Summer Research Project: Bats

By Randy Posada

Fig.cap="See attributions for link

My Report

Overview

In this report I use the Open Tree of Life alongside Physcraper to create and access an updated phylogentic tree of all bats.

There are over 1000 different species of bats. These extraordinary flying mammals use their hands to fly; granted their order name *chiroptera*, which translates in Greek to ‘Hand Wings’. Each of their fingers are connected to one another through a thin layer of skin which allows these nocturnal mammals to take off into flight. Chiroptera are the only mammals with the capability of continued flight.

TASK 1: Add an intro to the Open Tree of Life. TASK 2: How do we use tools from the Open Tree of Life with R? Talk about the rotl package.

TASK 3: What does the tnrs_match_names function does?.

```
my_taxa <- c("chiroptera")
resolved_names <- rotl::tnrs_match_names(names = my_taxa)

resolved_names
#>   search_string unique_name approximate_match ott_id is_synonym flags
#> 1    chiroptera  Chiroptera             FALSE 574724      FALSE
#>   number_matches
#> 1              1
```

It is useful to know the class of an object since it makes manipulating objects much easier with different functions.When we create a class we create a data structure that will house all the objects that belong to a specific class. THis is done for ease of access, organization, and clarity.

```
class(resolved_names)
#> [1] "match_names" "data.frame"
```

The class of the resolved_names object allows us to view the search string name, the unique name ,and the ott_id in respect to the open tree of life. The class of the resolved_names object which includes ‘trns_match_names’, allows us to view two outputs : “match_names” and “data_frame”

In the following chunk, We can subset to obtain certain columns and and if needed, manipulate the formula to extract a specific component from the row. Since there is no function that allows us to extract the values from a row of math_names, we need to use resolved names and indexing. Subsetting ultimately allows us to get values from all columns of one row.

```
resolved_names[1,]
#>   search_string unique_name approximate_match ott_id is_synonym flags
#> 1    chiroptera  Chiroptera             FALSE 574724      FALSE
#>   number_matches
#> 1              1
```

This is part of the previous chunk.

Our goal is to obtain the ott_id for bats (Chiroptera). In order to extract the information we need to subset using the column name ‘unique_name’ in the second part of the formula ‘resolved_names’. This way, we can extract one specific value ( ) from the column we want using the column name.

An OTT id is a unique numerical identifier assigned to a taxon in the Open Tree Taxonomy. Every taxon has a specific OTT id. These OTT ids allow us to interact with the Open Tree of Life.

If we want to obtain the unique name of the taxon used in the synthetic open tree of life, we can use the following function. This function takes into account our previous output of data and extract a specific value.

```
resolved_names[1,"unique_name"]
#> [1] "Chiroptera"
```

The next code gives all info from the current synthetic Open Tree:

```
rotl::tol_about()
#>
#> OpenTree Synthetic Tree of Life.
#>
#> Tree version: opentree12.3
#> Taxonomy version: 3.2draft9
#> Constructed on: 2019-12-23 11:41:23
#> Number of terminal taxa: 2391916
#> Number of source trees: 1216
#> Number of source studies: 1162
#> Source list present: false
#> Root taxon: cellular organisms
#> Root ott_id: 93302
#> Root node_id: ott93302
```

The previous code gave an output of the information from the Synthetic Opent Tree of Life (OTOL) using the package 'rotl'

This function assigns our matched name 'chiroptera' to "Chiroptera_ott_id" and will therefore extract the ott_id we wanted for chiroptera once we run it.

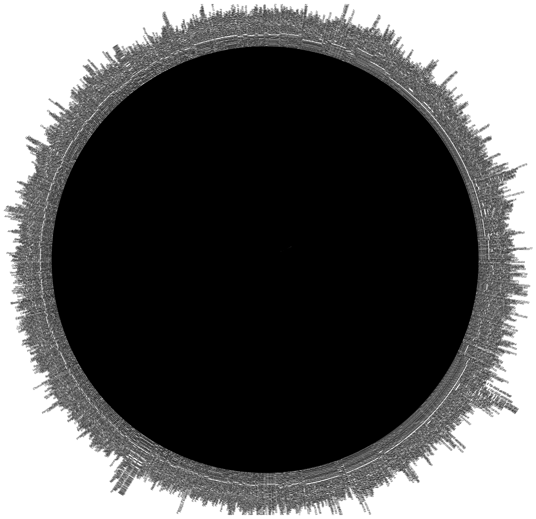
```
chiroptera_ott_id <- rotl::tnrs_match_names("Chiroptera")$ott_id
chiroptera_ott_id
#> [1] 574724
```

