PARAMO pipeline

***P**hylogenetic **A**ncestral **R**econstruction of **A**natomy by **M**apping **O**ntologies*

PARAMO pipeline requires three initial pieces of data: character matrix, dated phylogeny, and anatomy ontology. Herein, we use a of 9 characters and 87 species from the large-scale phylogeny of Hymenoptera (M. J. Sharkey et al. 2012), this dataset was slightly modified for the demonstrative purpose. For reconstructing character histories, we use the dated phylogeny of (Klopfstein et al. 2013), and for characters-ontology linking, we use Hymenoptera Anatomy Ontology [HAO (Yoder et al. 2010)]. In this demonstration, we are interested in constructing the amalgamated characters for the three leveles of amalgamation (=anatomical hierarchy): anatomical dependecies (ADs), body regions (BRs) and entire phenotype (EF). At the BR level, three main body regions are considered – "head", "legs" and "wings".

STEP 1. Initial character matrix

Our initial character matrix consists of a sample of 9 characters. The nexus file of the intial matrix can be found at `STEP_1/Step1_matrix.nex` and viewed using, for example, Mesquite. Below, in describing characters, the follwoing notation is used $C_{\#}\{S_1, S_2, ...\}$ where $C_{\#}$ stands for a character ID and $S_1, S_2, ...$ stand for character states. Let us have a look at the character report.

Table 1: Initial characters obtained at Step 1.

ID	ID*	CHAR_STATEMENT	STATE_0	STATE_1	DEPENDENCY
C1	3	Notch on medial margin of eye	absent	present	no
C2	23	Position of labrum	anterior	posterior	C2{0,1} < C3{0}
C3	25	Labrum	present	absent	C3{0} > C2{0,1}
C4	353	Forewing costal and radial vein fusion	not_fused	fused_along_their_lengths	no
C5	363	Hind wing subcostal SC vein, absent	no	yes	C5{0,1} <>C6{1,0}
C6	363	Hind wing subcostal SC vein, present	yes	no	C5{0,1} <>C6{1,0}
C7	380	Inner posterior mesotibial spur	simple	modified into a calcar	no
C8	381	Foretibial apical sensillum	present	absent	no
C9	382	Metatibial apical sensillum	present	absent	no

Note:
ID - character ID in this study.
IDR*- character ID in Sharkey et al. (2012).

Note, the two pairs of characters in the matrix $\{C_2, C_3\}$ and $\{C_5, C_6\}$ are subjected to anatomical dependencies.

- $C_2\{0, 1\}$ is hierrachically (anatomically) dependent on $C_3\{0\}$. This dependecy is indicated by $<$ or $>$ depending on the direction of the dependecy. The hierrachical dependecy means that states $C_2\{0, 1\}$ appear immediately as C_3 swithches to the state $C_3\{0\}$.
- C_5 and C_6 are subjected to synchronous changes, which means tha the states of this characters are mutually exclusive and hence dependent because one trait is coded using absent/present coding. The synchronous dpenedecy is indicated as $<>$; the notation $C_5\{0, 1\} <> C_6\{1, 0\}$ means – if C_5 is $\{0\}$ then C_6 is $\{0\}$, and if C_5 is $\{1\}$ then C_6 is $\{1\}$.

Step 2. Incorporating anatomical dependencies: constructing amalgamations at the AD level

The two pairs of anatomically depenent characters – C_2, C_3 and C_5, C_6 have to be apropriately amalgamated into single characters to adeuqtely model the dependecies. The amalgamation produces the following two

characters (see the text in the article):

- $C_3 \oplus C_2 = C_{3,2}\{00, 01, 10, 11\}$. $C_{\{}3, 2\}$ is coded in the matrix as $C_{3,2}\{0\&1, 0\&1, 2, 3\}$.
- C_5 and C_6 are combinded into $C_{\{}5, 6\}$. The synchronous dependency between these characters has to be eliminated that gives the character $C_{5,6}$ without changing the state pattern.

