

SSB 2020 Workshop: Using the Open Tree of Life for your Research

Get a dated tree for your *taxa of interest* (using R) with `datelife` and the Open Tree of Life

Luna L. Sánchez-Reyes

2019-12-30

Requirements for this tutorial

From R, install the packages `rotl`, `ape`, `devtools` and `stringr` with the function `install.packages()`, and the package `datelife` with the function `install_github()`. Load them into your workspace with `library()` or `require()`. If you do not want to load the packages, you can call functions specifying their package using two colons and the syntax `package_name::function_name()`. This implies more typing, but gives more clarity to reproduce the workflow later. So we will use that syntax for this tutorial. An exception to this are functions from packages that are “preloaded” (such as `library()` from `base` or `install.packages()`, from `utils`) that can be simply called by their name.

```
install.packages(c("rotl", "ape", "devtools", "stringr"))
library(rotl)
library(devtools)
library(stringr)
devtools::install_github("phylotastic/datelife")
library(datelife)
```

Find your taxa in the Open Tree of Life Taxonomy

Get the *ott ids* for your taxa with `tnrs_match_names()`. This will generate a “match_names” object.

```
my_taxa <- c("amphibians", "canis", "felis", "delphinidae", "spheniscidae")
resolved_names <- rotl::tnrs_match_names(my_taxa)
class(resolved_names)
> [1] "match_names" "data.frame"
resolved_names
>   search_string unique_name approximate_match ott_id is_synonym flags
> 1 amphibians    Amphibia          TRUE 544595      FALSE
> 2 canis         Canis          FALSE 372706      FALSE
> 3 felis         Felis          FALSE 563165      FALSE
> 4 delphinidae   Delphinidae     FALSE 698406      FALSE
> 5 spheniscidae  Spheniscidae     FALSE 494367      FALSE
>   number_matches
> 1              6
> 2              2
> 3              1
> 4              1
> 5              1
```

There are two ways to extract the *ott ids* from a “match_names” object:

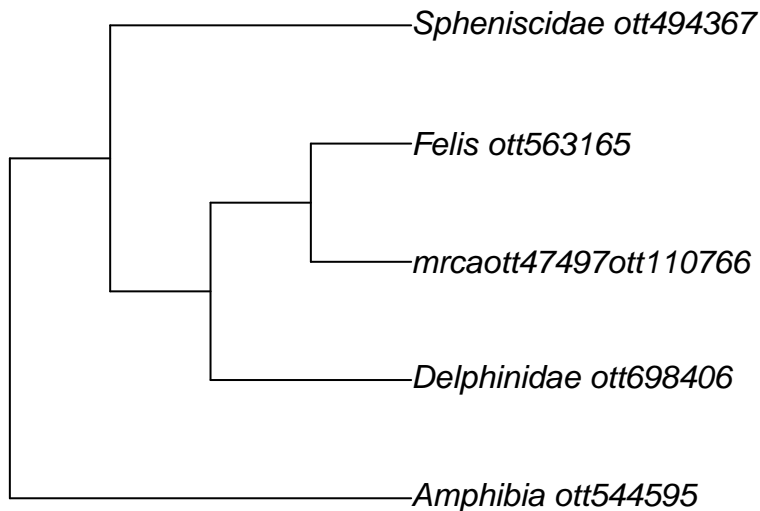
```
rotl::ott_id(resolved_names) # as a list
resolved_names$ott_id # as a vector
```

