rphenoscate: Tutorial 1

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

August, 2022

In this very first tutorial, we will be using *rphenoscape* and the new package *rphenoscate* to perform semantic-aware evolutionary analyses of multiple fish traits.

More specifically, we are going to assemble a data set of absences and presences of bones for Characidae fishes (e.g., catfishes), set up appropriate models accounting for trait dependencies based on prior knowledge available in an anatomy ontology (e.g., UBERON), and then sample character histories through stochastic mapping.

Finally, we are going to employ some tools for visually exploring the semantic properties of the data set.

STEP 1. Installing and loading the package.

If you haven't installed the package yet, then run the following:

```
devtools::install_github("uyedaj/rphenoscate")
```

It is also strongly recommended to install its companion package *rphenoscape* that allows access to the Phenoscape KB.

```
devtools::install_github("phenoscape/rphenoscape")
```

Now, let's load the packages *rphenoscape* and *rphenoscate*.

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

Let's load some other packages that might be useful as well.

```
library("ape")
library("phytools")
```

STEP 2. Assembling a data set.

For this step, we will use some functions from *rphenoscape* to access data available in the Phenoscape Knowledgebase (KB). Phenoscape KB is a database of curated semantic information for more than 6.5k fish species and about 14.5k phylogenetic characters, comprising a total of 256 phylogenetic character matrices annotated with ontology terms.

Stochastic character mapping will be performed using the dated phylogeny available from the R package *fishtree*. For details on how this tree was obtained, please refer to Rabosky et al. (2018): 10.1038/s41586-018-0273-1 and Chang et al. (2019): 10.1093/sysbio/syz081. If you do not have this package installed, please do so by running the following:

```
install.packages("fishtree")
```

First, let's get a phylogeny for Characidae.

```
ftree <- fishtree::fishtree_phylogeny(rank = "Characidae", type = "chronogram_mrca")
```

Second, let's set up some search parameters. For demonstrative purposes, let's assemble data available in the KB for some bones of the skull, pectoral girdle and postcranial axial skeleton of Characidae fishes.

```
taxa <- "Characidae"
terms <- c("antorbital", "infraorbital 1", "infraorbital 2", "infraorbital 3",
           "infraorbital 4", "infraorbital 5", "infraorbital 6", "scapula",
           "coracoid bone", "supraneural 1 bone", "supraneural 2 bone",
           "supraneural 3 bone", "supraneural 4 bone", "supraneural 5 bone",
           "uroneural 1", "uroneural 2")
```

Finally, let's then retrieve data from the Phenoscape KB.

```
nex <- lapply(terms, pk_get_ontotrace_xml, taxon = taxa, variable_only = T)
names(nex) <- terms
```

Now, let's build the OntoTrace matrix. This matrix is obtained by reasoning through the KB, which produces inferences of absences and presences of bones based on the ontology and semantic phenotype annotations. See Dececchi et al. (2015): doi.org/10.1093/sysbio/syv031 for more details.

```
onto <- suppressWarnings(lapply(nex, pk_get_ontotrace))
```

Then, let's organize the data by removing some unnecessary columns.

```
# Merge matrices and remove non-trait data.
mat <- purrr::reduce(onto, full_join, by = "taxa", suffix = c("", ".y"))
mat <- dplyr::select_at(mat, dplyr::vars(-contains("otu")))

# Removes duplicated columns.
dat <- dplyr::select_at(mat, vars(-contains(".y")))
```

