Creating Phylogenetic Trees with BEAST with Sequences Obtained from biomaRt in R

Roman Meneghini

Last Updated December 16, 2023

Video Tutorial: https://youtu.be/wH6m6RlYbqk

Phylogenetic trees are akin to the table of contents to the book of life. They can describe how different organisms connect to each other and when they may have split off from sister taxa. That said, when looking at organisms in nature, discerning what is and is not the same species can be difficult whether they are multiple species that appear to be one species, or one species that appears to be multiple (Bickford et al., 2007; Thompson et al., 2022). Using DNA from said organisms can make this task much easier as we can see on the genetic level how similar two individuals may be. Beyond that, the evolutionary relationships between them can be understood. Using tools like Bayesian Evolutionary Analysis Sampling Trees (BEAST) you can determine not only which organisms shared the most recent common ancestor but can also get an estimate on how long ago they shared that common ancestor.

This manual will demonstrate how to download BEAST and associated software, how to download and extract multiple sequences in R to export to a single FASTA file, how to align sequences using resources provided by EMBL-EBI, and finally how to use BEAST to produce phylogenetic trees using aligned sequence data and molecular clock calculations. This manual is aimed towards Windows users, so if using a different operating system, you may need to do some OS specific steps outside the manual. To demonstrate the effectiveness of BEAST, the tutorial will by default have you collect sequences from the reference genomes of several mammals. The tree produced by BEAST will then be compared to peer reviewed studies documenting the phylogenetic relationships of these animals to display the effectiveness of BEAST. Beyond BEAST using numerous iterations of possible evolutionary relationships to predict the most likely relationships, it is free software that has ongoing development support and troubleshooting available.

Video Link: https://youtu.be/wH6m6RlYbqk

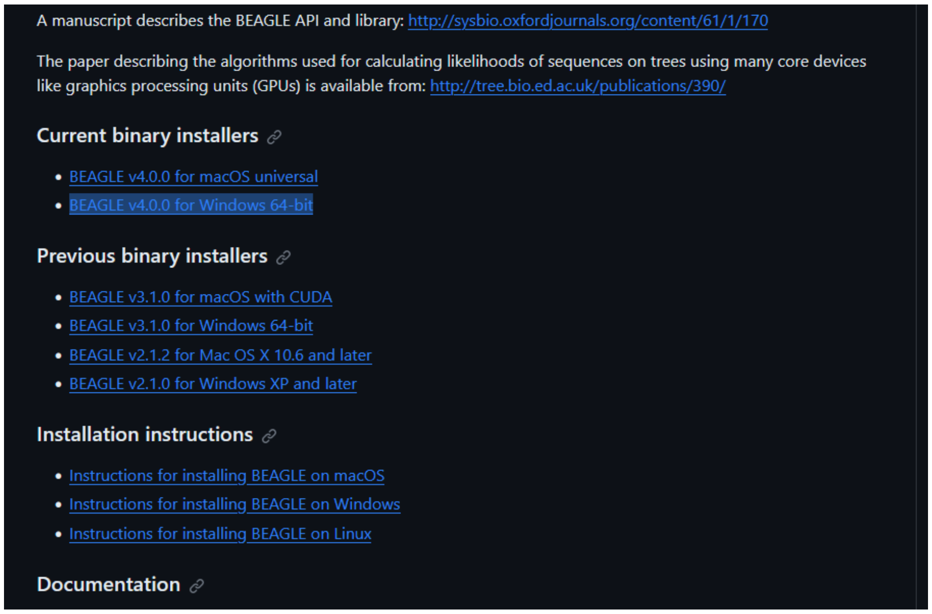
Tools and Resources Used

- BEAST https://beast.community/#
    - Dependencies
        - BEAGLE https://beast.community/beagle
        - Fig Tree http://tree.bio.ed.ac.uk/software/figtree/ https://github.com/rambaut/figtree/releases
            - Technically a soft dependency as it is only required to view tree
        - Java Development Kit and Java Runtime Environment JDK and JRE
            - https://www.oracle.com/java/technologies/downloads/ JDK
            - https://www.java.com/en/ JRE
- EMBL-EBI multiple sequence alignment services: https://www.ebi.ac.uk/jdispatcher/msa
    - Requires single FASTA file

- RStudio https://www.r-project.org/
  - Rscript used https://github.com/Bearypterid/BEAST-Manual
- Basic Text Editor

Setting up

1. Open the link to BEAST provided above
   a. Click on Install BEAST on Windows
   b. Download and extract BEAST in a directory you are comfortable working from
2. Open the links for Java provided above
   a. Download and install the latest Java Development Kit (JDK) using the Windows installer
   b. Download and install the Java Runtime Environment (JRE)
3. Open the link to BEAGLE
   a. Click the github link under downloading and installing BEAGLE
   b. Scroll all the way down and download the Binary installer for BEAGLE for Windows (see screenshot below).



A manuscript describes the BEAGLE API and library: http://sysbio.oxfordjournals.org/content/61/1/170

The paper describing the algorithms used for calculating likelihoods of sequences on trees using many core devices like graphics processing units (GPUs) is available from: http://tree.bio.ed.ac.uk/publications/390/

Current binary installers

- BEAGLE v4.0.0 for macOS universal
- BEAGLE v4.0.0 for Windows 64-bit

Previous binary installers

- BEAGLE v3.1.0 for macOS with CUDA
- BEAGLE v3.1.0 for Windows 64-bit
- BEAGLE v2.1.2 for Mac OS X 10.6 and later
- BEAGLE v2.1.0 for Windows XP and later

Installation instructions

- Instructions for installing BEAGLE on macOS
- Instructions for installing BEAGLE on Windows
- Instructions for installing BEAGLE on Linux

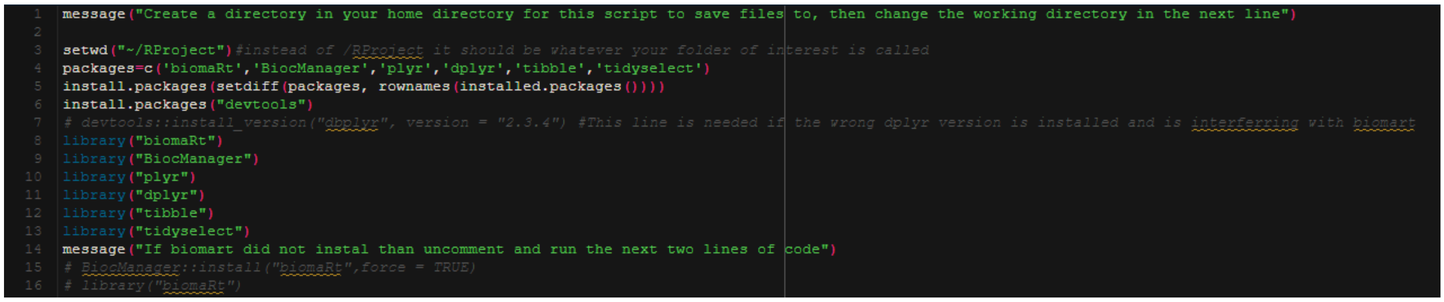
Documentation

   c. Use the installer
4. If you do not have RStudio installed, please download R and Rstudio using the link above
   a. When RStudio is installed, download the RScript linked above and place it in a new folder in your home directory. Make note of what the folder is called.
5. Download and extract FigTree https://github.com/rambaut/figtree/releases

Walkthrough

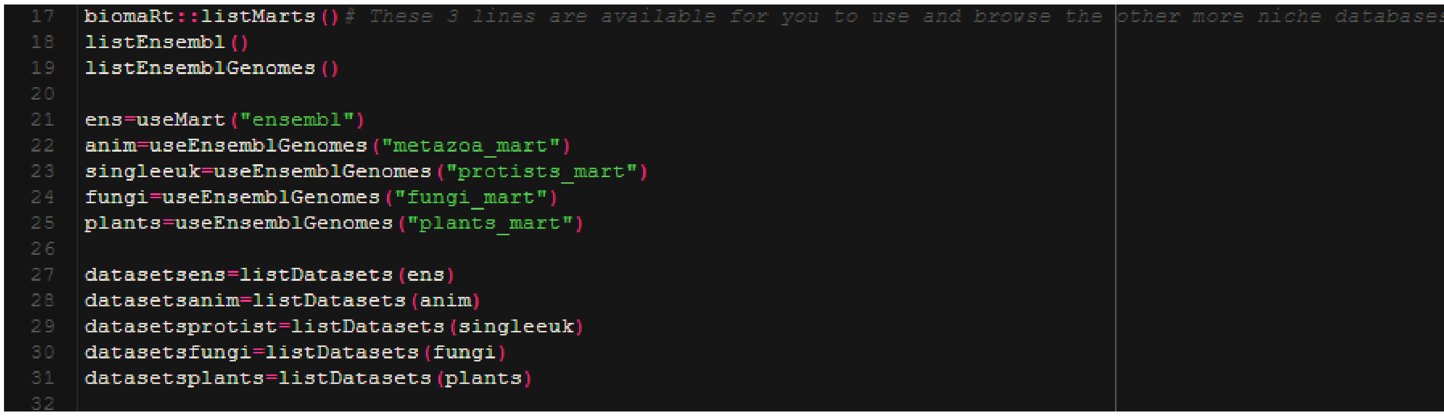
Getting organism specific DNA sequences in RStudio