Get an induced subtree for your set of taxa

```
mytree <- rotl::tol_induced_subtree(resolved_names$ott_id)
> Warning in collapse_singles(tr, show_progress): Dropping singleton nodes
> with labels: Mammalia ott244265, Theria (subclass in Deuterostomia)
> ott229558, Eutheria (in Deuterostomia) ott683263, Boreoeutheria ott5334778,
> Laurasiatheria ott392223, mrcaott1548ott6790, mrcaott1548ott3607484,
> mrcaott1548ott4942380, mrcaott1548ott4942547, mrcaott1548ott3021, Artiodactyla
> ott622916, mrcaott1548ott21987, mrcaott1548ott5256, mrcaott5256ott4944931,
> Whippomorpha ott7655791, Cetacea ott698424, mrcaott5256ott3615450,
> mrcaott5256ott44568, Odontoceti ott698417, mrcaott5256ott5269,
> mrcaott5269ott6470, mrcaott5269ott47843, mrcaott47843ott194312,
> mrcaott4697ott263949, Carnivora ott44565, Caniformia ott827263,
> Canidae ott770319, mrcaott47497ott3612617, mrcaott47497ott3612529,
> mrcaott47497ott3612596, mrcaott47497ott3612516, mrcaott47497ott3612589,
> mrcaott47497ott3612591, mrcaott47497ott3612592, mrcaott47497ott77889,
> Feliformia ott827259, mrcaott6940ott19397, mrcaott19397ott194349, Felidae
> ott563159, mrcaott54737ott660452, mrcaott54737ott86170, mrcaott54737ott86175,
> mrcaott54737ott442049, mrcaott54737ott86162, mrcaott54737ott86166, Sauropsida
> ott639642, Sauria ott329823, mrcaott246ott4128455, mrcaott246ott4127082,
> mrcaott246ott4129629, mrcaott246ott4142716, mrcaott246ott4126667,
> mrcaott246ott1662, mrcaott246ott2982, mrcaott246ott31216, mrcaott246ott4947920,
> mrcaott246ott4127428, mrcaott246ott4126230, mrcaott246ott4127421,
> mrcaott246ott664349, mrcaott246ott4126505, mrcaott246ott4127015,
> mrcaott246ott4129653, mrcaott246ott4127541, mrcaott246ott4946623,
> mrcaott246ott4126482, mrcaott246ott4128105, mrcaott246ott4127288,
> mrcaott246ott4132146, mrcaott246ott3602822, mrcaott246ott4143599,
> mrcaott246ott3600976, mrcaott246ott4132107, Aves ott81461, Neognathae
> ott241846, mrcaott246ott5481, mrcaott246ott5021, mrcaott246ott7145,
> mrcaott246ott5272, mrcaott5272ott9830, mrcaott9830ott86672, mrcaott9830ott90560,
> mrcaott9830ott18206, mrcaott18206ott60413, Sphenisciformes ott494366
```

What does this warning mean?

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



This is cool!

Now, what if you want a synthetic tree containing some or all descendants from your taxa of interest?

Get a subtree of one taxon

We can extract a subtree of all descendants of one taxon at a time using the function `tol_subtree()` and the amphibians *ott id*.

Try to extract a subtree of all amphibians. Get the *ott id* first. It is already stored in the `resolved_names` object, but you can run the function `tnrs_match_names()` again if you want.

```
amphibia_ott_id <- resolved_names$ott_id[1] # extract the ott id from resolved_names
# amphibian_ott_id <- rotl::tnrs_match_names("amphibians")$ott_id # OR run tnrs again
```

Now, extract the subtree from the Open Tree synthetic tree using `tol_subtree()`.

```
amphibia_subtree <- rotl::tol_subtree(ott_id = amphibian_ott_id)
```

```
amphibia_subtree
>
> Phylogenetic tree with 10012 tips and 3100 internal nodes.
>
> Tip labels:
>   Odorrana_geminata_ott114, Odorrana_supranarina_ott14375, Odorrana_narina_ott14379, Odorrana_amamiensis_ott14380, ...
> Node labels:
>   Amphibia ott544595, Batrachia ott471197, Anura ott991547, , , , ...
>
> Unrooted; no branch lengths.
```

This is a large tree!

Try to extract a subtree for the genus *Canis*. It should be way smaller!

```
canis <- rotl::tnrs_match_names("canis") # get the ott id of the genus Canis (dogs)
canis
>   search_string unique_name approximate_match ott_id is_synonym flags
> 1         canis         Canis             FALSE 372706         FALSE
```

```

> number_matches
> 1                2

subtree <- rotl::tol_subtree(canis$ott_id) # get the subtree
> Error: HTTP failure: 400
> list(contesting_trees = list(`ot_278@tree1` = list(attachment_points = list(list(children_from_taxon
> mrca = "mrcaott47497ott110766")[/v3/tree_of_life/subtree] Error: node_id was not found (broken ta

```

Why am I getting an error saying that the *node id* was not found and that my taxon is “broken”??

