OntoPhylo Tutorial: Application 3 - Rates of anatomical regions

Diego S. Porto and Sergei Tarasov

23 August, 2023

Load packages.

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

```
library(ontophylo)
```

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

```
library(phytools)
library(grImport)
library(ontologyIndex)
library(tidyverse)
library(RColorBrewer)
```

Load data.

First, load the data from the tutorial 3.

```
load("RData/step3_nhpp.RData")
```

Overview on graphics with OntoPhylo.

In this tutorial, we will explore some graphical functions from *grImport* Murrell (2009) to work with vector images in R. *OntoPhylo* is able to color-shade the layers of vector images annotated with ontology terms based on the evolutionary rates (or any other statistics of interest) of different anatomical entities or anatomical regions. To work with vector images, you have to install *GhostScript* in your computer to be able to convert PostScript files, for example those exported from Illustrator, to an XML file that can be read in R. Vector images of your organism of interest can be produced using software like Adobe Illustrator or GIMP and exported as a PostScript file (.ps or .eps).

If you do not want to install *GhostScript*, then just import the XML file already converted.

```
hym_img <- readRDS("data/hym_img.RDS")
```

STEP 1. Converting files and organizing data.

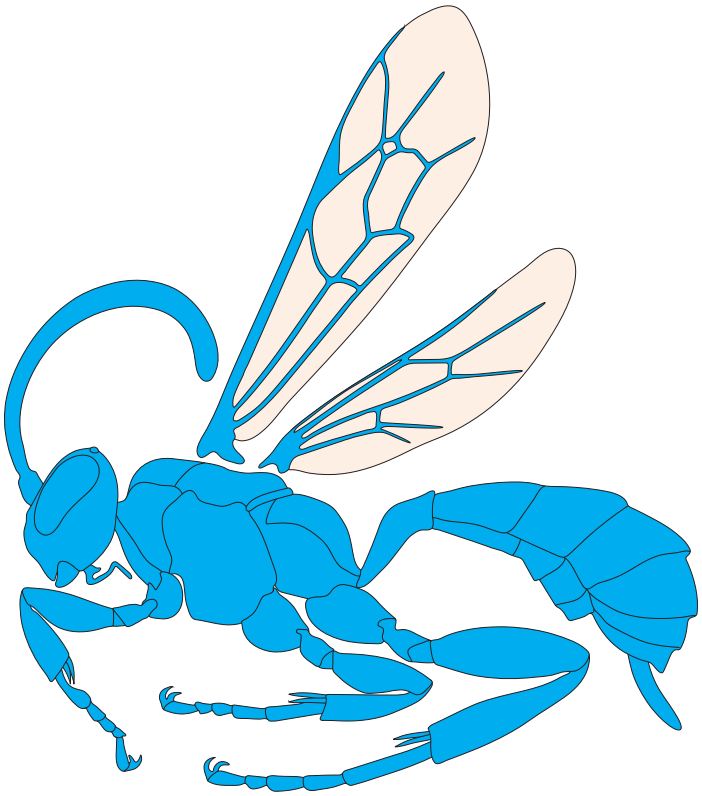
First, let's convert the PostScript file to XML (if necessary).

```
# Process postscript vector image.
PostScriptTrace(file = "data/hym.ps", outfilename = "data/hym.xml")

# Import XML file.
hym_img <- readPicture("data/hym.xml")
```

