

Re-analysis of Carballo (2022) co-phylogenetic plot

Liam J. Revell

26 February, 2023

As I mentioned in the Slack, I'm presently working on a new (extend) application note for the updated *phytools* R package. In this article I have included a number of different, fully-functional R workflows, similar to those that we use in our class exercises and that fill my book with Luke. *phytools* contains a very popular co-phylogenetic ("tanglegram") plotting method called *cophylo*. Unfortunately, I was having a lot of difficulty encountering published data to illustrate it in my paper!

I found one example that I liked in a paper by Caraballo (2022), giving the phylogenetic association of bats and their betacoronaviruses. Unfortunately, the author didn't publish the trees or original data in a usable format. The only "available" data was a set of tables (published in PDF format, no less) with *GenBank* accession numbers for all of the sequences used in the study!

I decided to start there anyway, and see what I could do to pull down each of these sequences based on their accession numbers, align the sequences, and do phylogenetic inference in R.

As an important sidenote, to replicate this analysis you will need to download the stand-alone program *MUSCLE* (here), and then (in Windows) add the directory of this program to your PATH environment variable. Other than that, I just pulled the PDFs from Caraballo's supplement, used Adobe to convert it to a Word doc, imported to Excel, and saved as a CSV file to read into R!

Load packages.

```
library(phytools)
```

```
## Loading required package: ape
```

```
## Loading required package: maps
```

```
library(phangorn)
```

Read my bat accession data.

```
Bats<-read.csv(file="Bats.csv")
```

```
Bats
```

##	Species	Accession	Host.type
## 1	Anoura geoffroyi	FJ155495	alphaCoV
## 2	Artibeus jamaicensis	U66503	alphaCoV
## 3	Artibeus lituratus	U66505	alphaCoV
## 4	Artibeus planirostris	U66508	alphaCoV
## 5	Carollia castanea	AF187021	alphaCoV
## 6	Carollia perspicillata	FJ589655	alphaCoV
## 7	Cynomops abrasus	GQ424038	alphaCoV
## 8	Cynomops planirostris	KP134563	alphaCoV
## 9	Desmodus rotundus	DQ077398	alphaCoV
## 10	Eptesicus brasiliensis	MW926801	alphaCoV
## 11	Glossophaga soricina	MN719299	alphaCoV
## 12	Mesophylla macconnelli	FJ154122	alphaCoV

```
## 13      Molossus molossus  KP134554  alphaCoV
## 14      Molossus rufus    MH058080  alphaCoV
## 15      Myotis lucifugus  AF376854  alphaCoV
## 16      Myotis nigricans  AF376864  alphaCoV
## 17      Myotis riparius   JX130474  alphaCoV
## 18      Perimyotis subflavus  AJ504449  alphaCoV
## 19      Phyllostomus discolor  KC783056  alphaCoV
## 20      Sturnira erythromos  FJ154179  alphaCoV
## 21      Sturnira lilium     KC753800  alphaCoV
## 22      Tadarida brasiliensis  JF489129  alphaCoV
## 23      Glossophaga soricina  MN719299  betaCoV
## 24      Pteronotus davyi     KX589762  betaCoV
## 25      Pteronotus parnellii  AY604456  betaCoV
## 26      Carollia perspicillata  FJ589655  betaCoV
## 27      Artibeus planirostris  U66508    betaCoV
## 28      Artibeus lituratus    U66505    betaCoV
## 29      Eumops glaucinus     KP134574  betaCoV
```

This contains accession numbers for both alpha- and betacoronaviruses. I decided to just do the betacoronaviruses.

```
betaAccessions<-setNames(
  Bats$Accession[which(Bats$Host.type=="betaCoV")],
  Bats$Species[which(Bats$Host.type=="betaCoV")])
betaAccessions
```

```
##      Glossophaga soricina      Pteronotus davyi      Pteronotus parnellii
##      "MN719299"              "KX589762"              "AY604456"
##      Carollia perspicillata  Artibeus planirostris      Artibeus lituratus
##      "FJ589655"              "U66508"              "U66505"
##      Eumops glaucinus
##      "KP134574"
```

Next, I'll use the *ape* function `read.GenBank` to pull the corresponding sequences from *GenBank*.

```
betaBats<-read.GenBank(betaAccessions,seq.names=names(betaAccessions))
betaBats
```

```
## 7 DNA sequences in binary format stored in a list.
##
## Mean sequence length: 1140.143
##   Shortest sequence: 1140
##   Longest sequence: 1141
##
## Labels:
## MN719299
## KX589762
## AY604456
## FJ589655
## U66508
## U66505
## ...
##
## Base composition:
##      a      c      g      t
## 0.295 0.280 0.134 0.291
## (Total: 7.98 kb)
```

