	
examples	15.11.2023 09:31	Dateiordner	
fxtemplates	15.11.2023 09:31	Dateiordner	
images	15.11.2023 09:31	Dateiordner	
jre	15.11.2023 09:31	Dateiordner	
lib	15.11.2023 09:31	Dateiordner	
AppLauncher	15.11.2023 09:31	Anwendung	208 KB
BEAST	15.11.2023 09:31	Anwendung	208 KB
BEAUti	15.11.2023 09:31	Anwendung	208 KB
DensiTree	15.11.2023 09:31	Anwendung	2.374 KB
LICENSE	15.11.2023 09:31	Textdokument	26 KB
LogCombiner	15.11.2023 09:31	Anwendung	208 KB
README	15.11.2023 09:31	Textdokument	10 KB
TreeAnnotator	15.11.2023 09:31	Anwendung	208 KB
VERSION HISTORY	15.11.2023 09:31	Textdokument	21 KB

- Navigate to the software folder
- A few java executables for different tasks (no need of installation!)
- BEAUti: help producing inputs for the analyses
- BEAST: run the mcmc chain

MCMC (Markov chain Monte Carlo): methods and algorithms for sampling from a continuous variable of known probability distribution. Used in Bayesian Statistics

Name	Änderungsdatum	Typ	Größe
bat	15.11.2023 09:31	Dateiordner	
examples	15.11.2023 09:31	Dateiordner	
fxtemplates	15.11.2023 09:31	Dateiordner	
images	15.11.2023 09:31	Dateiordner	
jre	15.11.2023 09:31	Dateiordner	
lib	15.11.2023 09:31	Dateiordner	
AppLauncher	15.11.2023 09:31	Anwendung	208 KB
BEAST	15.11.2023 09:31	Anwendung	208 KB
BEAUti	15.11.2023 09:31	Anwendung	208 KB
DensiTree	15.11.2023 09:31	Anwendung	2.374 KB
LICENSE	15.11.2023 09:31	Textdokument	26 KB
LogCombiner	15.11.2023 09:31	Anwendung	208 KB
README	15.11.2023 09:31	Textdokument	10 KB
TreeAnnotator	15.11.2023 09:31	Anwendung	208 KB
VERSION HISTORY	15.11.2023 09:31	Textdokument	21 KB

- Navigate to the software folder
- A few java executables for different tasks (no need of installation!)
- BEAUti: help producing inputs for the analyses
- BEAST: run the mcmc chain
- LogCombiner: combines log and tree files from different analyses

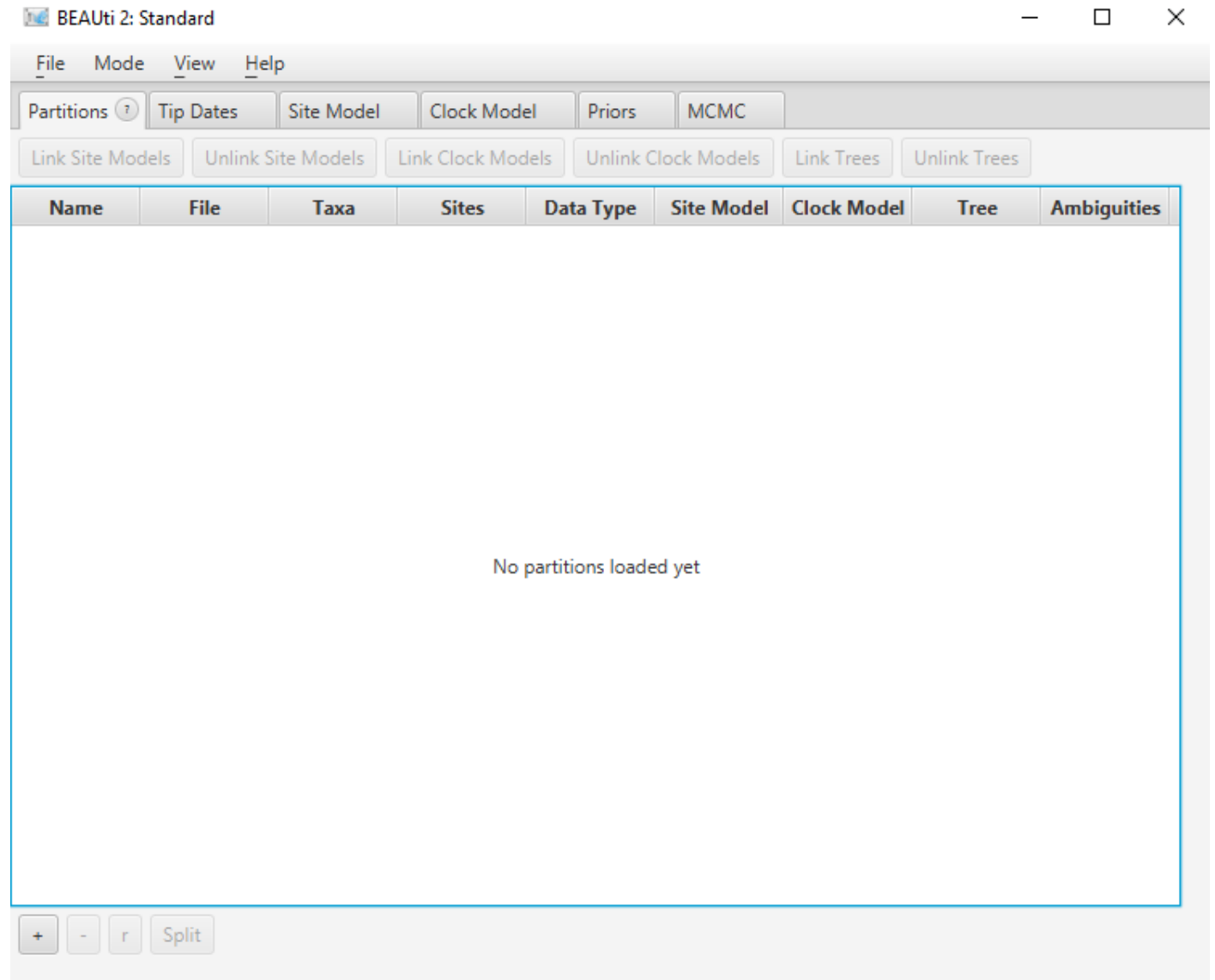


Name	Änderungsdatum	Typ	Größe
bat	15.11.2023 09:31	Dateiordner	
examples	15.11.2023 09:31	Dateiordner	
fxtemplates	15.11.2023 09:31	Dateiordner	
images	15.11.2023 09:31	Dateiordner	
jre	15.11.2023 09:31	Dateiordner	
lib	15.11.2023 09:31	Dateiordner	
AppLauncher	15.11.2023 09:31	Anwendung	208 KB
BEAST	15.11.2023 09:31	Anwendung	208 KB
BEAUti	15.11.2023 09:31	Anwendung	208 KB
DensiTree	15.11.2023 09:31	Anwendung	2.374 KB
LICENSE	15.11.2023 09:31	Textdokument	26 KB
LogCombiner	15.11.2023 09:31	Anwendung	208 KB
README	15.11.2023 09:31	Textdokument	10 KB
TreeAnnotator	15.11.2023 09:31	Anwendung	208 KB
VERSION HISTORY	15.11.2023 09:31	Textdokument	21 KB

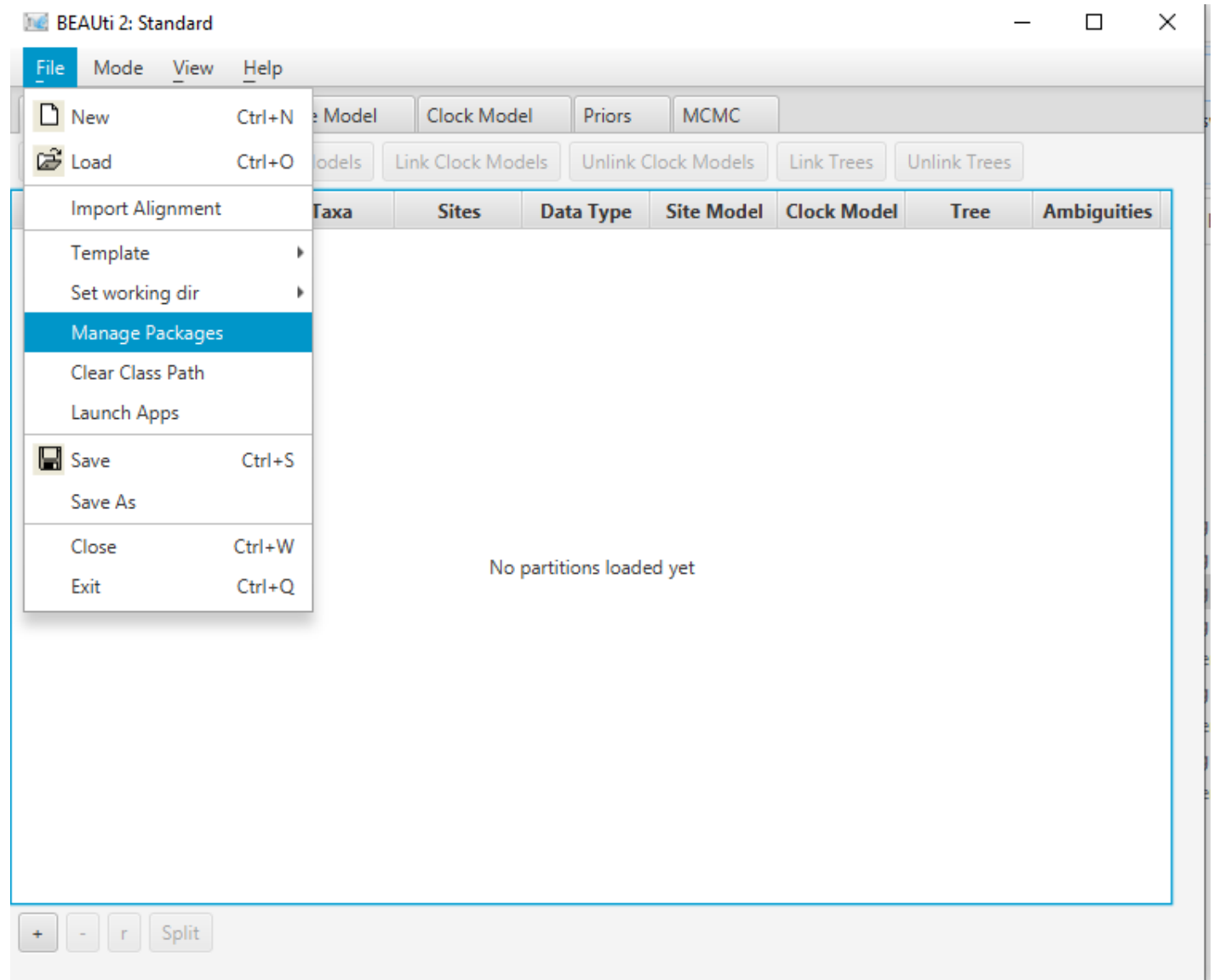
- Navigate to the software folder
- A few java executables for different tasks (no need of installation!)
- BEAUti: help producing inputs for the analyses
- BEAST: run the mcmc chain
- LogCombiner: combines log and tree files from different analyses
- TreeAnnotator: summarize phylogenetic trees from the tree files produced during the mcmc runs

Name	Änderungsdatum	Typ	Größe
bat	15.11.2023 09:31	Dateiordner	
examples	15.11.2023 09:31	Dateiordner	
fxtemplates	15.11.2023 09:31	Dateiordner	
images	15.11.2023 09:31	Dateiordner	
jre	15.11.2023 09:31	Dateiordner	
lib	15.11.2023 09:31	Dateiordner	
AppLauncher	15.11.2023 09:31	Anwendung	208 KB
BEAST	15.11.2023 09:31	Anwendung	208 KB
BEAUti	15.11.2023 09:31	Anwendung	208 KB
DensiTree	15.11.2023 09:31	Anwendung	2.374 KB
LICENSE	15.11.2023 09:31	Textdokument	26 KB
LogCombiner	15.11.2023 09:31	Anwendung	208 KB
README	15.11.2023 09:31	Textdokument	10 KB
TreeAnnotator	15.11.2023 09:31	Anwendung	208 KB
VERSION HISTORY	15.11.2023 09:31	Textdokument	21 KB

- launch BEAUti



- launch BEAUti
- Check in the package manager if the packages we need are installed
 - SPEEDEMON
 - StarBeast3



- SPEEDEMON is a BEAST 2 package for fast species delimitation under the multispecies coalescent.
 - It can be used on multi-locus sequence data (based on StarBeast3) or SNP data (based on SNAPPER)