Generally, this happens when phylogenetic information does not match taxonomic information. For example, extinct lineages are sometimes phylogenetically included within the taxon but are taxonomically excluded, making the taxon appear as paraphyletic. On the browser, you can still get to the subtree. From R, you need to do something else first.

Get a subtree of a “broken” taxon

There is a way to find out that the group is “broken” before trying to get the subtree and getting an error.

```
rotl::is_in_tree(canis$ott_id)
```

We say that a taxon is not in the tree when its *ott id* is not assigned to a node in the synthetic tree. This is the reason why we get an error when we try to get a subtree using the *ott id*.

Then, how do I get a subtree of my “broken” taxon? There are some options.

a) Get a subtree using the *node id* instead of the *ott id*

rotl has a function that gets for you all info from the node containing a taxon. That includes the actual *node id*.

```

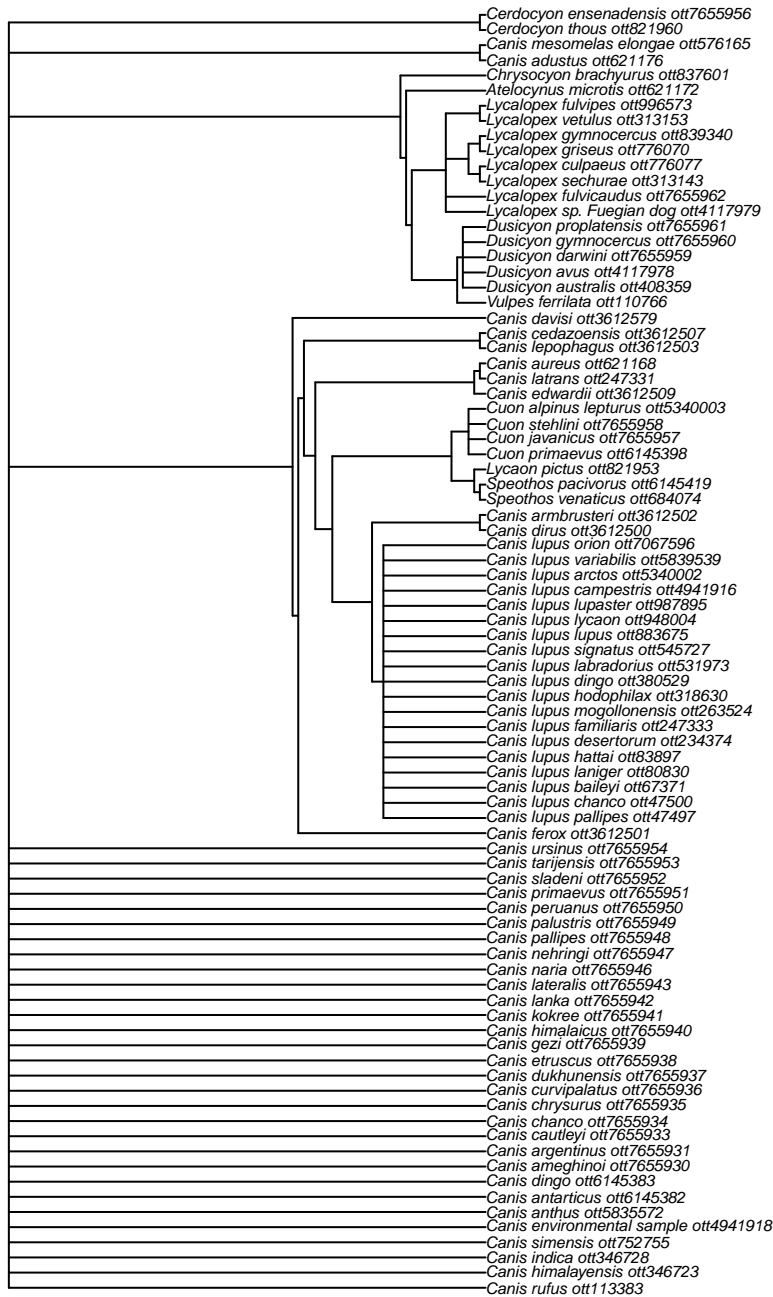
canis_node_info <- rotl::tol_node_info(canis$ott_id)
canis_node_info
>
> OpenTree node.
>
> Node id: mrcaott47497ott110766
> Number of terminal descendants: 85
> Is taxon: FALSE

