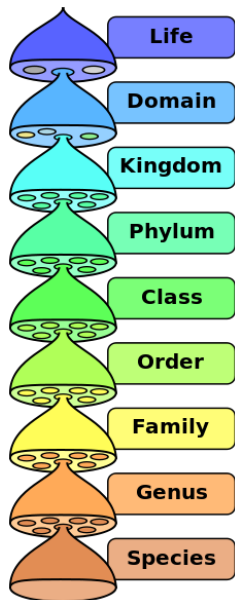# Taxa and metacoder: R packages for parsing, visualization, and manipulation of taxonomic data

Zachary Foster, Scott Chamberlain, Thomas Sharpton, and Niklaus Grünwald
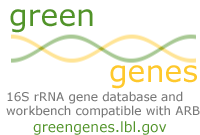
The challenges of taxonomic data

Life, Domain, Kingdom, Phylum, Class, Order, Family, Genus, Species

- ▸ Taxonomic data is hierarchical
- ▸ Associated with tabular data
- ▸ Can be names, classifications, or IDs
- ▸ Many different taxonomic systems
- ▸ Many different data formats
- ▸ Hierarchical visualization is difficult

Sources of taxonomic data

DNA sequence databases

Species occurrence databases

Museum records

Sources of taxonomic data: DNA sequences

NCBI Genbank

```
>AC073210.8 Homo sapiens BAC clone RP11-460N20 from 7, complete sequence
AACGAACGCTGGCGGCATGCCTAACACATGCAAGTCGAACGAGACCTTCGGGTC...
```

UNITE

```
>SH099456.05FU_FJ357315_refs k__Fungi;p__Ascomycota;c__Dothideomycetes
;o__Pleosporales;f__Pleosporaceae;g__Embellisia;s__Embellisia_planifunda
GCTGGCGGCGTGCCTAACACATGTAAGTCGAACGGGACTGGGGGCAACTCCAGT...
```

RDP

```
>S000448483 Sparassis crispa; MBUH-PIRJO&ILKKA94-1587/ss5
Lineage=Root;rootrank ;Fungi;domain;Basidiomycota;phylum;Agaricomycetes;
class;Polyporales;order ;Sparassidaceae;family;Sparassis;genus
AGAGTTTGATCCTGGCTCAGATTGAACGCTGGCGGAATGCTTAACACATGAAAC...
```

SILVA

```
>GCVF01000431.1.2369 Bacteria;Proteobacteria;Gammaproteobacteria;
Oceanospirillales;Alcanivoraceae;Alcanivorax;Thalassiosira rotula
AGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGTATGCTTAACACATGCAAG...
```

Sources of taxonomic data: Occurrence records

Global Biodiversity Information Facility : Archaea database

```
readr::read_tsv("datasets/gbif_archea.csv")[4:8]
```

```
# A tibble: 19,013 x 5
   kingdom phylum        class        order            family
   <chr>   <chr>         <chr>        <chr>            <chr>
 1 Archaea Euryarchaeota Halobacteria Halobacteriales  Halobacteriaceae
 2 Archaea Euryarchaeota Thermococci  Thermococcales   Thermococcaceae
 3 Archaea Euryarchaeota Thermococci  Thermococcales   Thermococcaceae
 4 Archaea Crenarchaeota Thermoprotei Desulfurococcales Desulfurococcaceae
 5 Archaea Crenarchaeota Thermoprotei Desulfurococcales Pyrodictiaceae
 6 Archaea Crenarchaeota Thermoprotei Thermoproteales  Thermoproteaceae
 7 Archaea Euryarchaeota Thermococci  Thermococcales   Thermococcaceae
 8 Archaea Euryarchaeota Thermococci  Thermococcales   Thermococcaceae
 9 Archaea Euryarchaeota Halobacteria Halobacteriales  Halobacteriaceae
10 Archaea Euryarchaeota Halobacteria Halobacteriales  Halobacteriaceae
# ... with 19,003 more rows
```

