Retrieving Taxonomic Info

Courtney Stepien

August 4, 2016

Contents

File status **1**

File goals **1**

Learning to use Taxize 2

Assessing the response of various APIs 2

How many Families per Order 2

Comparing AlgaeBase, NCBI, and BOLD 2

Last completed steps **2**

Next steps **2**

Improving functions and code 3

Analysis **3**

Data setup and packages 3

Genus List and Species List 3

ITIS taxonomy - inadequate 3

NCBI taxonomy - better than ITIS 4

BOLD taxonomy 9

File status

Current

File goals

In this file, I learned how to use taxize for Rhodophyta, and found that the large databases that the taxize package interfaces with were largely inaccurate for algal species with some Databases. I am using both higher taxonomy data from AlgaeBase and Class/Order/Family data from the better databases with taxize (BOLD and NCBI) to compare and see how different each of them are.

I also want to determine how many Orders, Families and Genera there are (and Families/Order, Genera/Family) so we can generate a table of how our dataset sampling compares to overall diversity levels in Florideophyceae.

Learning to use Taxize

Learn how to use the packages taxize and myTAI to get higher-level taxonomy for my genera, so we can table basic stats about the number of taxa per family, order and class.

taxize package github link for tutorials and source code

taxizesoap package github link for tutorials and source code

Assessing the response of various APIs

I want so see which databases have the best red algal coverage.

There are a number of different taxonomy databases:

- high coverage: Barcode of Life Data Systems (BOLD) <- taxize package
- high coverage: National Center for Biotechnology Information (NCBI) <- taxize package
- incomplete: Integrated Taxonomic Information Service (ITIS) <- taxize package
- too outdated: World Register of Marine Species (WoRMS) <- taxizesoap package
- not yet assessed: Encyclopedia of Life (EOL) <- taxize package

How many Families per Order

I want to investigate the coverage our tree will have compared to all possible families and orders (maybe genera too)

Comparing AlgaeBase, NCBI, and BOLD

Lastly, I want to compare these different databases to see what kinds of differences there are in taxonomy - Rhodophyta is constantly changing taxonomy.

Last completed steps

Compare my dataset to global NCBI dataset - what kind of coverage do we have at each level? Reformat NCBI global taxonomy dataset Get all children for each Class, Order, Family, Genus in NCBI Function for getting genera per family, family per order, etc. in the dataset Function for reformatting NCBI list data as dataframe How to transform data to appropriate format - making function in NCBI Exclude non-Protists and NAs in BOLD system Reporting results for BOLD (this many genera missing - only 2 I think)

Next steps

Next steps are:

0. Make code to subset my current isotope dataset by the actualy number of taxa we get gene sequences for (currently, I'm assuming that we get 100% of taxa represented from scraping GenBank). Only a subset of species we have isotope data for will also have DNA seqence data
1. Evaluate functions in reformatted NCBI global Rhodophyta dataset - genera per family, family per order, etc. (update code to reflect source global dataset) and compare to my dataset - create plots, ID key areas of low coverage (might be more appropriate for basic Dataset Stats file)

2. Pipeline for BOLD - 1st get genera IDs
3. Use IDs to get higher level taxonomy from BOLD
4. reformat taxonomy data in BOLD. BOLD is easier to recast the data
5. finding all genera in a family, all species in a genus, etc. for BOLD

Improving functions and code

My functions work but are clunky - fix the following functions:

1. tax_bins - could use aggregate to make it better, apply over a list
2. ncbi_downstream - make it generalizable to any level taxonomy, enter taxonomy name and level (eg. "Rhodomelaceae", "order") and it will return every taxa downstream of that, down to species
3. Adding new lines before and after print statements (especially "Missing Ballia genera children due to . . . ")

Analysis

Data setup and packages

```
library(taxize)
library(taxizesoap)
library(dplyr)
library(lazyeval)
library(tidyr)
library(knitr)
data <- read.csv("../data/mean_13c_per_species.csv")
```

Genus List and Species List

```
genus_list <- data %>% distinct(genus) %>% select(genus)
genus_list <- as.character(genus_list$genus)
genus_count <- data.frame(data %>% group_by(genus) %>% summarize(taxa = n()))

species_list <- data %>% distinct(truetaxa) %>% select(truetaxa)
species_list <- as.character(species_list$truetaxa)
```

ITIS taxonomy - inadequate

```
#taxreturn_s <- classification(species_list, db = 'itis')
#taxreturn_g <- classification(genus_list, db = 'itis')
```

Querying ITIS at the species level leaves the majority of taxa unidentified because there are very few seaweed species in the database. Querying at the genus level is also dodgy - and the reported results aren't consistent (the list returned has incorrect names connected to certain taxa search - eg taxonomy for Ballia, but it is linked to Bostrychia in the returned data). You also must choose the correct taxa if there are multiple taxa

with the same genus name (red algae and insect taxa frequently share the same genus name), and there is no information about which Acanthophora is the red algae and which is the invert, other than "accepted" versus "valid."