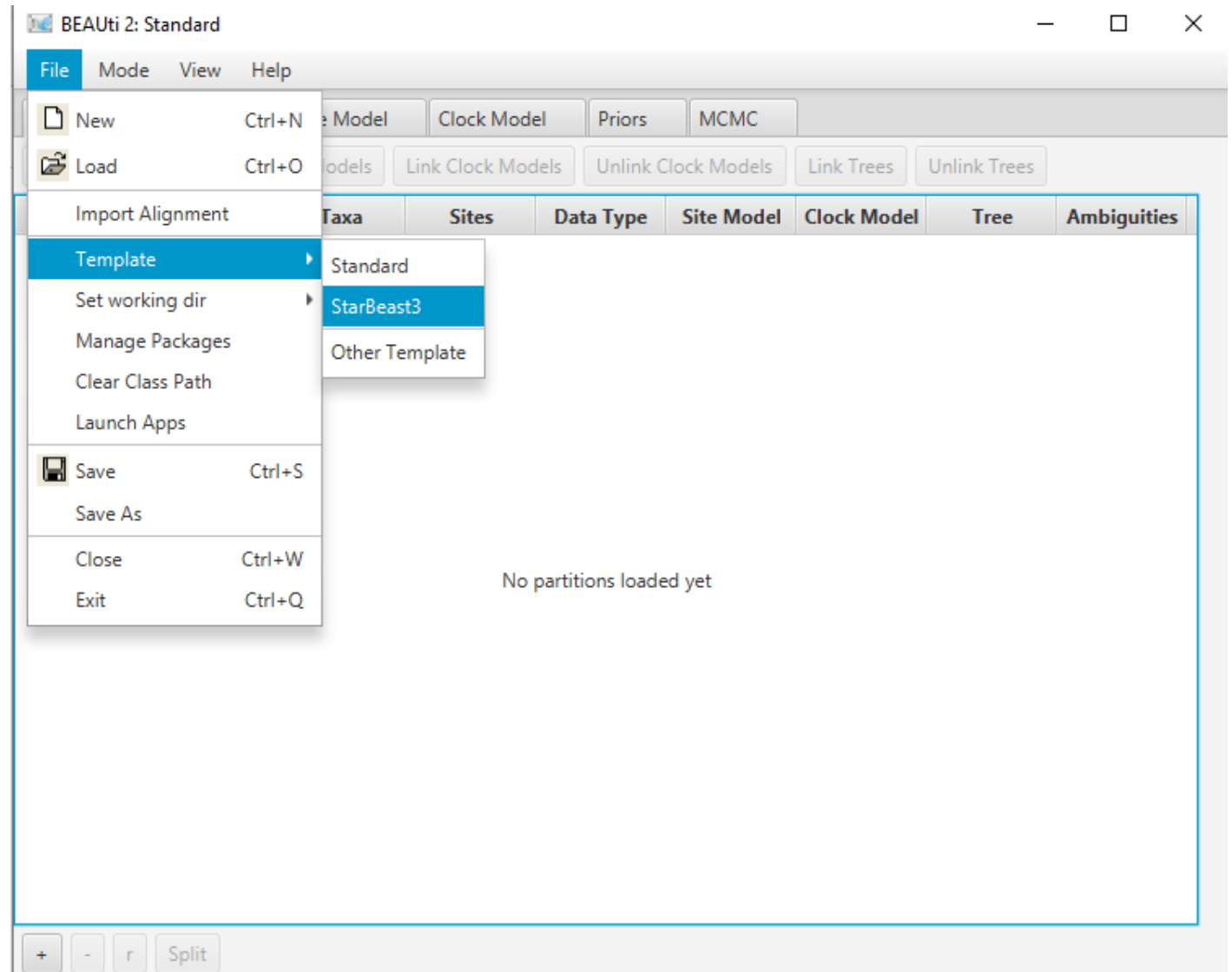
- SPEEDEMON is a BEAST 2 package for fast species delimitation under the multispecies coalescent.
 - It can be used on multi-locus sequence data (based on StarBeast3) or SNP data (based on SNAPPER)

Species boundaries are applied under the tree collapse model

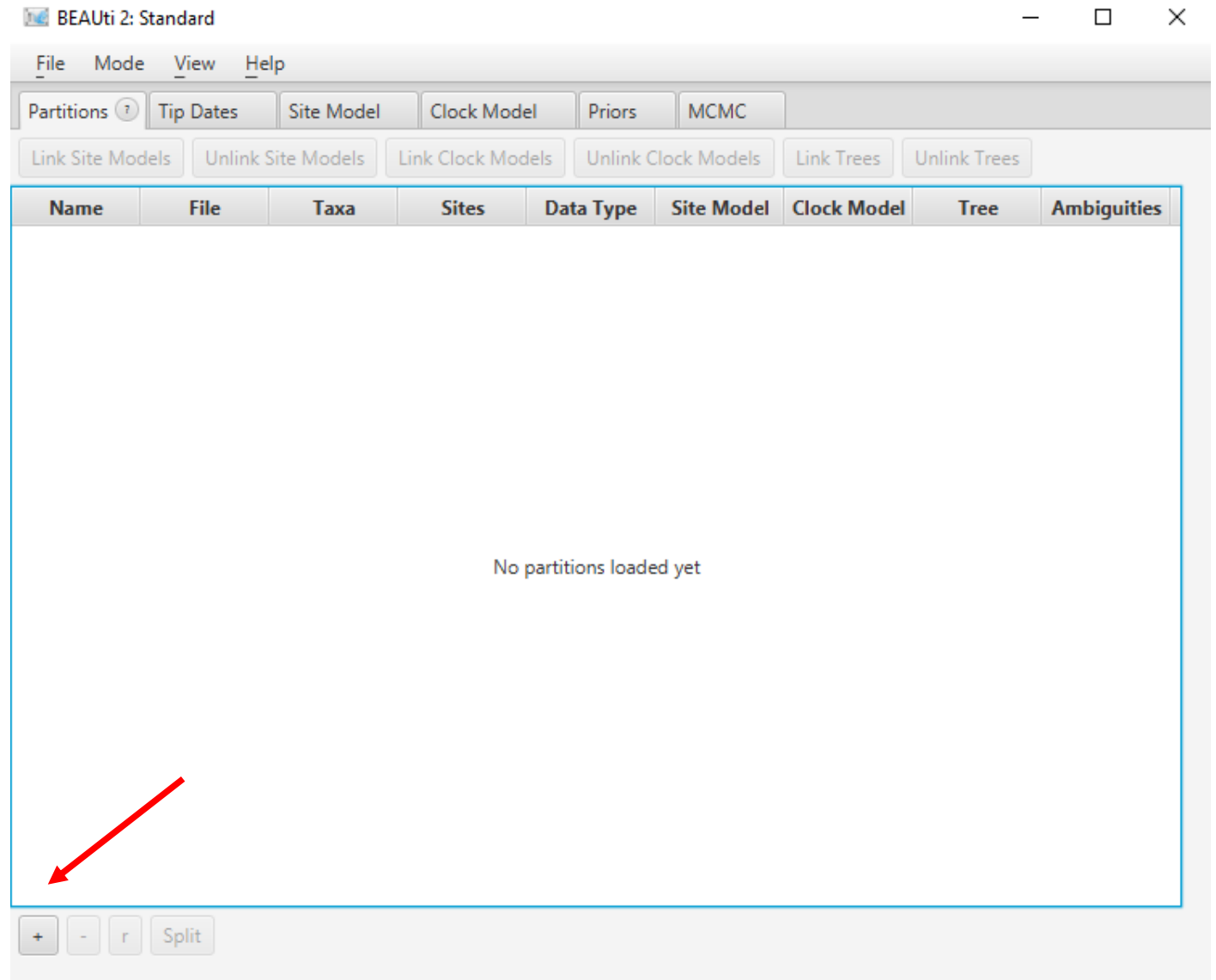
- samples whose ancestral species time falls below threshold epsilon are collapse into a single species
- Similar to STACEY, but allowing that speciation rate varies through time



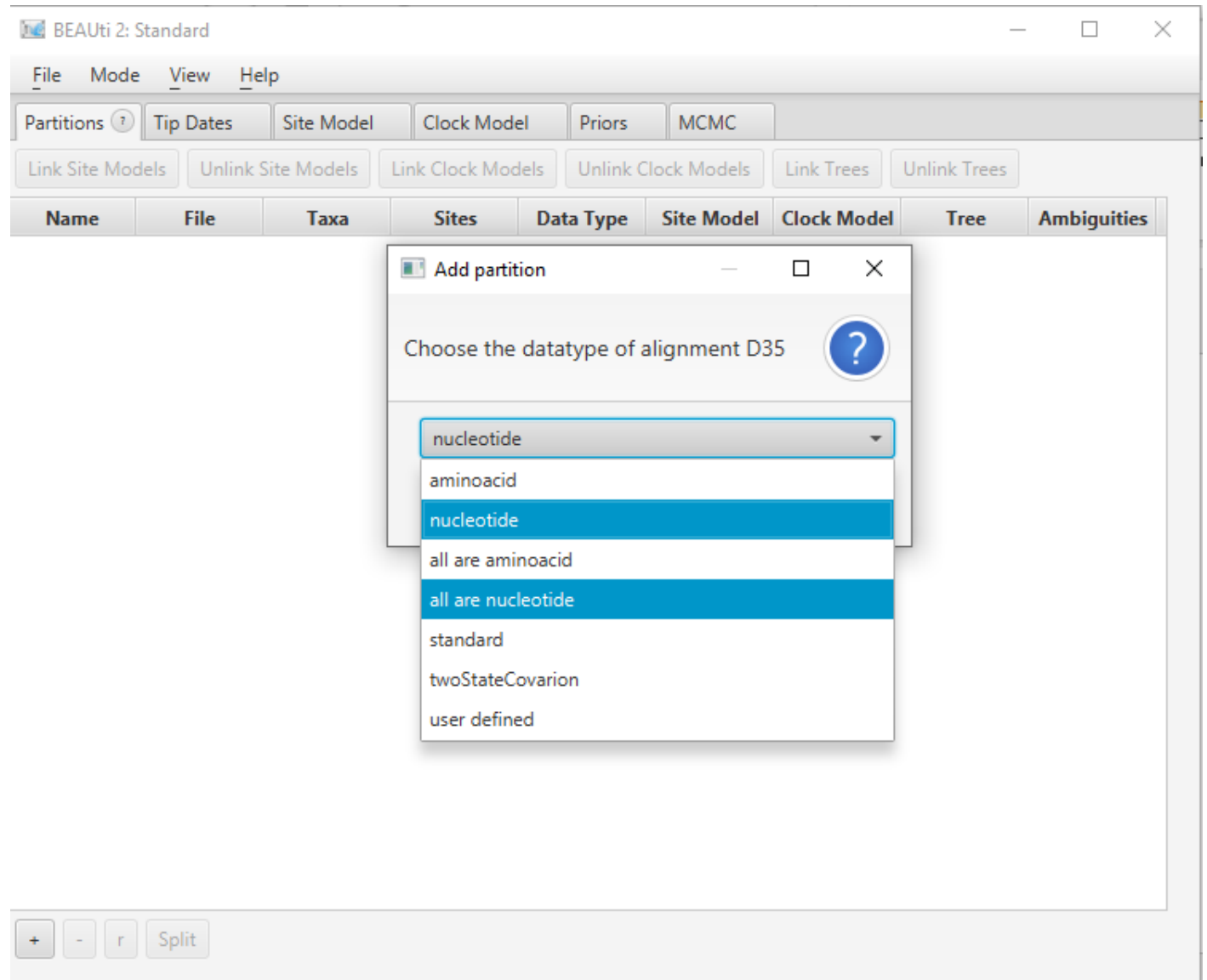
- Select the StarBeast3 template



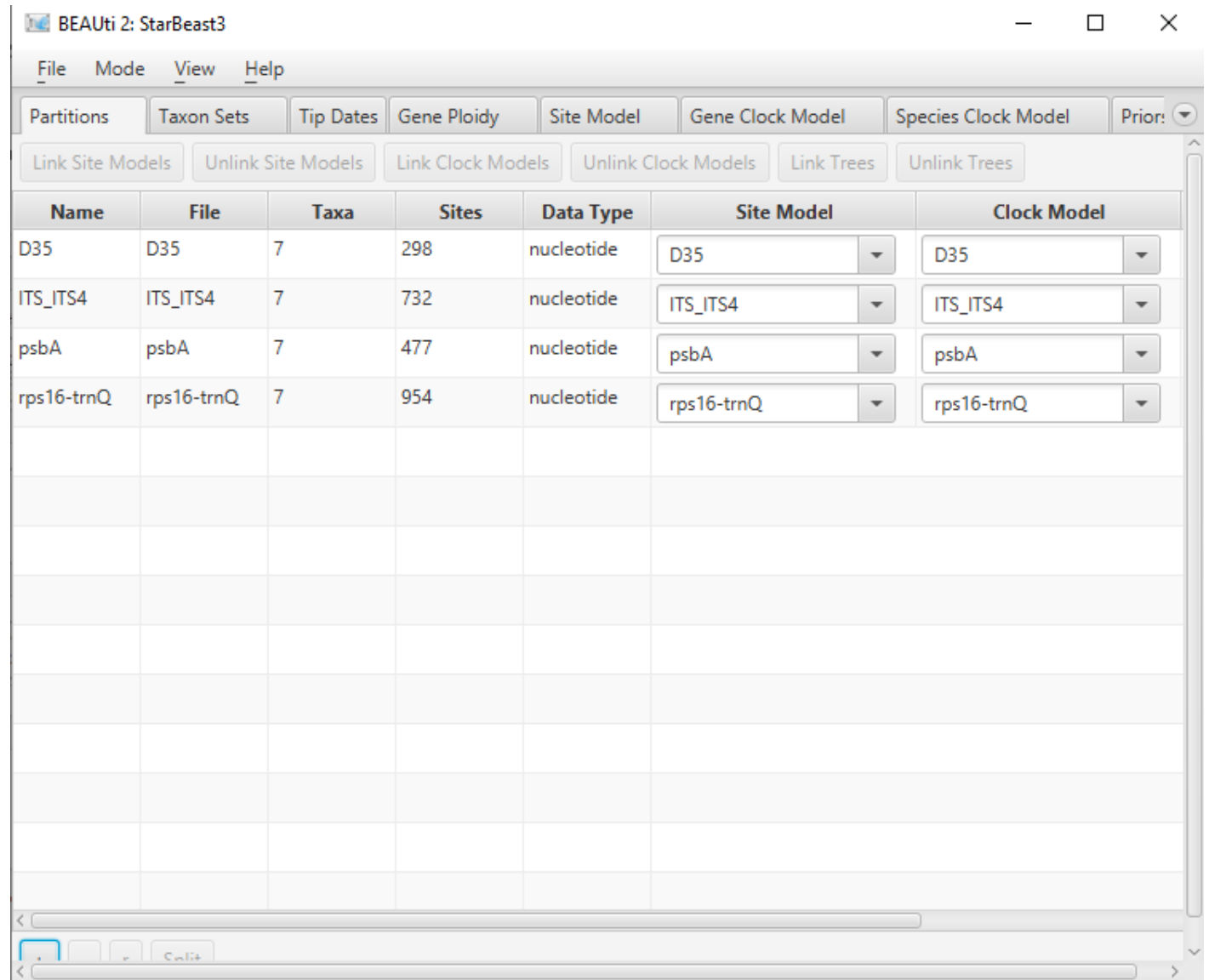
- Select the StarBeast3 template
- import the alignments:
- 4 in total
 - 2 plastid regions, *psbA-trnH* and *rsp16-trnQ*;
 - 2 nuclear, ITS and *D35* (a single-copy gene)



- Select the StarBeast3 template
- import the alignments:
- 4 in total
 - 2 plastid regions, *psbA-trnH* and *rsp16-trnQ*;
 - 2 nuclear, ITS and *D35* (a single-copy gene)