1. Open up the Rscript and run it. It will prompt you as needed with messages



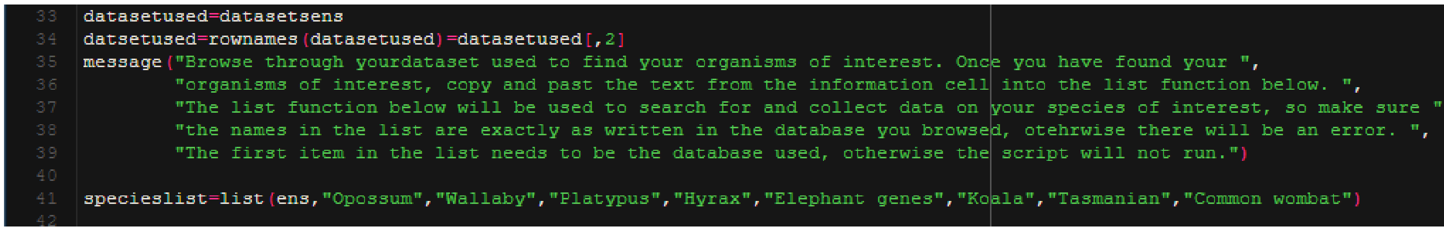
```r
message("Create a directory in your home directory for this script to save files to, then change the working directory in the next line")

setwd("~/RProject")#instead of /RProject it should be whatever your folder of interest is called
packages=c('biomaRt','BiocManager','plyr','dplyr','tibble','tidyselect')
install.packages(setdiff(packages, rownames(installed.packages())))
install.packages("devtools")
# devtools::install_version("dbplyr", version = "2.3.4") #This line is needed if the wrong dplyr version is installed and is interferring with biomart
library("biomaRt")
library("BiocManager")
library("plyr")
library("dplyr")
library("tibble")
library("tidyselect")
message("If biomart did not instal than uncomment and run the next two lines of code")
# BiocManager::install("biomaRt",force = TRUE)
# library("biomaRt")
```

2. These opening lines will install and load the packages needed.
3. The next set of lines allows you to browse the niche databases hosted by Ensembl as well as the more common and comprehensive ones.
    a. The default *datasetens* variable contains the database for vertebrates as well as *D. melanogaster* and *C. elegans*, while *datasetsanim* contains the database for all the remaining animals.



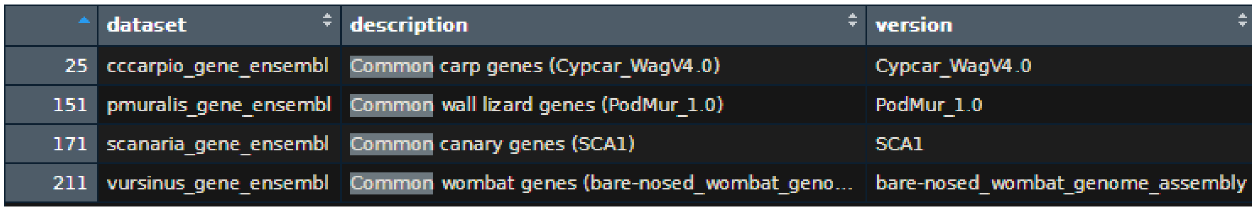
```r
biomaRt::listMarts()# These 3 lines are available for you to use and browse the other more niche databases
listEnsembl()
listEnsemblGenomes()

ens=useMart("ensembl")
anim=useEnsemblGenomes("metazoa_mart")
singleeuk=useEnsemblGenomes("protists_mart")
fungi=useEnsemblGenomes("fungi_mart")
plants=useEnsemblGenomes("plants_mart")

datasetsens=listDatasets(ens)
datasetsanim=listDatasets(anim)
datasetsprotist=listDatasets(singleeuk)
datasetsfungi=listDatasets(fungi)
datasetsplants=listDatasets(plants)
```

4. Here you can select the database you would like to use. By default it will use the main Ensembl vertebrate database. Feel free to browse this database by clicking on it in your environment.



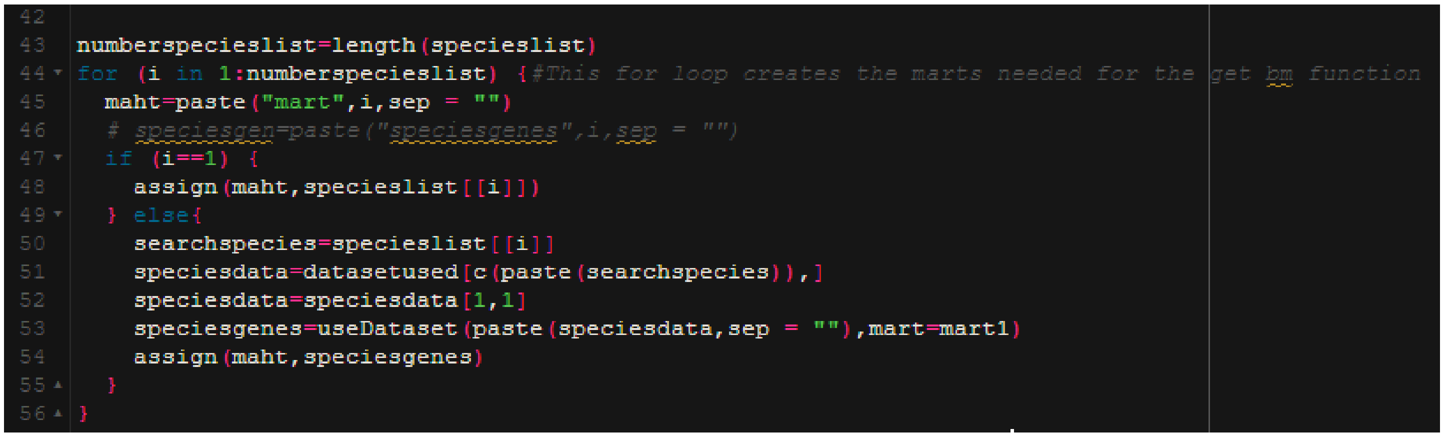
```r
datasetused=datasetsens
datsetused=rownames(datasetused)=datasetused[,2]
message("Browse through yourdataset used to find your organisms of interest. Once you have found your ",
        "organisms of interest, copy and past the text from the information cell into the list function below. ",
        "The list function below will be used to search for and collect data on your species of interest, so make sure "
        "the names in the list are exactly as written in the database you browsed, otehrwise there will be an error. ",
        "The first item in the list needs to be the database used, otherwise the script will not run.")

specieslist=list(ens,"Opossum","Wallaby","Platypus","Hyrax","Elephant genes","Koala","Tasmanian","Common wombat")
```

5. After browsing through the database, we will want to select our species of interest and make a list with them. The list needs to use the exact wording from the description column of the database for the keywords in the list to work correctly, however if you choose not to touch the list, the script will continue with the default animals.
    a. Notice that "common" is found in the names of multiple animals in the database, so we need "Common wombat". By contrast, Tasmanian is enough to get the Tasmanian Devil as that animal is the only one in the database at present with that adjective.
    b. Also note, the first word is always uppercase in the database, make sure your list uses the correct case as the code is case sensitive.