We can use *rphenoscate* to check the taxonomic coverage of information available for each anatomical entity.

```
print_coverage(dat)
```

```
##                                traits coverage    average
## infraorbital 1         infraorbital 1      359 0.98885794
## infraorbital 2         infraorbital 2      358 0.99162011
## infraorbital 4         infraorbital 4      358 0.95176849
## infraorbital 6         infraorbital 6      332 0.95757576
## infraorbital 3         infraorbital 3      323 0.99071207
## scapula                       scapula      297 0.99663300
## infraorbital 5         infraorbital 5      291 0.94501718
## coracoid bone           coracoid bone      288 0.99652778
## coracoid foramen     coracoid foramen      267 0.03734440
## uroneural 1               uroneural 1      246 0.98360656
## uroneural 2               uroneural 2      243 0.24890830
## scapular process     scapular process      225 0.99555556
## supraneural 3 bone supraneural 3 bone       18 0.23529412
## supraneural 4 bone supraneural 4 bone       13 0.07692308
## supraneural 5 bone supraneural 5 bone        7 0.14285714
```

Overall, the taxonomic coverage seems satisfactory, with the exception of some anatomical entities from the postcranial axial skeleton: 'supraneural 3-5 bones'. Moreover, let's also check for polymorphic data (i.e., species coded both as absent and present for a given bone).

```
round(apply(dat == "0 and 1" | dat == "1 and 0", 2, sum, na.rm = T)/dim(dat)[1], 2)[-1]
```

```
##         antorbital     infraorbital 1     infraorbital 2     infraorbital 3
##               0.01               0.00               0.00               0.00
##     infraorbital 4     infraorbital 5     infraorbital 6            scapula
##               0.11               0.00               0.00               0.00
##   scapular process      coracoid bone   coracoid foramen supraneural 1 bone
##               0.00               0.00               0.06               0.00
## supraneural 2 bone supraneural 3 bone supraneural 4 bone supraneural 5 bone
##               0.00               0.00               0.00               0.00
##        uroneural 1        uroneural 2
##               0.44               0.03
```

Most anatomical entities do not contain polymorphic character information, however for some of them, there is a substantial amount! For example: 'uroneural 1' (44%) and 'infraorbital 4' (11%). In order to gain some additional information, let's recode all polymorphisms as 'presences' (state:1) since the respective anatomical entities were annotated as 'present' in at least some individuals of the fish species. Also, let's recode all 'NAs' as missing information '?'.

```
dat[dat == "0 and 1" | dat == "1 and 0"] <- "1"
dat[is.na(dat)] <- "?"
```

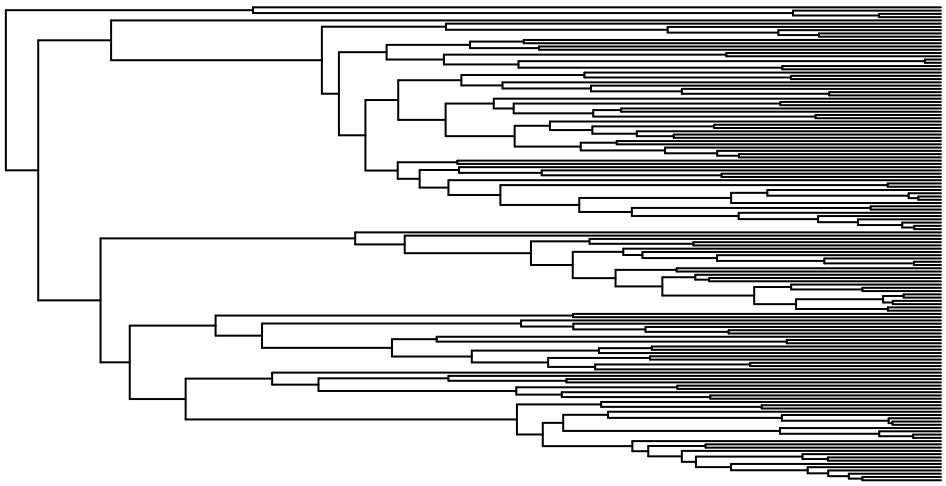
And build the final data set by concatenating information from all anatomical entities.

```
# Adjust tip labels.
ftree$tip.label <- gsub(ftree$tip.label, pattern = "_", replacement = " ")

# Build data set.
td <- suppressWarnings(make.treedata(ftree, dat))
```

Let's just visualize the Characidae tree by plotting it.

```
plot.phylo(td$phy, show.tip.label = F)
```



STEP 3. Check data set for dependencies.

