study_transition_matrix_MH.r

max

2022-05-29

```
get_csize <- function(clade){
  length(strsplit(clade, "")[[1]])
}
get_stuff <- function(clade){

  compute_all_errors <- function(clade){

    csize <- get_csize(clade)
    clade.ind <- clade_index(clade, all.splits)

    S1.c.pos <- which(complete.clade.matrix[, clade.ind] == 1)
    S0.c.pos <- setdiff(1:K, S1.c.pos)

    grid.pos <- subset(expand.grid(S1.c.pos, S1.c.pos),
                       Var1 < Var2)

    all.errors <- do.call(rbind,
                          apply(grid.pos, 1,
                                function(row){
                                  x.pos <- row[1]
                                  y.pos <- row[2]
                                  sum_x <- sum(trans.mat.MH[x.pos, S0.c.pos])
                                  sum_y <- sum(trans.mat.MH[y.pos, S0.c.pos])
                                  return(data.frame(
                                    index.x = x.pos,
                                    index.y = y.pos,
                                    Nx = neighbourhood.sizes[x.pos],
                                    Ny = neighbourhood.sizes[y.pos],
                                    A0cx = A0s.sizes[x.pos],
                                    A0cy = A0s.sizes[y.pos],
                                    sum.x = sum_x,
                                    sum.y = sum_y,
                                    error = sum_x - sum_y
                                  ))
                                })
    )
    all.errors$ntaxa <- n
    all.errors$clade_size <- csize
    all.errors$clade <- clade
    all.errors <- tibble::tibble(all.errors)
    return(all.errors)
  }
```

```
  compute_all_dists <- function(clade){
    clade.ind <- clade_index(clade, all.splits)
    S1.c.pos <- which(complete.clade.matrix[, clade.ind] == 1)
    grid.pos <- subset(expand.grid(S1.c.pos, S1.c.pos),
                       Var1 < Var2)
    all.dists <- do.call(rbind,
                         apply(grid.pos, 1,
                               function(row){
                                 x.pos <- row[1]
                                 y.pos <- row[2]
                                 return(data.frame(
                                   index.x = x.pos,
                                   index.y = y.pos,
                                   rspr_dist = rspr(
                                     all.trees[[x.pos]],
                                     all.trees[[y.pos]]
                                   )
                                 ))
                               })
    )
    all.dists$ntaxa <- n
    all.dists$clade <- clade
    all.dists <- tibble::tibble(all.dists)
    return(all.dists)
  }
  ###

  clade.nice <- paste0("t", strsplit(clade, "")[[1]])
  clade.ind <- clade_index(x = clade, all.splits = all.splits)
  hasclade <- which(complete.clade.matrix[, clade.ind] == 1)
  parts <- lapply(all.trees[hasclade],
                  safe_f_c, clade = clade.nice)

  Nxprimes <- unlist(lapply(parts,
                            function(x) neighbourhood_size(x$x_prime)$n_size))
  Nphixcs <- unlist(lapply(parts,
                           function(x) neighbourhood_size(x$phi_x_c)$n_size))

  AOs.sizes <- rep(NA, K)
  AOs.sizes[hasclade] <- neighbourhood.sizes[hasclade] - Nxprimes + Nphixcs

  all.errors <- compute_all_errors(clade)
  dists <- compute_all_dists(clade)

  all.errors.dt <- tibble::tibble(
    merge(all.errors, dists,
          by = c("index.x", "index.y",
                 "ntaxa", "clade") )
  )
  all.errors.dt <- all.errors.dt %>%
    mutate(bound = AOcx/Nx - (5*AOcy)/(6*Ny))
  return(all.errors.dt)
}
```

```
########################################
library(PhyloMarkovChains)
library(phangorn)
```