The following code will help us get a subtree: **TASK 4: A subtree of what?.**

```
chiroptera_subtree <- rotl::tol_subtree(ott_id = chiroptera_ott_id)

ape::Ntip(chiroptera_subtree)
#> [1] 1820

ape::plot.phylo(chiroptera_subtree, cex = 0.1, type = "fan")
```



```
# or just plot(my_tree, cex = 0.1)
# because it has no branch lengths, it does not plot pretty. We have to get branch lengths for it.
# One way way to do this is to use datelife::datelife_search()
# Another way to do it is to make up teh branch lengths with ape::compute.brlen()
```

This will tell you if the taxon is monophyletic:

It is relevant that our taxon is monophyletic since nonmonophyletic taxa contain 'invalid' or 'broken' data. When the taxon is 'broken', its ott_id is not assidigned to a node in the synthetic tree.

```
rotl::is_in_tree(chiroptera_ott_id)
#> [1] TRUE
```

The above code confirmed that indeed our taxon is monophyletic by giving the output 'TRUE'

OTT ids and node ids allow us to interact with the synthetic OTOL.

```
chiroptera_node_info <- rotl::tol_node_info(chiroptera_ott_id)
chiroptera_node_info
#> 
#> OpenTree node.
#> 
#> Node id: ott574724
#> Number of terminal descendants: 1820
#> Is taxon: TRUE
#> Name: Chiroptera
#> Rank: order
#> ott id: 574724
```

We can use another R package –`datelife`, to get and plot a tree of chiroptera families:

```
chiroptera_families <- datelife::get_ott_children(ott_ids = chiroptera_ott_id, ott_rank = "family")
#>   |                                                                               |
#>   |                                                                               |
ls(chiroptera_families)
#> [1] "Chiroptera"
chiroptera_families
#> $Chiroptera
#>                    ott_id   rank
#> Pteropodidae      574742 family
#> Myzopodidae         6788 family
#> Molossidae        238416 family
#> Vespertilionidae  238434 family
#> Thyropteridae     267980 family
#> Rhinopomatidae    267987 family
#> Hipposideridae    316928 family
#> Craseonycteridae   32051 family
#> Rhinolophidae     635025 family
#> Mystacinidae      759857 family
#> Noctilionidae     759861 family
#> Furipteridae     1060468 family
#> Emballonuridae    581454 family
#> Phyllostomidae    289151 family
#> Nycteridae       1018272 family
#> Natalidae        1018309 family
#> Mormoopidae       292475 family
#> Rhinonycteridae  5819794 family
#> Megadermatidae    813048 family
```

Using Chiroptera Families to create tree

```
chiroptera_families_subtree <- rotl::tol_induced_subtree(chiroptera_families$Chiroptera$ott_id)
#> Warning in collapse_singles(tr, show_progress): Dropping singleton nodes with
#> labels: Megachiroptera ott754606, mrcaott31957ott221782, mrcaott31957ott798260
```

Observing Tree Info

```
chiroptera_families_subtree
#> 
#> Phylogenetic tree with 18 tips and 17 internal nodes.
#> 
#> Tip labels:
```

```
#>  Vespertilionidae_ott238434, Molossidae_ott238416, Natalidae_ott1018309, Myzopodidae_ott6788, Phyllo
#> Node labels:
#>  Chiroptera ott574724, mrcaott6790ott6794, mrcaott6790ott6795, mrcaott6790ott130215, mrcaott6794ott7
#>
#> Rooted; no branch lengths.
```

Plotting Tree Of Chiroptera Families

```
ape::plot.phylo(chiroptera_families_subtree, cex = 0.8)
```