```

The *node* that contains *Canis* is “mrcaott47497ott110766”. We can use it to get a subtree with `tol_subtree()`

```
canis_node_subtree <- rotl::tol_subtree(node_id = canis_node_info$node_id)
```

```
ape::plot.phylo(canis_node_subtree, cex = 0.5)
```



Nice! We got a subtree of 85 tips, containing all descendants from the node that also contains *Canis*.

This includes species assigned to genera other than *Canis*.

What if I *really, really* want to get a tree containing species within the genus *Canis* only?

b) Get an induced subtree of taxonomic children

We can get the *ott ids* of the taxonomic children of our taxon of interest and use the function `tol_induced_subtree()`.

First, get the taxonomic children.

```
canis_taxonomy <- rotl::taxonomy_subtree(canis$ott_id)
```

```
canis_taxonomy
```

```
$tip_label [1] "Canis_dirus_ott3612500"  
[2] "Canis_anthus_ott5835572"  
[3] "Canis_rufus_ott113383"  
[4] "Canis_simensis_ott752755"  
[5] "Canis_aureus_ott621168"  
[6] "Canis_mesomelas_elongae_ott576165"  
[7] "Canis_adustus_ott621176"  
[8] "unclassified_Canis_ott7655955"  
[9] "Canis_latrans_ott247331"  
[10] "Canis_lupus_baileyi_ott67371"  
[11] "Canis_lupus_laniger_ott80830"  
[12] "Canis_lupus_orion_ott7067596"  
[13] "Canis_lupus_hodophilax_ott318630"  
[14] "Canis_lupus_signatus_ott545727"  
[15] "Canis_lupus_arctos_ott5340002"  
[16] "Canis_lupus_mogollonensis_ott263524"  
[17] "Canis_lupus_variabilis_ott5839539"  
[18] "Canis_lupus_lupus_ott883675"  
[19] "Canis_lupus_campestris_ott4941916"  
[20] "Canis_lupus_lycaon_ott948004"  
[21] "Canis_lupus_pallipes_ott47497"  
[22] "Canis_lupus_chancho_ott47500"  
[23] "Canis_lupus_x_Canis_lupus_familiaris_ott4941915" [24] "Canis_lupus_desertorum_ott234374"  
[25] "Canis_lupus_familiaris_ott247333"  
[26] "Canis_lupus_dingo_ott380529"  
[27] "Canis_lupus_labradorius_ott531973"  
[28] "Canis_lupus_hattai_ott83897"  
[29] "Canis_lupus_lupaster_ott987895"  
[30] "Canis_himalayensis_ott346723"  
[31] "Canis_indica_ott346728"  
[32] "Canis_environmental_samples_ott4941917"  
[33] "Canisssp.KEB-2016ott5925604"  
[34] "Canis_sp._CANInt1_ott470950"  
[35] "'Canisssp.Russia/33"  
[36] "500ott5338950"  
[37] "Canis_sp._ott247325"  
[38] "'Canisssp.Belgium/36"  
[39] "000ott5338951"  
[40] "Canis_environmental_sample_ott4941918"  
[41] "Canis_morenis_ott6145387"  
[42] "Canis_niger_ott6145388"  
[43] "Canis_palaeoplatensis_ott6145390"  
[44] "Canis_osorum_ott6145389"  
[45] "Canis_thooides_ott6145392"  
[46] "Canis_antarcticus_ott6145381"  
[47] "Canis_proplatensis_ott6145391"  
[48] "Canis_feneus_ott6145384"  
[49] "Canis_geismarianus_ott6145385"  
[50] "Canis_ameghinoi_ott7655930"
```

```

[51] "Canis_nehringi_ott7655947"
[52] "Canis_palustris_ott7655949"
[53] "Canis_lanka_ott7655942"
[54] "Canis_pallipes_ott7655948"
[55] "Canis_gezi_ott7655939"
[56] "Canis_montanus_ott7655945"
[57] "Canis_primaevus_ott7655951"
[58] "Canis_chrysurus_ott7655935"
[59] "Canis_dukhunensis_ott7655937"
[60] "Canis_kokree_ott7655941"
[61] "Canis_sladeni_ott7655952"
[62] "Canis_himalaicus_ott7655940"
[63] "Canis_chanco_ott7655934"
[64] "Canis_curvipalatus_ott7655936"
[65] "Canis_lateralis_ott7655943"
[66] "Canis_argentinus_ott7655931"
[67] "Canis_tarijensis_ott7655953"
[68] "Canis_naria_ott7655946"
[69] "Canis_peruanus_ott7655950"
[70] "Canis_cautleyi_ott7655933"
[71] "Canis_ursinus_ott7655954"
[72] "Canis_armbrusteri_ott3612502"
[73] "Canis_ferox_ott3612501"
[74] "Canis_lepophagus_ott3612503"
[75] "Canis_edwardii_ott3612509"
[76] "Canis_apolloniensis_ott3612508"
[77] "Canis_cedazoensis_ott3612507"
[78] "Canis_primigenius_ott3612506"
[79] "Canis_lydekkeri_ott7655944"
[80] "Canis_arnensis_ott7655932"
[81] "Canis_antarticus_ott6145382"
[82] "Canis_dingo_ott6145383"
[83] "Canis_etruscus_ott7655938"
[84] "Canis_spelaesus_ott3612504"