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.1.9000 --

## v ggplot2 3.3.5.9000     v purrr   0.3.4
## v tibble  3.1.6          v dplyr   1.0.8
## v tidyr   1.2.0          v stringr 1.4.0
## v readr   2.1.2          v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
n <- 6
all.splits <- sapply(allSplits(n + 1),  paste, collapse = "")
J <- length(all.splits)

all.trees <- allTrees(n, rooted = TRUE)
K <- length(all.trees)

compute <- FALSE

if(compute){
  comp.time <- system.time(
    SPR.mat <- as.matrix(rspr_matrix(all.trees) > 0) + 0
  )
  save(SPR.mat,
       file = paste0("saved_data/SPR_matrix_n=", n, ".RData"))
}else{
  load(paste0("saved_data/SPR_matrix_n=", n, ".RData"))
}

####
# Neighbourhood stuff
incidence.mat <- SPR.mat
neighbourhood.sizes <- colSums(incidence.mat)

neigh.ratios <- matrix(NA, nrow = K, ncol = K)
diag(neigh.ratios) <- 0
for(i in 1:K){
  neigh.ratios[i, ] <- sapply(neighbourhood.sizes[i]/neighbourhood.sizes,
                                          function(x) min(1, x))/
    neighbourhood.sizes[i]
}
```

```r
trans.mat.MH <- neigh.ratios * incidence.mat
diag(trans.mat.MH) <- 1-rowSums(trans.mat.MH)
diag(trans.mat.MH)
```

```
##   [1] 0.04428220 0.01307692 0.01865385 0.04428220 0.03245262 0.04428220
##   [7] 0.04428220 0.04428220 0.04428220 0.01307692 0.00000000 0.00000000
##  [13] 0.01307692 0.01307692 0.01307692 0.00000000 0.00000000 0.01865385
##  [19] 0.01865385 0.00000000 0.01865385 0.03245262 0.01865385 0.01865385
##  [25] 0.01865385 0.01865385 0.03245262 0.04428220 0.01307692 0.03245262
##  [31] 0.04428220 0.04428220 0.04428220 0.01865385 0.04428220 0.04428220
##  [37] 0.03245262 0.01307692 0.01865385 0.04428220 0.04428220 0.04428220
##  [43] 0.04428220 0.04428220 0.04428220 0.04428220 0.01307692 0.01865385
##  [49] 0.04428220 0.04428220 0.04428220 0.03245262 0.04428220 0.04428220
##  [55] 0.04428220 0.01865385 0.03245262 0.04428220 0.04428220 0.04428220
##  [61] 0.01307692 0.04428220 0.04428220 0.01307692 0.00000000 0.00000000
##  [67] 0.01307692 0.00000000 0.00000000 0.01307692 0.01307692 0.01865385
##  [73] 0.00000000 0.01307692 0.00000000 0.00000000 0.00000000 0.01307692
##  [79] 0.01307692 0.01307692 0.01865385 0.00000000 0.00000000 0.01307692
##  [85] 0.01307692 0.00000000 0.00000000 0.01307692 0.01307692 0.01865385
##  [91] 0.01307692 0.00000000 0.01307692 0.01307692 0.00000000 0.00000000
##  [97] 0.00000000 0.01307692 0.01865385 0.00000000 