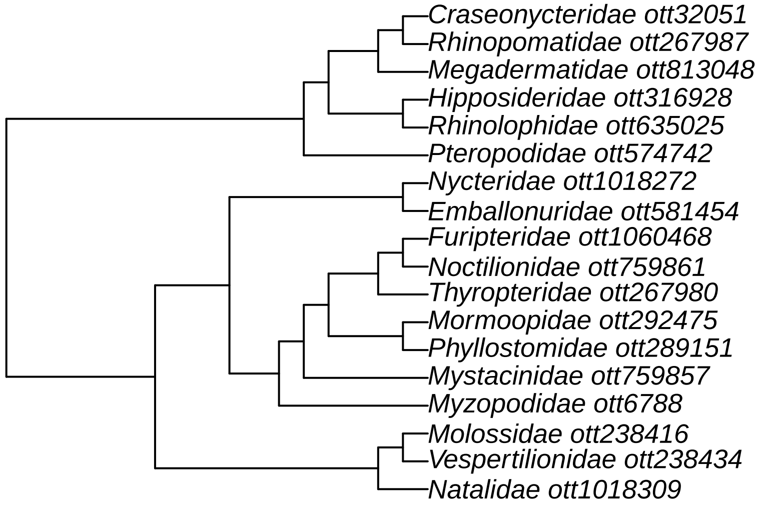


Figure out how to get the ott ids as a vector.

```
chiroptera_families$Chiroptera$ott_id
#>  [1]  574742    6788  238416  238434  267980  267987  316928   32051  635025
#> [10]  759857  759861 1060468  581454  289151 1018272 1018309  292475 5819794
#> [19]  813048
```

```
c(chiroptera_families$Chiroptera$ott_id)
#>  [1]  574742    6788  238416  238434  267980  267987  316928   32051  635025
#> [10]  759857  759861 1060468  581454  289151 1018272 1018309  292475 5819794
#> [19]  813048
```

To get an even smaller bat tree with 5 taxa that you like: First get the scientific names of families, genera or species of bats. Then run my_ott_ids <- rotl::tnrs_match_names to get the OTT ids

here I chose: “Megadermatidae”,“Mormoopidae”,“Vespertilionidae”,“Mystacinidae”,and “Furipteridae.”

```
my_ott_ids <- rotl::tnrs_match_names(c("Megadermatidae","Mormoopidae","Vespertilionidae","Mystacinidae"
```

We will need to extract the ott ids only, because now we have the whole table.

```
my_ott_ids
#>      search_string      unique_name approximate_match  ott_id is_synonym flags
#> 1   megadermatidae   Megadermatidae             FALSE  813048      FALSE
#> 2      mormoopidae      Mormoopidae             FALSE  292475      FALSE
#> 3 vespertilionidae Vespertilionidae             FALSE  238434      FALSE
#> 4     mystacinidae     Mystacinidae             FALSE  759857      FALSE
#> 5     furipteridae     Furipteridae             FALSE 1060468      FALSE
#>   number_matches
#> 1              1
#> 2              1
#> 3                1
#> 4                1
#> 5                1
```

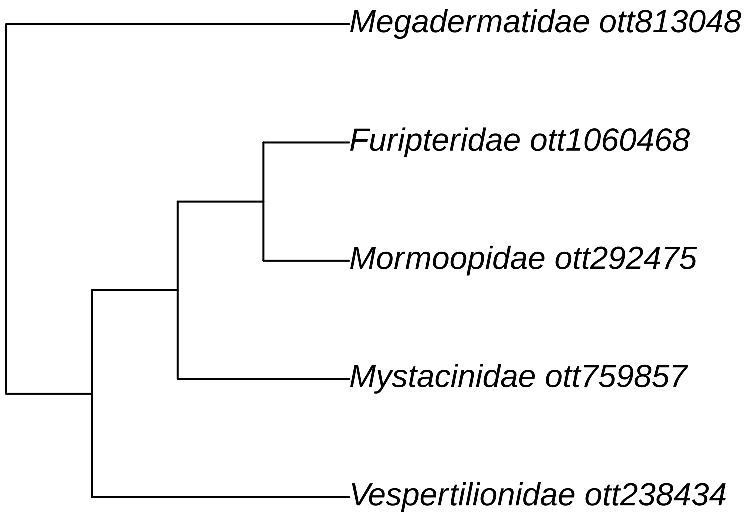
```
my_tree <- rotl::tol_induced_subtree(my_ott_ids$ott_id)
#> Warning in collapse_singles(tr, show_progress): Dropping singleton nodes
#> with labels: mrcaott6790ott6795, mrcaott6790ott130215, mrcaott6794ott73572,
#> mrcaott6794ott9379, mrcaott9379ott167316, mrcaott263938ott604404,
#> mrcaott604404ott1060469, mrcaott10730ott31957, mrcaott31957ott79793,
#> mrcaott79793ott289141
```