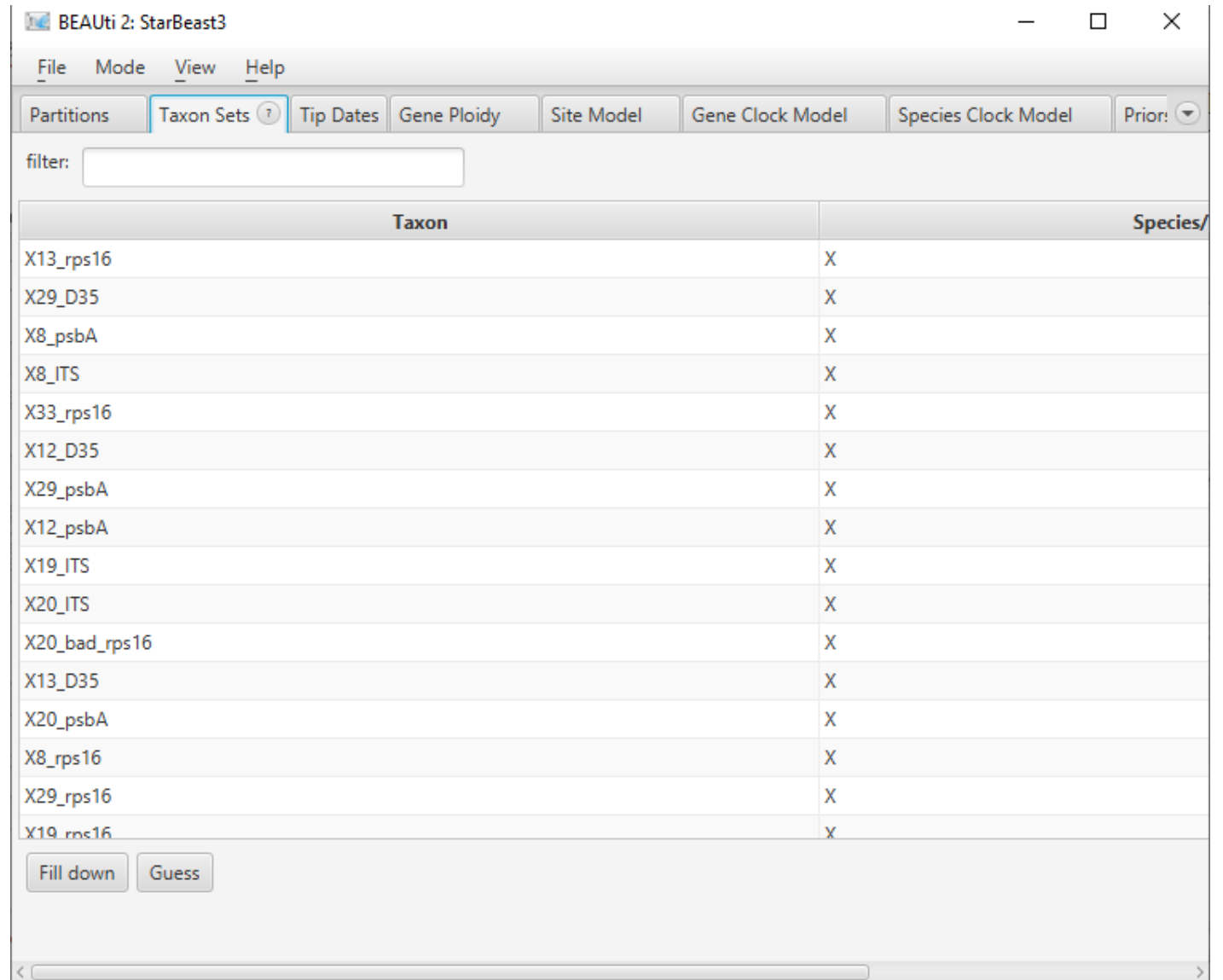


- click on one of the alignments!
 - Are there differences
 - How many sites?



Taxon Panel

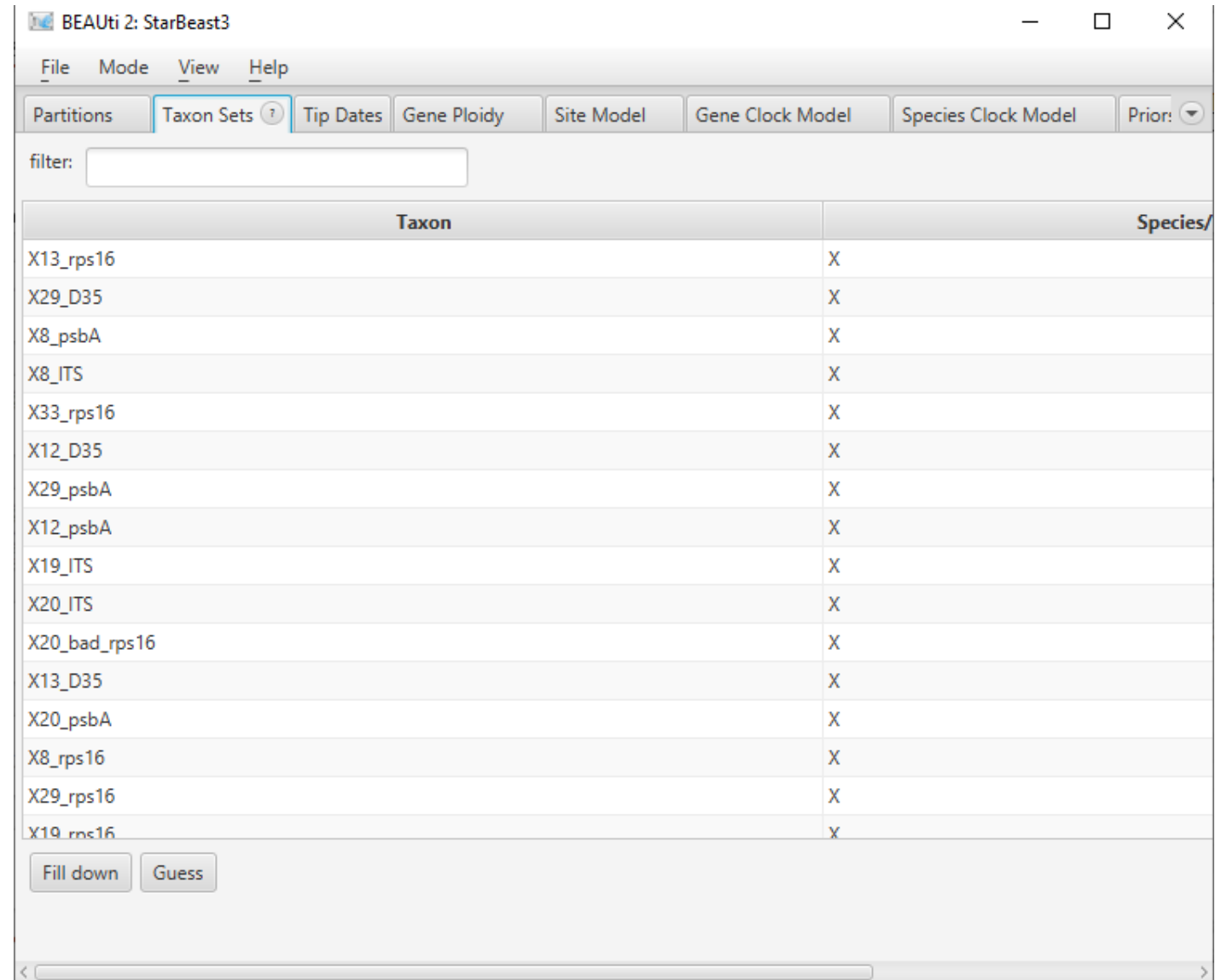
- Here we can assign the sequences of the alignments to samples/populations/species
- Sequences of the same samples have different names in different alignment!!



Taxon Panel

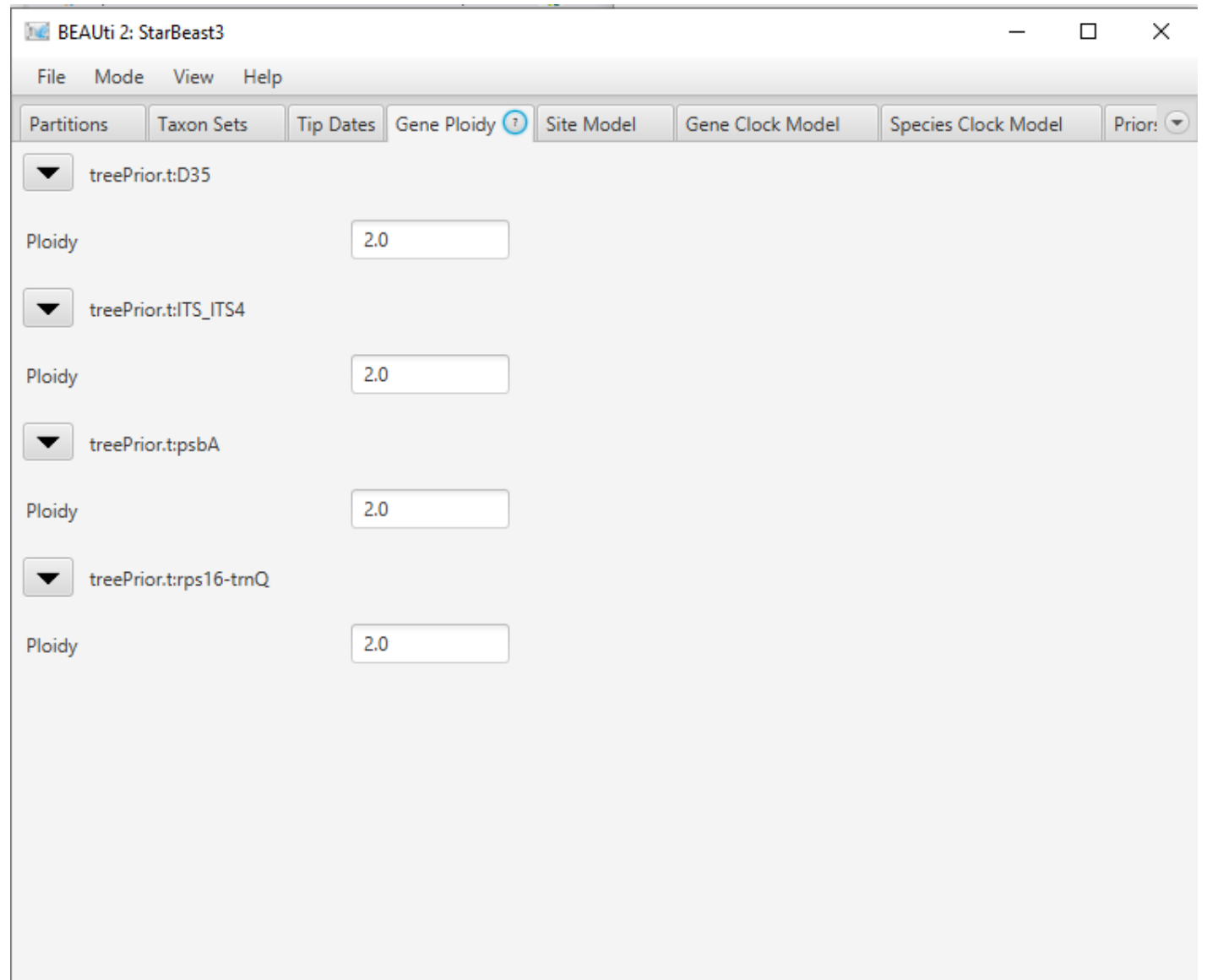
- you can either provide a “mapping” text file
- or you can “guess” it using (e.g.) characters or text expressions

Check if all sequences are assigned to a sample!!!



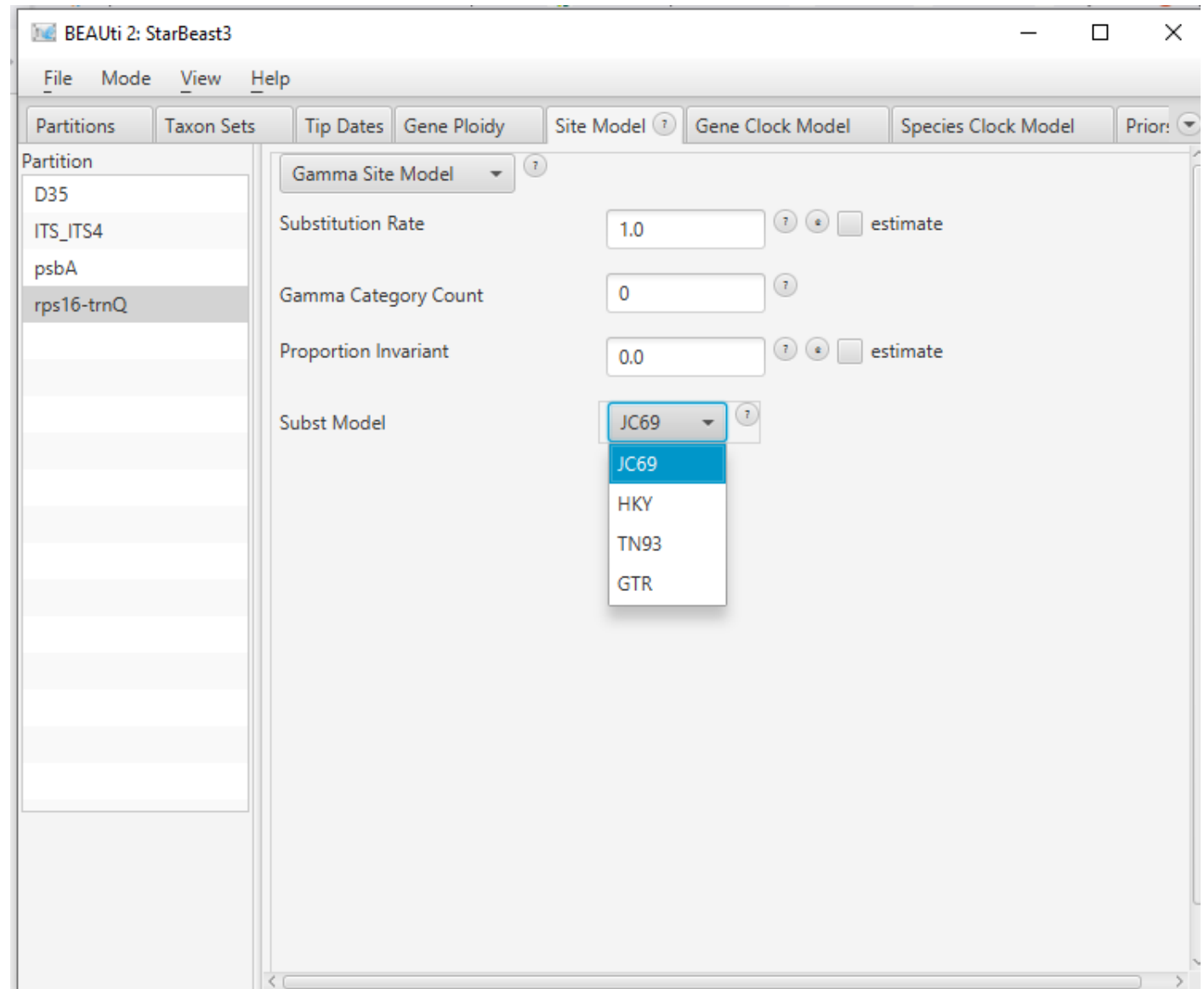
Ploidy Panel

- Change the ploidy of the genomic regions if necessary
- Leave 2.0 for the nuclear regions
- What should we put on the plastid ones??