The recoding of dependent characters constructs the amalgamated characters at the AD level. If a character does not display any dependencies then we treat it as correctly amalgamated at the AD level by default. Let's have a look at the new character report.

```
CH2<-read.csv(file="STEP_2/Char_info_step_2.csv", header = T, as.is=T, check.names=F)

kable(CH2, booktabs=TRUE,
      caption = 'Characters recoded for the AD level amalgamation (Step 2).')%>%
  kable_styling(full_width = F, font_size=8, latex_options="hold_position") %>%
  column_spec(1, bold = T) %>%
  column_spec(6, width = "5em") %>%
  column_spec(7, width = "5em") %>%
  footnote(c("ID - character ID in this study.",
    "IDR*- character ID in Sharkey et al. (2012)."))
```

Table 2: Characters recoded for the AD level amalgamation (Step 2).

ID	ID*	CHAR_STATEMENT	STATE_0	STATE_1	STATE_2	STATE_3
C1	3	Notch on medial margin of eye	absent	present		
C3,2	25, 23	Labrum + Position of labrum	absent, anterior	absent, posterior	present, anterior	present, posterior
C4	353	Forewing costal and radial vein fusion	not_fused	fused_along_their_lengths		
C5,6	363	Hind wing subcostal SC vein, present	present	absent		
C7	380	Inner posterior mesotibial spur	simple	modified into a calcar		
C8	381	Foretibial apical sensillum	present	absent		
C9	382	Metatibial apical sensillum	present	absent		

Note:
ID - character ID in this study.
IDR*- character ID in Sharkey et al. (2012).

The new matrix of the recoded characters can be found in `STEP_2/Step2_matrix.nex` and `STEP_2/matrix.csv`. Here is the part of this matrix.

```
# characters matrix
MT<-read.csv("STEP_4/matrix.csv", header = T, row.names=1, as.is=T, check.names=F)
kable(MT[c(1:10),], booktabs=TRUE,
      caption = 'Characters matrix obtained at Step 2.')%>%
  kable_styling(full_width = F, font_size=8, latex_options="hold_position")
```

STEP 3. Linking anatomical characters with ontology

Having initial characters properly coded to account for the anatomical dependecies, let's move on character-ontology linking. The table below shows the Hymenoptera characters linked with the terms of Hymenoptera Anatomy Ontology HAO. This table will be used in *"Retrieve all characters"* (*RAC*) query that retrives all characters asscoaited with an input ontology term.

```
AN<-read.csv(file="STEP_3/Char_annotation.csv", header = T, as.is=T, check.names=F)
kable(AN, booktabs=TRUE,
```

Table 3: Characters matrix obtained at Step 2.

	C1	C3-2	C4	C5-6	C7	C8	C9
Acanthochalcis	0	3	0	1	0	0	1
Aleiodes	1	3	1	1	0	?	?
Anacharis	0	3	0	1	0	?	?
Archaeoteleia	0	3	0	1	0	?	?
Athalia	0	3	0	1	0	0	1
Aulacus	0	3	0	1	0	1	1
Australomymar	?	3	?	?	?	?	?
Austroserphus	0	3	0	1	0	?	?
Belyta	0	3	0	1	0	?	?
Brachygaster	0	?	0	1	0	1	1

```
    caption = 'Characters linked with HAO terms (Step 3).')%>%
  kable_styling(full_width = F, font_size=8, latex_options="hold_position")
```

Table 4: Characters linked with HAO terms (Step 3).

ID	ID2	CHAR_STATEMENT	HAO_ID	HAO_ID_NAME
C1	1	Notch on medial margin of eye	HAO:0000234	cranium
C3,2	3,2	Labrum + Position of labrum	HAO:0000639	mouthparts
C4	4	Forewing costal and radial vein fusion	HAO:0000351	fore wing
C5,6	5,6	Hind wing subcostal SC vein, present	HAO:0000400	hind wing
C7	7	Inner posterior mesotibial spur	HAO:0001351	mesotibia
C8	8	Foretibial apical sensillum	HAO:0000350	fore tibia
C9	9	Metatibial apical sensillum	HAO:0000631	metatibia

