OntoPhylo Tutorial: PARAMO

Diego S. Porto and Sergei Tarasov

20 June, 2023

Load packages.

If you are starting a new R session, then reload *ontophylo*.

```
library(ontophylo)
```

And all the other packages. If you have not them installed, please do so by running `install.packages()`.

```
library(ape)
library(phytools)
library(geiger)
library(corHMM)
library(tidyverse)
```

Load data.

First, load the data from the previous tutorial.

```
load("RData/step1_annot.RData")
```

STEP 1. Sampling individual stochastic character histories.

Now, we will sample character histories for each individual character using stochastic character mapping (Huelsenbeck et al., 2003). For the purposes of the PARAMO pipeline (Tarasov et al., 2019), characters should be independent from each other. If dependencies are present, usually indicated as inapplicable character states in a phylogenetic character matrix, then models should be combined and characters recoded accordingly, as discussed in Tarasov (2019, 2020, 2023) (see also Simões et al. (2023)). For example, one character may describe the absence/presence of an anatomical structure (e.g. mandible) and another its shape (e.g. straight or curved). Therefore, any character state describing the shape of the mandible depends on the presence of the mandible. A discussion on how to model and code dependent characters is out of the scope of this workshop (see references above). For simplicity, here and in the following tutorials, the selected characters from the modified data set have no dependencies.

Finally, for performing stochastic character mapping we will use a phylogenetic tree modified from Klopfstein et al. (2013).

First, we import the tree, set up some parameters, and create folders to store objects. We will sample 100 stochastic maps per character. The resolution parameter controls the size of episodic bins to discretize the branches of the trees.

```r
# Import tree.
hym_tree <- readRDS("data/hym_tree.RDS")

# Set some parameters.
n_stm = 100
res = 500

# Create folder to store simmap objects.
dir.create("stmaps_discr")
```

Then, we run stochastic character mapping in *corHMM* (Beaulieu et al., 2013; Boyko and Beaulieu, 2021) or *phytools* (Revell, 2012). The same analyses can be performed in a Bayesian framework with RevBayes (Höhna et al., 2016). For simplicity, let's keep everything within R. In the following lines of code, for each character, we will fit three models with `corHMM` (ER, SYM, ARD), select the best one, and then obtain the mappings. Finally, we will discretize the trees and save them as RDS files. This last step is not strictly necessary, since everything can be kept within R, but exporting heavy objects to external folders helps to alleviate the memory cache. This step can take 5 to 10 minutes to resume depending on your computer. Alternatively, you can skip step 1 and load the RDS file starting step 2.

```r
# Set the same  RNG seed as last tutorial.
set.seed(42)

# Set a vector with all character names.
br_chars <- unlist(HYM_ANAT, use.names = FALSE)

for (i in 1:length(br_chars)) {

  cat(paste0("\n", "Working on: ", br_chars[i], ": ", Sys.time(), "\n"))

  # Get character vector.
  char <- cbind(hym_mat$taxa, hym_mat[[br_chars[i]]])

  # Set candidate models.
  models <- c("ER", "SYM", "ARD")

  fit_corHMM <- vector(mode = "list", length = length(models))

  for (j in 1:length(models)) {

    # Fit model with corHMM.
    fit_corHMM[[j]] <- corHMM(phy = hym_tree, data = char, model = models[[j]],
                              rate.cat = 1, root.p = "yang")

  }

  # Get best model.
  w <- aicw(sapply(fit_corHMM, function(x) x$AICc))[,3]

  # Set Q matrix.
  Q <- fit_corHMM[[min(which(w == max(w)))]]$solution

  # Simulate stochastic maps.
  stm <- makeSimmap(tree = hym_tree, data = char, model = Q, rate.cat = 1, nSim = n_stm)
  # Discretize trees.
  stm_discr <- lapply(stm, function(x) discr_Simmap_all(x, res = res) )
  stm_discr <- do.call(c, stm_discr)

  # Save RDS files.
  saveRDS(stm_discr, file = paste0("stmaps_discr/", br_chars[i], ".RDS"))

}
```

STEP 2. PARAMO: amalgamating stochastic character maps.

Now, we will finally amalgamate the individual character histories to describe the combined histories of all characters from each anatomical region.