These sequences are all *almost* the same length, so they may already be (mostly) aligned. We can use MUSCLE to align them anyway.

```
betaBats.aligned<-muscle5(betaBats)
betaBats.aligned
```

```
## 7 DNA sequences in binary format stored in a matrix.
##
## All sequences of same length: 1141
##
## Labels:
## MN719299
## KX589762
## AY604456
## FJ589655
## U66508
## U66505
## ...
##
## Base composition:
##      a      c      g      t
## 0.295 0.280 0.134 0.291
## (Total: 7.99 kb)
```

We're already ready to estimate our tree. We could do model selection, but I decided just to use "GTR+G(4)+I".

```
betaBat_pml<-pml_bb(betaBats.aligned,model="GTR+G(4)+I")

## optimize edge weights: -4975.334 --> -4936.397
## optimize base frequencies: -4936.397 --> -4915.004
## optimize rate matrix: -4915.004 --> -4711.746
## optimize invariant sites: -4711.746 --> -4657.845
## optimize shape parameter: -4657.845 --> -4657.633
## optimize edge weights: -4657.633 --> -4643.702
## optimize topology: -4643.702 --> -4643.587 NNI moves: 1
## optimize base frequencies: -4643.587 --> -4636.578
## optimize rate matrix: -4636.578 --> -4620.92
## optimize invariant sites: -4620.92 --> -4619.48
## optimize shape parameter: -4619.48 --> -4618.934
## optimize edge weights: -4618.934 --> -4616.322
## optimize topology: -4616.322 --> -4616.124 NNI moves: 1
## optimize base frequencies: -4616.124 --> -4613.988
## optimize rate matrix: -4613.988 --> -4610.864
## optimize invariant sites: -4610.864 --> -4609.685
## optimize shape parameter: -4609.685 --> -4608.994
## optimize edge weights: -4608.994 --> -4608.325
## optimize topology: -4608.325 --> -4608.325 NNI moves: 0
## optimize topology: -4672.3 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 1 , best pscore so far: -4608.32469651886"
## optimize topology: -4657.069 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 2 , best pscore so far: -4608.32469651886"
## optimize topology: -4693.143 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 3 , best pscore so far: -4608.32469543997"
## optimize topology: -4656.135 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 4 , best pscore so far: -4608.32469543997"
## optimize topology: -4711.687 --> -4608.325 NNI moves: 4
```

```

## [1] "Ratchet iteration 5 , best pscore so far: -4608.32469543997"
## optimize topology: -4780.042 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 6 , best pscore so far: -4608.32469543997"
## optimize topology: -4672.915 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 7 , best pscore so far: -4608.32469543997"
## optimize topology: -4765.338 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 8 , best pscore so far: -4608.32469543997"
## optimize topology: -4675.98 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 9 , best pscore so far: -4608.32469543997"
## optimize topology: -4705.612 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 10 , best pscore so far: -4608.32469543997"
## optimize topology: -4734.443 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 11 , best pscore so far: -4608.32469543997"
## optimize topology: -4744.254 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 12 , best pscore so far: -4608.32469543997"
## optimize topology: -4608.325 --> -4608.325 NNI moves: 0
## [1] "Ratchet iteration 13 , best pscore so far: -4608.32469541763"
## optimize topology: -4672.3 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 14 , best pscore so far: -4608.32469541763"
## optimize topology: -4658.912 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 15 , best pscore so far: -4608.32469541763"
## optimize topology: -4666.141 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 16 , best pscore so far: -4608.32469541763"
## optimize topology: -4683.924 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 17 , best pscore so far: -4608.32469541763"
## optimize topology: -4672.3 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 18 , best pscore so far: -4608.32469541763"
## optimize topology: -4621.646 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 19 , best pscore so far: -4608.32469541763"
## optimize topology: -4621.646 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 20 , best pscore so far: -4608.32469541763"
## optimize topology: -4744.254 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 21 , best pscore so far: -4608.32469541763"
## optimize topology: -4662.137 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 22 , best pscore so far: -4608.32469541763"
## optimize topology: -4622.044 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 23 , best pscore so far: -4608.32469541763"
## optimize topology: -4656.135 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 24 , best pscore so far: -4608.32469541763"
## optimize topology: -4623.018 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 25 , best pscore so far: -4608.32469541763"
## optimize topology: -4656.135 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 26 , best pscore so far: -4608.32469541763"
## optimize topology: -4702.041 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 27 , best pscore so far: -4608.32469541763"
## optimize topology: -4672.3 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 28 , best pscore so far: -4608.32469541763"
## optimize topology: -4697.748 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 29 , best pscore so far: -4608.32469541763"
## optimize topology: -4672.3 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 30 , best pscore so far: -4608.32469541763"
## optimize topology: -4668.448 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 31 , best pscore so far: -4608.32469541763"
## optimize topology: -4693.515 --> -4608.325 NNI moves: 3