0.00000000 0.00000000
## [103] 0.00000000 0.01307692 0.01307692 0.01307692 0.01307692 0.01865385
## [109] 0.00000000 0.01307692 0.00000000 0.00000000 0.01307692 0.01307692
## [115] 0.00000000 0.01307692 0.01865385 0.01865385 0.01865385 0.01865385
## [121] 0.01865385 0.01865385 0.01865385 0.00000000 0.03245262 0.03245262
## [127] 0.01865385 0.00000000 0.01865385 0.03245262 0.01865385 0.01865385
## [133] 0.01865385 0.01865385 0.03245262 0.00000000 0.00000000 0.01307692
## [139] 0.01307692 0.01307692 0.01307692 0.00000000 0.00000000 0.01865385
## [145] 0.01865385 0.01307692 0.04428220 0.04428220 0.03245262 0.04428220
## [151] 0.04428220 0.04428220 0.04428220 0.03245262 0.01307692 0.04428220
## [157] 0.04428220 0.04428220 0.04428220 0.01865385 0.04428220 0.04428220
## [163] 0.01865385 0.01307692 0.03245262 0.04428220 0.04428220 0.04428220
## [169] 0.04428220 0.04428220 0.04428220 0.01865385 0.01307692 0.04428220
## [175] 0.04428220 0.04428220 0.04428220 0.03245262 0.04428220 0.04428220
## [181] 0.03245262 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220
## [187] 0.01307692 0.04428220 0.04428220 0.04428220 0.01307692 0.03245262
## [193] 0.04428220 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220
## [199] 0.01307692 0.00000000 0.01307692 0.01307692 0.00000000 0.01307692
## [205] 0.00000000 0.00000000 0.01865385 0.03245262 0.01307692 0.04428220
## [211] 0.04428220 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220
## [217] 0.04428220 0.01307692 0.04428220 0.04428220 0.01865385 0.04428220
## [223] 0.03245262 0.04428220 0.04428220 0.01865385 0.00000000 0.01865385
## [229] 0.01865385 0.01865385 0.03245262 0.01865385 0.01865385 0.03245262
## [235] 0.04428220 0.01307692 0.04428220 0.04428220 0.03245262 0.04428220
## [241] 0.01865385 0.04428220 0.04428220 0.04428220 0.01865385 0.04428220
## [247] 0.04428220 0.03245262 0.04428220 0.01307692 0.04428220 0.04428220
## [253] 0.04428220 0.01865385 0.03245262 0.04428220 0.01307692 0.04428220
## [259] 0.04428220 0.04428220 0.04428220 0.01865385 0.01865385 0.01865385
## [265] 0.01865385 0.00000000 0.03245262 0.01865385 0.01865385 0.03245262
## [271] 0.03245262 0.01865385 0.04428220 0.04428220 0.01307692 0.04428220
## [277] 0.04428220 0.04428220 0.04428220 0.04428220 0.01865385 0.04428220
## [283] 0.04428220 0.01307692 0.04428220 0.03245262 0.04428220 0.04428220
## [289] 0.01307692 0.00000000 0.01307692 0.01307692 0.00000000 0.01307692
```