Sources of taxonomic data: Museum records

Smithsonian Museum of Natural History: Mammal database

```
readr::read_csv("datasets/SNMNH.csv")[9]
```

```
# A tibble: 5,000 x 1
   `Name Hierarchy`
   <chr>
 1 Abditomys latidens : Muridae : Rodentia : Mammalia : Chordata
 2 Abrawayaomys ruschii : Cricetidae : Rodentia : Mammalia : Chordata
 3 Abrawayaomys ruschii : Cricetidae : Rodentia : Mammalia : Chordata
 4 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
 5 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
 6 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
 7 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
 8 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
 9 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
10 Abrocoma bennettii bennettii : Abrocomidae : Rodentia : Mammalia : Chordata
# ... with 4,990 more rows
```
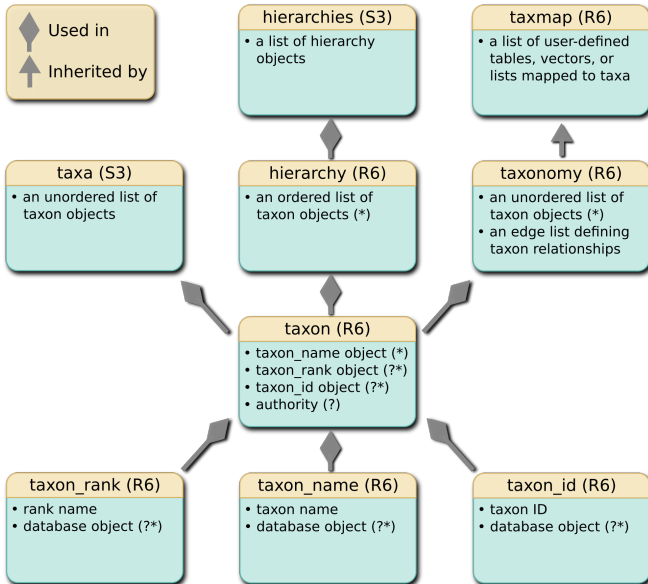
The `taxa` package

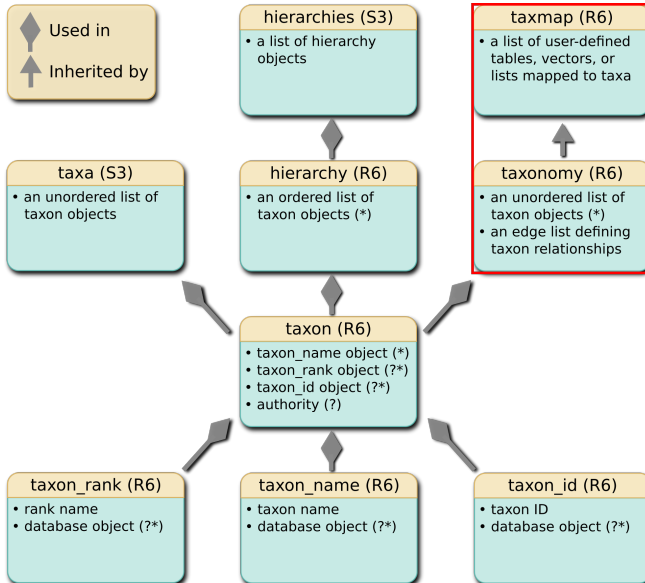The `taxa` package is designed to be a solid foundation for using taxonomic data in R.

- R6 classes to hold taxa, taxonomies, and associated data

- Flexible parsers to convert raw data to these classes

- Dplyr-inspired functions to manipulate these classes

- Functions to get data associated with each taxon in a taxonomy

The `metacoder` package: visualization of taxon data

```r
readr::read_tsv("datasets/gbif_archea.csv") %>%
  parse_tax_data(class_cols = 4:8) %>%
  filter_taxa(taxon_names != "") %>%
  heat_tree(node_label = taxon_names, node_color = n_obs,
            node_size = n_obs, layout = "da")
```

Classes defined by 'taxa'

Classes defined by `taxa`: Relationships

Classes defined by `taxa`: Relationships

Used in

Inherited by

hierarchies (S3)
- a list of hierarchy objects

taxmap (R6)
- a list of user-defined tables, vectors, or lists mapped to taxa

taxa (S3)
- an unordered list of taxon objects

hierarchy (R6)
- an ordered list of taxon objects (*)

taxonomy (R6)
- an unordered list of taxon objects (*)
- an edge list defining taxon relationships

taxon (R6)
- taxon_name object (*)
- taxon_rank object (?*)
- taxon_id object (?*)
- authority (?)

taxon_rank (R6)
- rank name
- database object (?*)

taxon_name (R6)
- taxon name
- database object (?*)

taxon_id (R6)
- taxon ID
- database object (?*)