```

\$edge_label [1] "Canis_mesomelas_ott666235" "Canis_lupus_ott247341"

[3] "Canis_ott372706"

Now, extract the *ott_ids*.

```

canis_taxonomy_ott_ids <- datelife::extract_ott_ids(x = canis_taxonomy$tip_label)
> After extracting ott ids, there are some non numeric elements:
> Canissp.KEB-2016ott5925604
> 'Canissp.Russia/33
> 500ott5338950'
> 'Canissp.Belgium/36
> 000ott5338951'
>
> NAs removed.

```

Try to get an induced subtree of *Canis* taxonomic children.

```

canis_taxonomy_subtree <- rotl::tol_induced_subtree(canis_taxonomy_ott_ids)
> Error: HTTP failure: 400
> [/v3/tree_of_life/induced_subtree] Error: node_id 'ott3612504' was not found!list(ott247325 = "pruned,
> ott7655944 = "pruned_ott_id", ott7655945 = "pruned_ott_id", ott7655955 = "pruned_ott_id")

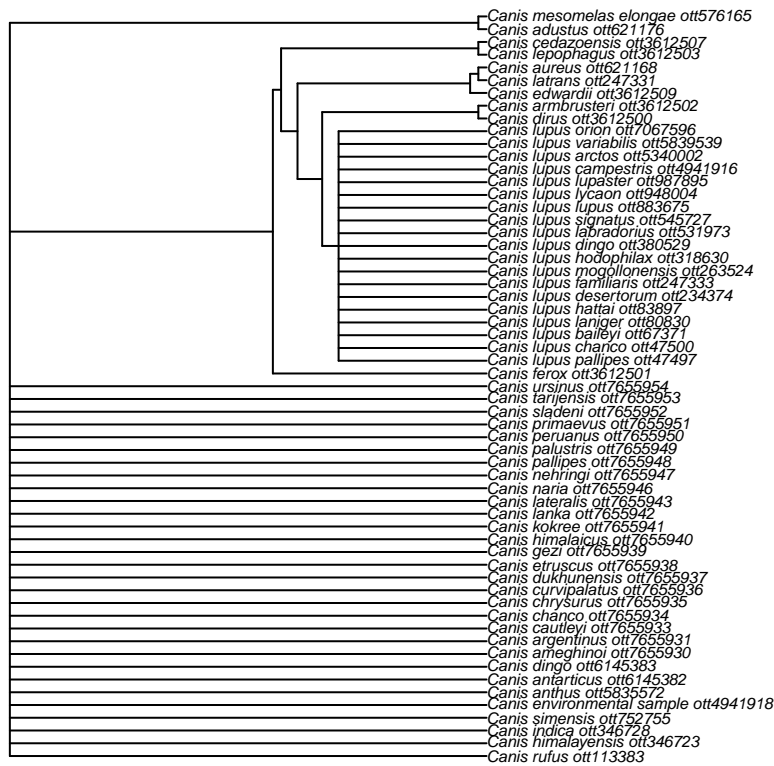
```