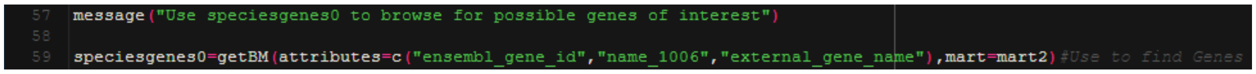
	dataset	description	version
25	cccarpio_gene_ensembl	Common carp genes (Cypcar_WagV4.0)	Cypcar_WagV4.0
151	pmuralis_gene_ensembl	Common wall lizard genes (PodMur_1.0)	PodMur_1.0
171	scanaria_gene_ensembl	Common canary genes (SCA1)	SCA1
211	vursinus_gene_ensembl	Common wombat genes (bare-nosed_wombat_geno…	bare-nosed_wombat_genome_assembly

6. Run the next set of code, it requires no input. It simply saves the information needed to use the specific datasets for each organism into a variable for each.

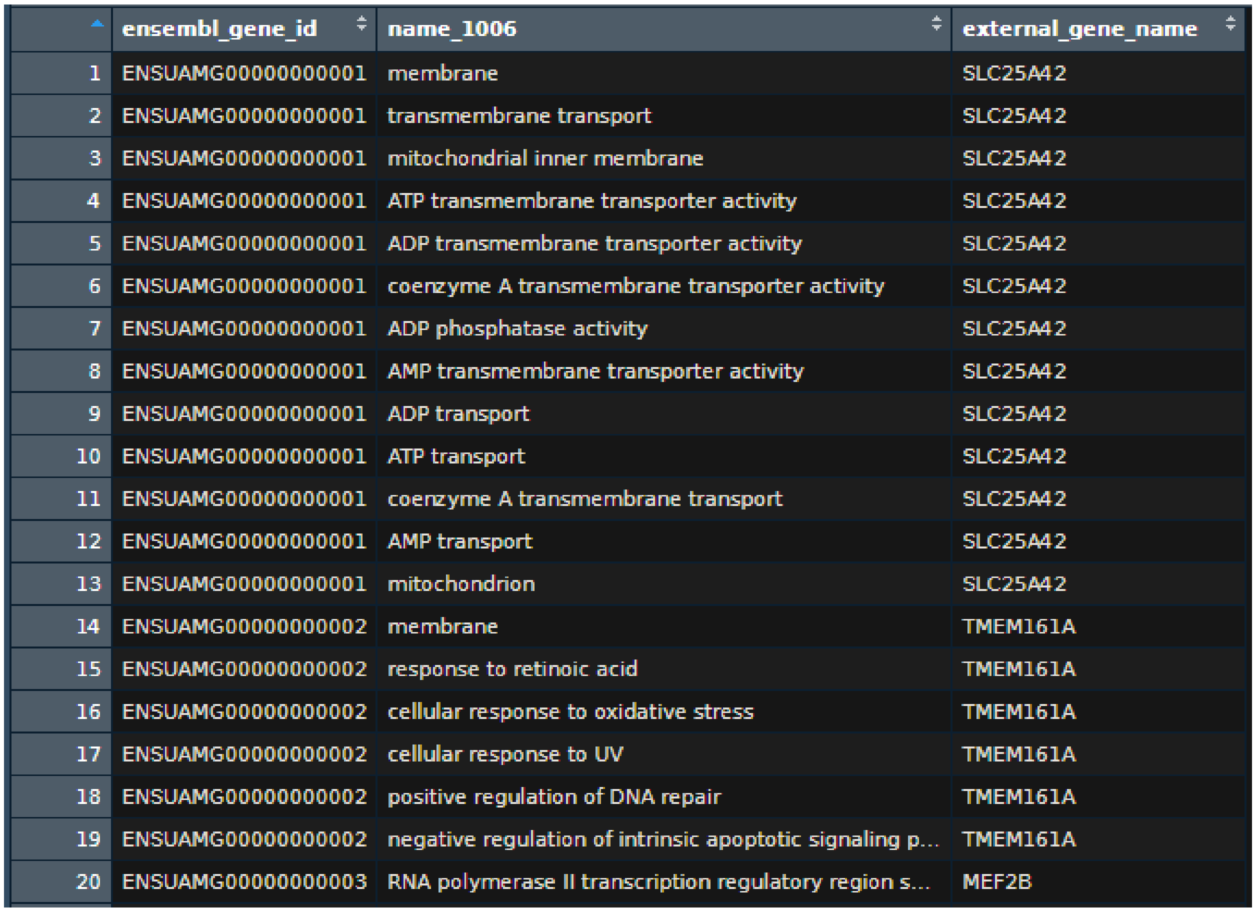


```r

numberspecieslist=length(specieslist)
for (i in 1:numberspecieslist) {#This for loop creates the marts needed for the get bm function
  maht=paste("mart",i,sep = "")
  # speciesgen=paste("speciesgenes",i,sep = "")
  if (i==1) {
    assign(maht,specieslist[[i]])
  } else{
    searchspecies=specieslist[[i]]
    speciesdata=datasetused[c(paste(searchspecies)),]
    speciesdata=speciesdata[1,1]
    speciesgenes=useDataset(paste(speciesdata,sep = ""),mart=mart1)
    assign(maht,speciesgenes)
  }
}
```

7. The next set of code saves the whole dataset for our first organism into a dataframe you can view by clicking on it in the environment. It will look similar to the dataframe below. The first column will contain the ensembl gene id, the second will contain the go enrichment term attached to the gene, and the last column will provide the gene name. The gene name is a term that will be shared between species. Feel free to browse around looking at what genes are named, and what are not.

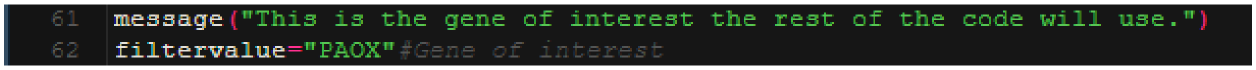


```r
message("Use speciesgenes0 to browse for possible genes of interest")

speciesgenes0=getBM(attributes=c("ensembl_gene_id","name_1006","external_gene_name"),mart=mart2)#Use to find Genes
```



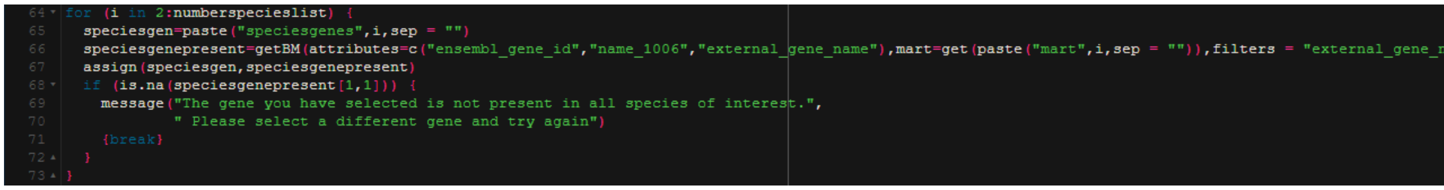
	ensembl_gene_id	name_1006	external_gene_name
1	ENSUAMG00000000001	membrane	SLC25A42
2	ENSUAMG00000000001	transmembrane transport	SLC25A42
3	ENSUAMG00000000001	mitochondrial inner membrane	SLC25A42
4	ENSUAMG00000000001	ATP transmembrane transporter activity	SLC25A42
5	ENSUAMG00000000001	ADP transmembrane transporter activity	SLC25A42
6	ENSUAMG00000000001	coenzyme A transmembrane transporter activity	SLC25A42
7	ENSUAMG00000000001	ADP phosphatase activity	SLC25A42
8	ENSUAMG00000000001	AMP transmembrane transporter activity	SLC25A42
9	ENSUAMG00000000001	ADP transport	SLC25A42
10	ENSUAMG00000000001	ATP transport	SLC25A42
11	ENSUAMG00000000001	coenzyme A transmembrane transport	SLC25A42
12	ENSUAMG00000000001	AMP transport	SLC25A42
13	ENSUAMG00000000001	mitochondrion	SLC25A42
14	ENSUAMG00000000002	membrane	TMEM161A
15	ENSUAMG00000000002	response to retinoic acid	TMEM161A
16	ENSUAMG00000000002	cellular response to oxidative stress	TMEM161A
17	ENSUAMG00000000002	cellular response to UV	TMEM161A
18	ENSUAMG00000000002	positive regulation of DNA repair	TMEM161A
19	ENSUAMG00000000002	negative regulation of intrinsic apoptotic signaling p...	TMEM161A
20	ENSUAMG00000000003	RNA polymerase II transcription regulatory region s...	MEF2B

8. The next two lines relate to our gene of interest. The variable should equal a gene name found in the above dataframe. The default gene used is PAOX as it was documented in all the organisms of interest, but you can and should try other genes as the script will check if all reference genomes used have it.