Classes defined by `taxa`: The `taxmap` class

```
> ex_taxmap
<Taxmap>
  17 taxa: b. Mammalia, c. Plantae, d. Felidae ... p. sapiens, q. lycopersicum, r. tuberosum
  17 edges: NA->b, NA->c, b->d, b->e, b->f, c->g, d->h ... h->m, i->n, j->o, k->p, l->q, l->r
  4 data sets:
    info:
      # A tibble: 6 x 4
        taxon_id name   n_legs dangerous
        <chr>    <chr>   <dbl> <lgl>
      1 m        tiger      4. TRUE
      2 n        cat        4. FALSE
      3 o        mole       4. FALSE
      # ... with 3 more rows
    phylopic_ids: a named vector of 'character' with 6 items
        m. e148eabb-f138-43c6-b1e4-5cda2180485a ... r. 63604565-0406-460b-8cb8-1abe954b3f3a
    foods: a list of 6 items named by taxa:
        m, n, o, p, q, r
    abund:
      # A tibble: 8 x 5
        taxon_id code  sample_id count taxon_index
        <chr>    <fct> <fct>     <dbl>       <int>
      1 m        T     A            1.           1
      2 n        C     A            2.           2
      3 o        M     B            5.           3
      # ... with 5 more rows
```

Parsing

Parsing

Input data format

Input type

	Simple	Embedded	Raw string
| | ```
> print(data)
[1] "input_1" "input_2"
[3] "input_3"
``` | ```
> print(data)
   x  input   y
1 a input_1 100
2 b input_2 200
3 c input_3 300
``` | ```
> print(data)
[1] ">id:a-tax:input_1"
[2] ">id:b-tax:input_2"
[3] ">id:c-tax:input_3"
``` |
| **Classification**<br>Primates;Hominidae;Homo;sapiens | ```
> print(data)
[1] "Primates;Hominidae;Hom..."
[2] "Primates;Haplorhini;Cr..."

> parse_tax_data(data,
    class_sep = ";")
``` | ```
> print(data)
   x                class  y
1 a Primates;Hominidae;... 100
2 b Primates;Haplorhini;.. 200

> parse_tax_data(data,
    class_cols = "class",
    class_sep = ";")
``` | ```
> print(data)
[1] ">id:a-tax:Primates;Hom..."
[2] ">id:b-tax:Primates;Hap..."

> extract_tax_data(data,
    regex = ">id:(.+)-tax:(.+)",
    key = c("info", "class"),
    class_sep = ";")
``` |
| **Taxon ID**<br>9606 | ```
> print(data)
[1] "9606" "100937" ...

> lookup_tax_data(data,
    type = "taxon_id")
``` | ```
> print(data)
   x     id    y
1 a 9606   100
2 b 100937 200

> lookup_tax_data(data,
    type = "taxon_id",
    column = "id")
``` | ```
> print(data)
[1] ">id:a-tax:9606"
[2] ">id:b-tax:100937"

> extract_tax_data(data,
    regex = ">id:(.+)-tax:(.+)",
    key = c("info", "taxon_id"),
    database = "ncbi")
``` |
| **Taxon name**<br>Homo sapiens | ```
> print(data)
[1] "Homo sapiens"
[2] "Primates" ...

> lookup_tax_data(data,
    type = "taxon_name")
``` | ```
> print(data)
   x       name    y
1 a Homo sapiens 100
2 b Primates     200

> lookup_tax_data(data,
    type = "taxon_name",
    column = "name")
``` | ```
> print(data)
[1] ">id:a-tax:Homo sapiens"
[2] ">id:b-tax:Primates"

> extract_tax_data(data,
    regex = ">id:(.+)-tax:(.+)",
    key = c("info","taxon_name"),
    database = "ncbi")
``` |
| **Sequence ID**<br>AC073210 | ```
> print(data)
[1] "AC073210" "KC312885" ...