Now that we have assembled our data set, we need to check for possible dependencies among anatomical entities. For that, we need a dependency matrix. This matrix describes the dependency structure among anatomical entities based on knowledge available in an anatomy ontology, in this case, UBERON (Mungall et al. 2012: 10.1186/gb-2012-13-1-r5). For example, the absence or presence of a 'dorsal fin ray' depends on the presence of a 'dorsal fin'.

```
# Get IRIs for anatomical terms.
IRI <- sapply(colnames(td$dat), function(x) pk_anatomical_detail(x)$id)

# Get the dependency matrix using rphenoscape.
dep.mat <- pa_dep_matrix(IRI, .names = "label", preserveOrder = T)
diag(dep.mat) <- NA
```

Then, if dependencies are found, the phylogenetic characters describing the absence (state:0) or presence (state:1) of the respective anatomical entities (i.e., fish bones) must be recoded accordingly. For that, we are using functions from *rphenoscate*. For more details on trait dependencies and character amalgamation in general, please refer to Tarasov(2019):doi.org/10.1093/sysbio/syz005, Tarasov (2020): doi.org/10.1093/sysbio/syz050, and Tarasov (2021):doi.org/10.1101/2021.04.26.441495. For a more in depth theoretical discussion on different types of dependencies among anatomical entities, please refer to Vogt (2017):doi/10.1111/cla.12209.

In our case, we found two pairs of dependent entities: 'scapula' and 'scapular process'; 'coracoid bone' and 'coracoid foramen'. In both cases, these are 'ontological dependencies' based on 'parthood relationships' (see Vogt 2017).

```r
# Amalgamate dependent traits.
amal.deps <- amalgamate_deps(dep.mat)

# Recode traits.
td$dat <- type.convert(as.data.frame(td$dat), as.is = T)
td.comb <- recode_traits(td, amal.deps)
```

STEP 4. Fitting models of trait evolution.

For simplicity, in this tutorial we are going to fit models of trait evolution using a maximum likelihood framework with *rphenoscate*. The function 'amalgamated_fits_corHMM' is a wrapper that uses *corHMM* and the dependency structure inferred from the dependency matrix to fit models of discrete trait evolution accounting for trait dependencies. We will need the model fit objects obtained for each trait to perform (faster) stochastic mapping. Note that we can also perform (slower) stochastic mapping under a Bayesian framework jointly sampling character histories and Q matrices, thus also accounting for uncertainty in transition rates estimation.

```r
corhmm.fits <- amalgamated_fits_corHMM(td.comb, amal.deps)
```

STEP 5. Sampling histories of trait evolution.

Now, let's use 'amalgamated_simmaps_corHMM' from *rphenoscate* to perform stochastic mapping in R. This is another wrapper function, which uses 'makeSimmap' from *corHMM* to sample character histories. Let's then sample 100 histories for each trait. This step can take a few minutes depending on the size of the data set and number of traits.

```r
stmaps <- amalgamated_simmaps_corHMM(corhmm.fits, nSim = 100)
names(stmaps) <- names(corhmm.fits)
```

Let's then just plot some samples of character histories from different traits.

```r
par(mfrow = c(2,4), mar = c(0.1,0.1,5.0,0.1))
plotSimmap(stmaps[[6]][[1]], ftype = "off")
```

```
## no colors provided. using the following legend:
##         1         2
##   "black" "#DF536B"
```

```r
title(main = names(stmaps)[6], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[7]][[1]], ftype = "off")
```

```
## no colors provided. using the following legend:
##         1         2
##   "black" "#DF536B"
```

```r
title(main = names(stmaps)[7], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[8]][[1]], ftype = "off")
title(main = names(stmaps)[8], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[9]][[1]], ftype = "off")
```