It is often not possible to get an induced subtree of all taxonomic children from a taxon, because some of them are not in the tree.

To verify which ones are giving us trouble, we can use the function `is_in_tree()` again.

```
canis_in_tree <- sapply(canis_taxonomy_ott_ids, rotl::is_in_tree)
canis_taxonomy_ott_ids_intree <- canis_taxonomy_ott_ids[canis_in_tree]
canis_taxonomy_subtree <- rotl::tol_induced_subtree(canis_taxonomy_ott_ids_intree)
```

```
ape::plot.phylo(canis_taxonomy_subtree, cex = 0.5)
```



There! We have a synthetic subtree (derived from phylogenetic information) containing only the taxonomic children of *Canis*.

It is sometimes thought that flags can help to detect taxa that are not in the tree, but they can be misleading. Get the flags of *Canis* taxonomic children.

```
canis_taxonomy_ott_ids_taxon_info <- rotl::taxonomy_taxon_info(canis_taxonomy_ott_ids)
canis_taxonomy_ott_ids_taxon_info_flags <- rotl::flags(canis_taxonomy_ott_ids_taxon_info)
in_tree <- sapply(canis_taxonomy_ott_ids, rotl::is_in_tree)
flags_not_in_tree <- canis_taxonomy_ott_ids_taxon_info_flags[!in_tree]
flags_not_in_tree <- unique(unlist(flags_not_in_tree))
flags_in_tree <- canis_taxonomy_ott_ids_taxon_info_flags[in_tree]
flags_in_tree <- unique(unlist(flags_in_tree))
```

```
flags_not_in_tree
> [1] "not_otu"          "was_container"    "hybrid"          "infraspecific"
> [5] "incertae_sedis"  "extinct"         "hidden"
flags_in_tree
> [1] "infraspecific" "extinct"
```

Some terms such as “infraspecific” and “extinct” are used to flag both taxa that are **not** in the tree and taxa that **are** in the tree.

Get a subtree of a rank

We can use the function `tax_rank()` to extract all children at or below the species level. First, get all ranks.

```
canis_ranks <- rotl::tax_rank(canis_taxonomy_ott_ids_taxon_info)
```

Then, get your desired rank only.

```
canis_species_index <- canis_ranks %in% "species"  
canis_species_in_tree <- canis_taxonomy_ott_ids[canis_species_index & canis_in_tree]
```