To run *RAC*, we use `ontologyIndex` package and a set of preccoked `R` functions located in `R_PARAMO/PARAMO_functions.R`. For our demonstrative purposes *RAC* is supposed to work with the BR and EF levels of the amalgamation. So, let's test our query for BR ("head", "wings" and "legs") and EF terms. First of all, we need to make character-ontology to be a part of the ontolgy graph.

```
library("ontologyIndex")
```

```
## Warning: package 'ontologyIndex' was built under R version 3.5.2
```

```
source("R_PARAMO/PARAMO_functions.R")

# opening HAO file ("BFO:0000050" is part_of relatinship)
ONT<-get_OBO("STEP_3/HAO.obo", extract_tags="everything", propagate_relationships = c("BFO:0000050", "is

# let's create  "annot" list of anotations from the annotation table
char_id<-paste0("CHAR:", AN$ID2)
annot<-set_names(table2list(AN[,c(2,4)]), char_id)

# next we make the annotations to be the part of the ontology object ONT
ONT$terms_selected_id<-annot
```

Now, we can construct and query the vectors of HAO terms that correspond to the focal BRs and EF. We will use the results of this query at Step 5.

```
# BR level, HAO terms
levelBR<-set_names(c("HAO:0000397", "HAO:0001089", "HAO:0000494"),
                   c("head", "wings", "legs") )
```

```
# EF level, HAO terms
levelEF<-set_names(c("HAO:0000012"),
                   c("whole_organism") )

# we use get_descendants_chars to get the set of all anatomcal characters that descend from a particula
#get_descendants_chars(ONT, annotations="manual", terms="HAO:0000012")

# now we can use RAC query for the BR and EF levels using the ontology
#BR level
BR<-lapply(levelBR, function(x)
  get_descendants_chars(ONT, annotations="manual", terms=x)  )
cat("BR LEVEL\n")
```

```
## BR LEVEL
```

```
BR
```

```
## $head
## [1] "CHAR:1"   "CHAR:3,2"
##
## $wings
## [1] "CHAR:4"   "CHAR:5,6"
##
## $legs
## [1] "CHAR:7" "CHAR:8" "CHAR:9"
```

```
#EF level
EF<-lapply(levelEF, function(x)
  get_descendants_chars(ONT, annotations="manual", terms=x)  )
cat("EF LEVEL\n")
```

```
## EF LEVEL
```

```
EF
```

```
## $whole_organism
## [1] "CHAR:1"   "CHAR:3,2" "CHAR:4"   "CHAR:5,6" "CHAR:7"   "CHAR:8"
## [7] "CHAR:9"
```

STEP 4. Inference: linking characters with models and tree

At this step, we need to construct data files for analysing the set of our seven individual characters (obtained at Step 2) using RevBayes. The three files have to be created for each character: (1) chraracter file `.char` (`STEP_4/RevBayes/data/`), (2) RevBayes script `.Rev` (`STEP_4/RevBayes/`), and (3) tree file `.tre` that is shared accross all chracters (`STEP_4/RevBayes/data/`). The process of file cretion can be automitized using the following scripts.

```
# reading chracter matrix
MT<-read.csv("STEP_4/matrix.csv", header = T, row.names=1, as.is=T, check.names=F)

# creating chracter files using the matrix
#setwd("~/Documents/Recon-Anc_Anat/Supplementary_materials/STEP_4/RevBayes/data")
for (i in 1:ncol(MT))
{
  C.rev<-MT[,i]
```

```
  C.rev<-gsub("&", " ", C.rev)

  out<-cbind(rownames(MT), C.rev)
  write.table(file=paste0(colnames(MT[i]), ".char"), out, quote=F, sep=" ",
              row.names=F, col.names=F)
}

# write Rev file for the two-state characters
#setwd("~/Documents/Recon-Anc_Anat/Supplementary_materials/STEP_4/RevBayes/")

# For constructing .Rev files we use the procooked template "PARAMO2_templ.Rev"
fl.in  <- readLines("PARAMO2_templ.Rev")

for (i in 1:ncol(MT))
{
  fl.in  <- readLines("PARAMO2_templ.Rev")
  fl.in  <- gsub(pattern = "@analysis_name@", replace = paste0(colnames(MT[i])),
                 x = fl.in)
  fl.in <- gsub(pattern = "@chrs_2_read@", replace = paste0("data/", colnames(MT[i]), ".char"), x = fl.

  cat(file=paste0(colnames(MT[i]), ".Rev"), sep="\n", fl.in)

}

# write Rev file for dependent four-state character C3-2
setwd("~/Documents/Recon-Anc_Anat/Supplementary_materials")


# I use precooked set of functions for constracting SMM from Tarasov (2019)
source("R_PARAMO/SMM_functions.R")

###################################
# same SMMs as for the tail color problem
###################################
char.state<-c("a", "p")
rate.param<-c(1, 1)
TL<-init_char_matrix(char.state, rate.param, diag.as=0)
char.state<-c("r", "b")
rate.param<-c(1, 1)
COL<-init_char_matrix(char.state, rate.param, diag.as=0)

#SMM-ind
TC.ind<-comb2matrices(TL, COL, controlling.state=NULL, name.sep="", diag.as="")
TC.ind
in.rev<-Mk_Rev(TC.ind)
cat(in.rev) # COPY the output and insert in Rev template PARAMO2_templ.Rev
#cat(in.rev, file="STEP_4/input_Rev.txt") # or save this outout to a file and then copy to Rev template
```