```

```

## [1] "Ratchet iteration 32 , best pscore so far: -4608.32469541763"
## optimize topology: -4668.448 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 33 , best pscore so far: -4608.32469541763"
## optimize topology: -4738.861 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 34 , best pscore so far: -4608.32469541763"
## optimize topology: -4708.062 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 35 , best pscore so far: -4608.32469541763"
## optimize topology: -4672.258 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 36 , best pscore so far: -4608.32469541763"
## optimize topology: -4683.924 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 37 , best pscore so far: -4608.32469541763"
## optimize topology: -4712.272 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 38 , best pscore so far: -4608.32469541763"
## optimize topology: -4622.331 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 39 , best pscore so far: -4608.32469541763"
## optimize topology: -4657.889 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 40 , best pscore so far: -4608.32469541763"
## optimize topology: -4692.768 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 41 , best pscore so far: -4608.32469541763"
## optimize topology: -4697.748 --> -4608.325 NNI moves: 3
## [1] "Ratchet iteration 42 , best pscore so far: -4608.32469541763"
## optimize topology: -4663.977 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 43 , best pscore so far: -4608.32469541763"
## optimize topology: -4683.924 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 44 , best pscore so far: -4608.32469541763"
## optimize topology: -4683.924 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 45 , best pscore so far: -4608.32469541763"
## optimize topology: -4644.089 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 46 , best pscore so far: -4608.32469541763"
## optimize topology: -4712.376 --> -4608.325 NNI moves: 4
## [1] "Ratchet iteration 47 , best pscore so far: -4608.32469541763"
## optimize topology: -4656.135 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 48 , best pscore so far: -4608.32469541763"
## optimize topology: -4621.646 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 49 , best pscore so far: -4608.32469541763"
## optimize topology: -4644.089 --> -4608.325 NNI moves: 2
## [1] "Ratchet iteration 50 , best pscore so far: -4608.32469541763"
## optimize base frequencies: -4608.325 --> -4607.835
## optimize rate matrix: -4607.835 --> -4606.957
## optimize invariant sites: -4606.957 --> -4606.147
## optimize shape parameter: -4606.147 --> -4605.815
## optimize edge weights: -4605.815 --> -4605.637
## optimize topology: -4605.637 --> -4605.637 NNI moves: 0
## optimize base frequencies: -4605.637 --> -4605.46
## optimize rate matrix: -4605.46 --> -4605.168
## optimize invariant sites: -4605.168 --> -4604.93
## optimize shape parameter: -4604.93 --> -4604.875
## optimize edge weights: -4604.875 --> -4604.809
## optimize base frequencies: -4604.809 --> -4604.767
## optimize rate matrix: -4604.767 --> -4604.651
## optimize invariant sites: -4604.651 --> -4604.616
## optimize shape parameter: -4604.616 --> -4604.613
## optimize edge weights: -4604.613 --> -4604.581
## optimize base frequencies: -4604.581 --> -4604.572

```

```

## optimize rate matrix: -4604.572 --> -4604.531
## optimize invariant sites: -4604.531 --> -4604.528
## optimize shape parameter: -4604.528 --> -4604.527
## optimize edge weights: -4604.527 --> -4604.508
## optimize base frequencies: -4604.508 --> -4604.506
## optimize rate matrix: -4604.506 --> -4604.484
## optimize invariant sites: -4604.484 --> -4604.484
## optimize shape parameter: -4604.484 --> -4604.482
## optimize edge weights: -4604.482 --> -4604.469
## optimize base frequencies: -4604.469 --> -4604.469
## optimize rate matrix: -4604.469 --> -4604.457
## optimize invariant sites: -4604.457 --> -4604.457
## optimize shape parameter: -4604.457 --> -4604.454
## optimize edge weights: -4604.454 --> -4604.446
## optimize base frequencies: -4604.446 --> -4604.446
## optimize rate matrix: -4604.446 --> -4604.439
## optimize invariant sites: -4604.439 --> -4604.438
## optimize shape parameter: -4604.438 --> -4604.437
## optimize edge weights: -4604.437 --> -4604.431
## optimize base frequencies: -4604.431 --> -4604.431
## optimize rate matrix: -4604.431 --> -4604.427
## optimize invariant sites: -4604.427 --> -4604.426
## optimize shape parameter: -4604.426 --> -4604.425
## optimize edge weights: -4604.425 --> -4604.422
## optimize base frequencies: -4604.422 --> -4604.422
## optimize rate matrix: -4604.422 --> -4604.419
## optimize invariant sites: -4604.419 --> -4604.419
## optimize shape parameter: -4604.419 --> -4604.418
## optimize edge weights: -4604.418 --> -4604.416
## optimize base frequencies: -4604.416 --> -4604.416
## optimize rate matrix: -4604.416 --> -4604.414
## optimize invariant sites: -4604.414 --> -4604.414
## optimize shape parameter: -4604.414 --> -4604.413
## optimize edge weights: -4604.413 --> -4604.412
## optimize base frequencies: -4604.412 --> -4604.412
## optimize rate matrix: -4604.412 --> -4604.41
## optimize invariant sites: -4604.41 --> -4604.41
## optimize shape parameter: -4604.41 --> -4604.41
## optimize edge weights: -4604.41 --> -4604.409

