rphenoscate: Study Case 2

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

07 February, 2023

In this second study case, 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 (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 character 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 packages.

If you have not installed the package yet, then run the following:

```
remotes::install_github("phenoscape/rphenoscape", build_vignettes = TRUE)
```

You should also install its companion package *rphenoscape* that allows access to the Phenoscape KB.

```
remotes::install_github("uyedaj/rphenoscate", build_vignettes = TRUE)
```

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. If you do not have they installed, please do so.

```
library("ape")
library("phytools")
library("treeplyr")
library("tibble")
library("stringr")
```

STEP 2. Assembling the data set.

For this step, we will use some functions from *rphenoscape* to access phenotype data available in the Phenoscape Knowledgebase (KB). Phenoscape KB is a database of curated semantic information for more than 6.5k vertebrate species (mostly fishes) and about 14.5k phylogenetic characters, comprising a total of 256 phylogenetic 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) and Chang et al. (2019, 2020). You need to install the package, 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.

```
# Set search parameters.
taxa <- "Characidae"
entities <- 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 and 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) for more details.

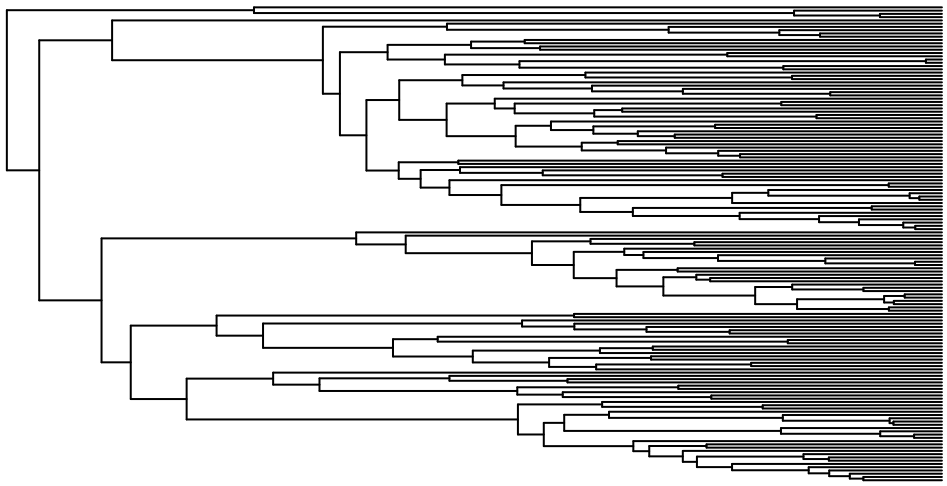
```
# Retrieve data from the Phenoscape KB.
nex <- get_ontotrace_data(taxon = taxa, entity = entities)
nex <- RNeXML::get_characters(nex)
```

Then, let's organize the data by adjusting taxon names and building the final data set.

```
# Organize labels.
mat <- bind_cols(taxon = rownames(nex), as_tibble(nex))
mat <- setNames(mat, c("taxon", colnames(nex)))

ftree$tip.label <- str_replace_all(ftree$tip.label, pattern = "_", replacement = " ")

# Build phylogenetic data set.
td <- make.treedata(ftree, mat, name_column = "taxon", as.is = TRUE)
```

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

```
# Plot tree.
plot.phylo(td$phy, show.tip.label = FALSE)
```



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

```
print_coverage(nex)
```

```
## # A tibble: 15 x 4
##    traits              coverage percentage average
##    <chr>                  <int>      <dbl>   <dbl>
##  1 infraorbital 1           361       0.86    0.99
##  2 infraorbital 2           361       0.86    0.99
##  3 infraorbital 4           360       0.85    0.95
##  4 infraorbital 6           334       0.79    0.95
##  5 infraorbital 3           325       0.77    0.99
##  6 scapula                  298       0.71    0.99
##  7 infraorbital 5           292       0.69    0.94
##  8 coracoid bone            289       0.68    0.99
##  9 coracoid foramen         269       0.64    0.04
## 10 uroneural 1              246       0.58    0.98
## 11 uroneural 2              243       0.58    0.25
## 12 scapular process         226       0.54    0.99
## 13 supraneural 3 bone        19       0.05    0.28
## 14 supraneural 4 bone        13       0.03    0.08
## 15 supraneural 5 bone         7       0.02    0.14
```

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