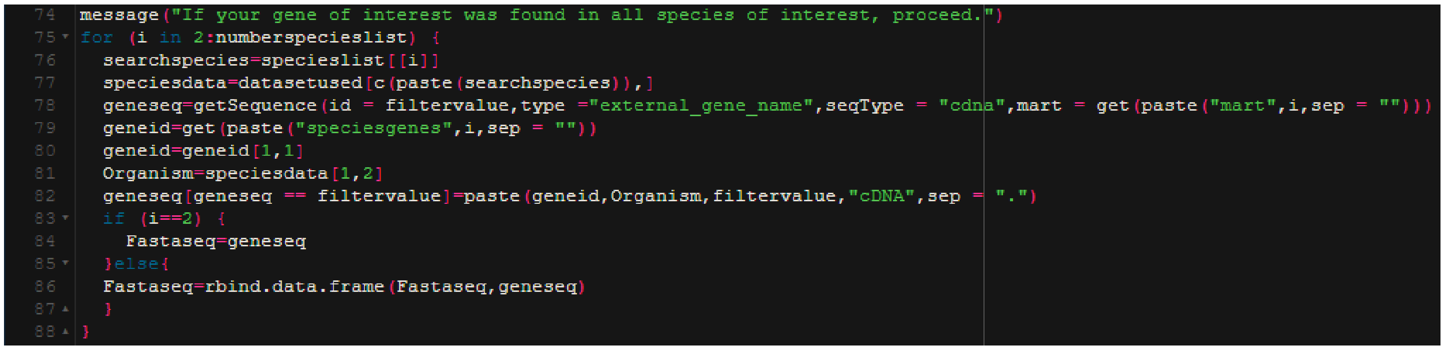
```
message("This is the gene of interest the rest of the code will use.")
filtervalue="PAOX"#Gene of interest
```

9. The next for loop checks for if our gene of interest is in each of our organisms of interest and then saves the database for that organism if it is. If the gene is not present in any of the organisms, the loop will end and you will be given a message prompting you to try a different gene.
    a. Note, this for loop will take a bit to run as scans through 100,000 of thousands of rows for each organism to determine whether or not the gene of interest is there.



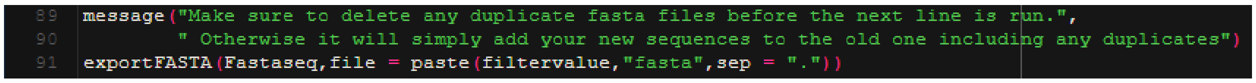
```
for (i in 2:numberspecieslist) {
  speciesgen=paste("speciesgenes",i,sep = "")
  speciesgenepresent=getBM(attributes=c("ensembl_gene_id","name_1006","external_gene_name"),mart=get(paste("mart",i,sep = "")),filters = "external_gene_
  assign(speciesgen,speciesgenepresent)
  if (is.na(speciesgenepresent[1,1])) {
    message("The gene you have selected is not present in all species of interest.",
            " Please select a different gene and try again")
    {break}
  }
}
```

10. If you were not given a message prompting you to use a different gene of interest, simply proceed with the code. The next for loop extracts the DNA sequences for the gene of interest for each organisms and binds them into one FASTA file.



```r
message("If your gene of interest was found in all species of interest, proceed.")
for (i in 2:numberspecieslist) {
  searchspecies=specieslist[[i]]
  speciesdata=datasetused[c(paste(searchspecies)),]
  geneseq=getSequence(id = filtervalue,type ="external_gene_name",seqType = "cdna",mart = get(paste("mart",i,sep = "")))
  geneid=get(paste("speciesgenes",i,sep = ""))
  geneid=geneid[1,1]
  Organism=speciesdata[1,2]
  geneseq[geneseq == filtervalue]=paste(geneid,Organism,filtervalue,"cDNA",sep = ".")
  if (i==2) {
    Fastaseq=geneseq
  }else{
  Fastaseq=rbind.data.frame(Fastaseq,geneseq)
  }
}
```

11. The last set of code will remind you to delete any files with the existing name of our FASTA file before exporting the file created by the script. The single file produced is needed for the alignment tools used further on.

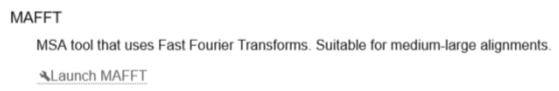


```r
message("Make sure to delete any duplicate fasta files before the next line is run.",
        " Otherwise it will simply add your new sequences to the old one including any duplicates")
exportFASTA(Fastaseq,file = paste(filtervalue,"fasta",sep = "."))
```

12. This script requires minimal input. The majority of work required on your end is to set up the packages, choose the organisms and gene of interest, and then run it. It is designed to reduce the time and effort needed on your end when changing the genes and organisms used.

Alignment

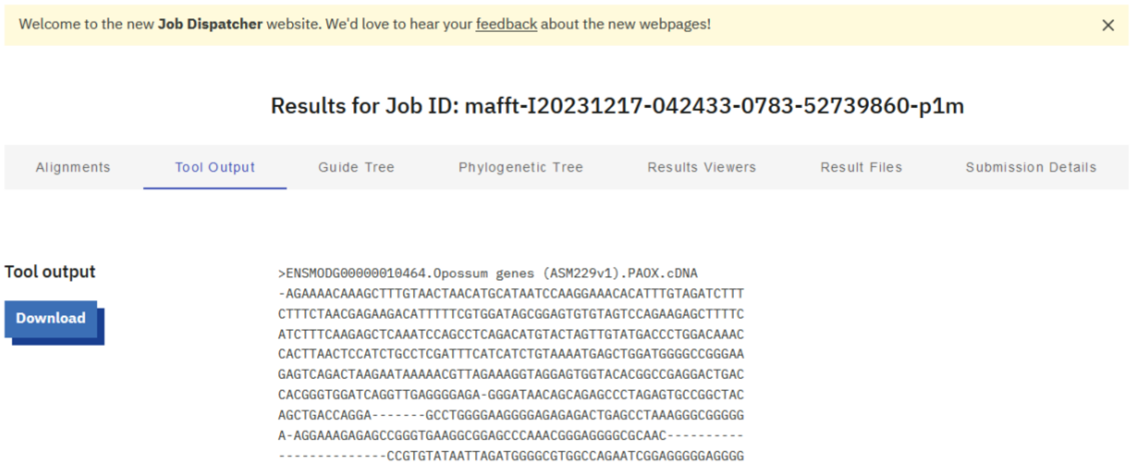
1. Navigate to EMBL-EBI multiple sequence alignment services using the link https://www.ebi.ac.uk/jdispatcher/msa
2. There are many tools available, but for this walkthrough, we will use MAFFT
    a. The tools vary in speed and the size of regions they can align. MAFFT is just suitable for the default data, if you are using more organisms or larger segments of DNA, consider one of the other MSA tools available on the site.



MAFFT

MSA tool that uses Fast Fourier Transforms. Suitable for medium-large alignments.

3. Change the setting from automatic to DNA, upload the FASTA file produced above and submit the job. It will only take a minute or two at most for this example, but if you are using hundreds of sequences, it will take longer.
4. After the job finishes, click tool output. It will produce an aligned FASTA file that looks like this.