> lookup_tax_data(data,
    type = "seq_id")
``` | ```
> print(data)
   x  ncbi_id   y
1 a AC073210 100
2 b KC312885 200

> lookup_tax_data(data,
    type = "seq_id",
    column = "ncbi_id")
``` | ```
> print(data)
[1] ">id:a-tax:AC073210"
[2] ">id:b-tax:KC312885"

> extract_tax_data(data,
    regex = ">id:(.+)-tax:(.+)",
    key = c("info","seq_id"),
    database = "ncbi")
``` |

Parsing: vectors of classifications

```r
x <- c("Mammalia;Theria;Metatheria;Diprotodontia;Macropodiformes",
       "Mammalia;Theria;Eutheria;Primates;Haplorrhini;Simiiformes")

parse_tax_data(x, class_sep = ";")
```

```
<Taxmap>
  9 taxa: b. Mammalia, c. Theria ... j. Simiiformes
  9 edges: NA->b, b->c, c->d, c->e, d->f, e->g, f->h, g->i, i->j
  1 data sets:
    tax_data: a named vector of 'character' with 2 items
      h. Mammalia;Theria;[truncated] ... j. Mammalia;Theria;[truncated]
  0 functions:
```

Parsing: vectors of names

```
x <- c("Homo sapiens", "Macropus", "Chordata")

lookup_tax_data(x, type = "taxon_name", database = "ncbi")
```

```
<Taxmap>
  35 taxa: 131567. cellular organisms ... 9606. Homo sapiens
  35 edges: NA->131567, 131567->2759 ... 207598->9605, 9605->9606
  2 data sets:
    tax_data:
      # A tibble: 35 x 4
        taxon_id ncbi_name         ncbi_rank     ncbi_id
        <chr>    <chr>             <chr>         <chr>
      1 131567   cellular organisms no rank       131567
      2 2759     Eukaryota         superkingdom 2759
      3 33154    Opisthokonta      no rank       33154
      # ... with 32 more rows
    query_data: a named vector of 'character' with 3 items
        9606. Homo sapiens, 9312. Macropus, 7711. Chordata
  0 functions:
```

Parsing: vectors of taxon or sequence IDs

Taxon IDs

```r
x <- c("9606", "207598", "7711") # NCBI taxon IDs
lookup_tax_data(x, type = "taxon_id", database = "ncbi")
```

Sequence IDs

```r
x <- c("AC073210", "MG014608", "AE006468") # NCBI sequence IDs
lookup_tax_data(x, type = "seq_id", database = "ncbi")
```

Parsing: tables

Global Biodiversity Information Facility : Archaea database

```r
readr::read_tsv("datasets/gbif_archea.csv")[4:8]
```

```
# A tibble: 19,013 x 5
   kingdom phylum        class        order             family
   <chr>   <chr>         <chr>        <chr>             <chr>
 1 Archaea Euryarchaeota Halobacteria Halobacteriales   Halobacteriaceae
 2 Archaea Euryarchaeota Thermococci  Thermococcales    Thermococcaceae
 3 Archaea Euryarchaeota Thermococci  Thermococcales    Thermococcaceae
 4 Archaea Crenarchaeota Thermoprotei Desulfurococcales Desulfurococcaceae
 5 Archaea Crenarchaeota Thermoprotei Desulfurococcales Pyrodictiaceae
 6 Archaea Crenarchaeota Thermoprotei Thermoproteales   Thermoproteaceae
 7 Archaea Euryarchaeota Thermococci  Thermococcales    Thermococcaceae
 8 Archaea Euryarchaeota Thermococci  Thermococcales    Thermococcaceae
 9 Archaea Euryarchaeota Halobacteria Halobacteriales   Halobacteriaceae
10 Archaea Euryarchaeota Halobacteria Halobacteriales   Halobacteriaceae
# ... with 19,003 more rows
```

Parsing: tables

```
x = readr::read_tsv("datasets/gbif_archea.csv")

parse_tax_data(x, class_cols = 4:8)
```

```
<Taxmap>
  95 taxa: ab. Archaea, ac. Euryarchaeota ... dr. Methermicoccaceae
  95 edges: NA->ab, ab->ac, ab->ad ... aw->dp, at->dq, ax->dr
  1 data sets:
    tax_data:
      # A tibble: 19,013 x 45
        taxon_id gbifid datasetkey occurrenceid kingdom phylum class
        <chr>      <dbl> <chr>       <chr>         <chr>   <chr>  <chr>
      1 br       1.84e9 3c6e7390-~ D7C42A39-3A~ Archaea Eurya~ Halo~
      2 bs       1.83e9 863efcc4-~ <NA>          Archaea Eurya~ Ther~
      3 bs       1.83e9 863efcc4-~ <NA>          Archaea Eurya~ Ther~
      # ... with 1.901e+04 more rows, and 38 more variables:
      #   order <chr>, family <chr>, genus <chr>, species <chr>,
      #   infraspecificepithet <lgl>, taxonrank <chr>,
      #   scientificname <chr>, countrycode <chr>, locality <chr>,
      #   publishingorgkey <chr>, ...
  0 functions:
```

Parsing: complex strings (NCBI Genbank)