```
##         antorbital      coracoid bone   coracoid foramen     infraorbital 1
##               0.01               0.00               0.03               0.00
##     infraorbital 2     infraorbital 3     infraorbital 4     infraorbital 5
##               0.00               0.00               0.21               0.00
##     infraorbital 6            scapula   scapular process supraneural 1 bone
##               0.01               0.00               0.00               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
##               0.62               0.08
```

Most anatomical entities do not contain polymorphic character information, however for some of them, there is a substantial amount! For example: 'uroneural 1' (62%) and 'infraorbital 4' (21%). 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.

```
td$dat[td$dat == "0 and 1" | td$dat == "1 and 0"] <- "1"
```

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). 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$dat), function(x) anatomy_term_info(x)$id)

# Get the dependency matrix using rphenoscape.
dep_mat <- pa_dep_matrix(IRI, .names = "label", preserveOrder = TRUE)
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 et al. (2019) and Tarasov (2019, 2022). For a more in depth theoretical discussion on different types of dependencies among anatomical entities, please refer to Vogt (2018). 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, 2018).

```
# Amalgamate dependent traits.
amal_deps <- amalgamate_deps(dep_mat)

# Recode traits.
td_comb <- recode_traits(td, amal_deps, as.is = TRUE)
```

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. The function 'amalgamated_fits_corHMM' from *rphenoscate* is a wrapper that uses *corHMM* and the dependency structure described in the dependency matrix to fit models of discrete trait evolution accounting for trait dependencies. We only need the model fit objects obtained for each trait to perform (faster) stochastic character 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.

```
corhmm_fits <- amalgamated_fits_corHMM(td_comb, amal_deps)
```

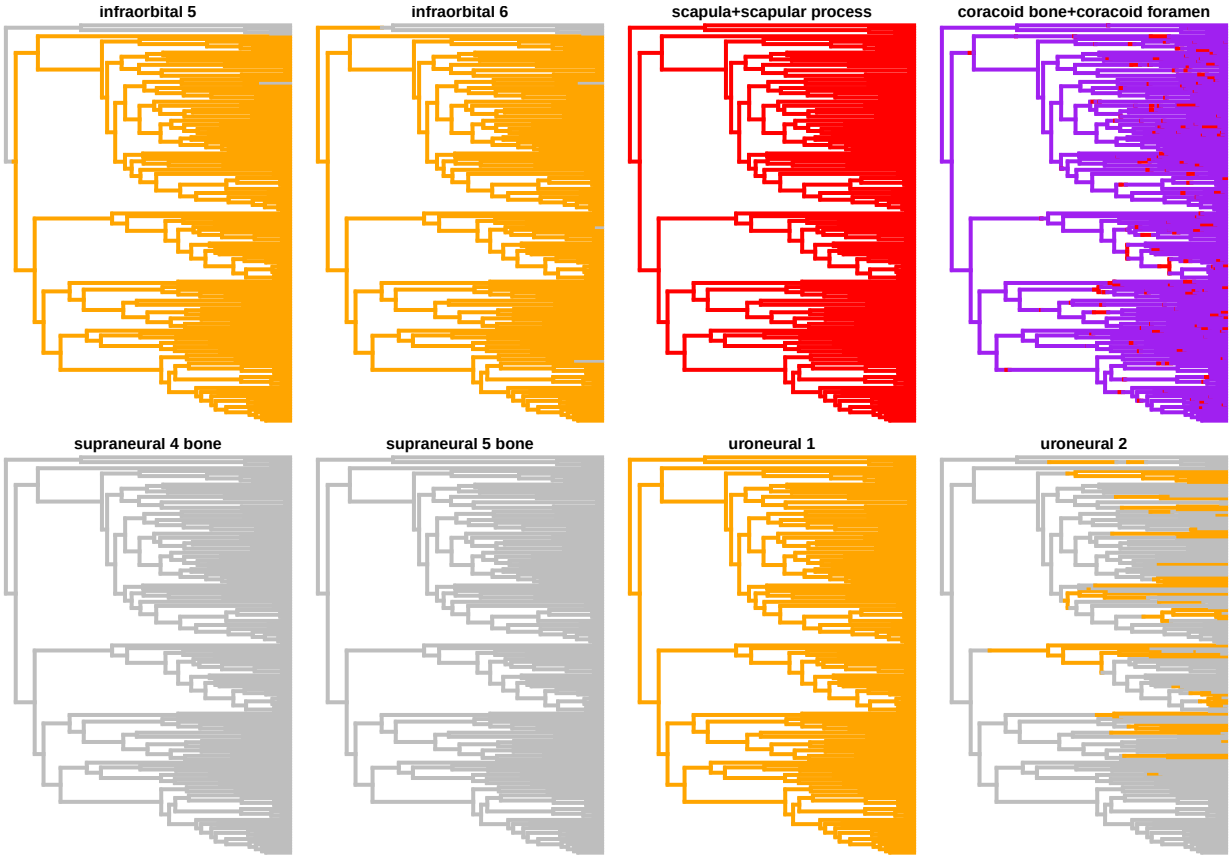
STEP 5. Sampling histories of trait evolution.

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.