```

Done! Let me pull out my tree.

```

bat_mltree_unrooted<-betaBat_pml$tree
bat_mltree_unrooted

```

```

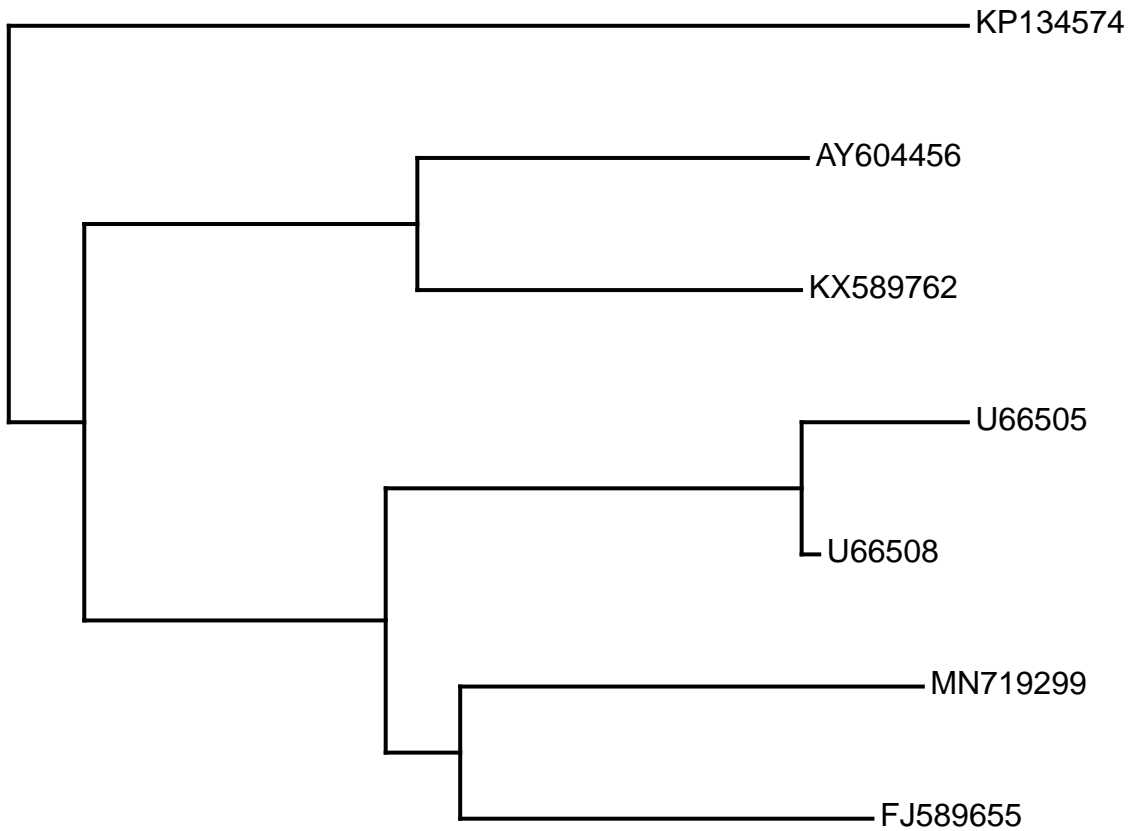
##
## Phylogenetic tree with 7 tips and 5 internal nodes.
##
## Tip labels:
##  MN719299, KX589762, AY604456, KP134574, FJ589655, U66508, ...
## Node labels:
##  1, 1, 1, 1, 1
##
## Unrooted; includes branch lengths.