ITIS terms:

- Accepted = Plantae, Chromista and Fungi taxa name (pick this one!)
- Valid = Animalia, Protozoa, Bacteria or Archaea taxa name (don't pick this one!)

NCBI taxonomy - better than ITIS

Initial query at genus and species level

```
taxreturn_s <- classification(species_list, db = 'ncbi')
#taxreturn_g <- classification(genus_list, db = 'ncbi')

#tax_ncbi <- data.frame(class = character(), order = character(), family = character(), genus = charact

#format_ncbi <- function(x){
#  for (j in 1:length(x)){
#    if (!is.na(x[j])){
#      if (length(which(x[j][[1]]$rank %in% c("class", "order", "family", "genus"))) == 4) {
#             tax_ncbi <<- rbind(tax_ncbi, data.frame(setNames(as.list(x[j][[1]]$name[which(x[j][[1]]$r
#      }
#    }
#  }
#  rownames(tax_ncbi) <- c()
#}

#write.csv(tax_ncbi, file = "../data/ncbi_taxonomy.csv", row.names = FALSE)
```

Like ITIS, you must choose the correct genus if there are multiple taxa with the same genus name (red algae and insect taxa frequently share the same genus name), so assessing the API response at the genus level must be manually updated (below). But the interface for choosing which taxa is the correct one is much better in NCBI vs ITIS, mostly because NCBI actually gives you the taxa division membership to help you choose (red algae, stick insect, etc.). This still means that the process can't be done completely automatically, as the user must choose everytime there is a problem.

Add higher taxonomy information to isotope dataset

```
tax_ncbi <- read.csv("../data/ncbi__dataset_taxonomy.csv")
data_ncbi <- data
data_ncbi$class <- tax_ncbi$class[match(data_ncbi$genus, tax_ncbi$genus)]
data_ncbi$order <- tax_ncbi$order[match(data_ncbi$genus, tax_ncbi$genus)]
data_ncbi$family <- tax_ncbi$family[match(data_ncbi$genus, tax_ncbi$genus)]
```

Dataset Species and Genus Coverage from NCBI

The NCBI database is way better than ITIS, and returns things in the correct scheme. Only 7 genera are missing out of 149 genera for the genus level dataset as of last query on Sun Aug 14 2016, a 95% coverage

rate - super cool! There is also one genus missing family-level data - Opuntiella family membership is listed as 'no rank', 'unclassified Gigartinales'

Missing red algal genera in NCBI:

"Bonnemaisonia" "Bornetia" "Dictyomenia" "Echinothamnion" "Halicnide" "Halopitys" "Jeannerettia" "Opuntiella" <- missing family ('unclassified Gigartinales')

For species, 55 species are missing out of 266 genera for the genus level dataset. 79.3233083% species coverage in NCBI, so clearly genus level is the way to go when assessing results.

How many Orders, Families and Genera total from Dataset, excluding Bangiaceae

```
tax_summary_ncbi <- apply(tax_ncbi[which(tax_ncbi$class == "Florideophyceae"),1:4], 2, function(x)length
```

The below table gives the results of how many classes, orders, families, genera and species are in the dataset (excluding the 7 genera in the previous section that NCBI doesn't have complete taxonomy data on).

```
tax_summary_ncbi
```

```
##  class  order family  genus
##      1     18     48    148
```

Genera per Family, Families per Order, Orders per Class from Dataset

Now let's see how many data points for each lineage I have.