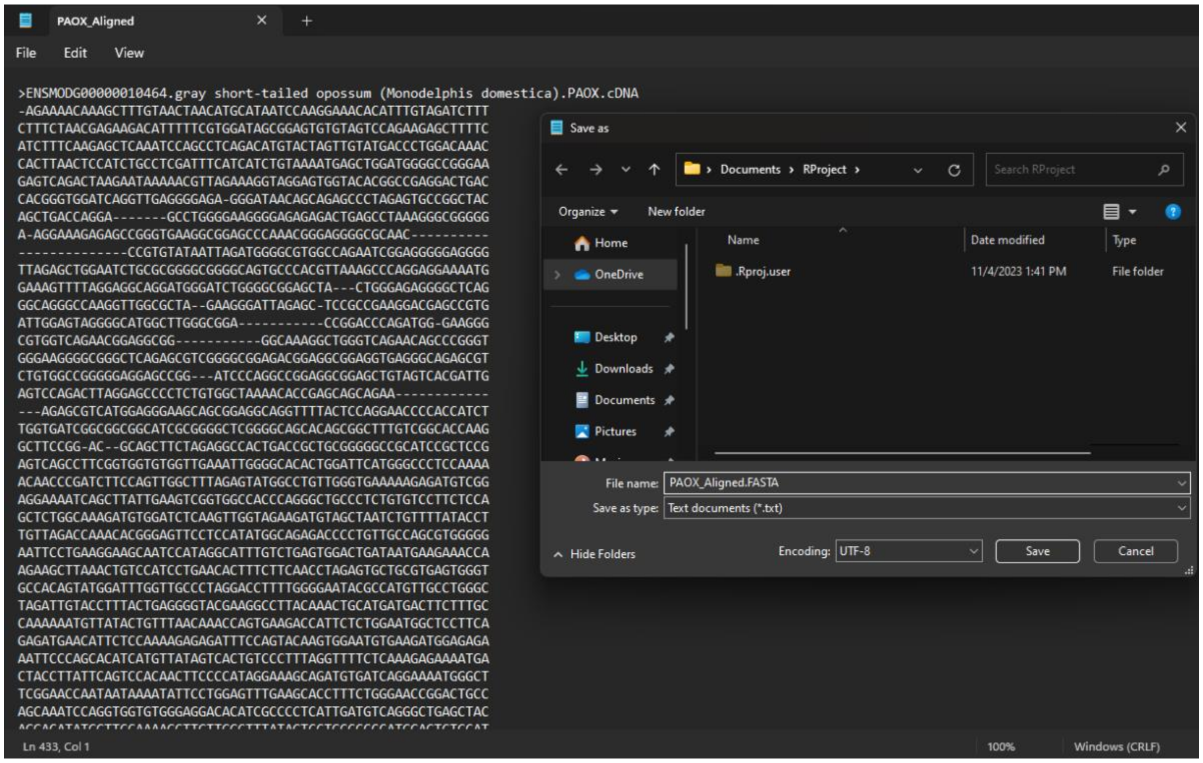
Welcome to the new **Job Dispatcher** website. We'd love to hear your feedback about the new webpages!

Results for Job ID: mafft-I20231217-042433-0783-52739860-p1m

Tool output

```
>ENSMODG00000010464.Opossum genes (ASM229v1).PAOX.cDNA
-AGAAAACAAAGCTTTGTAACTAACATGCATAATCCAAGGAAACACATTTGTAGATCTTT
CTTTCTAACGAGAAGACATTTTTCGTGGATAGCGGAGTGTGTAGTCCAGAAGAGCTTTTC
ATCTTTCAAGAGCTCAAATCCAGCCTCAGACATGTACTAGTTGTATGACCCTGGACAAAC
CACTTAACTCCATCTGCCTCGATTTCATCATCTGTAAAATGAGCTGGATGGGGCCGGGAA
GAGTCAGACTAAGAATAAAAACGTTAGAAAGGTAGGAGTGGTACACGGCCGAGGACTGAC
CACGGGTGGATCAGGTTGAGGGGAGA-GGGATAACAGCAGAGCCCTAGAGTGCCGGCTAC
AGCTGACCAGGA-------GCCTGGGGAAGGGGAGAGAGACTGAGCCTAAAGGGCGGGGG
A-AGGAAAGAGAGCCGGGTGAAGGCGGAGCCCAAACGGGAGGGGCGCAAC----------
--------------CCGTGTATAATTAGATGGGGCGTGGCCAGAATCGGAGGGGGAGGGG
```

5. It can also produce a simple phylogenetic tree, but this is typically not suitable for any serious analysis of relationships.
6. In the tool output tab, click download to download the aligned file. Open it in a program like notepad or notepad ++ and save it to either the work directory folder used by R or the folder BEAST is installed in. Save the file as your “Gene of interest_Aligned.FASTA”. This will save our alignment as a new aligned FASTA file to use with BEAST.



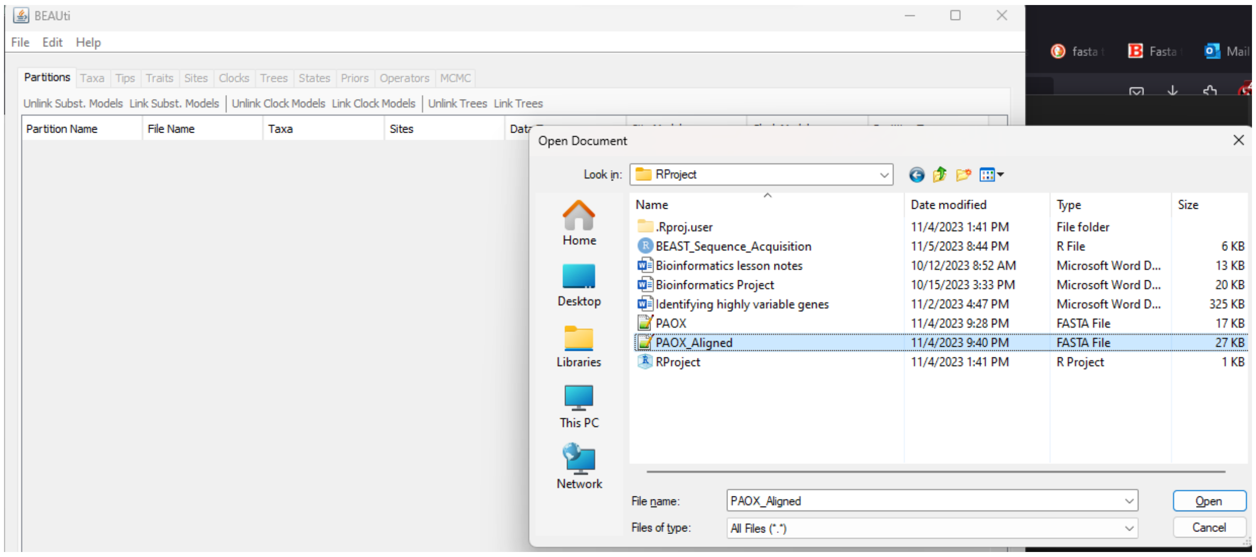
```
>ENSMODG00000010464.gray short-tailed opossum (Monodelphis domestica).PAOX.cDNA
-AGAAAACAAAGCTTTGTAACTAACATGCATAATCCAAGGAAACACATTTGTAGATCTTT
CTTTCTAACGAGAAGACATTTTTCGTGGATAGCGGAGTGTGTAGTCCAGAAGAGCTTTTC
ATCTTTCAAGAGCTCAAATCCAGCCTCAGACATGTACTAGTTGTATGACCCTGGACAAAC
CACTTAACTCCATCTGCCTCGATTTCATCATCTGTAAAATGAGCTGGATGGGGCCGGGAA
GAGTCAGACTAAGAATAAAAACGTTAGAAAGGTAGGAGTGGTACACGGCCGAGGACTGAC
CACGGGTGGATCAGGTTGAGGGGAGA-GGGATAACAGCAGAGCCCTAGAGTGCCGGCTAC
AGCTGACCAGGA-------GCCTGGGGAAGGGGAGAGAGACTGAGCCTAAAGGGCGGGGG
A-AGGAAAGAGAGCCGGGTGAAGGCGGAGCCCAAACGGGAGGGGCGCAAC----------
--------------CCGTGTATAATTAGATGGGGCGTGGCCAGAATCGGAGGGGGAGGGG
TTAGAGCTGGAATCTGCGCGGGGCGGGGCAGTGCCCACGTTAAAGCCCAGGAGGAAAATG
GAAAGTTTTAGGAGGCAGGATGGGATCTGGGGCGGAGCTA---CTGGGAGAGGGGCTCAG
GGCAGGGCCAAGGTTGGCGCTA--GAAGGGATTAGAGC-TCCGCCGAAGGACGAGCCGTG
ATTGGAGTAGGGGCATGGCTTGGGCGGA-----------CCGGACCCAGATGG-GAAGGG
CGTGGTCAGAACGGAGGCGG-----------GGCAAAGGCTGGGTCAGAACAGCCCGGGT
GGGAAGGGGCGGGCTCAGAGCGTCGGGGCGGAGACGGAGGCGGAGGTGAGGGCAGAGCGT
CTGTGGCCGGGGGAGGAGCCGG---ATCCCAGGCCGGAGGCGGAGCTGTAGTCACGATTG
AGTCCAGACTTAGGAGCCCCTCTGTGGCTAAAACACCGAGCAGCAGAA------------
---AGAGCGTCATGGAGGGAAGCAGCGGAGGCAGGTTTTACTCCAGGAACCCCACCATCT
TGGTGATCGGCGGCGGCATCGCGGGGCTCGGGGCAGCACAGCGGCTTTGTCGGCACCAAG
GCTTCCGG-AC--GCAGCTTCTAGAGGCCACTGACCGCTGCGGGGGCCGCATCCGCTCCG
AGTCAGCCTTCGGTGGTGTGGTTGAAATTGGGGCACACTGGATTCATGGGCCCTCCAAAA
ACAACCCGATCTTCCAGTTGGCTTTAGAGTATGGCCTGTTGGGTGAAAAAGAGATGTCGG
AGGAAAATCAGCTTATTGAAGTCGGTGGCCACCCAGGGCTGCCCTCTGTGTCCTTCTCCA
GCTCTGGCAAAGATGTGGATCTCAAGTTGGTAGAAGATGTAGCTAATCTGTTTTATACCT
TGTTAGACCAAACACGGGAGTTCCTCCATATGGCAGAGACCCCTGTTGCCAGCGTGGGGG
AATTCCTGAAGGAAGCAATCCATAGGCATTTGTCTGAGTGGACTGATAATGAAGAAACCA
AGAAGCTTAAACTGTCCATCCTGAACACTTTCTTCAACCTAGAGTGCTGCGTGAGTGGGT
GCCACAGTATGGATTTGGTTGCCCTAGGACCTTTTGGGGAATACGCCATGTTGCCTGGGC
TAGATTGTACCTTTACTGAGGGGTACGAAGGCCTTACAAACTGCATGATGACTTCTTTGC
CAAAAAATGTTATACTGTTTAACAAACCAGTGAAGACCATTCTCTGGAATGGCTCCTTCA
GAGATGAACATTCTCCAAAAGAGAGATTTCCAGTACAAGTGGAATGTGAAGATGGAGAGA
AATTCCCAGCACATCATGTTATAGTCACTGTCCCTTTAGGTTTTCTCAAAGAGAAAATGA
CTACCTTATTCAGTCCACAACTTCCCCATAGGAAAGCAGATGTGATCAGGAAAATGGGCT
TCGGAACCAATAATAAAATATTCCTGGAGTTTGAAGCACCTTTCTGGGAACCGGACTGCC
AGCAAATCCAGGTGGTGTGGGAGGACACATCGCCCCTCATTGATGTCAGGGCTGAGCTAC
```