```

We can proceed to root it. Generally speaking, this wouldn't be advised, but I happen to know that the

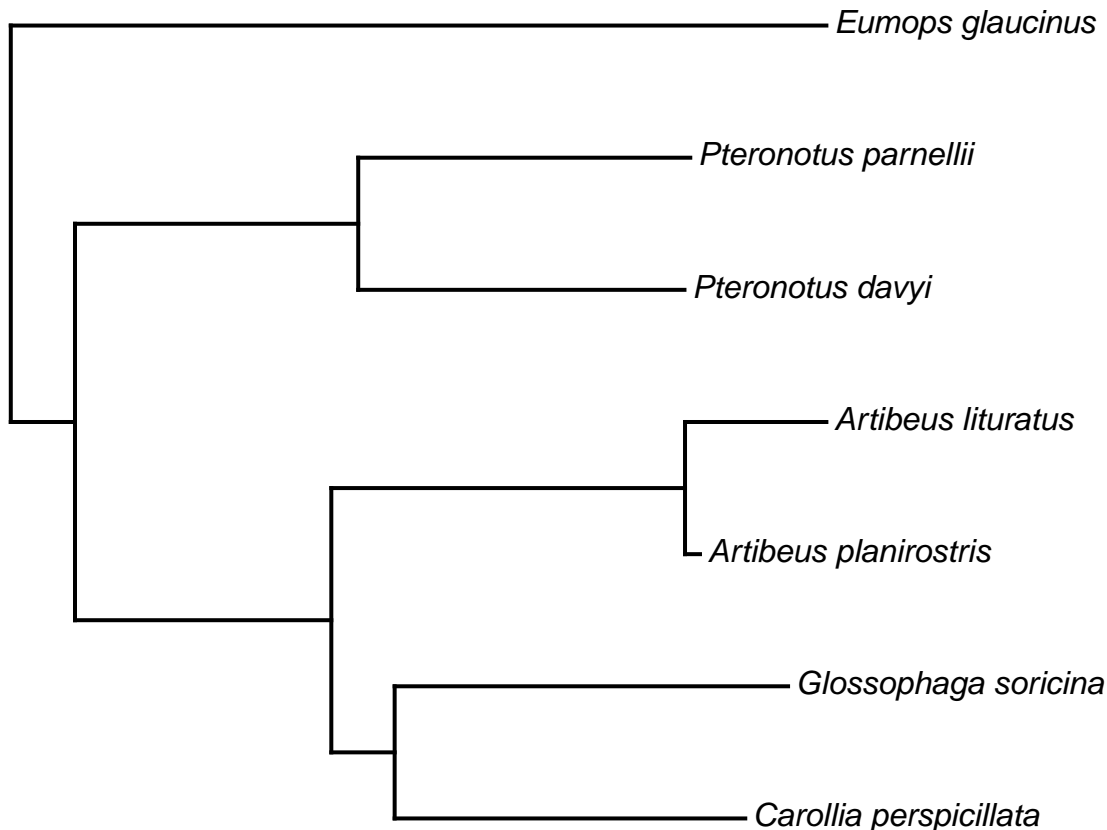
method of *midpoint rooting* will give me the tree of Caraballo (2022).

```
plotTree(bat.tree<-untangle(midpoint(bat_mltree_unrooted),"read.tree"))
```



This tree has accession numbers instead of species labels – but we know how they’re related, so let’s update them.

```
spp<-setNames(names(betaAccessions),betaAccessions)
bat.tree$tip.label<-spp[bat.tree$tip.label]
plotTree(bat.tree,ftype="i")
```



Great. So far, so good.

Next, let's do much the same with our betacoronavirus sequences. In this case, we don't have named lineages – all we've got are the *GenBank* labels, so we can just keep these as our OTU names.

```

betaCovs<-read.csv(file="betaCovs.csv")
betaCovs_seqs<-read.GenBank(betaCovs$Accession.Number)
betaCovs_seqs.aligned<-muscle5(betaCovs_seqs)
betaCovs_tree<-pml_bb(betaCovs_seqs.aligned,
  model="GTR+G(4)+I")

## optimize edge weights: -3211.334 --> -3100.881
## optimize base frequencies: -3100.881 --> -3097.157
## optimize rate matrix: -3097.157 --> -3017.458
## optimize invariant sites: -3017.458 --> -3007.4
## optimize shape parameter: -3007.4 --> -3007.312
## optimize edge weights: -3007.312 --> -3001.531
## optimize topology: -3001.531 --> -3001.531 NNI moves: 0
## optimize topology: -3056.148 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 1 , best pscore so far: -3001.53126363834"
## optimize topology: -3059.174 --> -3001.531 NNI moves: 9
## [1] "Ratchet iteration 2 , best pscore so far: -3001.53126363834"
## optimize topology: -3022.192 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 3 , best pscore so far: -3001.53126325208"
## optimize topology: -3032.828 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 4 , best pscore so far: -3001.53126325208"
## optimize topology: -3049.097 --> -3001.531 NNI moves: 6
  
