rphenoscate: Tutorial 4

Diego S. Porto, Sergei Tarasov, Caleb Charpentier, and SCATE team

August, 2022

In this last tutorial we will be exploring the potential of semantic phenotypes in phylogenetic inference. The anatomy of organisms can be described using semantic phenotypes, collections of semantic statements about their anatomical entities, structural relations, and qualities. In a sense, semantic phenotypes serve a purpose similar to that of standard character matrices, which summarize anatomical information with potential phylogenetic utility. The advantage of using semantic phenotypes, however, is that they can also be reasoned using computer logic. However, how can we use semantic phenotypes fo infer phylogenies? One simple approach would be to calculate the similarity (or distance) among semantic phenotypes describing different species. For that, we can use semantic similarity, a metric that describes how similar two or more concepts are. Particularlly, each species can be conceived as a profile of semantic statements describing its anatomical entities and qualities, and the semantic similarity among different profiles can be calculated using the function 'profile_similarity' from *rphenoscape*.

Note that this is just an initial exploration of the phylogenetic information available in semantic phenotypes. This approach does not assume an explicit model for 'how semantic phenotypes evolve through time'. Our goal here is just to demonstrate that semantic statements in semantic phenotypes can also carry relevant information to phylogenetic inference, as standard morphological characters in phylogenetic character matrices do. Therefore, we agree that appropriate phylogenetic algorithms for tree inference still need to be developed to deal with semantic representations of organism anatomy (for more discussions see Vogt 2018: 10.1111/cla.12195).

The first two steps of this tutorial are the same as in Tutorial 3.

STEP 1. Loading the packages.

First, let's load *rphenoscape* and *rphenoscate*.

```
library("rphenoscape")
library("rphenoscate")
```

STEP 2. Assembling the data set from a given study.

Then, let's retrieve the phylogenetic character matrix from Dillman et al. (2016).

```
# Get a list of all annotated studies available at Phenoscape KB.
studies <- pk_get_study_list()

# Get a particular study # (Change this part to get a particular study).
study <- studies$id[studies$label == 'Dillman et al. (2016)']

# Get NeXML data.
selected_study <- pk_get_study_xml(study)

# Build the original character matrix.
```

```
char.mat <- RNeXML::get_characters(selected_study[[1]])

# Get rownames and colnames from data set.
row.mat <- rownames(char.mat)
col.mat <- colnames(char.mat)
```

Sometimes, data sets might have rows and/or columns represented as IRIs or character IDS instead of actual human-readable labels. Let's check for that.

```
# Check rownames and colnames from data set.
row.mat[1:3]
```

```
## [1] "Abramites hypselonotus" "Anostomoides laticeps"  "Anostomus anostomus"
```

```
col.mat[1:3]
```

```
## [1] "Alignment of border between frontal and parietal on dorsal surface of cranium"
## [2] "Alignment of dorsal process of fourth epibranchial"
## [3] "Alignment of ectopterygoid"
```

If rownames and/or colnames are IRIs and/or character IDs, then run the code below to extract the original labels.

```
# Get metadata the original character matrix.
selected_study.meta <- pk_get_study_meta(selected_study)

# Get rownames and colnames from data set.
row.mat <- selected_study.meta[[1]]$id_taxa$label[
  match(rownames(char.mat),selected_study.meta[[1]]$id_taxa$otu)]
col.mat <- selected_study.meta[[1]]$id_entities$label[
  match(colnames(char.mat),selected_study.meta[[1]]$id_entities$char)]
```

The approach used here can be quite computationally intensive since we are going to calculate the 'profile semantic similarity' for all pairwise comparisons among taxa. Each taxon, in the case of this data set, can be described by more than 400 phenotype statements. Thus, each comparison can potentially involve more than 800 distinct phenotype statements producing very large subsumer matrices. For more details about what a subsumer matrix is, please refer to *rphenocape* documentation. Therefore, let's get a small sample of taxa from the original data set and retrieve all phenotypes.

```
# Set a sample size.
N <- 10

# Get a sample.
set.seed(123)
s <- sample(row.mat,10)

# Get all phenotypes for each taxon.
tax.pheno <- list()

for(i in 1:N){

  tax.pheno[[i]] <- get_phenotypes(taxon = s[i], study = study)$id

}

# Name list elements.
names(tax.pheno) <- s
```

STEP 3. Calculating semantic similarity among taxon profiles.