Now having created the files for the inference, we run RevBayes. Each RevBayes output consists of four files located in `STEP_4/RevBayes/output/`: log file, ancestral character state reconstruction (asr), and stochastim maps (stm).

Before starting ontology-informed amalgamation of characters, let us first fix potential issues with data maipulation. To amalgamate stochastic maps, we first have to descriteze them – each tree branch is split into

small bins, whereas each bin indicates the state of a charatcer. This dicretization facilitates stochastic map amalgamation but may critically increseas memory usage in `R` (if map samples and trees are large). To make the memory usage efficient, we put each stochastic map in a seprate `.rds` file, then we put all `.rds` files belonging to the same character into a seprate `.zip` archive. This trick prevents `R` to collapse and, at the same time, allows getting access to the individual maps upon demand. You may want to skip this substep and proceed directly to the next step dealiang with amalgamation of the stochastic maps (the `.zip` archives obtained at this substep are in `STEP_4/RevBayes/Discr_maps`).

```
library("phytools")
# we use a set of precooked functions to work with stoch. maps
source("R_PARAMO/Functions_Discr_maps.R")
# let's make character list

c=paste0("C", AN$CHAR_ID2)
c<-sub(",", "-", c  )

# dir to write and read files
dirW= ("STEP_5/Discr_maps/")
dirR= ("STEP_4/RevBayes/output/")

#####################################
# Read a sample of 100 maps from .stm files and save them in the poper format .stmR
#####################################

for (i in 1:length(c))
{
  tree<-read_Simmap_Rev(paste0(dirR, c[i], ".stm"),
                        start=400, end=500,
                        save = NULL) %>% read.simmap(text=., format="phylip")


  write.simmap(tree, file=paste0(dirW, c[i], ".stmR"))
}
##########

#####################################
# Read stmR, discretize maps, and save each map as a separate rds file;
#all rds filea for a chracter are stored in a zip archive
#####################################

for (i in 1:length(c))
  {
  # read in undesritezed trees
  print(paste0("Reading ", c[i]))
  sim=read.simmap(file=paste0(dirW, c[i], ".stmR"), format="phylip")

  # descritize trees by looping over sample and saving as rds

  for (j in 1:length(sim)){
    tryCatch({

      print(paste0("Descritizing tree ", j))

      ## errors with na
```

```
      ##

      ##### make trees equal with template
      sim.d<-make_tree_eq(tree.tmp.final, sim[[j]], round=5)
      ###

      #sim.d<-discr_Simmap_all(sim[[j]], 1000)
      sim.d<-discr_Simmap_all(sim.d, 1000)

      saveRDS(sim.d, file =  paste0(dirW,c[i], "_", j, ".rds") )

    }, error=function(e){
      cat("ERROR :",conditionMessage(e), "\n")
      #errors<-rbind(errors, c(ii,jj))
    }  )

  }

  # putting rds files into archive
  files<-paste0(dirW, c[i], "_", c(1:length(sim)), ".rds")
  zip(paste0(dirW, c[i], ".zip"), files=files)
  file.remove(files)

}

# close connections
 showConnections (all=T)
 closeAllConnections()
#########################
```