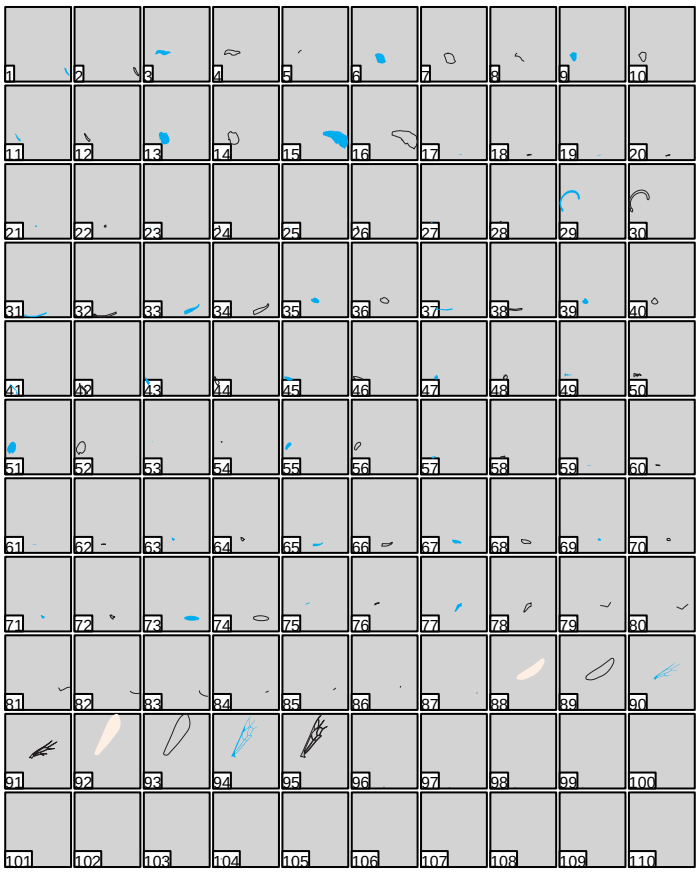
Let's check our wasp image.

```
grid.picture(hym_img)
```



You can also check all the layers of the image (picture paths) by using the function `picturePaths` from *grImport*. The picture paths are important because they will allow us to link each layer to a particular ontology term. For example, if your vector image has several layers, each corresponding to a particular anatomical entity. This will allow us to recover the layers corresponding to ontology terms from a particular query of interest. In our case, we want to query all layers that correspond to anatomical entities pertaining to the head, mesosoma and metasoma. Let's first check the figure paths.

```
picturePaths(hym_img)
```



Note that figure paths will depend on each particular vector image. The matches between layers and anatomy ontology terms has to be done manually. See the example data `hym_graph` below.

```
# Load csv table with information on vector image layers.
hym_graph <- readRDS("data/hym_graph.RDS")
hym_graph
```

```
## # A tibble: 46 x 4
##    Term             ID          layer_id       wing_blades
##    <chr>            <chr>       <chr>                <dbl>
##  1 antenna          HAO:0000101 29                      NA
##  2 coxa             HAO:0000228 <NA>                    NA
##  3 cranium          HAO:0000234 29, 51, 53, 55          NA
##  4 eye              HAO:0000217 55                      NA
##  5 female genitalia HAO:0000324 1                       NA
##  6 femur            HAO:0000327 <NA>                    NA
##  7 fore leg         HAO:0000349 <NA>                    NA
##  8 fore wing        HAO:0000351 94                      92
##  9 head             HAO:0000397 <NA>                    NA
## 10 hind leg         HAO:0000399 <NA>                    NA
## # ... with 36 more rows
```

Now, let's query for the layers corresponding to the anatomical entities of interest.

```r
# Define query terms for anatomical regions.
query_terms <- set_names(as.list(c("HAO:0000397", "HAO:0000576", "HAO:0000626")), query_anat)

# Query layers.
anat_layers <- get_vector_ids_list(terms = query_terms, ONT = onto, GR = hym_graph)
anat_layers
```

```
##     head     head     head     head     head mesosoma mesosoma mesosoma
##       29       51       53       55       49       94       90       39
## mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma
##       67        3       13       37       59       61       65       21
## mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma
##       63       35       73       75        6       31       17       19
## mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma mesosoma
##       33       69       71       47       45        9       11       41
## mesosoma mesosoma mesosoma mesosoma mesosoma metasoma metasoma metasoma
##       23       25       43       27       57        1       15       77
```

STEP 2. Get statistics from branches.