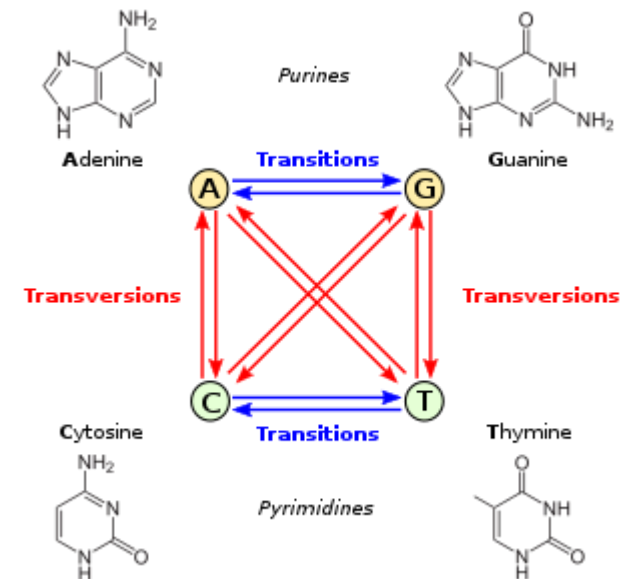


Site Model Panel

- Choose a model of sequence evolution for each of the regions
- You can let BEAST estimate parameters (e.g., Gamma, Proportion of invariant sites)
- Or fix them to a predetermined values (e.g., found in ModelTest or similar...)

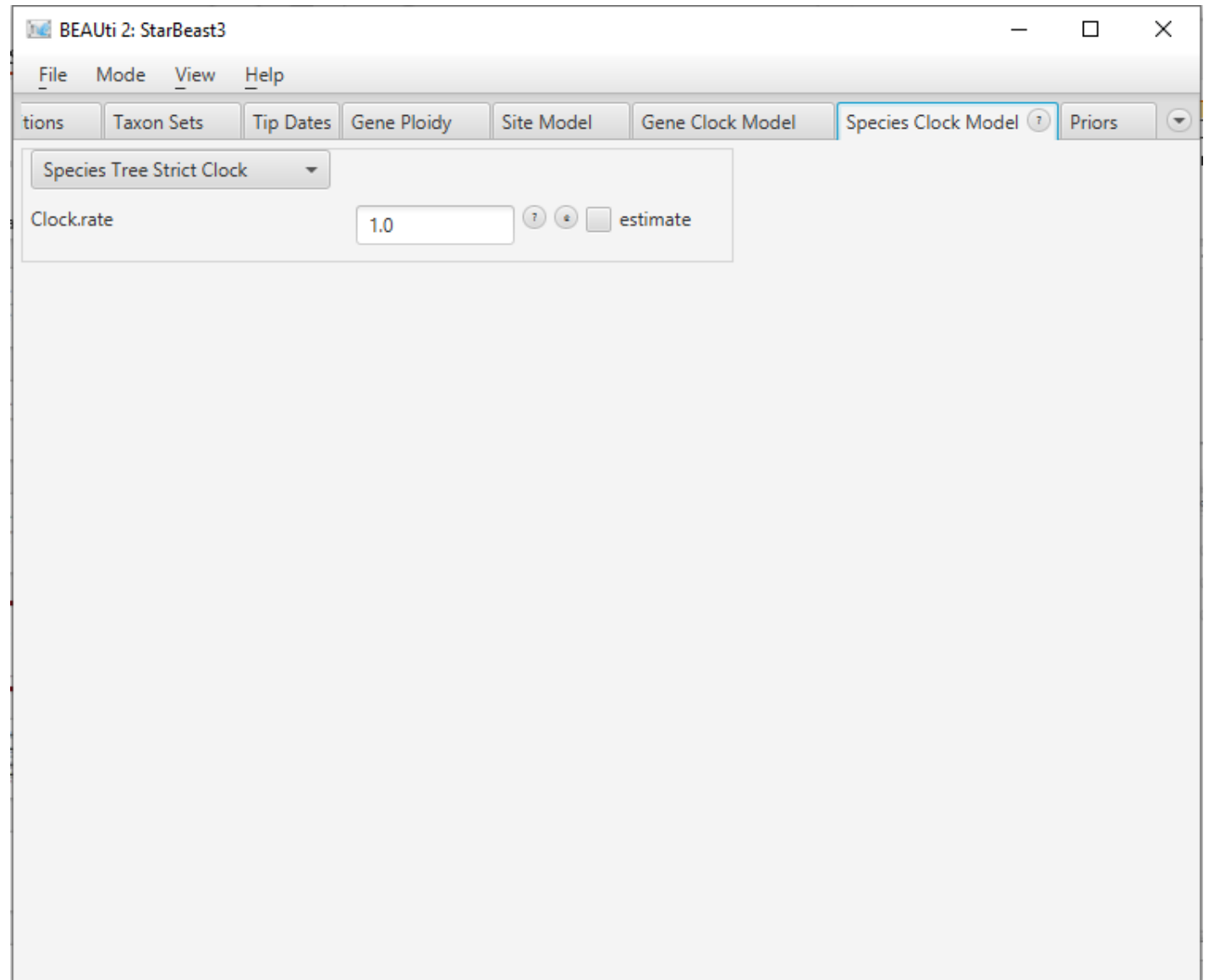


- JC69 model (Jukes and Cantor 1969): equal base frequencies and equal mutation rates. The only parameter of this model is the overall substitution rate
- HKY (Hasegawa, Kishino and Yano 1985): allows unequal base frequencies. It distinguishes between the rate of transitions and transversions (k parameter ratio between them)
- TN93 model (Tamura and Nei 1993): distinguishes between the two different types of transition
- GTR (Generalised time-reversible model, Tavaré 1986): The most complex and neutral. Six substitution rate parameters, as well as 4 equilibrium base frequency parameters.



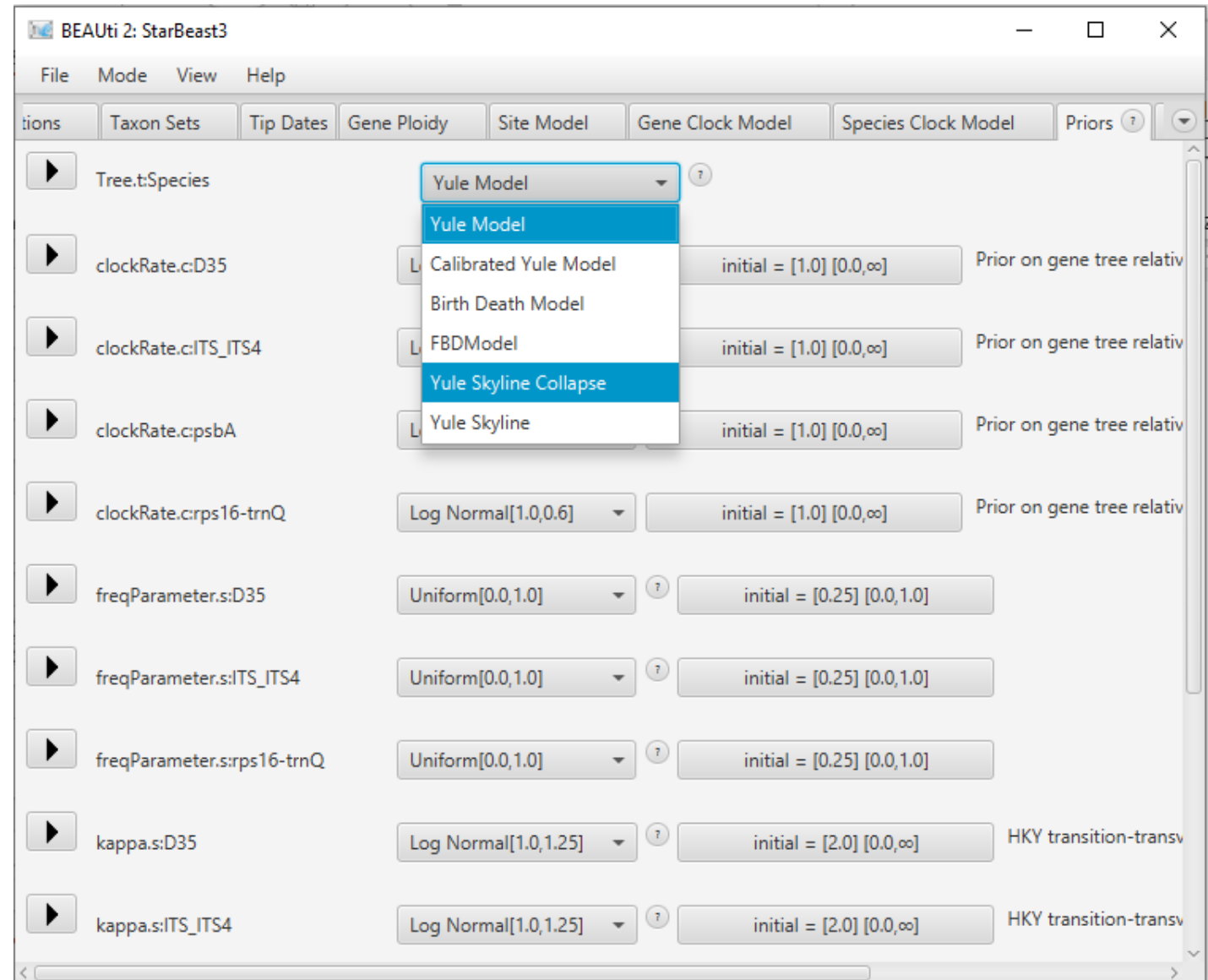
Clock Model Panel

- Here you can select a clock model.
- Strict: equal rates overall
- Relaxed: allowed different rates for different branches of the tree



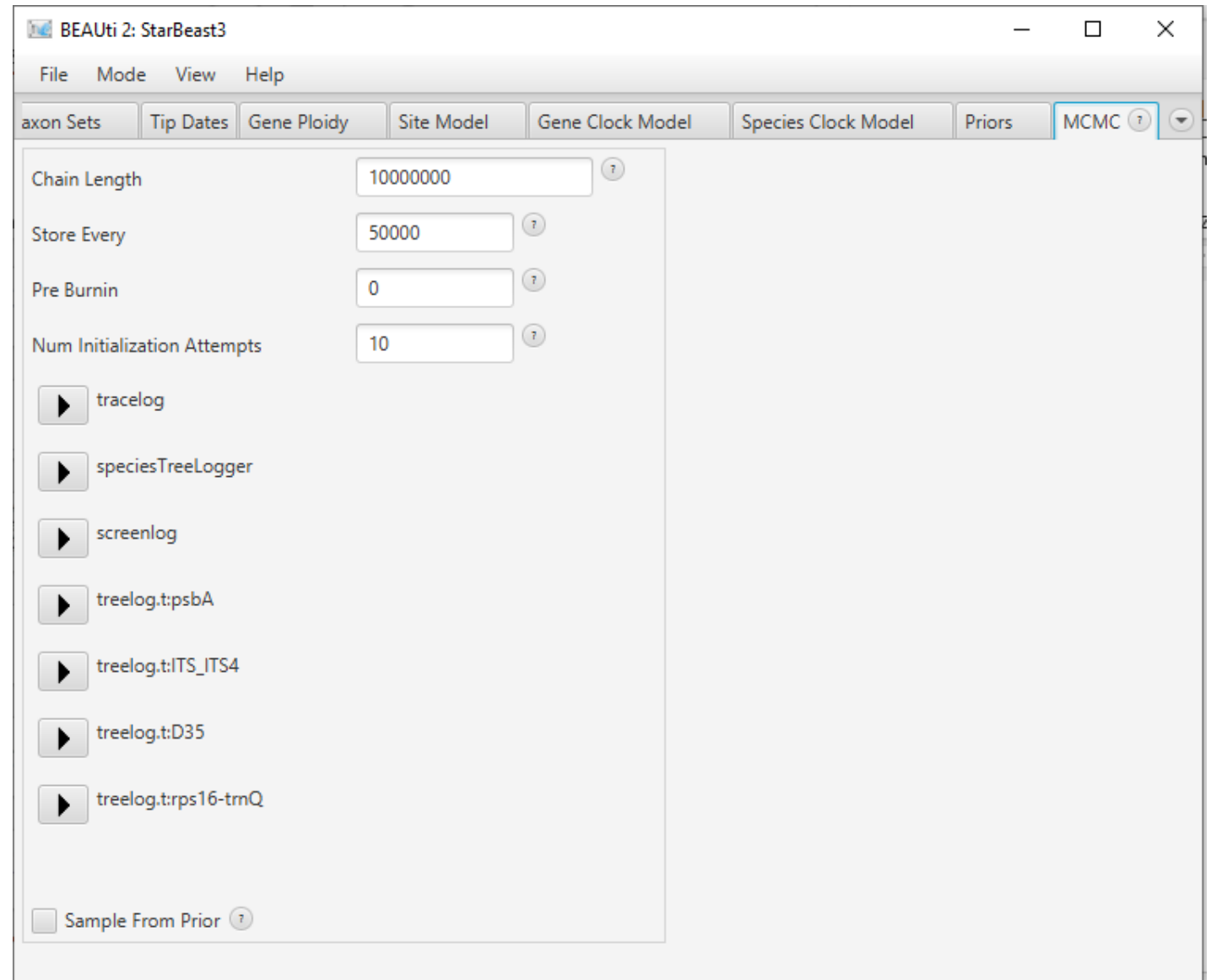
Prior Panel

- In the species tree Prior, set the Yule Skyline Collapse (the model used by SPEEDEMON)
- Leave the rest as default