```



```

## [1] "Ratchet iteration 5 , best pscore so far: -3001.53126325208"
## optimize topology: -3036.536 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 6 , best pscore so far: -3001.53126318693"
## optimize topology: -3048.227 --> -3001.531 NNI moves: 7
## [1] "Ratchet iteration 7 , best pscore so far: -3001.53126314175"
## optimize topology: -3040.522 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 8 , best pscore so far: -3001.53126314175"
## optimize topology: -3055.733 --> -3003.529 NNI moves: 8
## [1] "Ratchet iteration 9 , best pscore so far: -3001.53126314175"
## optimize topology: -3062.204 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 10 , best pscore so far: -3001.53126314175"
## optimize topology: -3032.254 --> -3001.531 NNI moves: 3
## [1] "Ratchet iteration 11 , best pscore so far: -3001.53126314175"
## optimize topology: -3019.942 --> -3001.531 NNI moves: 7
## [1] "Ratchet iteration 12 , best pscore so far: -3001.53126314175"
## optimize topology: -3064.971 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 13 , best pscore so far: -3001.53126314175"
## optimize topology: -3023.13 --> -3001.531 NNI moves: 3
## [1] "Ratchet iteration 14 , best pscore so far: -3001.53126314175"
## optimize topology: -3068.447 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 15 , best pscore so far: -3001.53126314175"
## optimize topology: -3042.785 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 16 , best pscore so far: -3001.53126314175"
## optimize topology: -3057.379 --> -3003.432 NNI moves: 8
## [1] "Ratchet iteration 17 , best pscore so far: -3001.53126314175"
## optimize topology: -3023.839 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 18 , best pscore so far: -3001.53126314175"
## optimize topology: -3047.678 --> -3011.475 NNI moves: 7
## [1] "Ratchet iteration 19 , best pscore so far: -3001.53126314175"
## optimize topology: -3050.13 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 20 , best pscore so far: -3001.53126314175"
## optimize topology: -3048.556 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 21 , best pscore so far: -3001.53126314175"
## optimize topology: -3034.603 --> -3001.531 NNI moves: 7
## [1] "Ratchet iteration 22 , best pscore so far: -3001.53126314175"
## optimize topology: -3067.636 --> -3001.531 NNI moves: 7
## [1] "Ratchet iteration 23 , best pscore so far: -3001.53126314175"
## optimize topology: -3047.109 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 24 , best pscore so far: -3001.53126314175"
## optimize topology: -3015.986 --> -3001.531 NNI moves: 4
## [1] "Ratchet iteration 25 , best pscore so far: -3001.53126314175"
## optimize topology: -3072.036 --> -3005.959 NNI moves: 6
## [1] "Ratchet iteration 26 , best pscore so far: -3001.53126314175"
## optimize topology: -3037.558 --> -3003.136 NNI moves: 5
## [1] "Ratchet iteration 27 , best pscore so far: -3001.53126314175"
## optimize topology: -3031.78 --> -3001.531 NNI moves: 4
## [1] "Ratchet iteration 28 , best pscore so far: -3001.53126314175"
## optimize topology: -3050.125 --> -3001.531 NNI moves: 7
## [1] "Ratchet iteration 29 , best pscore so far: -3001.53126314175"
## optimize topology: -3045.157 --> -3018.841 NNI moves: 7
## [1] "Ratchet iteration 30 , best pscore so far: -3001.53126314175"
## optimize topology: -3044.238 --> -3003.432 NNI moves: 6
## [1] "Ratchet iteration 31 , best pscore so far: -3001.53126314175"
## optimize topology: -3037.26 --> -3001.531 NNI moves: 4