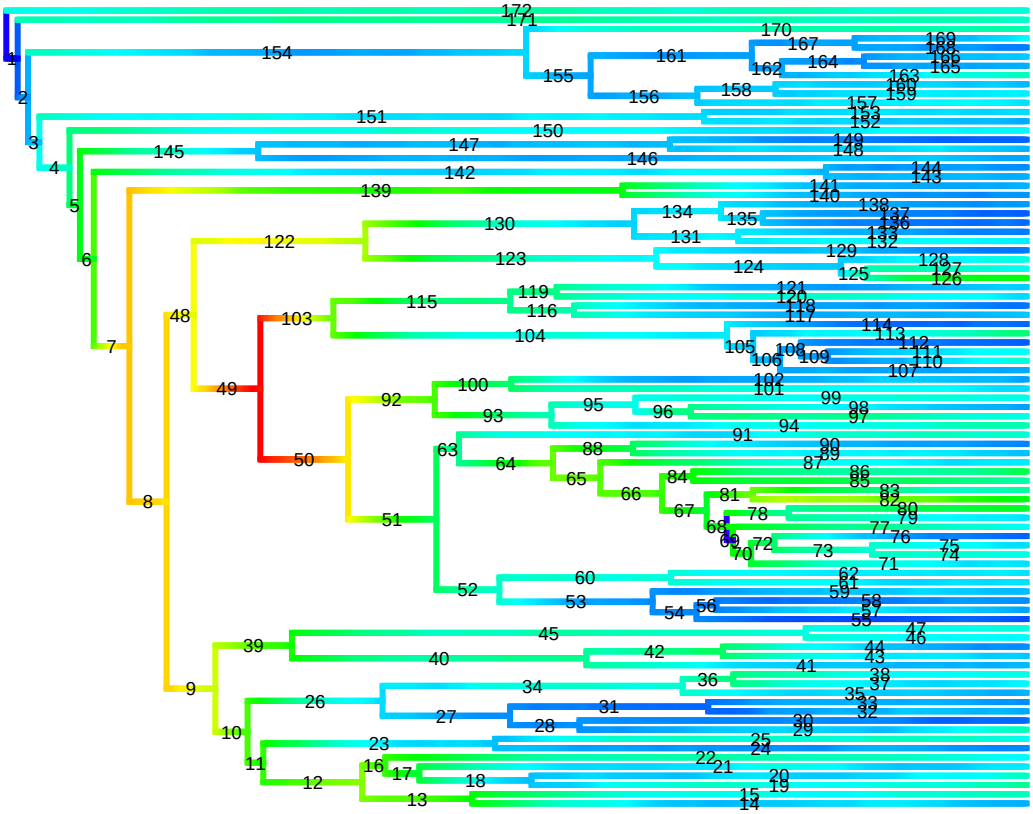
Then, we process the branch data to extract the mean rates estimated in tutorial 3.

```r
# Extract mean lambda from branches and calculate mean across all maps.
hd_stat <- lapply(edge_KDE_hd$lambda.mean.loess, function(x) mean(x) )  %>% unlist()
ms_stat <- lapply(edge_KDE_ms$lambda.mean.loess, function(x) mean(x) )  %>% unlist()
mt_stat <- lapply(edge_KDE_mt$lambda.mean.loess, function(x) mean(x) )  %>% unlist()
ph_stat <- lapply(edge_KDE_ph$lambda.mean.loess, function(x) mean(x) )  %>% unlist()
```

STEP3. Plot the figures.

Now we can select some of the branches to see how fast different anatomical regions are evolving. First, let's check the branch rates for the entire phenome and get the edge ids.

```r
# Plot contmap.
plot.contMap(nhpp_lambda_mean_ph, lwd = 3, outline = F, legend = F,
             ftype = "off", plot = F, mar = c(0.1, 3.45, 0.1, 0.35))
edgelabels(frame = "none", col = "black", cex = 0.6)
```



The phenome seems to be evolving at higher rates on some edges, for example 7 and 8, and at much higher rates on 49 and 50. Let's check what is happening in different anatomical regions.

Set some preliminary parameters for the figures.

```
# Select target branches.
br = c(7,8,49,50)

# Set limits for scale (based on phenome rates).
scale_lim <- range(ph_stat)

# Set a nice color palette.
hm_palette <- colorRampPalette(brewer.pal(9, "Spectral") %>% rev(), space = "Lab")
```

Export and plot all figures.

```
for (i in 1:length(br)) {

  # Get branch-specific statistics for each anatomical region
  plot_stat <- set_names(c(hd_stat[[br[[i]]]], ms_stat[[br[[i]]]],
                           mt_stat[[br[[i]]]]), query_anat)

  # Save a png.
  png(paste0("figures/", "pheno_br_", br[[i]],".png"), units = "in",
      width = 7, height = 7, res = 300)
```

```
  # Plot image.
  anat_plot(hym_img, anat_layers, plot_stat, color_palette = hm_palette(100),
            scale_lim = scale_lim)
  title(main = paste0("BR_", br[[i]]), font.main = 2, line = 1)

  dev.off()

}
```