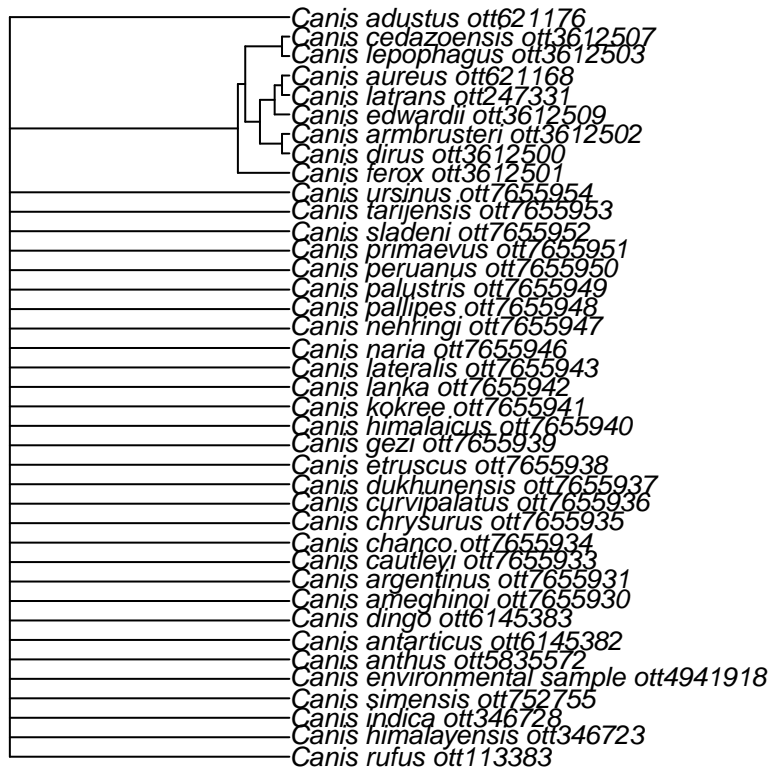
Get an induced subtree of these *ott ids*.

```
canis_species_subtree <- rotl::tol_induced_subtree(canis_species_in_tree)
```

```
canis_species_subtree
```

```
>  
> Phylogenetic tree with 39 tips and 8 internal nodes.  
>  
> Tip labels:  
> Canis_dirus_ott3612500, Canis_armbrusteri_ott3612502, Canis_latrans_ott247331, Canis_aureus_ott621176  
> Node labels:  
> mrcaott47497ott110766, mrcaott47497ott3612501, mrcaott47497ott3612503, mrcaott47497ott247331, mrcaott47497ott621176  
>  
> Unrooted; no branch lengths.
```

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



What if we want to get all taxa of a taxonomic rank above the species level? Try to get all amphibian families.

Extract ott ids from tip labels of subtree of all amphibians.

```
amphibia_subtree_ott_ids <- datelife::extract_ott_ids(x = amphibia_subtree$tip.label)
```

Get their taxon information. This might take about 10 minutes or so.

```
amphibia_ott_ids_taxon_info <- rotl::taxonomy_taxon_info(amphibia_subtree_ott_ids)
```

We already know that all children of amphibia are in the tree. So we only need to extract the ones belonging to our desired rank.

```
amphibia_ranks <- rotl::tax_rank(amphibia_subtree_ott_ids_taxon_info)
unique(amphibia_ranks)
# [[1]]
# [1] "species"
#
# [[2]]
# [1] "subspecies"
#
# [[3]]
# [1] "no rank - terminal"
```

We have implemented a function in datelife to extract all taxa of a given rank. It is also slightly faster.

```
amphibia_families <- datelife::get_ott_children(ott_ids = amphibia_ott_id, ott_rank = "family")
```

Now, get the subtree of families only.

```
amphibia_families_subtree <- rotl::tol_induced_subtree(amphibia_families)
> Error in FUN(X[[i]], ...): only 1 element should be provided
```

```
ape::plot.phylo(amphibia_families_subtree, cex = 0.5)
> Error in ape::plot.phylo(amphibia_families_subtree, cex = 0.5): object 'amphibia_families_subtree' not found
amphibia_families_subtree
> Error in eval(expr, envir, enclos): object 'amphibia_families_subtree' not found
```

Get branch length information (proportional to time) for you taxa

You can explore the studies supporting the synthetic tree, and find out which ones have branch lengths.