First, let’s import all discretized maps of individual characters. This may require a lot memory depending on the size of trees, number of stochastic maps, and resolution parameter.

```
# Set a vector with all character names.
br_chars <- unlist(HYM_ANAT, use.names = FALSE)

# Create temporary list to store discretized maps from individual characters.
MAPS <- setNames(vector(mode = "list", length = length(br_chars)), br_chars)

# Import all discretized maps from characters of a given anatomical region.
for (k in 1:length(br_chars)) {

  MAPS[[k]] <- readRDS(paste0("stmaps_discr/", br_chars[[k]], ".RDS"))

}
```

Then we amalgamate stochastic maps by anatomical regions and by the entire phenome by running the main functions of the PARAMO pipeline.

The function `paramo` amalgamate stochastic character maps given a list of partitions `HYM_ANAT`, here defined in the previous tutorial by querying the HAO ontology, and the list of stochastic maps `MAPS`. Each element of `HYM_ANAT` should be a vector of names matching the names of the elements in `MAPS`. Each element in `HYM_ANAT` is a partition of the original data representing a group of characters.

Let’s check our partitions again.

```
HYM_ANAT
```

```
## $head
##  [1] "CH73" "CH57" "CH1"  "CH42" "CH11" "CH54" "CH31" "CH41" "CH7"  "CH67"
##
## $mesosoma
##  [1] "CH198" "CH372" "CH228" "CH104" "CH358" "CH233" "CH229" "CH252" "CH130"
## [10] "CH383"
##
## $metasoma
##  [1] "CH284" "CH285" "CH390" "CH396" "CH397" "CH280" "CH389" "CH288" "CH283"
## [10] "CH386"
```

And character statements.

```
lapply(HYM_ANAT, function(x) hym_annot %>% filter(char_id %in% x) %>% select(char) )
```

```
## $head
## # A tibble: 10 x 1
##    char
##    <chr>
##  1 Ocellar corona
##  2 Inner margin of torulus
##  3 Occipital sulcus and ridge
##  4 Tormae
##  5 Left mandible dentition
##  6 Dentition of right mandible relative to left
##  7 Glossal margin
##  8 Paraglossal basal setation
##  9 Sixth maxillary palpomere count
## 10 Galea division
##
## $mesosoma
## # A tibble: 10 x 1
##    char
##    <chr>
##  1 Propleuron  cervical lines
##  2 Foreleg  probasitarsal spur
##  3 Mesofurca  arms proximally
##  4 Metapleuron  paracoxal notches
##  5 Metapleuron  median longitudinal carina projection
##  6 Metapleuron  anterior paracoxal sulci ridges
##  7 Hind leg  tibial preapical spurs
##  8 Forewing  second branch of radial sector  RS2  vein presence
##  9 Hind wing  presence of sclerotized glabrous plate posterior to hamuli
## 10 Midleg tibial comb
##
## $metasoma
## # A tibble: 10 x 1
##    char
##    <chr>
##  1 Petiole  posteriorly
##  2 Petiole  transverse carina on T2
##  3 Petiole  T2 longitudinal internal carina
##  4 Petiole  S2
##  5 Petiole  longitudinal carina anteriorly on S2
##  6 Presence of glymma  laterope
##  7 Second abdominal sternum  first metasomal
##  8 Abdominal tergum 2  metasomal 1  overlapping abdominal tergum 3  metasomal 2
##  9 Valvilli presence
## 10 Second valvifer structure
```

In this case, the groups represent the main anatomical regions of the hymenopteran anatomy, but PARAMO can be used to amalgamated any group of characters based on different hypotheses. For example, instead of anatomical regions, a researcher can groups traits associated with living in different types of environments (e.g. aquatic, terrestrial) or different types of traits (e.g., morphology, behavior). Each element in `MAPS` is a list of N stochastic maps obtained for each character. In our example, we have 10 characters per anatomical region, so 30 characters in total, 100 maps each.

Let's amalgamate characters by anatomical regions first.

```r
# Amalgamate by anatomical regions.
stm_amalg_anato <- paramo(rac_query = HYM_ANAT, tree.list = MAPS, ntrees = n_stm)
```

The function `paramo.list` is more flexible. Given a list of stochastic maps `MAPS`, you can simply provide a vector with the names of characters to amalgamate. Since we want to amalgamate all the characters to obtain the amalgamation of the entire phenome, we provide all names `br_chars`.

```r
# Amalgamate all individual characters as a single complex character.
stm_amalg_pheno <- paramo.list(br_chars, tree.list = MAPS, ntrees = n_stm)
```