```

```

## [1] "Ratchet iteration 32 , best pscore so far: -3001.53126314175"
## optimize topology: -3062.236 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 33 , best pscore so far: -3001.53126314175"
## optimize topology: -3064.26 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 34 , best pscore so far: -3001.53126314175"
## optimize topology: -3058.552 --> -3010.074 NNI moves: 8
## [1] "Ratchet iteration 35 , best pscore so far: -3001.53126314175"
## optimize topology: -3050.958 --> -3001.531 NNI moves: 7
## [1] "Ratchet iteration 36 , best pscore so far: -3001.53126314175"
## optimize topology: -3008.5 --> -3006.45 NNI moves: 3
## [1] "Ratchet iteration 37 , best pscore so far: -3001.53126314175"
## optimize topology: -3053.629 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 38 , best pscore so far: -3001.53126314175"
## optimize topology: -3083.412 --> -3001.531 NNI moves: 9
## [1] "Ratchet iteration 39 , best pscore so far: -3001.53126314175"
## optimize topology: -3032.572 --> -3001.531 NNI moves: 4
## [1] "Ratchet iteration 40 , best pscore so far: -3001.53126314175"
## optimize topology: -3038.124 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 41 , best pscore so far: -3001.53126314175"
## optimize topology: -3032.304 --> -3001.531 NNI moves: 5
## [1] "Ratchet iteration 42 , best pscore so far: -3001.53126314175"
## optimize topology: -3039.986 --> -3003.432 NNI moves: 6
## [1] "Ratchet iteration 43 , best pscore so far: -3001.53126314175"
## optimize topology: -3065.314 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 44 , best pscore so far: -3001.53126314175"
## optimize topology: -3044.361 --> -3001.531 NNI moves: 6
## [1] "Ratchet iteration 45 , best pscore so far: -3001.53126314175"
## optimize topology: -3041.247 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 46 , best pscore so far: -3001.53126314175"
## optimize topology: -3016.917 --> -3003.136 NNI moves: 6
## [1] "Ratchet iteration 47 , best pscore so far: -3001.53126314175"
## optimize topology: -3033.285 --> -3006.238 NNI moves: 5
## [1] "Ratchet iteration 48 , best pscore so far: -3001.53126314175"
## optimize topology: -3057.541 --> -3008.134 NNI moves: 8
## [1] "Ratchet iteration 49 , best pscore so far: -3001.53126314175"
## optimize topology: -3053.94 --> -3001.531 NNI moves: 8
## [1] "Ratchet iteration 50 , best pscore so far: -3001.53126314175"
## optimize base frequencies: -3001.531 --> -2999.472
## optimize rate matrix: -2999.472 --> -2996.325
## optimize invariant sites: -2996.325 --> -2995.03
## optimize shape parameter: -2995.03 --> -2994.999
## optimize edge weights: -2994.999 --> -2993.999
## optimize topology: -2993.999 --> -2993.999 NNI moves: 0
## optimize base frequencies: -2993.999 --> -2993.825
## optimize rate matrix: -2993.825 --> -2993.275
## optimize invariant sites: -2993.275 --> -2992.952
## optimize shape parameter: -2992.952 --> -2992.951
## optimize edge weights: -2992.951 --> -2992.672
## optimize base frequencies: -2992.672 --> -2992.657
## optimize rate matrix: -2992.657 --> -2992.511
## optimize invariant sites: -2992.511 --> -2992.456
## optimize shape parameter: -2992.456 --> -2992.45
## optimize edge weights: -2992.45 --> -2992.349
## optimize base frequencies: -2992.349 --> -2992.348

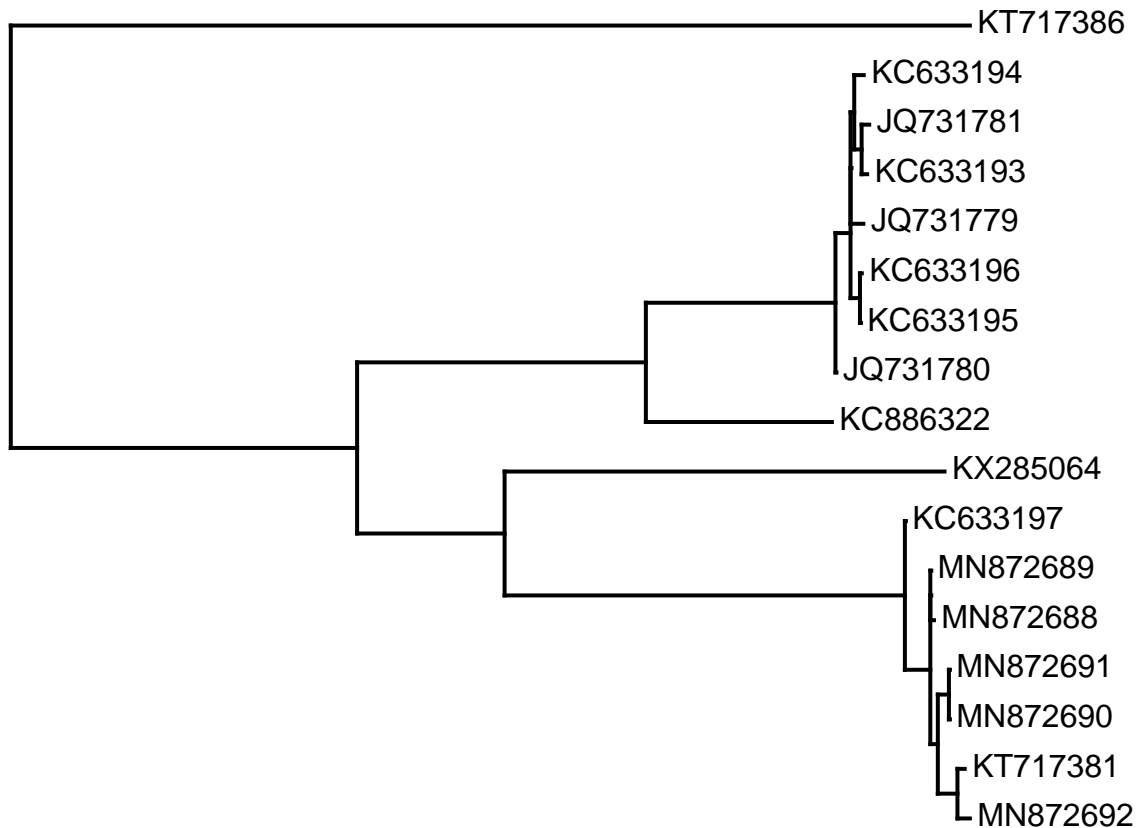
```

```

## optimize rate matrix: -2992.348 --> -2992.301
## optimize invariant sites: -2992.301 --> -2992.297
## optimize shape parameter: -2992.297 --> -2992.283
## optimize edge weights: -2992.283 --> -2992.24
## optimize base frequencies: -2992.24 --> -2992.239
## optimize rate matrix: -2992.239 --> -2992.22
## optimize invariant sites: -2992.22 --> -2992.22
## optimize shape parameter: -2992.22 --> -2992.203
## optimize edge weights: -2992.203 --> -2992.183
## optimize base frequencies: -2992.183 --> -2992.183
## optimize rate matrix: -2992.183 --> -2992.175
## optimize invariant sites: -2992.175 --> -2992.171
## optimize shape parameter: -2992.171 --> -2992.155
## optimize edge weights: -2992.155 --> -2992.144
## optimize base frequencies: -2992.144 --> -2992.144
## optimize rate matrix: -2992.144 --> -2992.14
## optimize invariant sites: -2992.14 --> -2992.133
## optimize shape parameter: -2992.133 --> -2992.119
## optimize edge weights: -2992.119 --> -2992.113
## optimize base frequencies: -2992.113 --> -2992.113
## optimize rate matrix: -2992.113 --> -2992.111
## optimize invariant sites: -2992.111 --> -2992.104
## optimize shape parameter: -2992.104 --> -2992.091
## optimize edge weights: -2992.091 --> -2992.088
## optimize base frequencies: -2992.088 --> -2992.088
## optimize rate matrix: -2992.088 --> -2992.086
## optimize invariant sites: -2992.086 --> -2992.079
## optimize shape parameter: -2992.079 --> -2992.069
## optimize edge weights: -2992.069 --> -2992.067
## optimize base frequencies: -2992.067 --> -2992.067
## optimize rate matrix: -2992.067 --> -2992.066
## optimize invariant sites: -2992.066 --> -2992.059
## optimize shape parameter: -2992.059 --> -2992.05
## optimize edge weights: -2992.05 --> -2992.049
## optimize base frequencies: -2992.049 --> -2992.049
## optimize rate matrix: -2992.049 --> -2992.049
## optimize invariant sites: -2992.049 --> -2992.044
## optimize shape parameter: -2992.044 --> -2992.036
## optimize edge weights: -2992.036 --> -2992.036

plotTree(virus.tree<-untangle(midpoint(betaCovs_tree$tree),
"read.tree"))