STEP 5. Ontology-informed amalgamation of the stochastic maps

Now having the stochastic maps in the proper format, we can start with their ontology-informed amalgamation. Our goal is to construct the amalgamated characters for the AD, BR and EF levels of anatomical hierarchy. The AD level exibits the individual stochastic maps obtained at the previous step (`STEP_4/RevBayes/Discr_maps`). At this step, we will construct characters for BR and EF levels using ontology-informed amalgamatiion of the stochastic maps. Remeber that, at the BR level, we considered three main body regions – "head", "legs" and "wings". The stochstic map amalagantion is done using the results of the *RAC* query from Step 3.

```
source("R_PARAMO/Functions_Stack_maps.R")

# dir to write and read files
dirW= ("STEP_5/Discr_maps/")
dirR= ("STEP_4/RevBayes/output/")
#############
# Amalgamation at the BR level
#############
# we use the ouput `BR` from the RAC query obtained at Step 3.
# This ouput contains character IDs for BR terms
# Let's rename those IDs to match the file names of the stochastic maps
cc<-lapply(BR, function(x) sub("CHAR:", "C", x) )
cc<-lapply(cc, function(x) sub(",", "-", x) )
```

```
# creating BR.maps to store the amalagamations
BR.maps<-vector("list", length(BR))
names(BR.maps)<-names(BR)

# run amalgamation using the renamed outputs from RAC query
# this loop construct one amalgamation for each BR term
# the number of amalgamations per term can be specified using `ntrees=`
for (i in 1:length(BR.maps))
{
  map<-paramo(cc[[i]], ntrees=1, dirW=dirW)
  BR.maps[[i]]<-map
}

##############
# Amalgamation at the EF level
##############
# we use the ouput `EF` from the RAC query obtained at Step 3.
# This ouput contains character IDs for EF term
# Let's rename those IDs to match the file names of the stochastic maps
cc3<-lapply(EF, function(x) sub("CHAR:", "C", x) )
cc3<-lapply(cc3, function(x) sub(",", "-", x) )

# creating EF.maps to store the amalagamations
EF.maps<-vector("list", length(EF))
names(EF.maps)<-names(EF)

# run amalgamation using the renamed outputs from RAC query
# this code will return 10 amalgamated stochastic maps of the EF character
for (i in 1:length(EF.maps))
{
  map<-paramo(cc3[[i]], ntrees=10, dirW=dirW)
  EF.maps[[i]]<-map
}
```

Now let us plot the amalgamated character hostories for BR and EF characters.

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

```
## Loading required package: ape
```