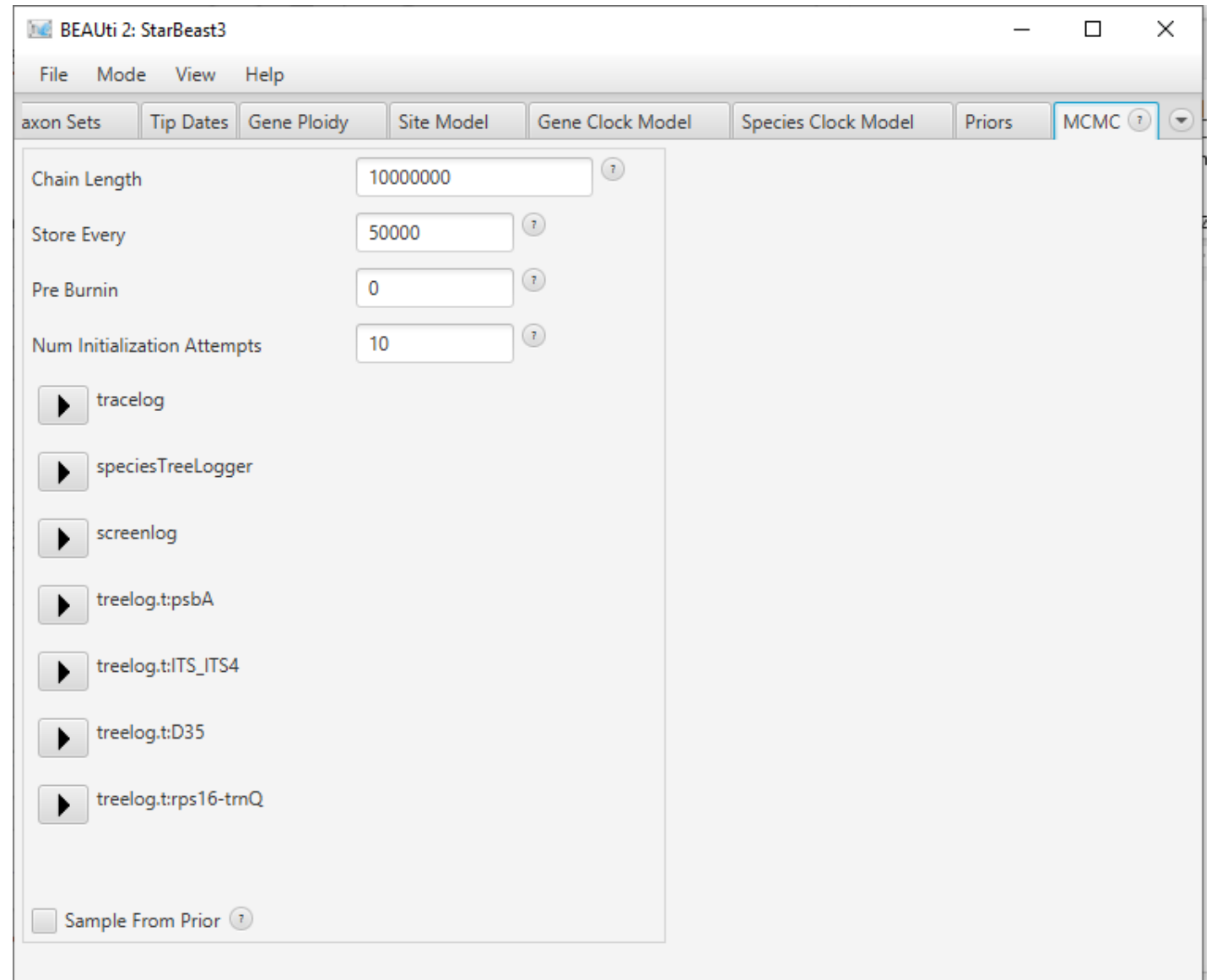
mcmc Panel

- set the length of the mcmc chain
- The intervals for sampling (store every)
- Burnin: initial fraction of the chain that will be discharged (no need to specify it now...)
- Sample From Prior: if you want to sample just from the priors, not from the data (not in our case)



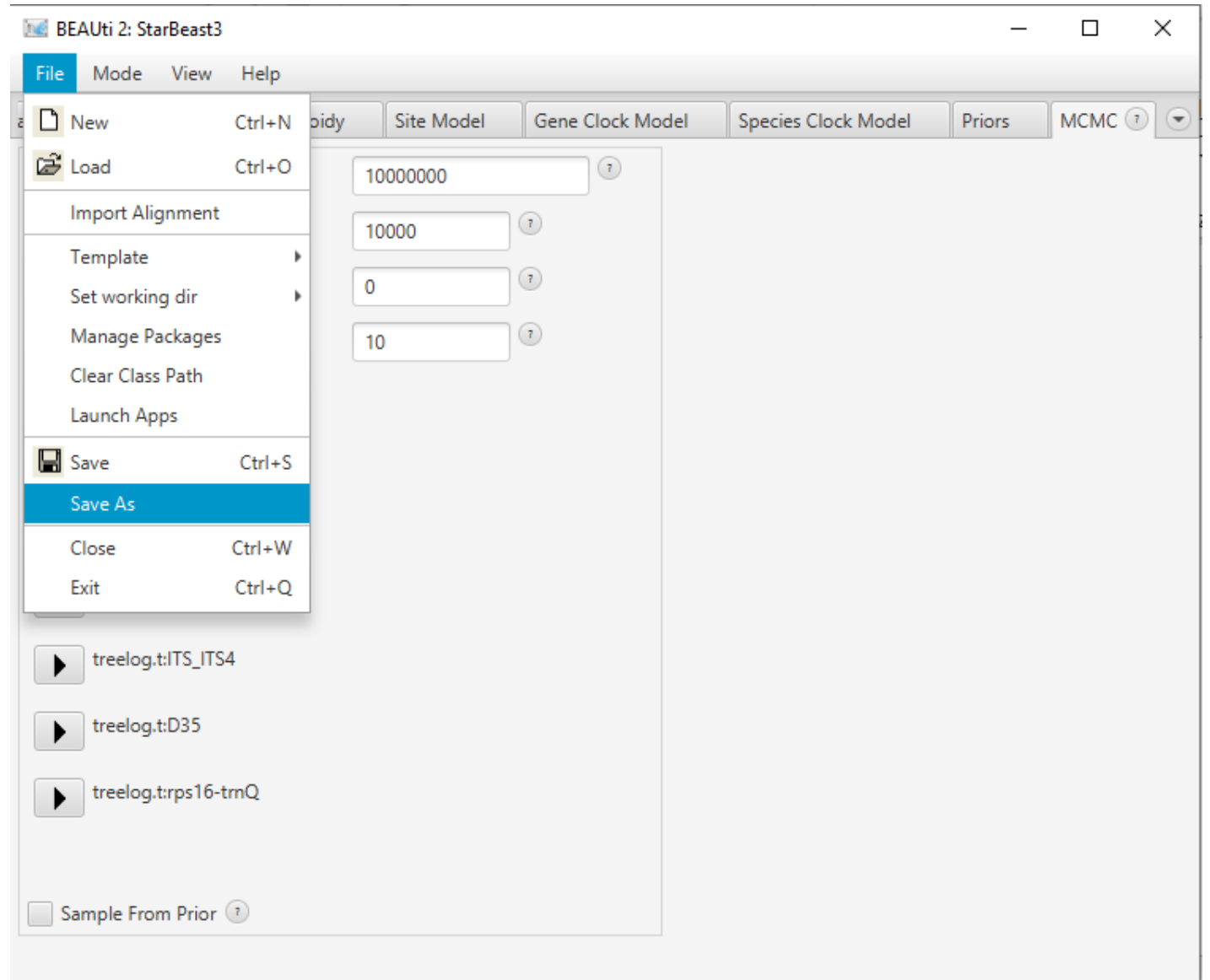
mcmc Panel

- The longer the chain, the longer the analyses.
- Sometime long chains are needed to reach good mixing and convergence.
- Do not sample too often. Make it in order to get 10.000 trees.



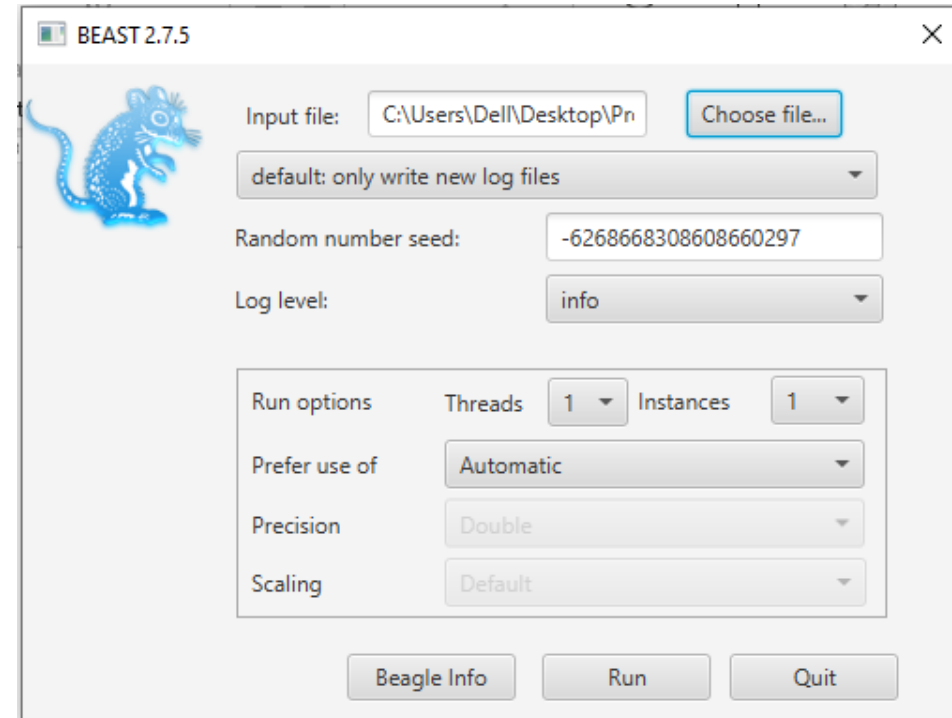
Save in the file

It will produce a .xml file,
which is the input for BEAST



Open BEAST

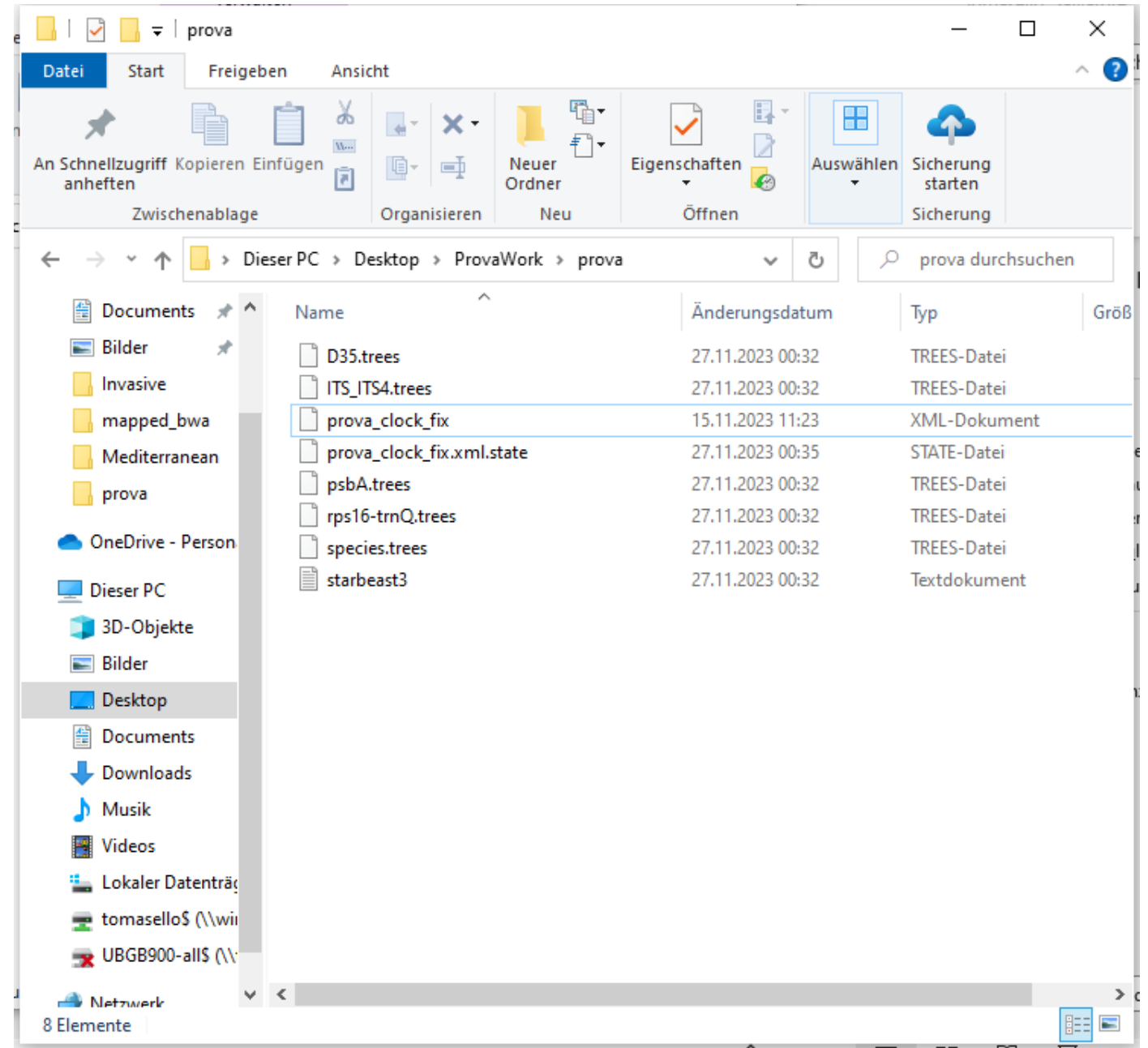
Chose the input .xml file, and
run BEAST



Coffe?