Then, let's calculate the semantic similarity for all possible pairwise comparisons of profiles of semantic statements using *rphenoscape*.

```
# Get all possible pairwise comparisons among taxon profiles.
q <- combn(names(tax.pheno), m = 2, function(x) tax.pheno[x] )

# Run all profile pairwise ss calculations.
# This might take some hours depending on the size of the sample! (~ +1h).
prof <- numeric()

for(i in 1:dim(q)[2]){

  st <- Sys.time()
  a <- subsumer_matrix(terms = c(unlist(q[1,i][[1]]),unlist(q[2,i][[1]])),
                       preserveOrder = T)
  b <- c(rep("A",length(q[1,i][[1]])), rep("B",length(q[2,i][[1]])))
  a <- profile_similarity(jaccard_similarity, a, f = as.factor(b))

  prof[i] <- a[1,2]
  ed <- Sys.time()

  print(paste0("Cycle: ",i, "   Completed: ",round((i/dim(q)[2]),2),"
               Cycle Time: ", round((ed - st),2) ))

}

# Convert vector to a matrix.
tax.ss <- matrix(data = NA, ncol = length(s), nrow = length(s), byrow = T)

# Fill-in the matrix.
tax.ss[lower.tri(tax.ss)] <- prof
tax.ss[upper.tri(tax.ss)] <- t(tax.ss)[upper.tri(t(tax.ss))]
diag(tax.ss) <- 1
colnames(tax.ss) <- rownames(tax.ss) <- s

# Save temporary object.
saveRDS(tax.ss, file = "ssmat.RDS")

# Clean workspace.
rm(i,a,b,st,ed)
```

STEP 4. Geting the semantic similarity species tree.

Finally, let's build a distance tree with the semantic similarity values and compare with the consensus tree obtained from the analysis of the actual character matrix from Dillman et al. (2016).

```
# Get hierarchical clusters and convert to phylo object.
phylo <- as.phylo(hclust(as.dist(1 - tax.ss)))

# Import MJ consensus tree from the analysis of the full data set from Tutorial 3.
mjtree <- read.nexus(file = "./data/fishtree.tre")
mjtree$tip.label <- gsub(mjtree$tip.label, pattern = "_", replacement = " ")
```

```r
# Prune tree to match sample of taxa.
mjtree.p <- keep.tip(mjtree, s)

# Set some colors for tip labels.
cols1 <- phylo$tip.label
cols2 <- mjtree.p$tip.label

cols1[grep(cols1, pattern = "Steindachnerina", invert = T)] <- "black"
cols1[grep(cols1, pattern = "Steindachnerina")] <- "purple2"

cols2[grep(cols2, pattern = "Steindachnerina", invert = T)] <- "black"
cols2[grep(cols2, pattern = "Steindachnerina")] <- "purple2"

# Set some graphical parameters.
par(mfrow = c(1,2), mar = c(0.1,0.1,1.0,0.1), pty = "m")

# Plot semantic species tree.
plot.phylo(phylo, use.edge.length = F, cex.main = 0.9, main = "semantic similarity",
           edge.width = 1.5, cex = 0.8, tip.color = cols1)

# Plot prunned original tree.
plot.phylo(mjtree.p, use.edge.length = F, cex.main = 0.9, main = "standard characters",
           edge.width = 1.5, cex = 0.8, tip.color = cols2, direction = "leftwards")
```