This code chunk provides us with the info of our tree.

```
my_tree
#>
#> Phylogenetic tree with 5 tips and 4 internal nodes.
#>
#> Tip labels:
#> [1] "Vespertilionidae_ott238434" "Mormoopidae_ott292475"
#> [3] "Furipteridae_ott1060468"    "Mystacinidae_ott759857"
#> [5] "Megadermatidae_ott813048"
#> Node labels:
#> [1] "Chiroptera ott574724" "mrcaott6790ott6794"   "mrcaott9379ott604409"
#> [4] "mrcaott9379ott263938"
#>
#> Rooted; no branch lengths.
```

To plot the above tree, the ape functiopn “plot.phylo” is used.

```
ape::plot.phylo(my_tree, cex = 1)
```



TASK 5: Describe how do you get help to use a function in R?

TASK 6: Run the datelife::get_datelife_result function for the Chiroptera families.

```
# YOUR_DATELIFE_RESULT_OBJECT <- datelife::get_datelife_result
```

TASK 7: Take the output from datelife::get_datelife_result and run the following code chunk.

```
 # chiroptera_phylo_all <- datelife::summarize_datelife_result(YOUR_DATELIFE_RESULT_OBJECT, summary_for
# datelife::plot_phylo_all(trees = chiroptera_phylo_all)
```

```
chiroptera_node_subtree <- rotl::tol_subtree(node_id = chiroptera_node_info$node_id, label = "name")
#> Warning in collapse_singles(tr, show_progress): Dropping singleton nodes with
#> labels: Murina aurata, Murina huttoni, Eudiscopus, Myotis cf. nipalensis,
#> Myotis muricola, Myotis siligorensis, Myotis bombinus, Myotis myotis, Myotis
#> bocagii, Myotis nesopolus, Myotis oxyotus, Myotis martiniquensis, Myotis evotis,
#> Myotis brandtii, Submyotodon, Scotomanes, Ia, Lasionycteris, Nycticeinops,
#> Philetor, Pipistrellus javanicus, Nyctalus leisleri, Plecotus teneriffae,
#> Plecotus austriacus, Plecotus macrobullaris, Euderma, Idionycteris, Perimyotis,
#> Parastrellus, Lasiurus blossevillii, Dasypterus intermedius, Aeorestes
#> cinereus, Scotophilus viridis, Miniopterinae, Miniopterus griveaudi, Miniopterus
#> natalensis, Niumbaha, Scotozous, Scoteanax, Pharotis, Mimetillus, Atalapha,
#> Cynomops abrasus, Cynomops paranus, Eumops glaucinus, Eumops bonariensis,
#> Sauromys, Molossus currentium, Tomopeas, Platymops, Natalus stramineus,
#> Nyctiellus, Myzopodidae, Centurio senex, Sphaeronycteris, Pygoderma, Ametrida,
#> Ariteus, Ardops, Cubanycteris, Ectophylla, Platyrrhinus helleri, Platyrrhinus
#> lineatus, Mesophylla (genus in Opisthokonta), Sturnira lilium, Lionycteris,
#> Platalina, Xeronycteris, Lophostoma silvicolum, Macrophyllum, Vampyrum,
#> Chrotopterus, Brachyphyllinae, Brachyphylla nana, Lichonycteris, Musonycteris,
#> Choeronycteris, Hylonycteris, Diaemus, Neonycteris, Scleronycteris,
#> Dryadonycteris, Pteronotus davyi, Pteronotus personatus, Thyropteridae,
#> Noctilionidae, Amorphochilus, Mystacinidae, Emballonura semicaudata, Cormura,
#> Cyttarops, Mosia, Rhynchonycteris, Nycteridae, Megachiroptera, Boneia,
#> Mirimiri, Nyctimene albiventer, Pteropus pelewensis, Pteropus admiralitatum,
#> Pteropus rayneri, Pteropus samoensis, Pteropus anetianus, Pteropus capistratus,
#> Pteropus dasymallus, Pteropus melanotus, Melonycteris woodfordi, Nanonycteris,
#> Scotonycteris zenkeri, Chironax, Penthetor, Haplonycteris, Alionycteris,
#> Latidens, Sphaerias, Aproteles, Neopteryx, Plerotes, Hipposideros pomona,
#> Hipposideros ater, Hipposideros caffer, Hipposideros diadema, Anthops,
#> Macronycteris, Rhinonicteris, Cloeotis, Rhinolophidae, Rhinolophinae,
#> Rhinolophus lepidus, Rhinolophus rouxii, Rhinolophus sinicus, Rhinopomatidae,
#> Rhinopoma hardwickii, Craseonycteridae, Craseonycteris, Macroderma (genus in
#> Holozoa), Cardioderma, Lavia, Eudiscoderma, Archaeonycteridae, Icaronycteris,
#> Onychonycteridae, Onychonycteris
head(chiroptera_node_subtree$tip.label)
#> [1] "Kerivoula_hardwickii"  "Kerivoula_titania"     "Kerivoula_kachinensis"
#> [4] "Kerivoula_intermedia"  "Kerivoula_minuta"      "Kerivoula_whiteheadi"
```