```
x = c("AC073210.8 Homo sapiens BAC clone RP11-460N20 from 7, complete sequence"
      "AE006468.2 Salmonella enterica subsp. enterica serovar Typhimurium",
      "MG014608.1 Macropus fuliginosus Csf1r gene, enhancer")

extract_tax_data(x, database = "ncbi", regex = "([A-Z0-9.]+) (.+)",
                 key = c(my_ncbi_id = "seq_id", my_desc = "info"))
```

```
<Taxmap>
  46 taxa: 131567. cellular organisms ... 9606. Homo sapiens
  46 edges: NA->131567, 131567->2759 ... 207598->9605, 9605->9606
  2 data sets:
    tax_data:
      # A tibble: 46 x 4
        taxon_id ncbi_name         ncbi_rank      ncbi_id
        <chr>    <chr>             <chr>          <chr>
      1 131567   cellular organisms no rank        131567
      2 2759     Eukaryota         superkingdom   2759
      3 33154    Opisthokonta      no rank        33154
      # ... with 43 more rows
    query_data:
      # A tibble: 3 x 4
        taxon_id my_ncbi_id my_desc           input
        <chr>    <chr>      <chr>             <chr>
      1 9606     AC073210.8 Homo sapiens BAC cl~ AC073210.8 Homo sap~
```

Taxon attributes
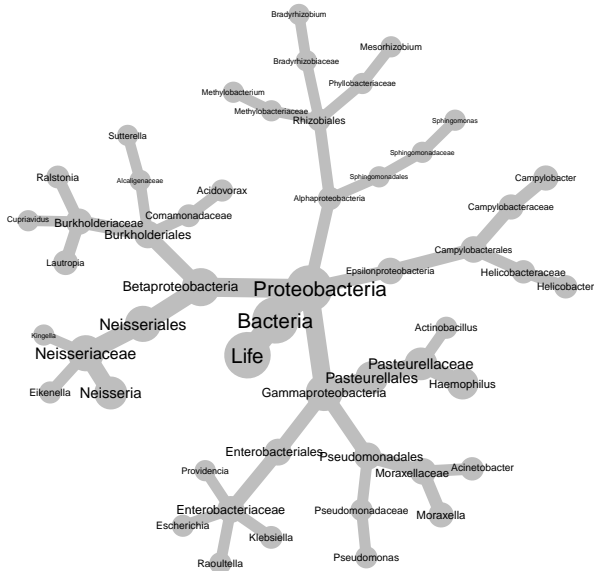
Taxon attributes: subtaxa and supertaxa

Subtaxa (recursive = T)

Subtaxa (recursive = F)
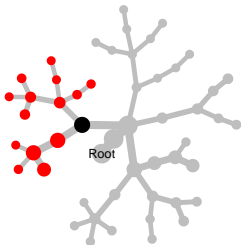
Supertaxa (recursive = T)
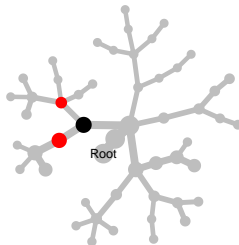
Supertaxa (recursive = F)
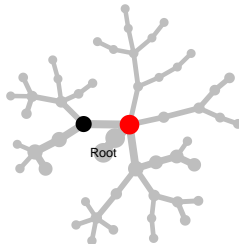
Root

Taxon attributes: parts of a tree

Leaves
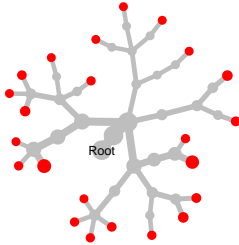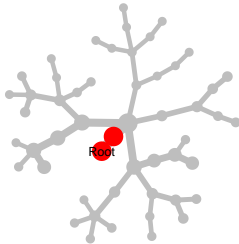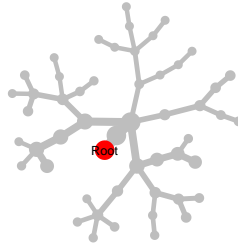
Roots
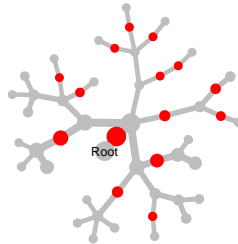
Stems

Internodes

Taxon attributes: functions

Ranks, names, and IDs

taxon_names, taxon_ranks, taxon_ids

Parts of the tree

branches, internodes, leaves, roots, stems, supertaxa,
subtaxa

Filtering helpers

is_branch, is_internode, is_leaf, is_root, is_stem

Numbers of supertaxa/subtaxa/data

n_supertaxa, n_subtaxa, n_obs, n_supertaxa_1,
n_subtaxa_1, n_obs_1

Taxon attributes: Ranks, names, and IDs

These are derived from the list of `taxon` objects.