In the package datelife, we have implemented a workflow that extracts all studies containing information from at least two taxa. It only works at the species level.

```
apes <- c("Pongo", "Pan", "Gorilla", "Hoolock", "Homo")
resolved_names <- rotl::tnrs_match_names(apes)
```

```
apes_dr <- datelife::get_datelife_result(resolved_names$unique_name, get_spp_from_taxon = TRUE)
```

```
names(apes_dr)[1]
> [1] "Bininda-Emonds, Olaf R. P., Marcel Cardillo, Kate E. Jones, Ross D. E. MacPhee, Robin M. D. Beck"
apes_dr[[1]] # look at the first element of the list
>
> Gorilla gorilla Homo sapiens Pan paniscus Pan troglodytes
> Gorilla gorilla      0.0      25.4      25.4      25.4
> Homo sapiens        25.4      0.0      19.4      19.4
> Pan paniscus        25.4      19.4      0.0      7.8
> Pan troglodytes     25.4      19.4      7.8      0.0
> Pongo pygmaeus      39.4      39.4      39.4      39.4
> Hoolock hoolock     47.0      47.0      47.0      47.0
```

```

>                                Pongo pygmaeus Hoolock hoolock
> Gorilla gorilla                39.4                47
> Homo sapiens                   39.4                47
> Pan paniscus                   39.4                47
> Pan troglodytes                39.4                47
> Pongo pygmaeus                 0.0                47
> Hoolock hoolock                47.0                0
names(apes_dr)[length(apes_dr)]
> [1] "Springer, Mark S., Robert W. Meredith, John Gatesy, Christopher A. Emerling, Jong Park, Daniel L.
apes_dr[[length(apes_dr)]] # look at the last element of the list
>                                Pongo abelii Pongo pygmaeus Pan troglodytes Pan paniscus
> Pongo abelii                   0.0000                1.6408                24.1506                24.1506
> Pongo pygmaeus                 1.6408                0.0000                24.1506                24.1506
> Pan troglodytes                24.1506                24.1506                0.0000                2.0326
> Pan paniscus                  24.1506                24.1506                2.0326                0.0000
> Homo sapiens                  24.1506                24.1506                10.8756                10.8756
> Gorilla beringei              24.1506                24.1506                12.6052                12.6052
> Gorilla gorilla               24.1506                24.1506                12.6052                12.6052
> Hoolock hoolock               27.2042                27.2042                27.2042                27.2042
> Hoolock leuconedys            27.2042                27.2042                27.2042                27.2042
>                                Homo sapiens Gorilla beringei Gorilla gorilla
> Pongo abelii                   24.1506                24.1506                24.1506
> Pongo pygmaeus                 24.1506                24.1506                24.1506
> Pan troglodytes               10.8756                12.6052                12.6052
> Pan paniscus                  10.8756                12.6052                12.6052
> Homo sapiens                   0.0000                12.6052                12.6052
> Gorilla beringei              12.6052                0.0000                1.8636
> Gorilla gorilla               12.6052                1.8636                0.0000
> Hoolock hoolock               27.2042                27.2042                27.2042
> Hoolock leuconedys            27.2042                27.2042                27.2042
>                                Hoolock hoolock Hoolock leuconedys
> Pongo abelii                   27.2042                27.2042
> Pongo pygmaeus                 27.2042                27.2042
> Pan troglodytes                27.2042                27.2042
> Pan paniscus                   27.2042                27.2042
> Homo sapiens                   27.2042                27.2042
> Gorilla beringei              27.2042                27.2042
> Gorilla gorilla               27.2042                27.2042
> Hoolock hoolock                0.0000                2.3368
> Hoolock leuconedys            2.3368                0.0000

```

We have now a list of matrices storing time of lineage divergence data for all taxon pairs. Lists are named by the study citation, so we have that information handy at all times.

We can summarize it.

```

apes_phylo_median <- datelife::summarize_datelife_result(apes_dr,summary_format = "phylo_median")
apes_phylo_median

```

Phylogenetic tree with 8 tips and 7 internal nodes.

Tip labels: Homo_sapiens, Pan_paniscus, Gorilla_beringei, Gorilla_gorilla, Pongo_abelii, Pongo_pygmaeus, ... Node labels: n1, n2, n3, n4, n5, n6, ...

Rooted; includes branch lengths.

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