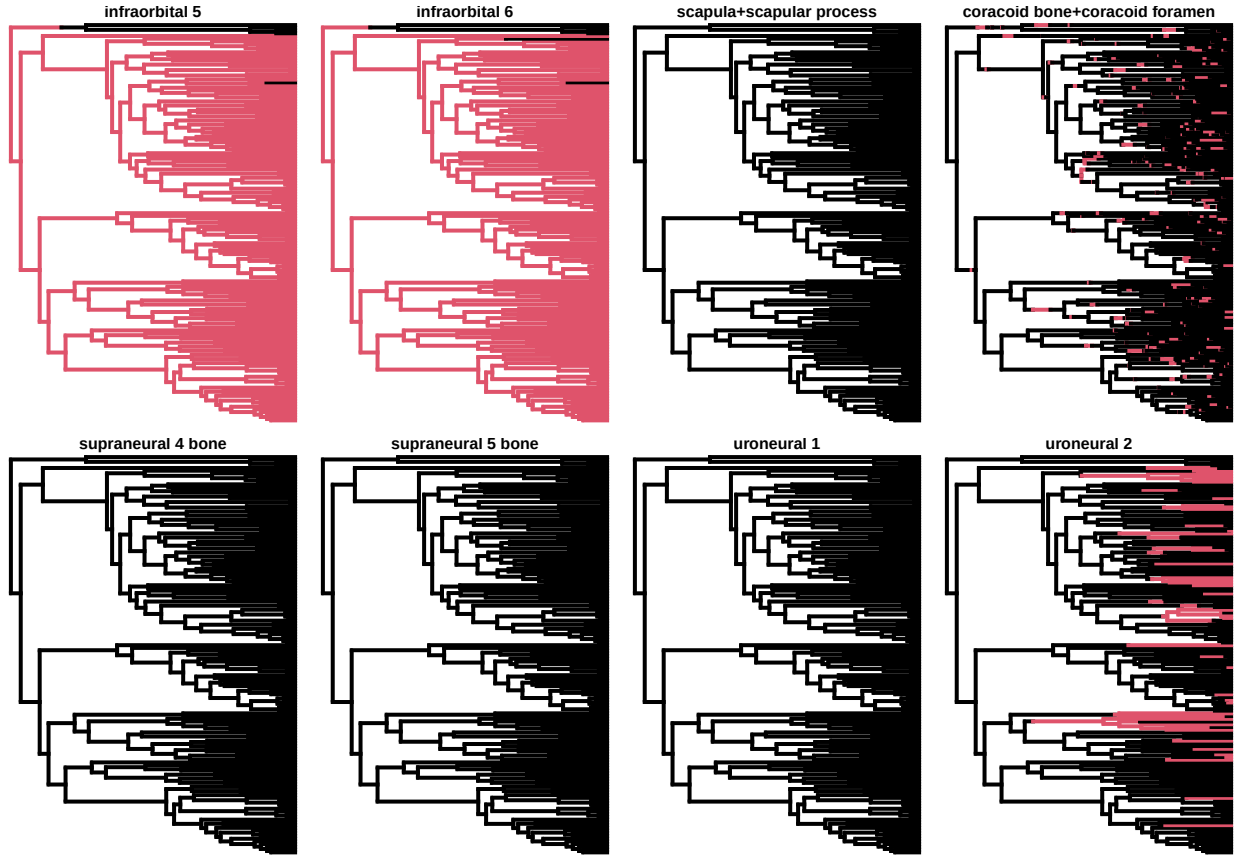
```
## no colors provided. using the following legend:
##         3         4
##   "black" "#DF536B"
```

```r
title(main = names(stmaps)[9], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[13]][[1]], ftype = "off")
```

```
title(main = names(stmaps)[13], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[14]][[1]], ftype = "off")
title(main = names(stmaps)[14], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[15]][[1]], ftype = "off")
title(main = names(stmaps)[15], font.main = 2, cex.main = 0.75, line = -0.3)
plotSimmap(stmaps[[16]][[1]], ftype = "off")
```

```
## no colors provided. using the following legend:
##         1         2
##   "black" "#DF536B"
```

```
title(main = names(stmaps)[16], font.main = 2, cex.main = 0.75, line = -0.3)
```