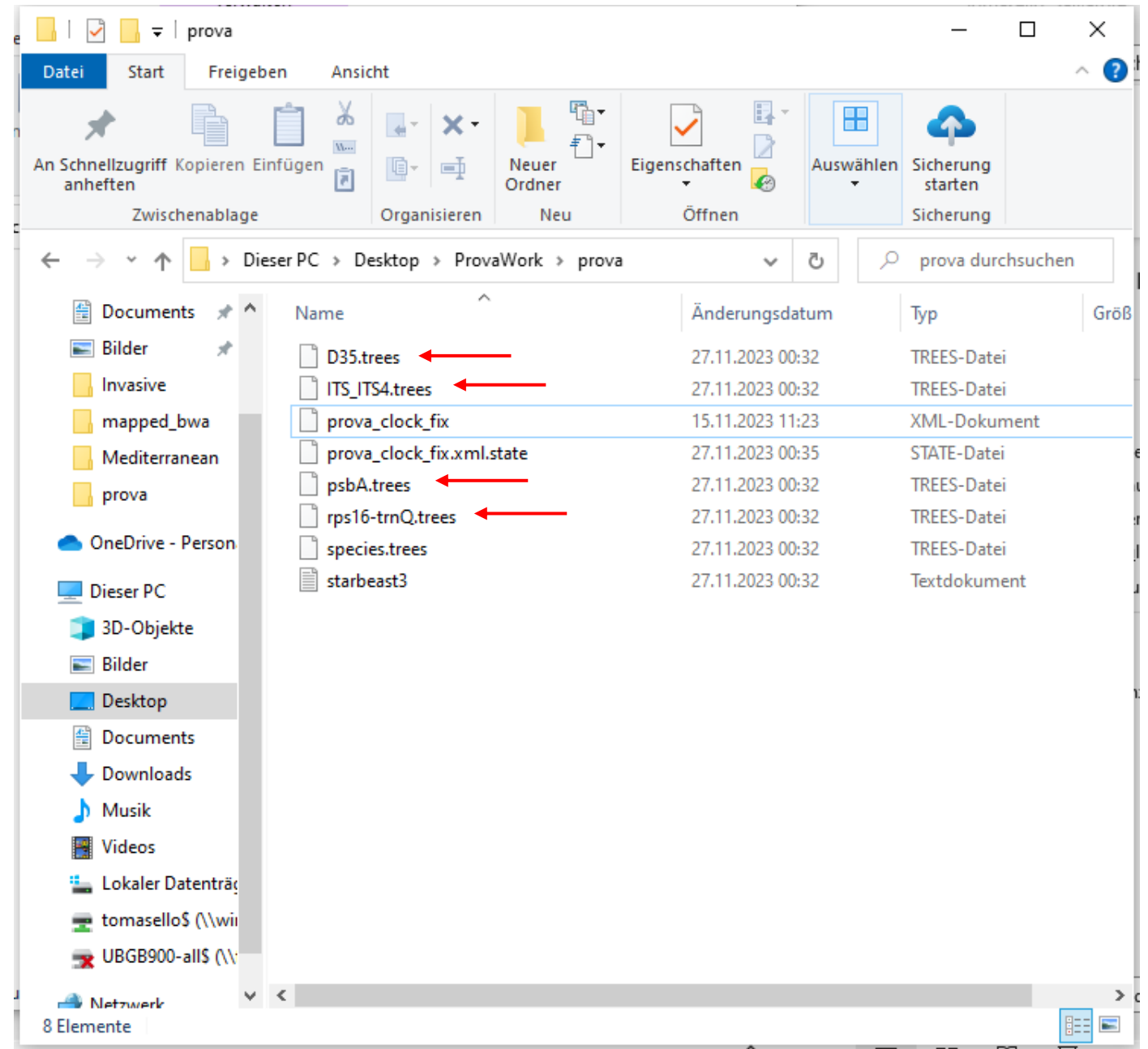


Different files have been created in the input folder:



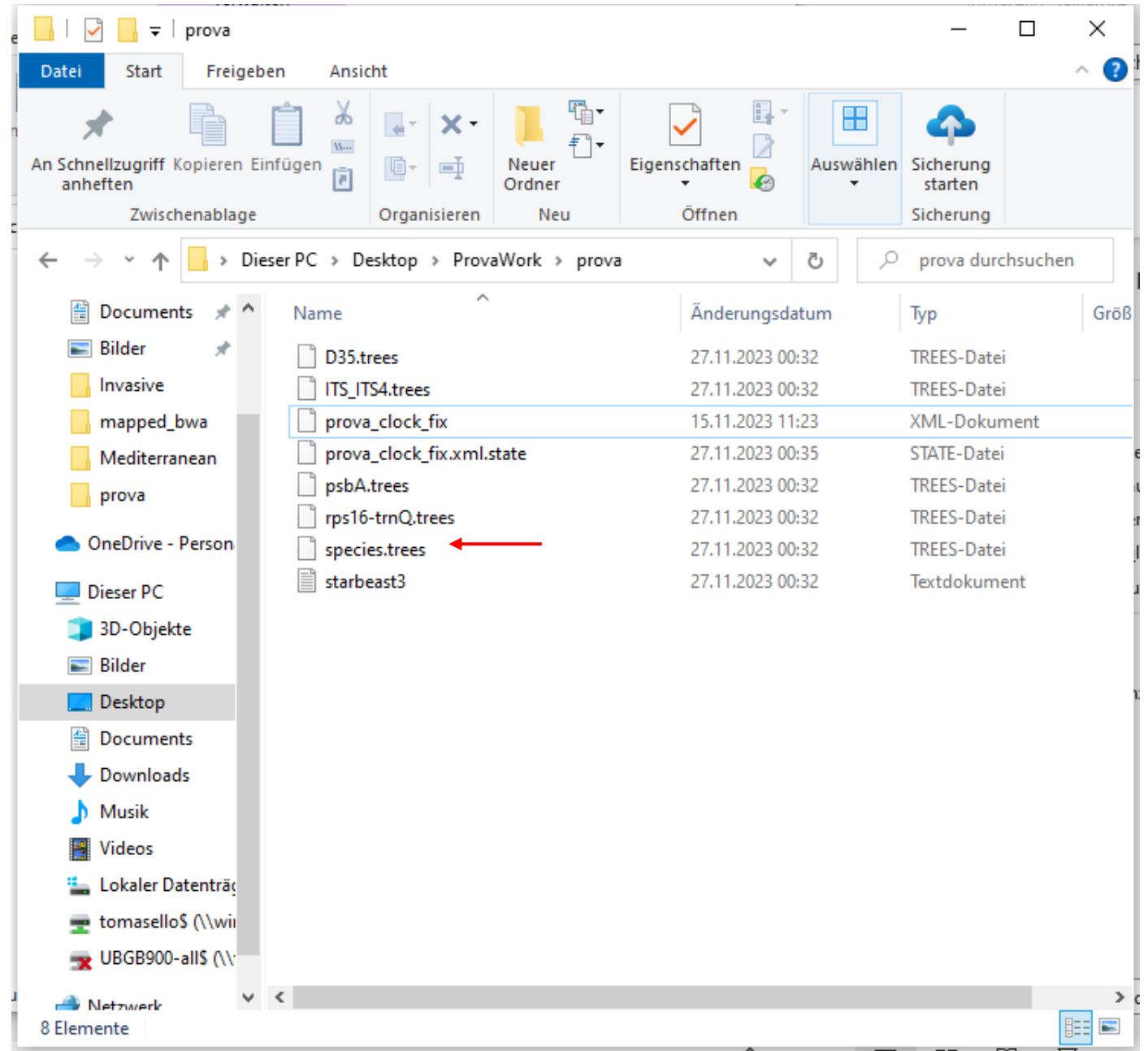
Different files have been created in the input folder:

- gene-tree files for each region



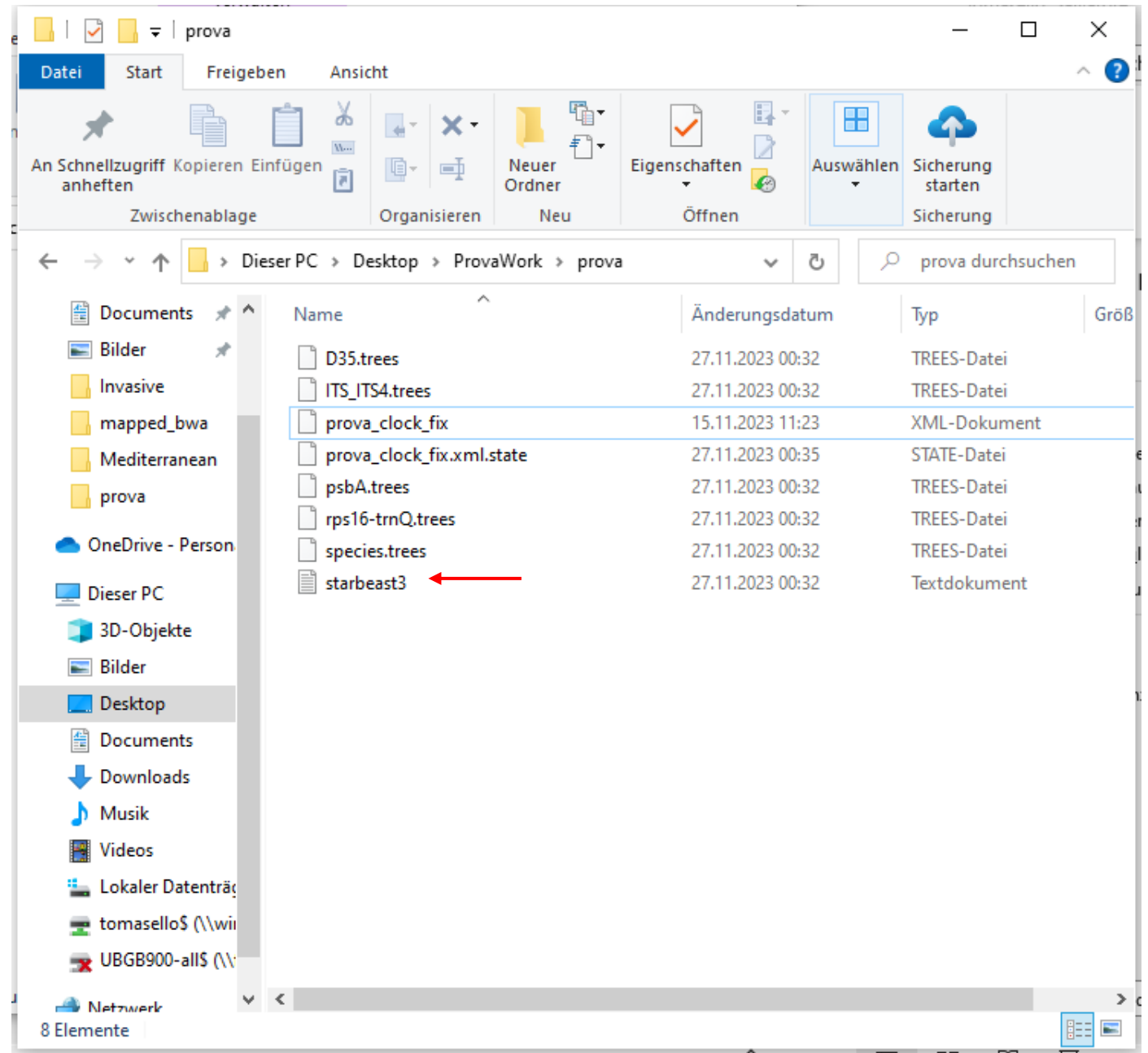
Different files have been created in the input folder:

- gene-tree files for each region
- a species-tree file



Different files have been created in the input folder:

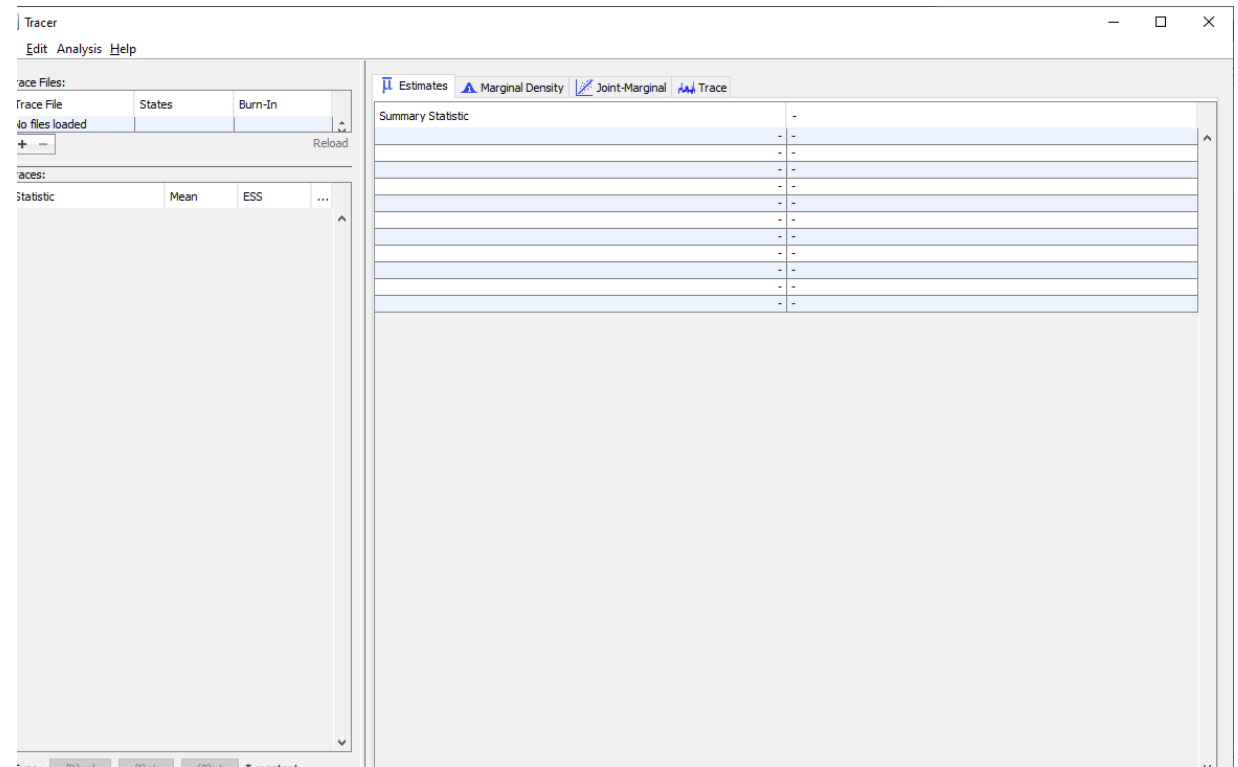
- gene-tree files for each region
- a species-tree file
- a log file



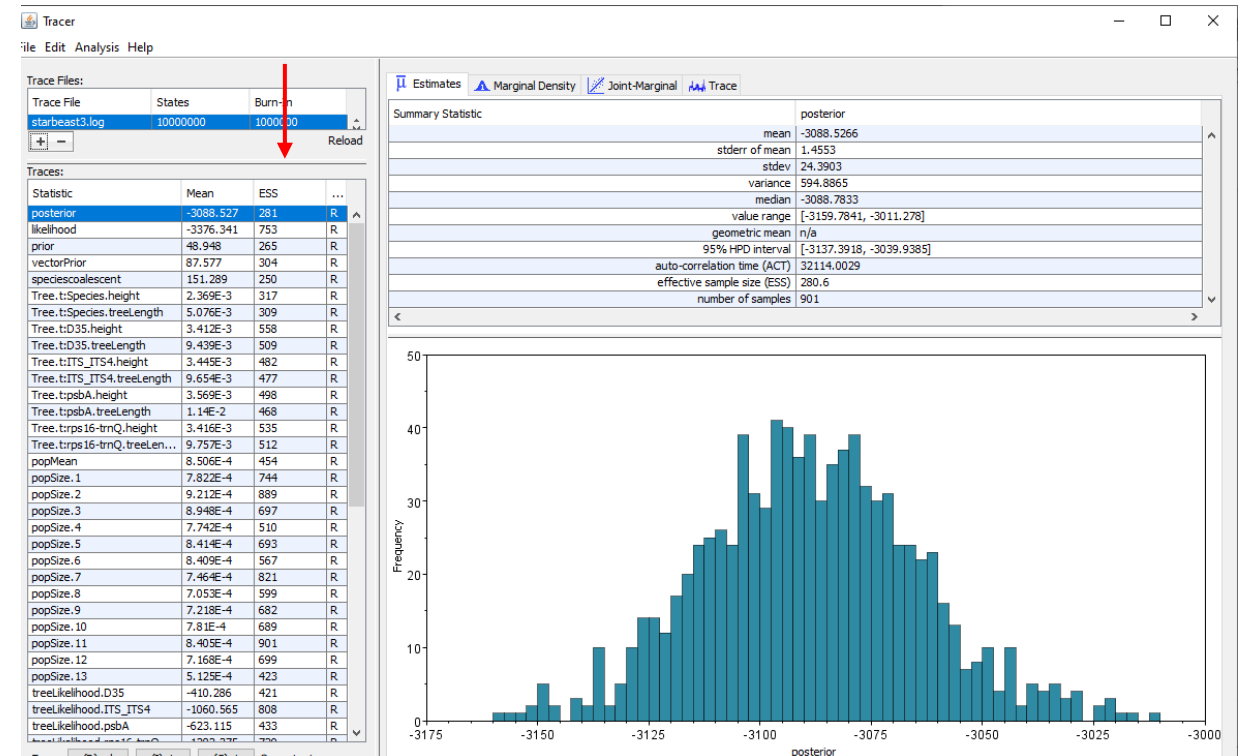
How to check if the analyses went well?