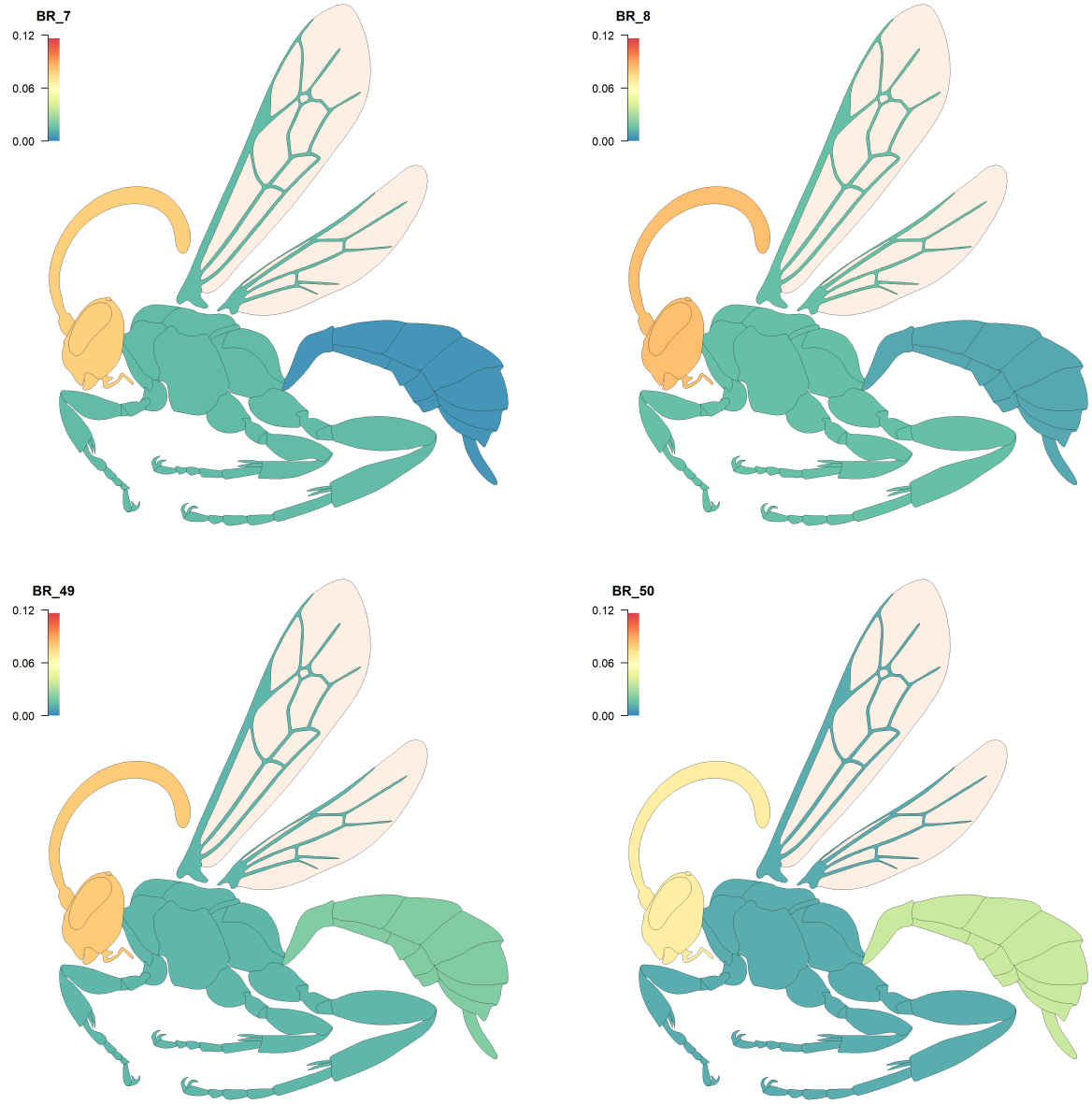


Figure 1: From left to right, top to bottom, rates for the anatomical regions on branches 7, 8, 49, and 50.

And finally, save all the results obtained so far.

```r
save.image("RData/step5_anato.RData")
```

References

Murrell, P. (2009). Importing vector graphics: The grimport package for r. *Journal of Statistical Software*, 30:1–37.