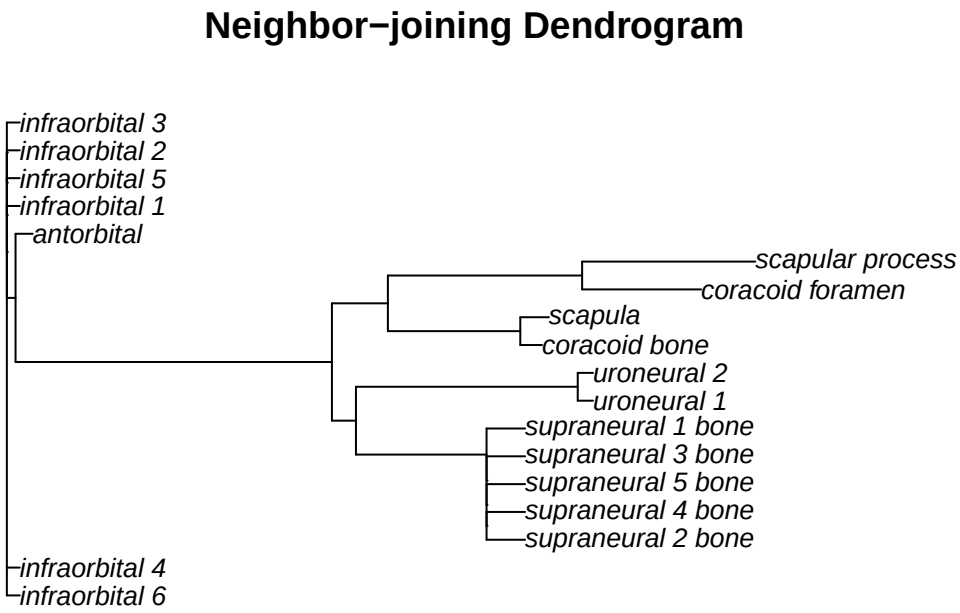
STEP 6. Exploring semantic properties of the data set.

One promising feature of semantically enriched data sets, i.e., phenotypes and anatomical entities from phylogenetic character matrices annotated with ontology terms, is that we can explore semantic properties such as different types of relations (e.g., **part_of**, **is_a**, **develops_from**) among anatomical terms.

First, we can visually assess the semantic similarity among ontology terms annotated to anatomical entities in our data set. Semantic similarity is a measure of relatedness between ontology terms based on their shared properties and the underlying ontology structure. In our example, we can use semantic similarity to investigate evolutionary patterns observed in our data set. For example, if anatomical entities evolving in a similar fashion share any underlying properties based on the semantic similarity of ontology terms used to describe them. For instance, all bones that are '**part_of**' the 'cranium' might evolve faster/slower than bones that are '**part_of**' the 'pectoral girdle'.