- Check in the log file to see if the analysis reached convergence
- Usually more independent analyses are run to check in they converge to similar results

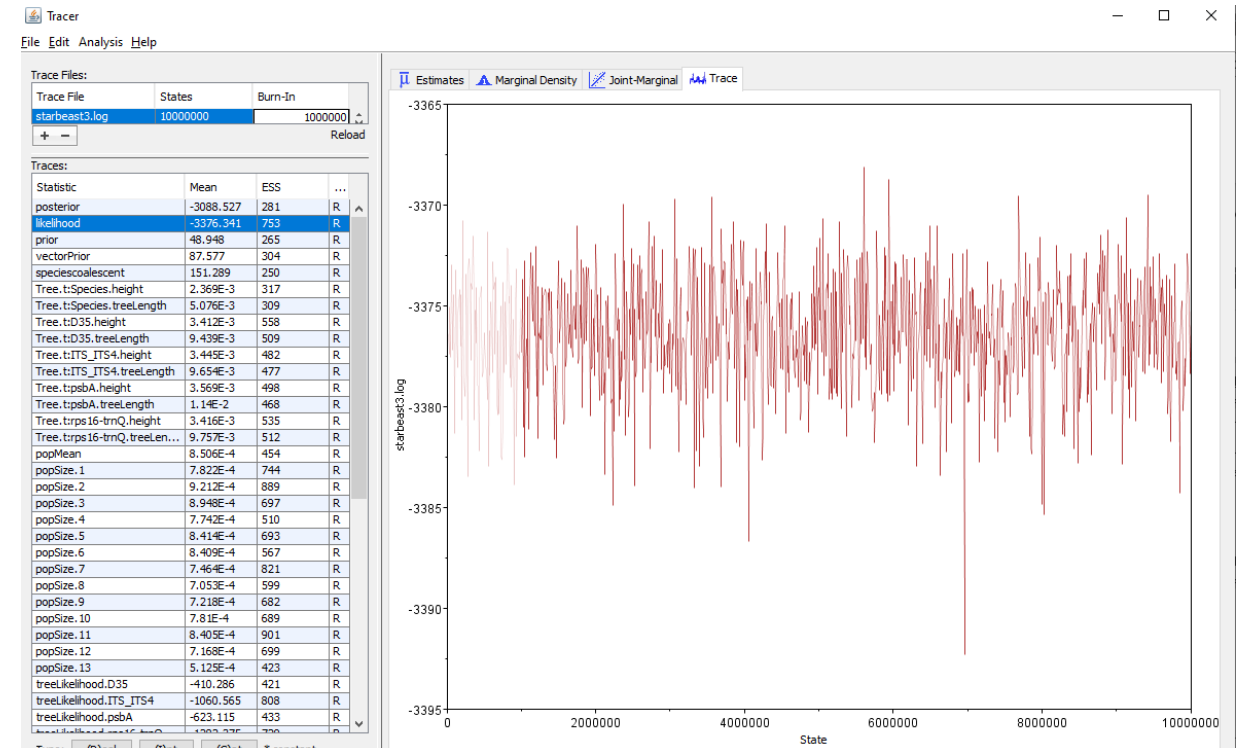
- Check in the log file to see if the analysis reached convergence
 - Open Tracer and load the .log file



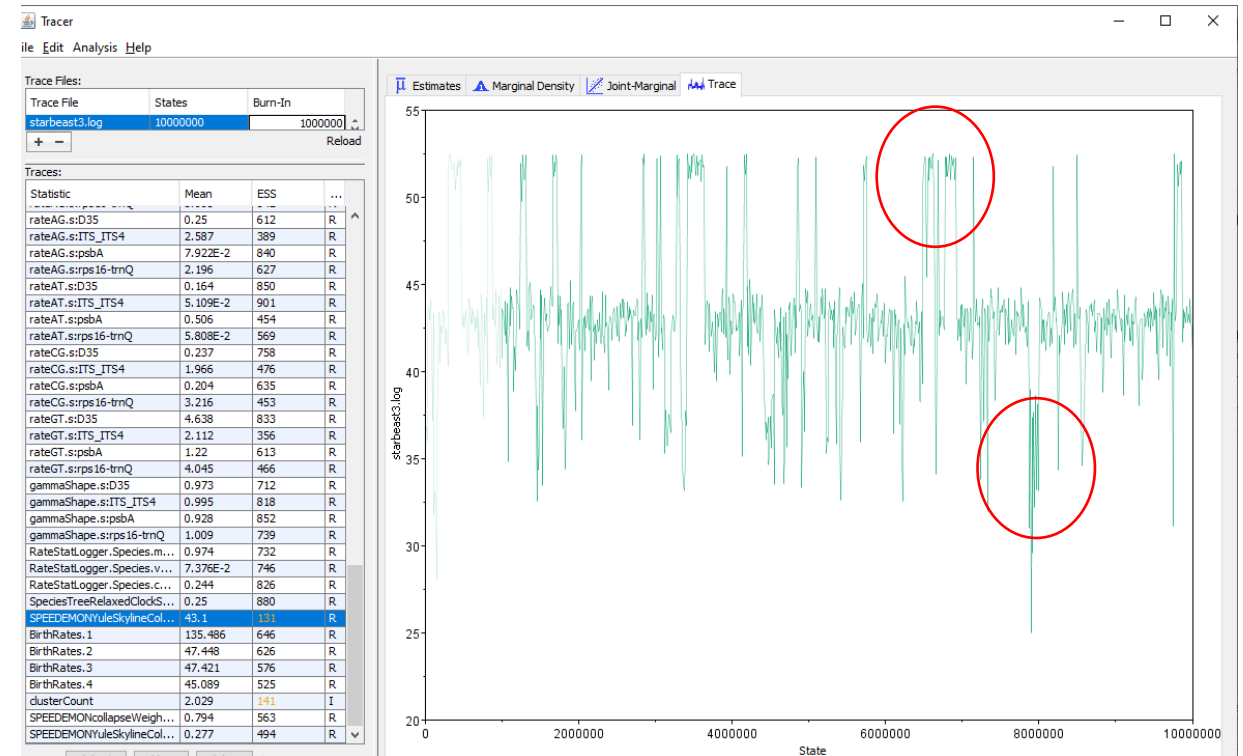
- Check in the log file to see if the analysis reached convergence
 - Open Tracer and load the .log file
 - ESS (Estimated Sample Size) should be always above 100 (better above 200) for all estimated parameters



- Check in the log file to see if the analysis reached convergence
 - Open Tracer and load the .log file
 - ESS (Estimated Sample Size) should be always above 100 (better above 200) for all estimated parameters
 - Check the trace for different parameters. It should have a “caterpillar” shape, indicating good mixing



- Check in the log file to see if the analysis reached convergence
 - Open Tracer and load the .log file
 - ESS (Effective Sample Size) should be always above 100 (better above 200) for all estimated parameters
- Check the trace for different parameters. It should have a “caterpillar” shape, indicating good mixing
- not jumping from a sub-optimum to another



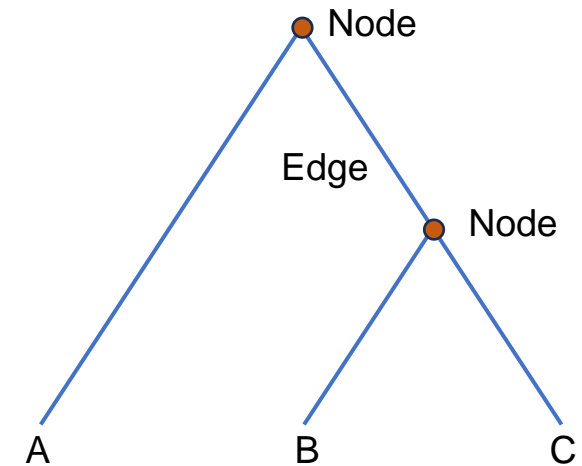
Summarize a phylogenetic trees:

- the mcmc has run for n iterations sampling every m generation
- It produced tree files containing n/m trees

Summarize a phylogenetic trees:

- the mcmc has run for n iterations sampling every m generation
- It produced tree files containing n/m trees

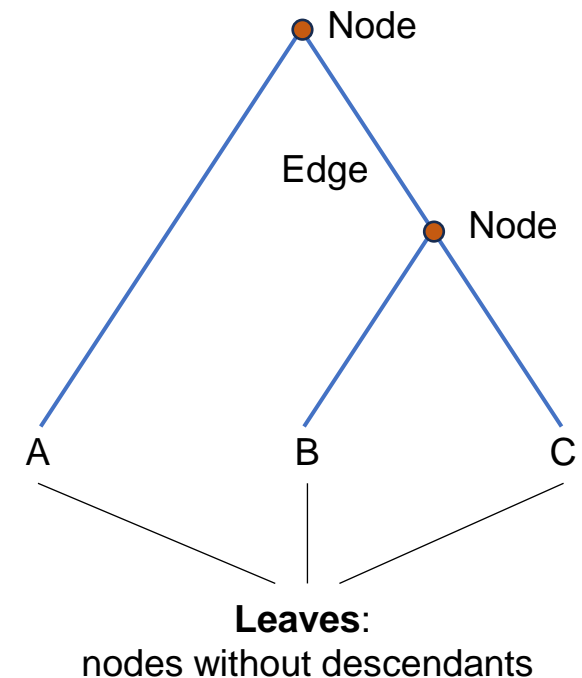
A tree is a graph having edges and nodes and can be represented by commas and parenthesis (**Newick format**)



Summarize a phylogenetic trees:

- the mcmc has run for n iterations sampling every m generation
- It produced tree files containing n/m trees

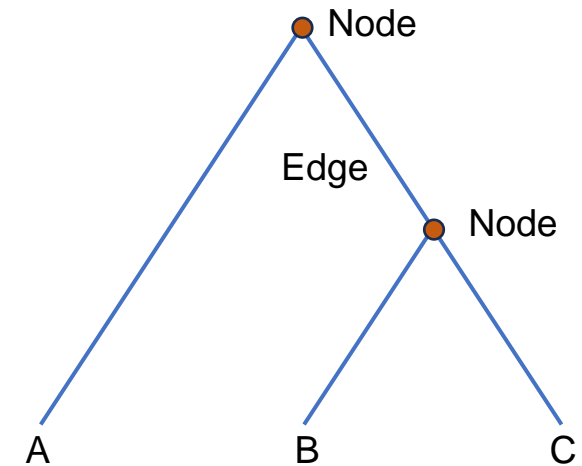
A tree is a graph having edges and nodes and can be represented by commas and parenthesis (**Newick format**)



Summarize a phylogenetic trees:

- the mcmc has run for n iterations sampling every m generation
- It produced tree files containing n/m trees

A tree is a graph having edges and nodes and can be represented by commas and parenthesis (**Newick format**)



(A,(B,C));

(A:2,(B:1,C:1):1);

„:....“ information on branch length

Summarize a phylogenetic trees:

- The mcmc has run for n iterations sampling every m generation
- It produced tree files containing n/m trees
- We can summarize the information contained in all these trees in a single tree using TreeAnnotator

TreeAnnotator v2.7.5, 2002-2023
MCMC Output analysis
by
Andrew Rambaut and Alexei J. Drummond

Institute of Evolutionary Biology
University of Edinburgh
a.rambaut@ed.ac.uk

Department of Computer Science
University of Auckland
alexei@cs.auckland.ac.nz

TreeAnnotator 2.7.5

Burn in percentage:

Posterior probability limit:

Target tree type:

Node heights:

Target tree file:

Input Tree File:

Output File:

Low memory: ☐

usually fine. You can select other values based on Tracer

Not exactly a *consensus* tree...
select "mean height"

input tree(s) file
output tree file

- Look at the tree file
- Open it in FigTree ...