```
stmaps <- amalgamated_simmaps_corHMM(corhmm_fits, nSim = 100)
names(stmaps) <- colnames(td_comb$dat)
```

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

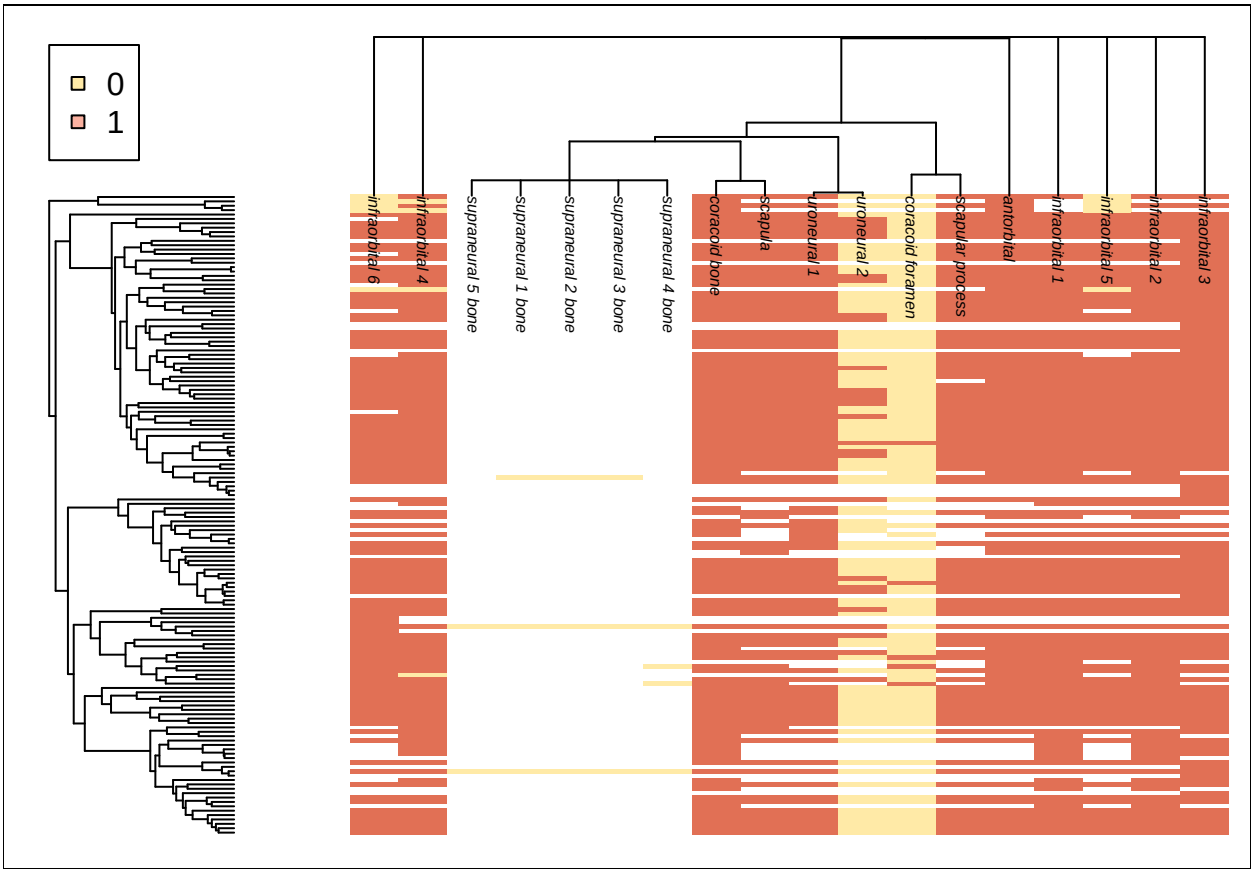
```
st_cols <- setNames(c("grey", "orange", "purple", "red"), c(0:3))