Let's make a function to determine the number of x in higher taxonomy y in the dataset z. I chose to use the dplyr package, but I also could have made a function with aggregate, aggregating by different taxonomy levels (by a list of taxonomy levels to automate it) and then binding those data frames together. This way I learned some new things about dplyr - notably the underscored group_by, which is used when you have variable inputs.

```
tax_bins <- function(x,y,z){
  data <- data.frame(z %>% filter(!is.na(class)) %>% group_by_(y) %>% summarize_(n_uniq=interp(~n_disti
  #print(data)
  colnames(data) <- c(as.character(y), paste("n_", as.name(x), sep = ""))
  #print(colnames(data))
  #print(data)
  assign(paste(as.name(x), "_per_", as.name(y), sep = ""), data, envir = .GlobalEnv)
  print(paste(as.name(x), "_per_", as.name(y), sep = ""))
}

#Example usage: tax_bins("order", "class", tax_ncbi) would give classes PER order in the dataset tax_nc
```

First, Orders, Families and Genera per Class in Dataset

```
tax_bins("order", "class", data_ncbi)
```

```
## [1] "order_per_class"
```

```
tax_bins("family", "class", data_ncbi)
```

```
## [1] "family_per_class"
```

```
tax_bins("genus", "class", data_ncbi)
```

```
## [1] "genus_per_class"
```

```
tax_bins("species", "class", data_ncbi)
```

```
## [1] "species_per_class"
```

```
class_children_dataset <- cbind(order_per_class, n_family = family_per_class$n_family, n_genus = genus_
```

Next, Families and Genera per Order in Dataset

```
tax_bins("family", "order", data_ncbi)
```

```
## [1] "family_per_order"
```

```
tax_bins("genus", "order", data_ncbi)
```

```
## [1] "genus_per_order"
```

```
tax_bins("species", "order", data_ncbi)
```

```
## [1] "species_per_order"
```

```
order_children_dataset <- cbind(family_per_order, n_genus = genus_per_order$n_genus, n_species = specie
```

Finally, Genera per Family in Dataset

```
tax_bins("genus", "family", data_ncbi)
```

```
## [1] "genus_per_family"
```

```
tax_bins("species", "family", data_ncbi)
```

```
## [1] "species_per_family"
```

```
family_children_dataset <- cbind(genus_per_family, n_species = species_per_family$n_species)
```

Querying the global Rhodophyta dataset from NCBI - all children of Rhodophyta class Florideophyceae

Here I'm getting the total number of children for each Class, Order, Family and Genus. This way we can determine how much coverage we have over all families - for example, do we have 100% sampling at the Order level, 80% coverage at the Family level? 60% at the genus level? How representative is our phylogeny and our isotope dataset?

clean this function up to use apply - I could use apply, start with any taxa, just enter the taxa, and the taxa rank as the arguments, then I can get everything downstream.

```
ncbi_downstream <- function(startertaxa){
  df <- ncbi_children(name = startertaxa, ancestor = "Rhodophyta")[1][[1]] %>% filter(childtaxa_rank ==
  df$parent <- startertaxa
  df$parent_rank <- "class"
  print(df)
  df_c2f <- df

  for (i in 1:nrow(df)){
    family <- ncbi_children(name = df$childtaxa_name[i], ancestor = "Rhodophyta")[1][[1]] #%>% filter(c
    family$parent <- df$childtaxa_name[i]
    family$parent_rank <- "order"
    print(family)
    df_c2f <- rbind(df_c2f, family)
  }

  ncbi_order <- filter(df_c2f, childtaxa_rank == "order") %>% arrange(childtaxa_name)

  ncbi_family <- filter(df_c2f, childtaxa_rank == "family") %>% arrange(childtaxa_name)

  df_c2g <- df_c2f
  missing_families <- c()

  for (j in 1:nrow(ncbi_family)){
    genus <- ncbi_children(name = ncbi_family$childtaxa_name[j], ancestor = "Florideophyceae")[1][[1]]
    if (nrow(genus) > 0) {
      genus$parent <- ncbi_family$childtaxa_name[j]
      genus$parent_rank <- "family"
      print(genus)
      df_c2g <- rbind(df_c2g, genus)
    }
    else {
      missing_families <- append(missing_families, ncbi_family$childtaxa_name[j])
      print(paste("Missing ", as.character(ncbi_family$childtaxa_name[j]), " family children due to API
    }
  }

  ncbi_genus <<- filter(df_c2g, childtaxa_rank == "genus") %>% arrange(childtaxa_name)

  df_c2s <- df_c2g
  missing_genera <- c()

  for (k in 1:nrow(ncbi_genus)){
  species <- ncbi_children(name = ncbi_genus$childtaxa_name[k], ancestor = "Florideophyceae")[1][[1]] #
```

```
    if (nrow(species) > 0) {
      species$parent <- ncbi_genus$childtaxa_name[k]
      species$parent_rank <- "genus"
      print(species)
      df_c2s <- rbind(df_c2s, species)
    }
    else {
      missing_genera <- append(missing_genera, ncbi_genus$childtaxa_name[k])
      print(paste("Missing ", as.character(ncbi_genus$childtaxa_name[k]), " genus children due to API qu
    }
  }

  ncbi_species <<- filter(df_c2s, childtaxa_rank == "species") %>% arrange(childtaxa_name)

write.csv(df_c2s, file = "../data/ncbi_global_Rhodophyta_tax.csv", row.names = FALSE)
```