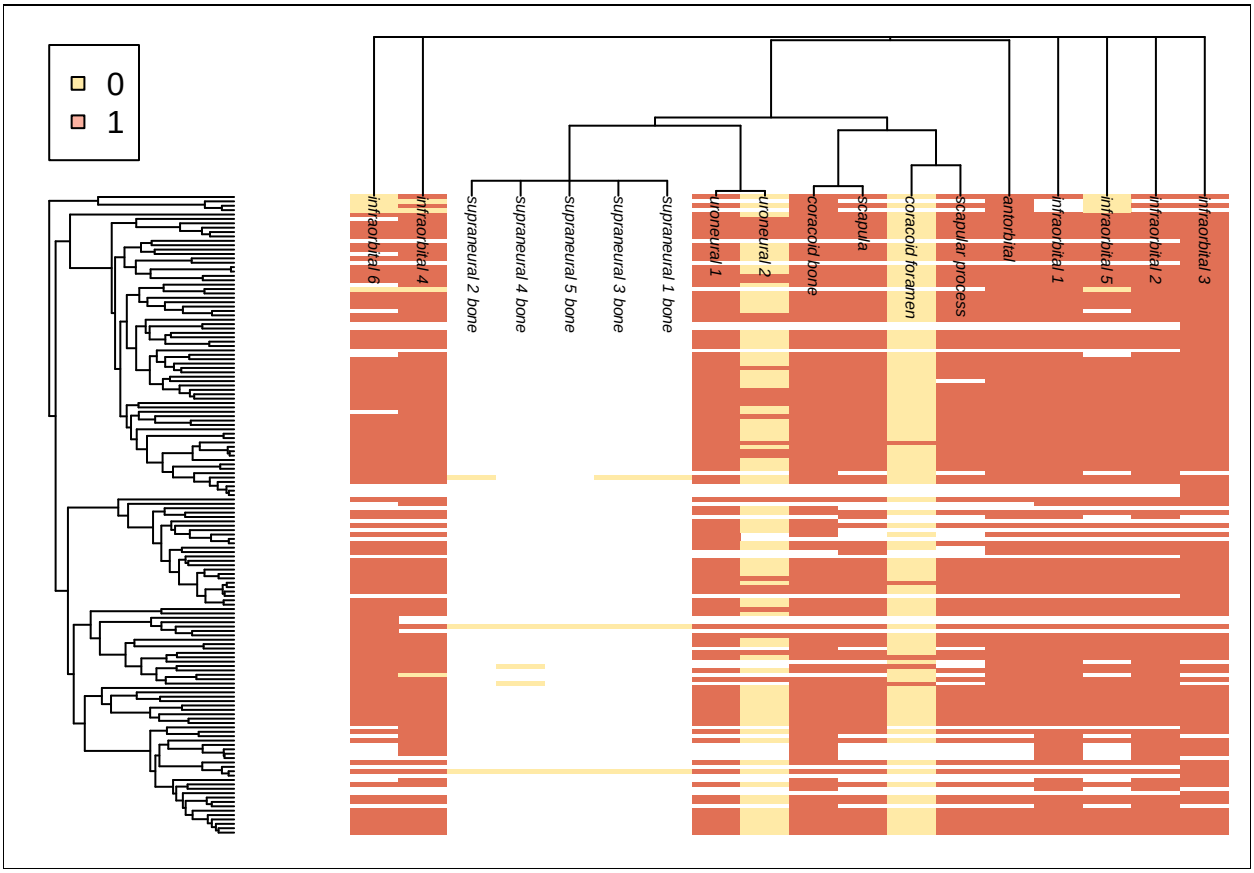
Let's build a clustering dendrogram based on the semantic similarity among ontology terms annotated to the anatomical entities in our data set.

```
trait.tree <- makeTraitTree(td, method = "nj")
plot(trait.tree, cex = 0.8, main = "Neighbor-joining Dendrogram")
```



We can also investigate if semantic properties of the anatomical terms in our data set are associated with the phylogenetic structure of the Characidae phylogeny. For example, if some regions of the Characidae phylogeny show different patterns of absences or presences of bones; or if some semantically similar terms referring to related anatomical entities (i.e., groups in the clustering dendrogram) show similar patterns across different regions of the fish phylogeny (i.e., clades). For instance, bones that are '**part_of**' the 'cranium' might be lost more frequently in some groups of Characidae whereas bones that are '**part_of**' the 'pectoral girdle' might be lost more frequently in others.

```
ontologyHeatMap(td, njt = trait.tree, start = 1)
```


0
1
infraorbital 6
infraorbital 4
Supraneural 2 bone
supraneural 4 bone
supraneural 5 bone
supraneural 3 bone
supraneural 1 bone
Uroneural 1
uroneural 2
coracod bone
scapula
coracod foramen
scapular process
antorbital
infraorbital 1
infraorbital 5
infraorbital 2
infraorbital 3

NULL