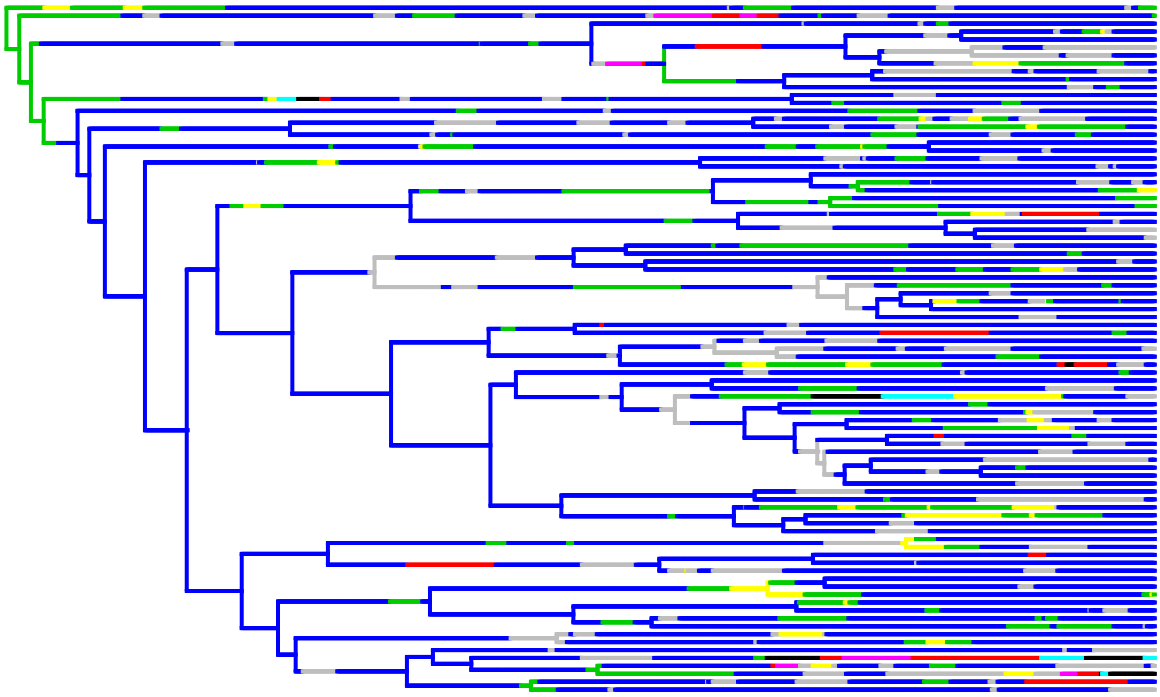
```
## Loading required package: maps
```

```
#########
# BR level
########

# plot one stochastic maps for the head character
plotSimmap(BR.maps$head[[1]], pts=F,ftype="off", ylim=c(0,100) )
```

```
## no colors provided. using the following legend:
##        00        01        02        03        10        11        12
##   "black"     "red"  "green3"    "blue"    "cyan" "magenta"  "yellow"
##        13
##    "gray"
```

```
title("\n Head character")
```

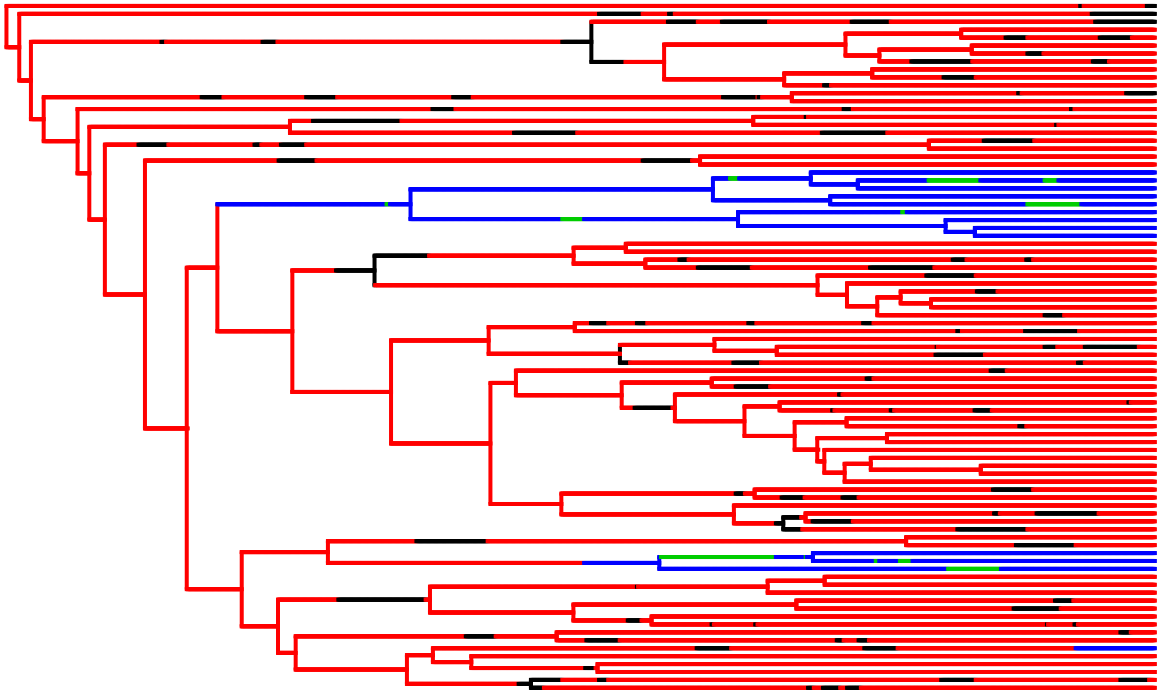



```
# plot one stochastic maps for the wings character
plotSimmap(BR.maps$wings[[1]], pts=F,ftype="off", ylim=c(0,100) )
```

```
## no colors provided. using the following legend:
##       00       01       10       11
##  "black"    "red" "green3"   "blue"
```

```
title("\n Wings character")
```