```



We can next build a table of associations between the taxa in our bat tree and the taxa in our virus tree. Once again, this is from the supplemental material of Caraballo (2022).

```
assoc<-data.frame(Bats=betaCovs$Host.species,
  betaCoVs=betaCovs$Accession.Number)
assoc
```

```
##           Bats betaCoVs
## 1 Carollia perspicillata KC633197
## 2   Artibeus lituratus KT717381
## 3   Artibeus planirostris MN872692
## 4   Artibeus planirostris MN872690
## 5   Artibeus planirostris MN872691
## 6   Artibeus planirostris MN872689
## 7   Artibeus planirostris MN872688
## 8       Glossophaginae KX285064
## 9   Pteronotus davyi KC886322
## 10  Pteronotus parnellii JQ731780
## 11  Pteronotus parnellii JQ731779
## 12  Pteronotus parnellii JQ731781
## 13  Pteronotus parnellii KC633193
## 14  Pteronotus parnellii KC633194
## 15  Pteronotus parnellii KC633195
## 16  Pteronotus parnellii KC633196
## 17    Eumops glaucinus KT717386
```

One thing I noticed right away is that the tip labeled "Glossophaga soricina" in the tree is actually coded by the subfamily, "Glossophaginae", in our association table.

```
assoc[assoc$Bats=="Glossophaginae",1]<-"Glossophaga soricina"
```

```
obj<-cophylo(bat.tree,virus.tree,assoc=assoc)
```

```
## Rotating nodes to optimize matching...
## Done.
```

```
obj
```

```
## Object of class "cophylo" containing:
```

```
##
```

```
## (1) 2 (possibly rotated) phylogenetic trees in an object of class "multiPhylo".
```

```
##
```

```
## (2) A table of associations between the tips of both trees.
```

We're done!

```
plot(obj,link.lty="solid",link.type="curved",fsize=c(0.9,0.8))
```