Formatting global NCBI dataset for using tax_bins function

```
df <- read.csv("../data/ncbi_global_Rhodophyta_tax.csv")
df <- filter(df, childtaxa_rank %in% c("order", "family", "genus")) %>% distinct(childtaxa_name)
global_genus <- filter(df, parent_rank == "family") %>% select(childtaxa_id, childtaxa_name, childtaxa_r
colnames(global_genus) <- c("childtaxa_id", "childtaxa_name", "childtaxa_rank", "family", "source")
family <- filter(df, parent_rank == "order")
class <- filter(df, parent_rank == "class")
global_genus$order <- family$parent[match(global_genus$family, family$childtaxa_name)]
global_genus$class <- class$parent[match(global_genus$order, class$childtaxa_name)]
colnames(global_genus) <- c("childtaxa_id", "genus", "childtaxa_rank", "family", "source", "order", "cla
```

How many Orders, Families and Genera total from Flordeophyceae children

```
tax_summary_ncbi_global <- apply(global_genus, 2, function(x)length(unique(x)))
tax_summary_ncbi_global <- tax_summary_ncbi_global[c("class", "order", "family", "genus")]
```

The below table gives the results of how many classes, orders, families, genera and species are in the dataset (excluding the 7 genera in the previous section that NCBI doesn't have complete taxonomy data on).

```
tax_summary_ncbi_global
```

```
##  class  order family  genus
##      1     25     83    547
```

How many children total (globally) in Florideophyceae from each taxonomy level

First, Orders, Families and Genera per Class Globally

```
tax_bins("order", "class", global_genus)
```

```
## [1] "order_per_class"
```

```
tax_bins("family", "class", global_genus)
```

```
## [1] "family_per_class"
```

```
tax_bins("genus", "class", global_genus)
```

```
## [1] "genus_per_class"
```

```
class_children_global <- cbind(order_per_class, n_family = family_per_class$n_family, n_genus = genus_pe
```

Next, Families and Genera per Order Globally

```
tax_bins("family", "order", global_genus)
```

```
## [1] "family_per_order"
```

```
tax_bins("genus", "order", global_genus)
```

```
## [1] "genus_per_order"
```

```
order_children_global <- cbind(family_per_order, n_genus = genus_per_order$n_genus)
```

Finally, Genera per Family Globally

```
tax_bins("genus", "family", data_ncbi)
```

```
## [1] "genus_per_family"
```

```
tax_bins("species", "family", data_ncbi)
```

```
## [1] "species_per_family"
```

```
family_children_global <- cbind(genus_per_family)
```

Comparing Global Children vs Our Dataset

```
dataset_coverage <- rbind(tax_summary_ncbi[2:4], tax_summary_ncbi_global[2:4])
rownames(dataset_coverage) <- c("Dataset Coverage", "All Florideophyceae Children")
Fraction_Coverage <- dataset_coverage[1,]/dataset_coverage[2,]
ncbi_coverage <- rbind(dataset_coverage, Fraction_Coverage)
```

BOLD taxonomy

Initial query at genus and species level

```
taxreturn_s <- bold_search(name = species_list, includeTree = TRUE)
taxreturn_g <- bold_search(name = genus_list, includeTree = TRUE)
```

The BOLD database returns a dataframe initially, as opposed to a list of dataframes from the ITIS and NCBI databases. It also doesn't require choosing the correct taxa if there are multiple taxa with the same genus name (red algae and insect taxa frequently share the same genus name), but rather just reports them both. This means we can filter out the non-algal taxa easily with dplyr.

Filter out non-algal taxa at Genus Level

```
taxreturn <- taxreturn_g %>% filter(tax_division == "Protists", !is.na(taxon))
```

Species and Genus Coverage from BOLD

Only 3 genera are missing out of 149 genera for the genus level dataset - super cool! 97.9865772% genus coverage in BOLD.

Missing red algal genera in BOLD:

```
taxreturn_g$input[(which(is.na(taxreturn_g$taxon)))]
```

```
## [1] "Halopitys"    "Jeannerettia" "Rytiphlaea"
```

For species, 61 species are missing out of 266 genera for the genus level dataset -77.0676692% species coverage in BOLD. So clearly genus level is the way to go.

Get higher taxonomy info by ID