Wings character

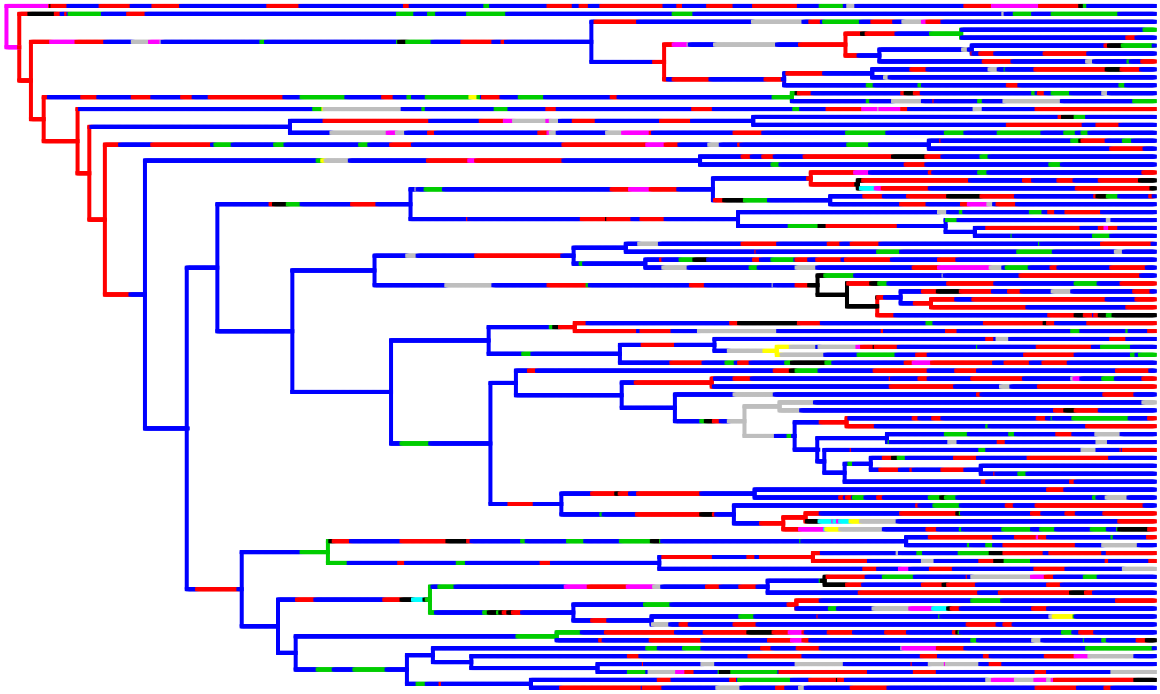


```
# plot one stochastic maps for the legs character
plotSimmap(BR.maps$legs[[1]], pts=F,ftype="off", ylim=c(0,100) )
```

```
## no colors provided. using the following legend:
##       000       001       010       011       100       101       110
##   "black"     "red"  "green3"    "blue"    "cyan" "magenta"  "yellow"
##       111
##    "gray"
```

```
title("\n Legs character")
```