Datelife Functions

When you run the get datelife result function it will give node ages from published trees that contain at least two taxa from your search:

```
chiroptera_dr <- datelife::get_datelife_result(chiroptera_node_subtree)
```

The datelife result object is not a tree but a list of tables with the node ages for each pair of taxa from your search. For our 1800 species in the Chiroptera, we got the following trees with node ages:

```
names(chiroptera_dr)
#> [1] "Shi, Jeff J., Daniel L. Rabosky. 2015. Speciation dynamics during the global radiation of extan
#> [2] "Bininda-Emonds, Olaf R. P., Marcel Cardillo, Kate E. Jones, Ross D. E. MacPhee, Robin M. D. Bec
#> [3] "Bininda-Emonds, Olaf R. P., Marcel Cardillo, Kate E. Jones, Ross D. E. MacPhee, Robin M. D. Bec
#> [4] "Bininda-Emonds, Olaf R. P., Marcel Cardillo, Kate E. Jones, Ross D. E. MacPhee, Robin M. D. Bec
#> [5] "Hedges, S. Blair, Julie Marin, Michael Suleski, Madeline Paymer, Sudhir Kumar. 2015. Tree of li
#> [6] "Lack J.B., & Van den bussche R.A. 2010. Identifying the Confounding Factors in Resolving Phylog
#> [7] "Dumont E.R., Davalos L.M., Goldberg A., Santana S.E., Rex K., & Voigt C.C. 2012. Morphological
```

We have 7 studies in OpenTree with ages for the Chiroptera. The code above provided all references of the seven studies as an output.

To get the actual chronograms we need to run another function:

```
chiroptera_phylo_all <-  datelife::summarize_datelife_result(chiroptera_dr, summary_format = "phylo_all
# We will write this object into a file, bc it takes a long time to run
save(chiroptera_phylo_all, file="data/chiroptera_phylo_all.RData")
```

Now, we have to load it into the R work space so it is available for the next part

```
load("../data/chiroptera_phylo_all.RData")
```

The following function will allow us to plot the Tree with the ages.

```
datelife::plot_phylo_all(trees = chiroptera_phylo_all, write="pdf")
```

However, they are quite large, so we will not show them here for now.

Summarizing node ages is slow so we will save the output of `datelife::summarize_datelife_result` in the data folder. This function summarizes the node information from all the chronograms in chiroptera_phylo_all.

```
chiroptera_phylo_median <-  datelife::summarize_datelife_result(chiroptera_dr, summary_format = "phylo_

chiroptera_phylo_median
```

To plot the chronogram, Plotting Chronogram

```
ape::plot.phylo(chiroptera_phylo_median, cex = 1.2)
# Add the time axis:
ape::axisPhylo()
# And a little hack to add the axis name:
graphics::mtext("Time (myrs)", side = 1, line = 2, at = max(get("last_plot.phylo",envir = .PlotPhyloEnv
```

running python

** The Physcraper software allow to update a published phylogeny with new DNA sequences from GenBank. This can be your task for the fall if you are interested.**

Reproducibility

Do you want to reproduce this report yourself?

The following piece of code will render this report as pdf:

Attributions

bat image