```
taxon_names(ex_taxmap) %>% head
```

```
          b           c           d           e           f
  "Mammalia"    "Plantae"    "Felidae" "Notoryctidae" "Hominidae"
          g
"Solanaceae"
```

```
taxon_ranks(ex_taxmap) %>% head
```

```
      b        c        d        e        f        g
 "class" "kingdom" "family" "family" "family" "family"
```

```
taxon_ids(ex_taxmap) %>% head
```

```
  b   c   d   e   f   g
"b" "c" "d" "e" "f" "g"
```

Taxon attributes: `subtaxa`

These return a list of vectors named by taxon IDs.

```
subtaxa(ex_taxmap, value = "taxon_names")[1:3]
```

```
$b
            d              h              m              i              n
    "Felidae"     "Panthera"      "tigris"        "Felis"        "catus"
            e              j              o              f              k
"Notoryctidae"  "Notoryctes"    "typhlops"    "Hominidae"         "homo"
            p
    "sapiens"

$c
            g              l              q              r
 "Solanaceae"      "Solanum" "lycopersicum"   "tuberosum"

$d
          h          m          i          n
 "Panthera"   "tigris"    "Felis"    "catus"
```

Taxon attributes: `subtaxa`

These return a list of vectors named by taxon IDs.

```
subtaxa(ex_taxmap, value = "taxon_names", recursive = FALSE)[1:3]
```

```
$b
            d                e                f
    "Felidae" "Notoryctidae"    "Hominidae"

$c
          g
"Solanaceae"

$d
        h              i
"Panthera"      "Felis"
```

Taxon attributes: counts

These return counts of things per taxon.

```
n_subtaxa(ex_taxmap)
```

```
 b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r
11  4  4  2  2  3  1  1  1  1  2  0  0  0  0  0  0
```

```
n_supertaxa(ex_taxmap)
```

```
b c d e f g h i j k l m n o p q r
0 0 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3
```

```
n_obs(ex_taxmap, "info")
```

```
b c d e f g h i j k l m n o p q r
4 2 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1
```

```
n_obs(ex_taxmap, "abund")
```

```
b c d e f g h i j k l m n o p q r
8 0 4 2 2 0 2 2 2 2 0 2 2 2 2 0 0
```

Manipulating

Manipulating: example data

Here is the example object that will be used:

```
print(obj)
```

```
<Taxmap>
  17 taxa: b. Mammalia, c. Plantae ... q. lycopersicum, r. tuberosum
  17 edges: NA->b, NA->c, b->d, b->e ... j->o, k->p, l->q, l->r
  3 data sets:
    info:
      # A tibble: 6 x 4
        taxon_id name   n_legs dangerous
        <chr>    <chr>   <dbl> <lgl>
      1 m        tiger     4.  TRUE
      2 n        cat       4.  FALSE
      3 o        mole      4.  FALSE
      # ... with 3 more rows
    phylopic_ids: a named vector of 'character' with 6 items
       m. e148eabb-f138-43[truncated] ... r. 63604565-0406-46[truncated]
    foods: a list of 6 items named by taxa:
       m, n, o, p, q, r
  1 functions:
    reaction
```

Manipulating: example data

Here is the example object that will be used:

```
heat_tree(obj, node_label = taxon_names, layout = "da")
```
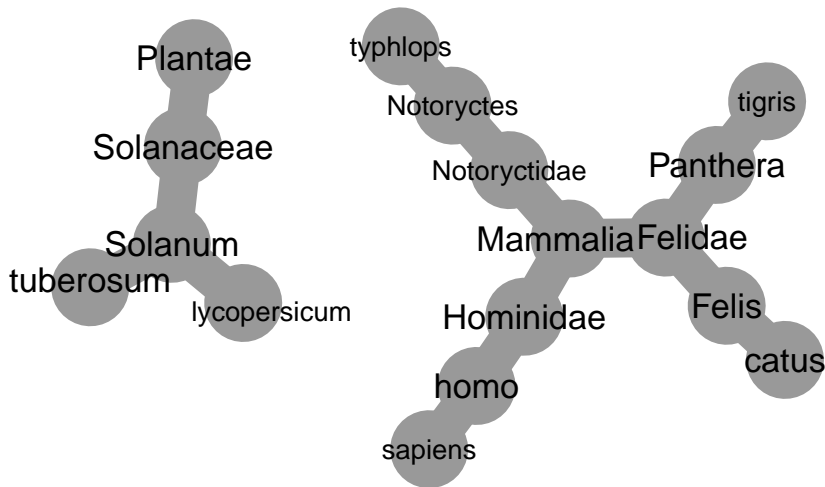
Manipulating: Subsetting the taxonomy

Subset taxonomy and user-defined data to one taxon:

```
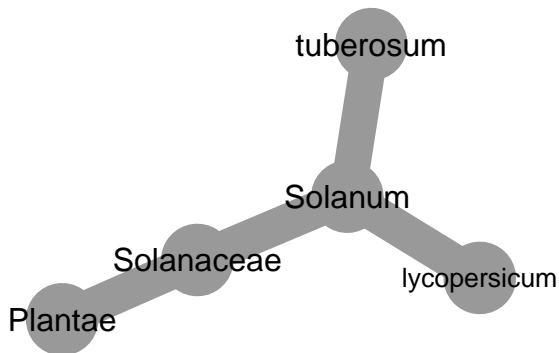filter_taxa(obj, taxon_names == "Plantae", subtaxa = TRUE)
```

```
<Taxmap>
  5 taxa: c. Plantae, g. Solanaceae, l. Solanum, q. lycopersicum, r. tuberosum
  5 edges: NA->c, c->g, g->l, l->q, l->r
  3 data sets:
    info:
      # A tibble: 2 x 4
        taxon_id name   n_legs dangerous
        <chr>    <chr>   <dbl> <lgl>
      1 q        tomato     0. FALSE
      2 r        potato     0. FALSE
    phylopic_ids: a named vector of 'character' with 2 items
       q. b6400f39-345a-47[truncated] ... r. 63604565-0406-46[truncated]
    foods: a list of 2 items named by taxa:
       q, r
  1 functions:
    reaction
```

Manipulating: Subsetting the taxonomy

Subset taxonomy and user-defined data to one taxon:

```
filter_taxa(obj, taxon_names == "Plantae", subtaxa = TRUE) %>%
  heat_tree(node_label = taxon_names, layout = "da")
```

Manipulating: Subsetting the taxonomy

Subset taxonomy to one rank:

```r
filter_taxa(obj, taxon_ranks == "family", supertaxa = TRUE)
```

```
<Taxmap>
  6 taxa: b. Mammalia, c. Plantae ... f. Hominidae, g. Solanaceae
  6 edges: NA->b, NA->c, b->d, b->e, b->f, c->g
  3 data sets:
    info:
      # A tibble: 6 x 4
        taxon_id name   n_legs dangerous
        <chr>    <chr>   <dbl> <lgl>
      1 d        tiger   4.    TRUE
      2 d        cat     4.    FALSE
      3 e        mole    4.    FALSE
      # ... with 3 more rows
    phylopic_ids: a named vector of 'character' with 6 items
       d. e148eabb-f138-43[truncated] ... g. 63604565-0406-46[truncated]
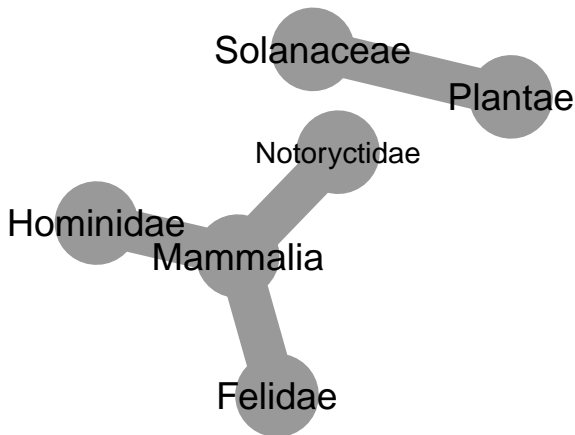    foods: a list of 6 items named by taxa:
       d, d, e, f, g, g
  1 functions:
    reaction
```

Manipulating: Subsetting the taxonomy

Subset taxonomy to one rank:

```
filter_taxa(obj, taxon_ranks == "family", supertaxa = TRUE) %>%
  heat_tree(node_label = taxon_names, layout = "da")
```

Manipulating: Subsetting user data

Subset user-defined data and remove any taxa not in subset:

```
filter_obs(obj, "info", n_legs == 4, drop_taxa = TRUE)
```

```
<Taxmap>
  9 taxa: b. Mammalia, d. Felidae ... n. catus, o. typhlops
  9 edges: NA->b, b->d, b->e, d->h, d->i, e->j, h->m, i->n, j->o
  3 data sets:
    info:
      # A tibble: 3 x 4
        taxon_id name   n_legs dangerous
        <chr>    <chr>  <dbl>  <lgl>
      1 m        tiger    4.   TRUE
      2 n        cat      4.   FALSE
      3 o        mole     4.   FALSE
    phylopic_ids: a named vector of 'character' with 3 items
      m. e148eabb-f138-43[truncated] ... o. 11b783d5-af1c-4f[truncated]
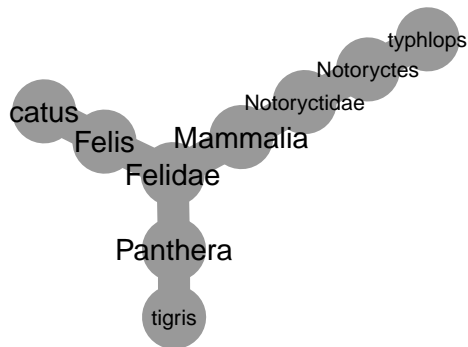    foods: a list of 3 items named by taxa:
      m, n, o
  1 functions:
    reaction
```

Subset data and remove any taxa not in subset:

```
filter_obs(obj, "info", n_legs == 4, drop_taxa = TRUE) %>%
  heat_tree(node_label = taxon_names, layout = "da")
```

Manipulating: Adding user data

Add a column to a dataset:

```
mutate_obs(obj, "info", bipedal = n_legs == 2)
```

```
<Taxmap>
  17 taxa: b. Mammalia, c. Plantae ... q. lycopersicum, r. tuberosum
  17 edges: NA->b, NA->c, b->d, b->e ... j->o, k->p, l->q, l->r
  3 data sets:
    info:
      # A tibble: 6 x 5
        taxon_id name  n_legs dangerous bipedal
        <chr>    <chr> <dbl>  <lgl>     <lgl>
      1 m        tiger  4.    TRUE      FALSE
      2 n        cat    4.    FALSE     FALSE
      3 o        mole   4.    FALSE     FALSE
      # ... with 3 more rows
    phylopic_ids: a named vector of 'character' with 6 items
      m. e148eabb-f138-43[truncated] ... r. 63604565-0406-46[truncated]
    foods: a list of 6 items named by taxa:
      m, n, o, p, q, r
  1 functions:
    reaction
```

Manipulating: values accessible to NSE

The following can be used in manipulation functions as if they were
independent variables using Non-Standard Evaluation (NSE):

- Functions that return per-taxon information

- User-defined table columns

- User-defined vectors and lists

- User-defined functions

```
unname(all_names(obj))
```

```
 [1] "taxon_names"      "taxon_ids"        "taxon_indexes"    "classifications"
 [5] "n_supertaxa"      "n_supertaxa_1"    "n_subtaxa"        "n_subtaxa_1"
 [9] "n_leaves"         "n_leaves_1"       "taxon_ranks"      "is_root"
[13] "is_stem"          "is_branch"        "is_leaf"          "is_internode"
[17] "n_obs"            "n_obs_1"          "name"             "n_legs"
[21] "dangerous"        "phylopic_ids"     "foods"            "reaction"
```

Compatiblity with other packages

taxa is still being developed, but we hope it will become a
foundation for an ecosystem of R packages that use taxonomic data.

- ▶ Flexible parsers can read most formats
- ▶ Working on making `taxize` compatible with `taxa`
- ▶ Fully compatible with `metacoder`

Metacoder: visulization of taxonomic data

Metacoder: visulization of taxonomic data

Acknowledgements

Questions?