Legs character



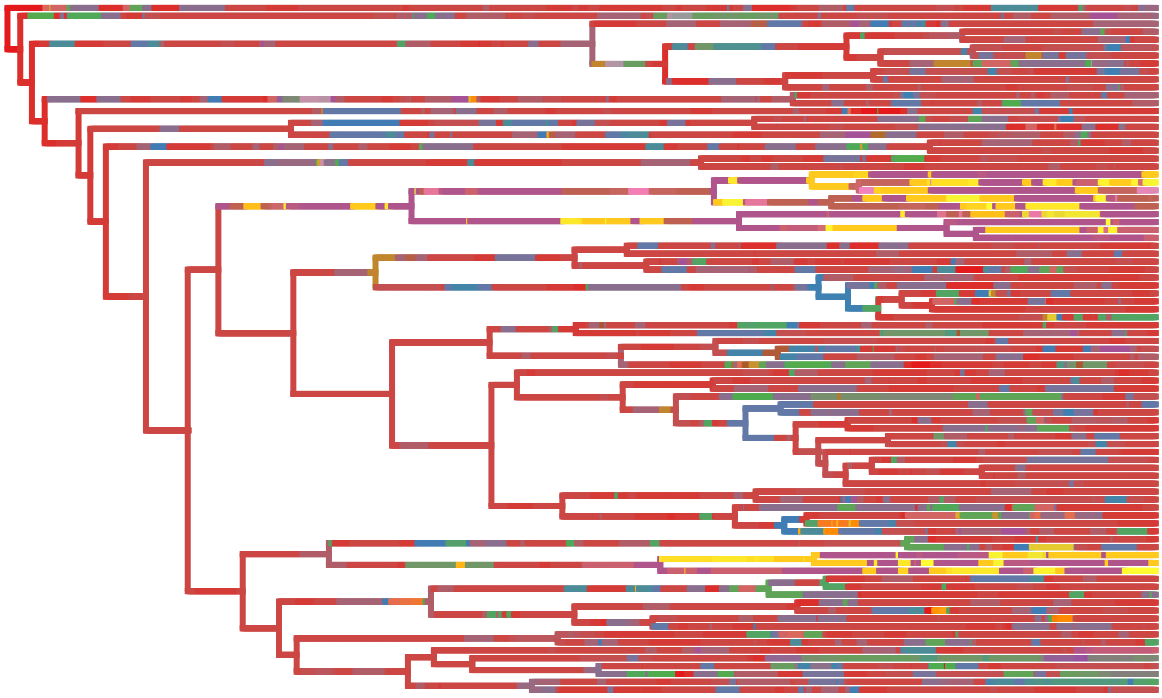
```
#########
# EF level
#########

# plot one stochastic maps for the entire phenotype character
# first, let's define color pallette for the characters sonce it contains many states
library("RColorBrewer")
tmm<-EF.maps$whole_organism[[1]]
lapply(tmm$maps, names) %>% unlist %>% unique->states
# number of states in the character
#length(states)

hm.palette <- colorRampPalette(brewer.pal(9, 'Set1'), space='Lab')
color<-hm.palette(length(states))

plotSimmap(tmm, setNames(color, states),  lwd=3, pts=F,ftype="off", ylim=c(0,100))
title("\n Entire Phenotype character")
```

Entire Phenotype character



References

Klopfstein, Seraina, Lars Vilhelmsen, John M Heraty, Michael Sharkey, and Fredrik Ronquist. 2013. "The Hymenopteran Tree of Life: Evidence from Protein-Coding Genes and Objectively Aligned Ribosomal Data." *PLoS One* 8 (8). Public Library of Science: e69344.

Sharkey, Michael J, James M Carpenter, Lars Vilhelmsen, John Heraty, Johan Liljeblad, Ashley PG Dowling, Susanne Schulmeister, et al. 2012. "Phylogenetic Relationships Among Superfamilies of Hymenoptera." *Cladistics* 28 (1). Wiley Online Library: 80–112.

Yoder, Matthew J, Istvan Miko, Katja C Seltmann, Matthew A Bertone, and Andrew R Deans. 2010. "A Gross Anatomy Ontology for Hymenoptera." *PloS One* 5 (12). Public Library of Science: e15991.