Let's plot a sample of stochastic map from the head, mesosoma, and metasoma.

```r
# Create a folder to store figures.
dir.create("figures")

# HEAD.
png(paste0("figures/stm_head.png"),
    units = "in", width = 7, height = 7, res = 300)
plotSimmap(stm_amalg_anato$head[[5]],
           get_rough_state_cols(stm_amalg_anato$head[[5]]),
           lwd = 3, pts = F,ftype = "off", ylim = c(0,90))
title(main = "HEAD", font.main = 2, line = -1)
dev.off()

# MESOSOMA.
png(paste0("figures/stm_meso.png"),
    units = "in", width = 7, height = 7, res = 300)
plotSimmap(stm_amalg_anato$mesosoma[[5]],
           get_rough_state_cols(stm_amalg_anato$mesosoma[[5]]),
           lwd = 3, pts = F,ftype = "off", ylim = c(0,90))
title(main = "MESOSOMA", font.main = 2, line = -1)
dev.off()

# METASOMA.
png(paste0("figures/stm_meta.png"),
    units = "in", width = 7, height = 7, res = 300)
plotSimmap(stm_amalg_anato$metasoma[[5]],
           get_rough_state_cols(stm_amalg_anato$metasoma[[5]]),
           lwd = 3, pts = F,ftype = "off", ylim = c(0,90))
title(main = "METASOMA", font.main = 2, line = -1)
dev.off()
```

And finally, save all the results.

```r
# Save RDS files.
saveRDS(stm_amalg_anato, file = paste0("data/stm_amalg_anato.RDS"))
saveRDS(stm_amalg_pheno, file = paste0("data/stm_amalg_pheno.RDS"))
save.image("RData/step2_paramo.RData")
```

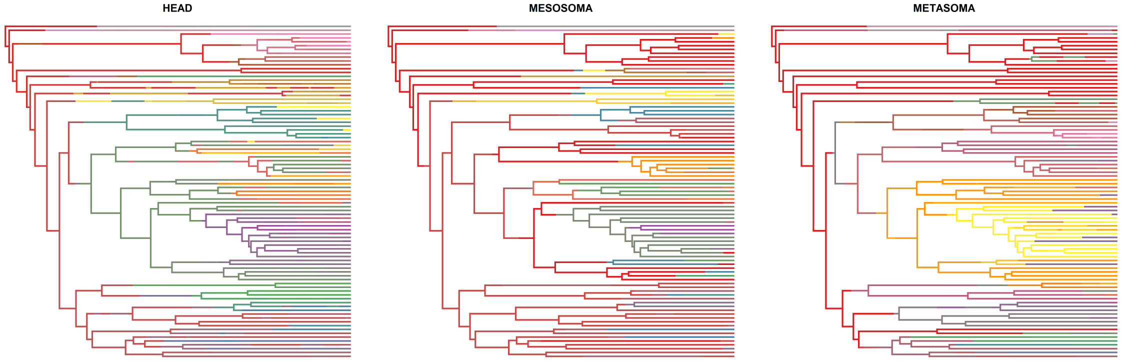

Figure 1: A sample of stochastic map of the head, mesosoma, and metasoma characters.

References

Beaulieu, J. M., O'Meara, B. C., and Donoghue, M. J. (2013). Identifying hidden rate changes in the evolution of a binary morphological character: the evolution of plant habit in campanulid angiosperms. *Systematic biology*, 62(5):725–737.

Boyko, J. D. and Beaulieu, J. M. (2021). Generalized hidden markov models for phylogenetic comparative datasets. *Methods in Ecology and Evolution*, 12(3):468–478.

Höhna, S., Landis, M. J., Heath, T. A., Boussau, B., Lartillot, N., Moore, B. R., Huelsenbeck, J. P., and Ronquist, F. (2016). Revbayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic biology*, 65(4):726–736.

Huelsenbeck, J. P., Nielsen, R., and Bollback, J. P. (2003). Stochastic mapping of morphological characters. *Systematic biology*, 52(2):131–158.

Klopfstein, S., Vilhelmsen, L., Heraty, J. M., Sharkey, M., and Ronquist, F. (2013). The hymenopteran tree of life: evidence from protein-coding genes and objectively aligned ribosomal data. *PLoS One*, 8(8):e69344.

Revell, L. J. (2012). phytools: an r package for phylogenetic comparative biology (and other things). *Methods in ecology and evolution*, (2):217–223.

Simões, T. R., Vernygora, O. V., de Medeiros, B. A., and Wright, A. M. (2023). Handling logical character dependency in phylogenetic inference: Extensive performance testing of assumptions and solutions using simulated and empirical data. *Systematic biology*, page syad006.

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. (2020). The invariant nature of a morphological character and character state: insights from gene regulatory networks. *Systematic biology*, 69(2):392–400.

Tarasov, S. (2023). New phylogenetic markov models for inapplicable morphological characters. *Systematic biology*.

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.