File name: PAOX_Aligned.FASTA
Save as type: Text documents (*.txt)
Encoding: UTF-8

Ln 433, Col 1

BEAST

1. Navigate to the file where BEAST was extracted and launch BEAUTi
   a. BEAST requires a specific xml file to run, and you obtain that xml file using BEAUTi.


BEAUti
File Edit Help
Partitions Taxa Tips Traits Sites Clocks Trees States Priors Operators MCMC
Unlink Subst. Models Link Subst. Models Unlink Clock Models Link Clock Models Unlink Trees Link Trees
Partition Name File Name Taxa Sites
Open Document
Home
Desktop
Libraries
This PC
Network
Open
Cancel

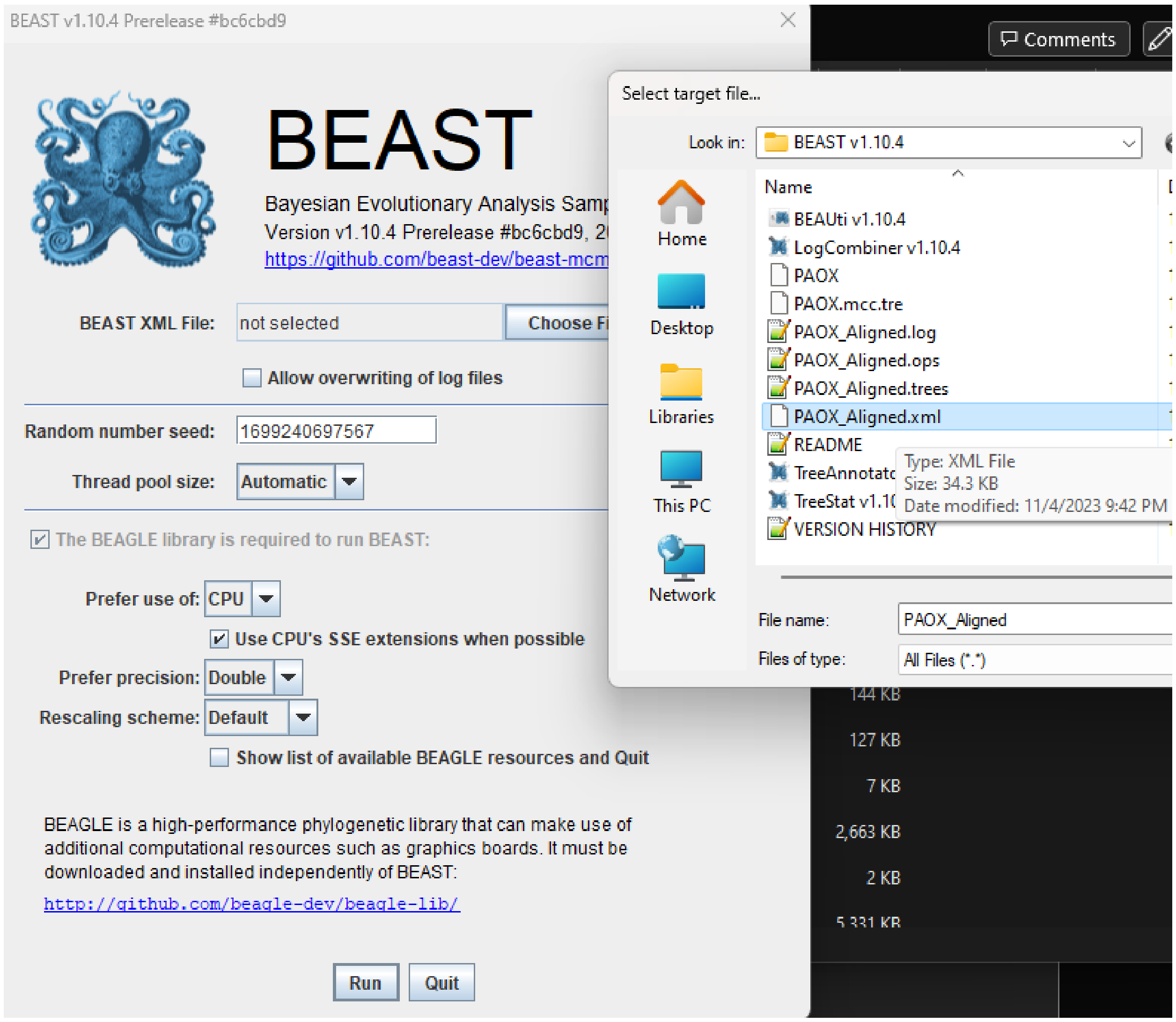
Look in: RProject

Name	Date modified	Type	Size
.Rproj.user	11/4/2023 1:41 PM	File folder	
BEAST_Sequence_Acquisition	11/5/2023 8:44 PM	R File	6 KB
Bioinformatics lesson notes	10/12/2023 8:52 AM	Microsoft Word D...	13 KB
Bioinformatics Project	10/15/2023 3:33 PM	Microsoft Word D...	20 KB
Identifying highly variable genes	11/2/2023 4:47 PM	Microsoft Word D...	325 KB
PAOX	11/4/2023 9:28 PM	FASTA File	17 KB
PAOX_Aligned	11/4/2023 9:40 PM	FASTA File	27 KB
RProject	11/4/2023 1:41 PM	R Project	1 KB

File name: PAOX_Aligned

Files of type: All Files (*.*)

2. Click file, then import data then select the aligned FASTA file made above and then click “generate BEAST file” once it has loaded the file. If you would like to alter the evolutionary model or the clock model used you may do so now, but we will stick with the default for now. Once you have generated the BEAST file, you may close BEAUTi
3. Launch BEAST and choose the newly generated XML file from part 2. Toggle “Allow overwriting of log files”, the default is off and can produce an error. Click run. This will take some time and can be intensive on your computer. As it performs many repeated calculations over and over to produce numerous sets of possible trees.


BEAST v1.10.4 Prerelease #bc6cbd9

BEAST

Bayesian Evolutionary Analysis Samp

Version v1.10.4 Prerelease #bc6cbd9, 2

https://github.com/beast-dev/beast-mcm

BEAST XML File: not selected

Allow overwriting of log files

Random number seed: 1699240697567

Thread pool size: Automatic

☑ The BEAGLE library is required to run BEAST:

Prefer use of: CPU

☑ Use CPU's SSE extensions when possible

Prefer precision: Double

Rescaling scheme: Default

☐ Show list of available BEAGLE resources and Quit

BEAGLE is a high-performance phylogenetic library that can make use of additional computational resources such as graphics boards. It must be downloaded and installed independently of BEAST:

http://github.com/beagle-dev/beagle-lib/

Select target file...