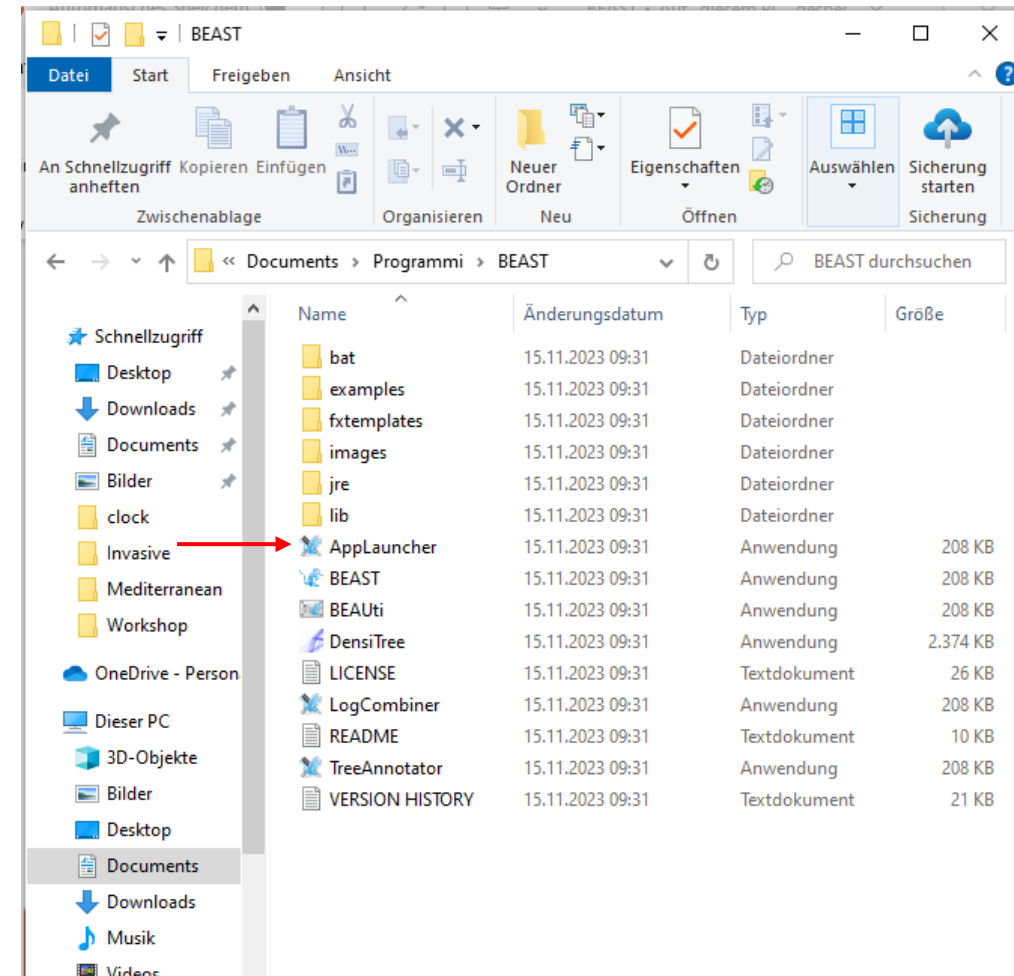


Post Processing (Species Delimitation!)

- We can use the ClusterTreeSetAnalyser provided in BEAST
- Or using speciesDA.jar (Species Delimitation Analyser; Jones et al., 2015)

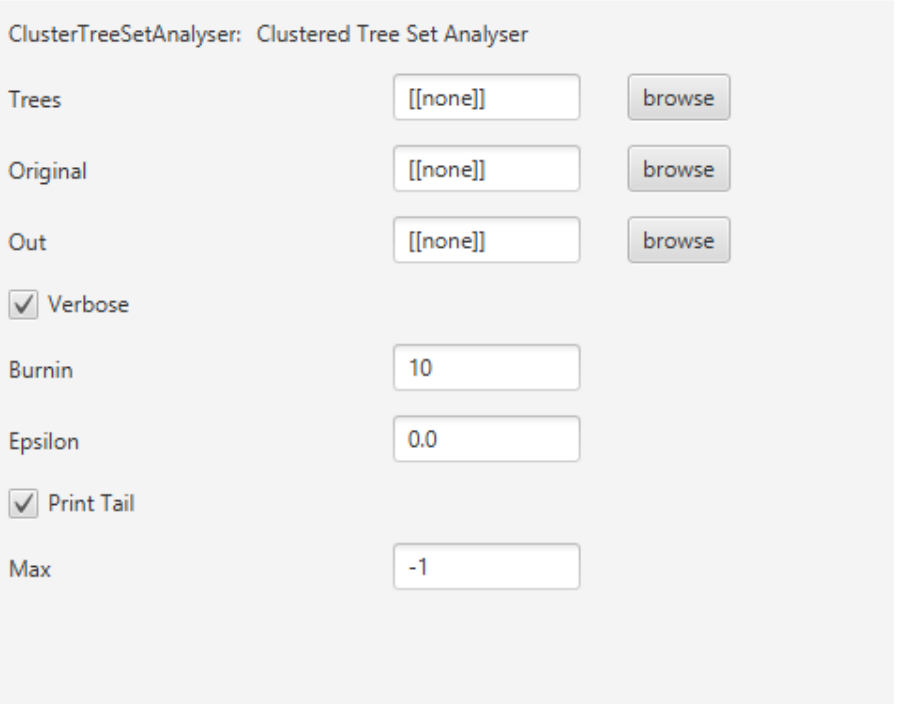
Post Processing (Species Delimitation!)

- We can use the ClusterTreeSetAnalyser provided in BEAST
 - Open the BEAST AppLauncher



Post Processing (Species Delimitation!)

- We can use the ClusterTreeSetAnalyser provided in BEAST
 - Open the BEAST AppLauncher
 - Launch the ClusterTreeSetAnalyser
 - Chose input and output
 - use the same Epsilon as in the analyses (usually $1e-4$)
 - Run the analyses



The screenshot shows the 'ClusterTreeSetAnalyser: Clustered Tree Set Analyser' window. It contains several input fields and checkboxes. The 'Trees' field is set to '[[none]]' with a 'browse' button next to it. The 'Original' field is also set to '[[none]]' with a 'browse' button. The 'Out' field is set to '[[none]]' with a 'browse' button. There are two checked checkboxes: 'Verbose' and 'Print Tail'. The 'Burnin' field is set to '10'. The 'Epsilon' field is set to '0.0'. The 'Max' field is set to '-1'.

Parameter	Value	Action
Trees	[[none]]	browse
Original	[[none]]	browse
Out	[[none]]	browse
Verbose	<input checked="" type="checkbox"/>	
Burnin	10	
Epsilon	0.0	
Print Tail	<input checked="" type="checkbox"/>	
Max	-1	

Check the results:

- How many species in the results with the higher posterior?
- What's the posterior probability of the clusters (species)?
- What's the best tree topology?
- Represent it graphically ...

Post Processing (Species Delimitation!)

- We can use the ClusterTreeSetAnalyser provided in BEAST
- Or using speciesDA.jar (Species Delimitation Analyser; Jones et al., 2015)
 - Copy the tree file (containing the tree samples during the mcmc) in the speciesDA folder
 - open the Windows command prompt
 - Check the options `java -jar speciesDA.jar -help`
 - [-collapseheight] = Epsilon

Post Processing (Species Delimitation!)

- We can use the ClusterTreeSetAnalyser provided in BEAST
- Or using speciesDA.jar (Species Delimitation Analyser; Jones et al., 2015)
 - Copy the tree file (containing the tree samples during the mcmc) in the speciesDA folder
 - open the Windows command prompt
 - Check the options `java -jar speciesDA.jar -help`
 - [-collapseheight] = Epsilon

```
java -jar speciesDA.jar -burnin 0.1 -collapseheight 0.0001 <input> <output>
```

Check the output:

- compare it with the one from ClusterTreeSetAnalyser ...
- Represent it graphically ...

Build a Similarity Matrix

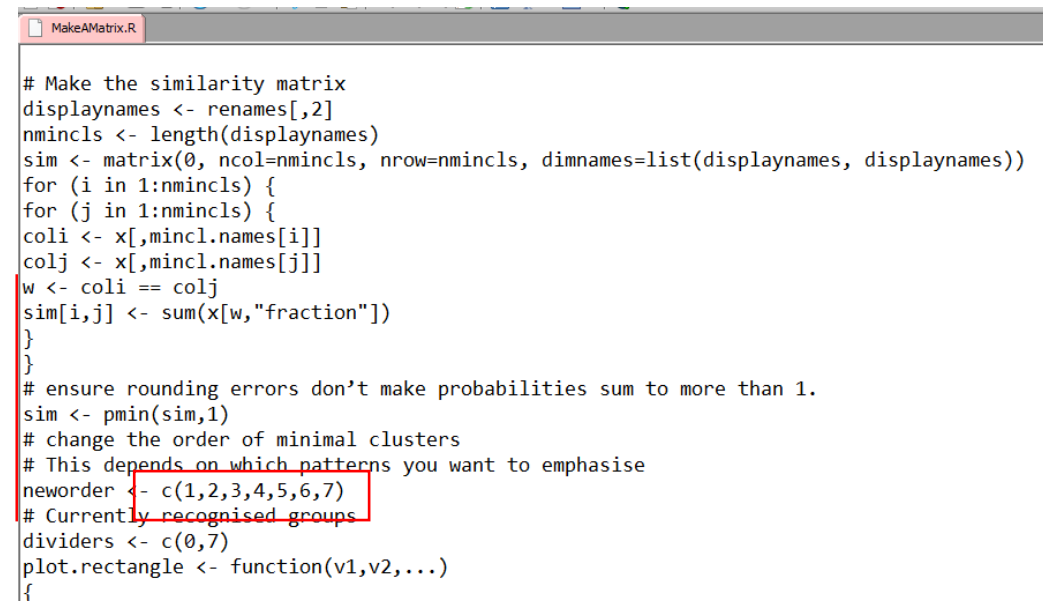
- Open the R script “MakeAMatrix.R”
- Change the marked parts in the script
 - work directory
 - input file names (the output of speciesDA.jar)
 - names of the samples you want to have in the matrix

```
MakeAMatrix.R
### R code for displaying the similarity matrix ###

# Read in the table of clusterings
workdir <- "C:\\Users\\Dell\\Desktop\\ProvaWork\\clock\\speciesDA_provaWork"
x <- read.table(paste(workdir, "output.txt", sep=""), header=TRUE)
# Abbreviations for display
renames <- matrix(c(
  "X8", "X8",
  "X19", "X19",
  "X20", "X20",
  "X12", "X12",
  "X33", "X33",
  "X13", "X13",
  "X29", "X29"),
  nrow=7, ncol=2, byrow=TRUE)
# Minimal cluster names are column names omitting first 4
```

Build a Similarity Matrix

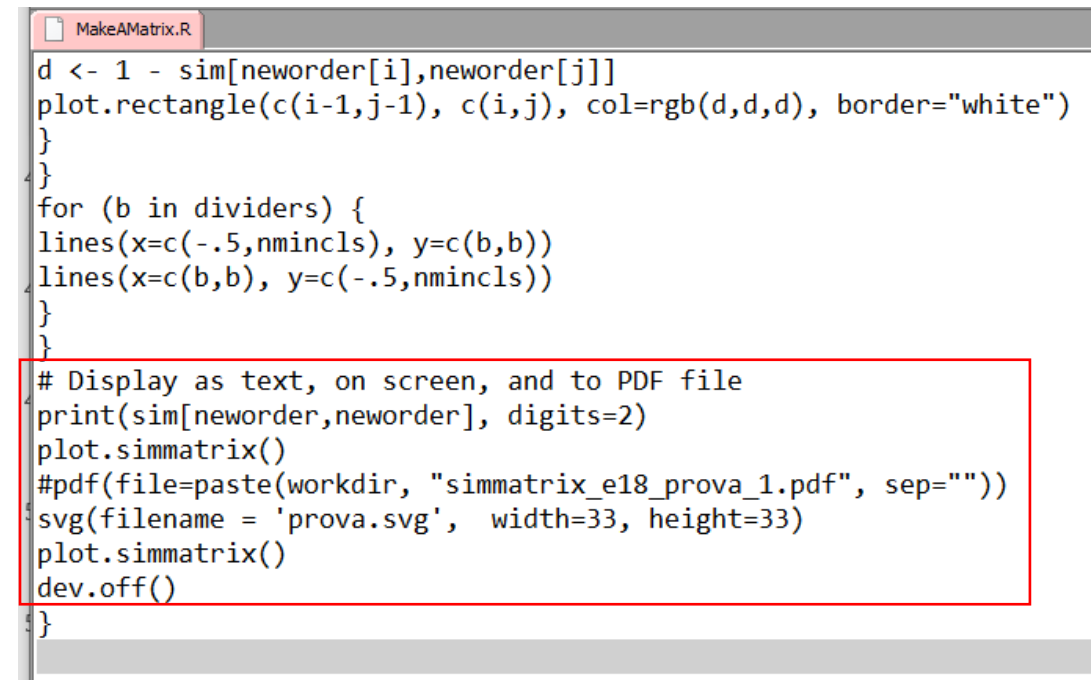
- Open the R script “MakeAMatrix.R”
- Change the marked parts in the script
 - work directory
 - input file names (the output of speciesDA.jar)
 - names of the samples you want to have in the matrix
- Change the order in which samples are shown (if you want...)



```
# Make the similarity matrix
displaynames <- renames[,2]
nmincls <- length(displaynames)
sim <- matrix(0, ncol=nmincls, nrow=nmincls, dimnames=list(displaynames, displaynames))
for (i in 1:nmincls) {
  for (j in 1:nmincls) {
    coli <- x[,mincl.names[i]]
    colj <- x[,mincl.names[j]]
    w <- coli == colj
    sim[i,j] <- sum(x[w,"fraction"])
  }
}
# ensure rounding errors don't make probabilities sum to more than 1.
sim <- pmin(sim,1)
# change the order of minimal clusters
# This depends on which patterns you want to emphasise
neworder <- c(1,2,3,4,5,6,7)
# Currently recognised groups
dividers <- c(0,7)
plot.rectangle <- function(v1,v2,...)
{
```

Build a Similarity Matrix

- Open the R script “MakeAMatrix.R”
- Change the marked parts in the script
 - work directory
 - input file names (the output of speciesDA.jar)
 - names of the samples you want to have in the matrix
- Change the order in which samples are shown (if you want...)
- Write in a file



```
MakeAMatrix.R
d <- 1 - sim[neworder[i],neworder[j]]
plot.rectangle(c(i-1,j-1), c(i,j), col=rgb(d,d,d), border="white")
}
}
for (b in dividers) {
  lines(x=c(-.5,nmincls), y=c(b,b))
  lines(x=c(b,b), y=c(-.5,nmincls))
}
}

# Display as text, on screen, and to PDF file
print(sim[neworder,neworder], digits=2)
plot.simmatrix()
#pdf(file=paste(workdir, "simmatrix_e18_prova_1.pdf", sep=""))
svg(filename = 'prova.svg', width=33, height=33)
plot.simmatrix()
dev.off()
}
```

Build a Similarity Matrix

- Run the script

```
> "PathToRscriptbin\Rscript.exe" PathToMakeAMatrix.R\MakeAMatrix.R
```