par(mfrow = c(2,4), mar = c(0.1,0.1,5.0,0.1))

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

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

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

plotSimmap(stmaps[[2]][[1]], ftype = "off", colors = st_cols)
title(main = names(stmaps)[2], font.main = 2, cex.main = 0.75, line = -0.3)

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

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

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

plotSimmap(stmaps[[16]][[1]], ftype = "off", colors = st_cols)
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. We can visually assess the relationships among ontology terms annotated to anatomical entities in our data set using some measure of similarity. Semantic similarity is a measure of relatedness between ontology terms based on their shared properties and underlying ontology structure (e.g., Resnik, Jaccard). In our example, we can use semantic similarity to investigate evolutionary patterns observed in our data set. For example, do bones from particular body regions get lost more frequently than others in this particular group of fishes? 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, do some clades in the Characidae phylogeny show different patterns of absences or presences of bones? 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.

```
trait.tree <- makeTraitTree(td, method = "nj")
suppressWarnings(ontologyHeatMap(td, njt = trait.tree, start = 1, show.tip.label = FALSE))
```




```
## NULL
```

References

Chang, J., Rabosky, D. L., and Alfaro, M. E. (2020). Estimating diversification rates on incompletely sampled phylogenies: theoretical concerns and practical solutions. *Systematic Biology*, 69(3):602–611.

Chang, J., Rabosky, D. L., Smith, S. A., and Alfaro, M. E. (2019). An r package and online resource for macroevolutionary studies using the ray-finned fish tree of life. *Methods in Ecology and Evolution*, 10(7):1118–1124.

Dececchi, T. A., Balhoff, J. P., Lapp, H., and Mabee, P. M. (2015). Toward synthesizing our knowledge of morphology: using ontologies and machine reasoning to extract presence/absence evolutionary phenotypes across studies. *Systematic biology*, 64(6):936–952.

Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E., and Haendel, M. A. (2012). Uberon, an integrative multi-species anatomy ontology. *Genome biology*, 13(1):1–20.

Rabosky, D. L., Chang, J., Title, P. O., Cowman, P. F., Sallan, L., Friedman, M., Kaschner, K., Garilao, C., Near, T. J., Coll, M., et al. (2018). An inverse latitudinal gradient in speciation rate for marine fishes. *Nature*, 559(7714):392–395.

Tarasov, S. (2019). Integration of anatomy ontologies and evo-devo using structured markov models suggests a new framework for modeling discrete phenotypic traits. *Systematic biology*, 68(5):698–716.

Tarasov, S. (2022). New phylogenetic markov models for inapplicable morphological characters. *bioRxiv*.

Tarasov, S., Mikó, I., Yoder, M. J., and Uyeda, J. C. (2019). Paramo: A pipeline for reconstructing ancestral anatomies using ontologies and stochastic mapping. *Insect Systematics and Diversity*, 3(6):1.

Vogt, L. (2018). The logical basis for coding ontologically dependent characters. *Cladistics*, 34(4):438–458.