Look in: BEAST v1.10.4

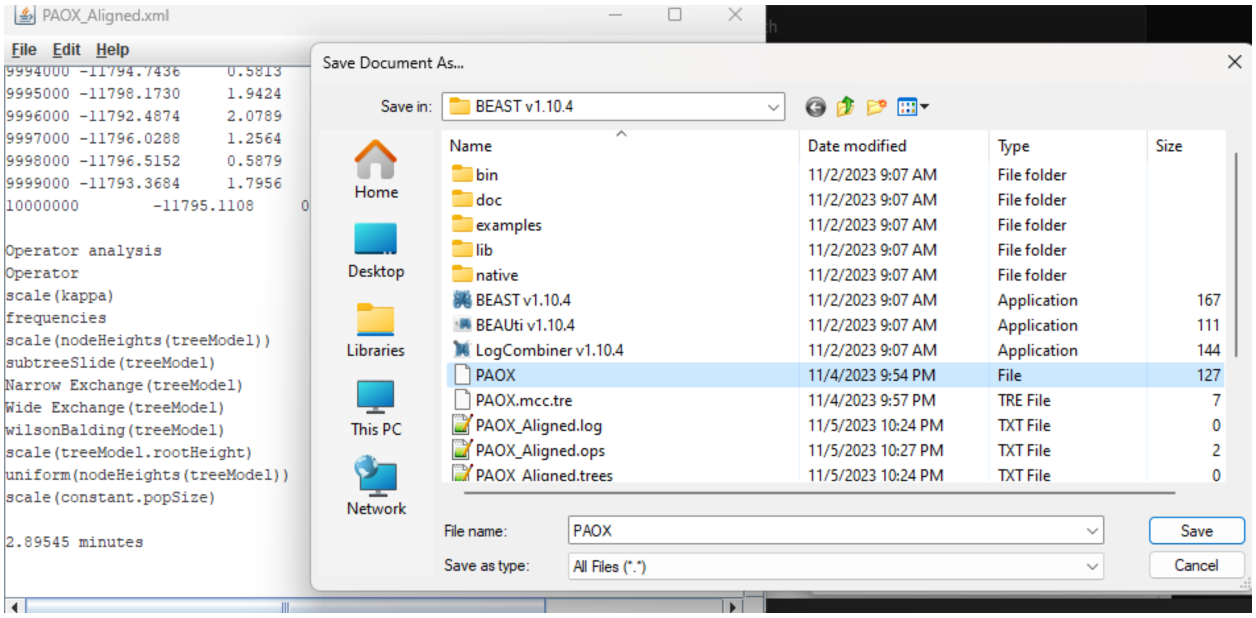
Name
BEAUti v1.10.4
LogCombiner v1.10.4
PAOX
PAOX.mcc.tre
PAOX_Aligned.log
PAOX_Aligned.ops
PAOX_Aligned.trees
PAOX_Aligned.xml
README
TreeAnnotato
TreeStat v1.10
VERSION HISTORY

Type: XML File
Size: 34.3 KB
Date modified: 11/4/2023 9:42 PM

File name: PAOX_Aligned

Files of type: All Files (*.*)

4. Save the File. I used the gene of interest as the name for this example



```
9994000 -11794.7436     0.5813
9995000 -11798.1730     1.9424
9996000 -11792.4874     2.0789
9997000 -11796.0288     1.2564
9998000 -11796.5152     0.5879
9999000 -11793.3684     1.7956
10000000        -11795.1108     0

Operator analysis
Operator
scale(kappa)
frequencies
scale(nodeHeights(treeModel))
subtreeSlide(treeModel)
Narrow Exchange(treeModel)
Wide Exchange(treeModel)
wilsonBalding(treeModel)
scale(treeModel.rootHeight)
uniform(nodeHeights(treeModel))
scale(constant.popSize)

2.89545 minutes
```

Save Document As...

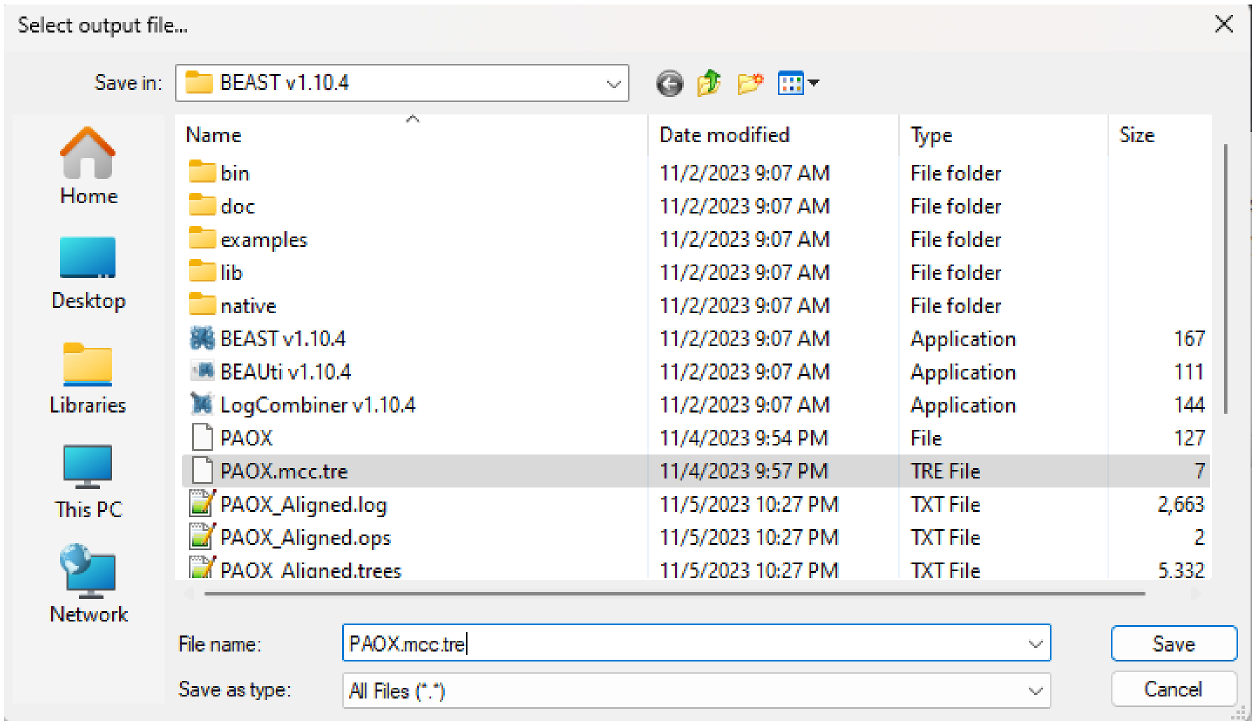
Save in: BEAST v1.10.4

Name	Date modified	Type	Size
bin	11/2/2023 9:07 AM	File folder	
doc	11/2/2023 9:07 AM	File folder	
examples	11/2/2023 9:07 AM	File folder	
lib	11/2/2023 9:07 AM	File folder	
native	11/2/2023 9:07 AM	File folder	
BEAST v1.10.4	11/2/2023 9:07 AM	Application	167
BEAUti v1.10.4	11/2/2023 9:07 AM	Application	111
LogCombiner v1.10.4	11/2/2023 9:07 AM	Application	144
PAOX	11/4/2023 9:54 PM	File	127
PAOX.mcc.tre	11/4/2023 9:57 PM	TRE File	7
PAOX_Aligned.log	11/5/2023 10:24 PM	TXT File	0
PAOX_Aligned.ops	11/5/2023 10:27 PM	TXT File	2
PAOX_Aligned.trees	11/5/2023 10:24 PM	TXT File	0

File name: PAOX

Save as type: All Files (*.*)

5. Close BEAST and open TreeAnnotator. You will find it in the same folder as BEAUTi and BEAST. Set the “Burnin (as states) to 100000 as this was the default used for BEAST. Load the file produced by BEAST as the input tree file. For the output file, click choose file, then type in the new window the name you want for the file, with the file extension of .mcc.tre as seen in the example below


Select output file...

Save in: BEAST v1.10.4

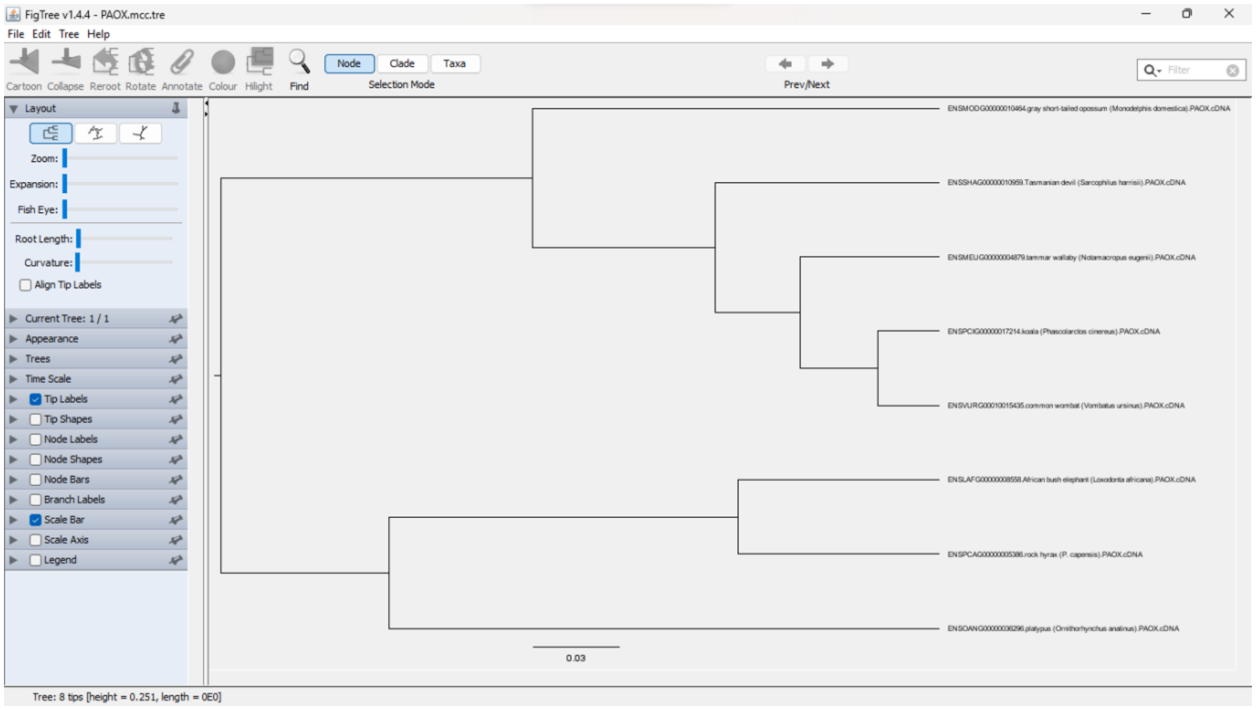
Name	Date modified	Type	Size
bin	11/2/2023 9:07 AM	File folder	
doc	11/2/2023 9:07 AM	File folder	
examples	11/2/2023 9:07 AM	File folder	
lib	11/2/2023 9:07 AM	File folder	
native	11/2/2023 9:07 AM	File folder	
BEAST v1.10.4	11/2/2023 9:07 AM	Application	167
BEAUti v1.10.4	11/2/2023 9:07 AM	Application	111
LogCombiner v1.10.4	11/2/2023 9:07 AM	Application	144
PAOX	11/4/2023 9:54 PM	File	127
PAOX.mcc.tre	11/4/2023 9:57 PM	TRE File	7
PAOX_Aligned.log	11/5/2023 10:27 PM	TXT File	2,663
PAOX_Aligned.ops	11/5/2023 10:27 PM	TXT File	2
PAOX_Aligned.trees	11/5/2023 10:27 PM	TXT File	5,332

File name: PAOX.mcc.tre

Save as type: All Files (*.*)

6. Click run. The program will now generate the data for trees based off of the median outcome of possibilities produced by BEAST.

7. To actually visualize the tree, you need to navigate to where FigTree was installed and open FigTree. Once FigTree is open, click file, then open the newly generated file from TreeAnnotator to see the newly created tree.


FigTree v1.4.4 - PAOX.mcc.tre
File Edit Tree Help
Cartoon Collapse Reroot Rotate Annotate Colour Hilight Find
Node Clade Taxa
Selection Mode
Prev/Next
Filter
Layout
Zoom:
Expansion:
Fish Eye:
Root Length:
Curvature:
Align Tip Labels
Current Tree: 1 / 1
Appearance
Trees
Time Scale
Tip Labels
Tip Shapes
Node Labels
Node Shapes
Node Bars
Branch Labels
Scale Bar
Scale Axis
Legend
ENSMODG00000010464.gray short-tailed opossum (Monodelphis domestica).PAOX.cDNA
ENSSHAG00000010959.Tasmanian devil (Sarcophilus harrisii).PAOX.cDNA
ENSMEUG00000004879.tammar wallaby (Notamacropus eugenii).PAOX.cDNA
ENSPCIG00000017214.koala (Phascolarctos cinereus).PAOX.cDNA
ENSVURG00010015435.common wombat (Vombatus ursinus).PAOX.cDNA
ENSLAFG00000008558.African bush elephant (Loxodonta africana).PAOX.cDNA
ENSPCAG00000005386.rock hyrax (P. capensis).PAOX.cDNA
ENSOANG00000036296.platypus (Ornithorhynchus anatinus).PAOX.cDNA
0.03

Tree: 8 tips [height = 0.251, length = 0E0]

8. When compared to the phylogenetic tree found in Deakin & O'Neil (2020), we can see our small example phylogeny lines up fairly well to theirs when looking at the marsupials. Opossums are placed on their own branch, then Tasmanian Devils, then Wallabies, and finally Wombats and Koalas make up their own clade.

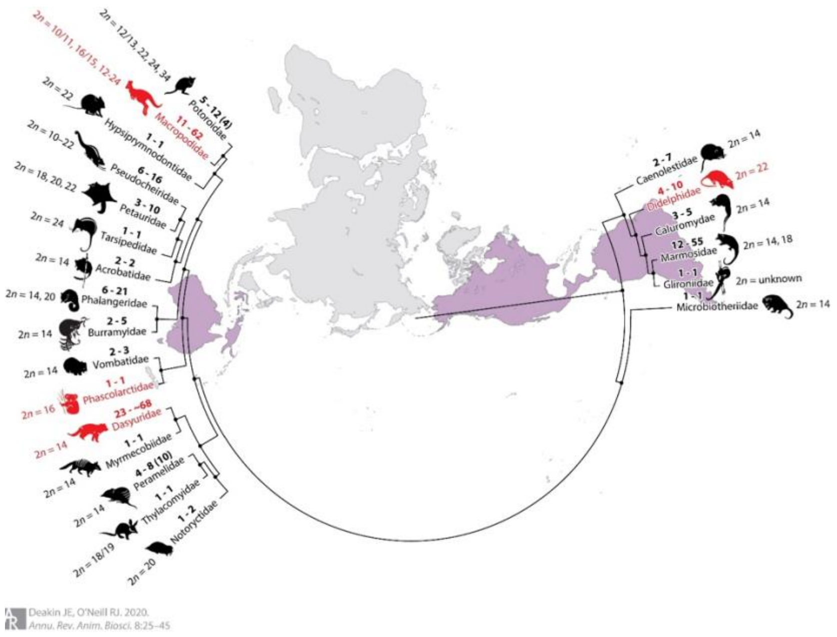


Figure 1 (Deakin & O'Neill, 2020)

References

Bickford, D., Lohman, D. J., Sodhi, N. S., Ng, P. K. L., Meier, R., Winker, K., Ingram, K. K., & Das, I. (2007). Cryptic species as a window on diversity and conservation. *Trends in Ecology & Evolution*, *22*(3), 148–155. https://doi.org/10.1016/j.tree.2006.11.004

Deakin, J. E., & O'Neill, R. J. (2020). Evolution of Marsupial Genomes. *Annual Review of Animal Biosciences*, *8*(1), 25–45. https://doi.org/10.1146/annurev-animal-021419-083555

Thompson, M. J., Capilla-Lasheras, P., Dominoni, D. M., Réale, D., & Charmantier, A. (2022). Phenotypic variation in urban environments: Mechanisms and implications. *Trends in Ecology & Evolution*, *37*(2), 171–182. https://doi.org/10.1016/j.tree.2021.09.009