```
## [295] 0.00000000 0.00000000 0.01865385 0.04428220 0.03245262 0.04428220
## [301] 0.04428220 0.01307692 0.04428220 0.01865385 0.04428220 0.04428220
## [307] 0.04428220 0.03245262 0.04428220 0.04428220 0.01865385 0.04428220
## [313] 0.01307692 0.04428220 0.04428220 0.01307692 0.00000000 0.00000000
## [319] 0.00000000 0.00000000 0.01307692 0.01307692 0.01307692 0.01865385
## [325] 0.00000000 0.01307692 0.00000000 0.01307692 0.00000000 0.00000000
## [331] 0.01307692 0.01307692 0.01865385 0.00000000 0.00000000 0.01307692
## [337] 0.01307692 0.00000000 0.00000000 0.01307692 0.01307692 0.01865385
## [343] 0.00000000 0.01307692 0.01307692 0.01307692 0.00000000 0.00000000
## [349] 0.00000000 0.01307692 0.01865385 0.00000000 0.00000000 0.00000000
## [355] 0.00000000 0.01307692 0.01307692 0.01307692 0.01307692 0.01865385
## [361] 0.01307692 0.00000000 0.00000000 0.00000000 0.01307692 0.01307692
## [367] 0.00000000 0.01307692 0.01865385 0.01865385 0.01865385 0.01865385
## [373] 0.01865385 0.01865385 0.01865385 0.00000000 0.03245262 0.03245262
## [379] 0.00000000 0.01307692 0.00000000 0.01307692 0.01307692 0.01307692
## [385] 0.00000000 0.00000000 0.01865385 0.01307692 0.04428220 0.01865385
## [391] 0.04428220 0.03245262 0.04428220 0.04428220 0.04428220 0.04428220
## [397] 0.00000000 0.01865385 0.01865385 0.03245262 0.01865385 0.01865385
## [403] 0.01865385 0.01865385 0.03245262 0.01307692 0.04428220 0.03245262
## [409] 0.04428220 0.04428220 0.04428220 0.01865385 0.04428220 0.04428220
## [415] 0.01307692 0.03245262 0.01865385 0.04428220 0.04428220 0.04428220
## [421] 0.04428220 0.04428220 0.04428220 0.01307692 0.04428220 0.01865385
## [427] 0.04428220 0.04428220 0.04428220 0.03245262 0.04428220 0.04428220
## [433] 0.01865385 0.04428220 0.03245262 0.04428220 0.04428220 0.04428220
## [439] 0.01307692 0.04428220 0.04428220 0.00000000 0.00000000 0.01307692
## [445] 0.01307692 0.01307692 0.01307692 0.00000000 0.00000000 0.01865385
## [451] 0.00000000 0.01865385 0.01865385 0.03245262 0.01865385 0.01865385
## [457] 0.01865385 0.01865385 0.03245262 0.01307692 0.01865385 0.04428220
## [463] 0.04428220 0.03245262 0.04428220 0.04428220 0.04428220 0.04428220
## [469] 0.01307692 0.03245262 0.04428220 0.04428220 0.04428220 0.04428220
## [475] 0.01865385 0.04428220 0.04428220 0.01307692 0.01865385 0.03245262
## [481] 0.04428220 0.04428220 0.04428220 0.04428220 0.04428220 0.04428220
## [487] 0.01307692 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220
## [493] 0.03245262 0.04428220 0.04428220 0.01865385 0.03245262 0.04428220
## [499] 0.04428220 0.04428220 0.04428220 0.01307692 0.04428220 0.04428220
## [505] 0.00000000 0.01307692 0.01307692 0.01307692 0.00000000 0.01307692
## [511] 0.00000000 0.00000000 0.01865385 0.01307692 0.04428220 0.03245262
## [517] 0.04428220 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220
## [523] 0.01307692 0.03245262 0.04428220 0.04428220 0.01865385 0.04428220
## [529] 0.04428220 0.04428220 0.04428220 0.01307692 0.04428220 0.04428220
## [535] 0.04428220 0.01865385 0.04428220 0.03245262 0.04428220 0.04428220
## [541] 0.00000000 0.01865385 0.01865385 0.01865385 0.01865385 0.03245262
## [547] 0.01865385 0.01865385 0.03245262 0.01307692 0.04428220 0.04428220
## [553] 0.04428220 0.03245262 0.04428220 0.01865385 0.04428220 0.04428220
## [559] 0.01865385 0.04428220 0.04428220 0.04428220 0.03245262 0.04428220
## [565] 0.01307692 0.04428220 0.04428220 0.01865385 0.01865385 0.01865385
## [571] 0.01865385 0.00000000 0.03245262 0.01865385 0.01865385 0.03245262
## [577] 0.01865385 0.04428220 0.03245262 0.04428220 0.01307692 0.04428220
## [583] 0.04428220 0.04428220 0.04428220 0.01865385 0.03245262 0.04428220
## [589] 0.04428220 0.01307692 0.04428220 0.04428220 0.04428220 0.04428220
## [595] 0.01865385 0.04428220 0.04428220 0.04428220 0.01307692 0.04428220
## [601] 0.03245262 0.04428220 0.04428220 0.00000000 0.01307692 0.01307692
## [607] 0.01307692 0.00000000 0.01307692 0.00000000 0.00000000 0.01865385
## [613] 0.03245262 0.04428220 0.04428220 0.04428220 0.01307692 0.04428220
```

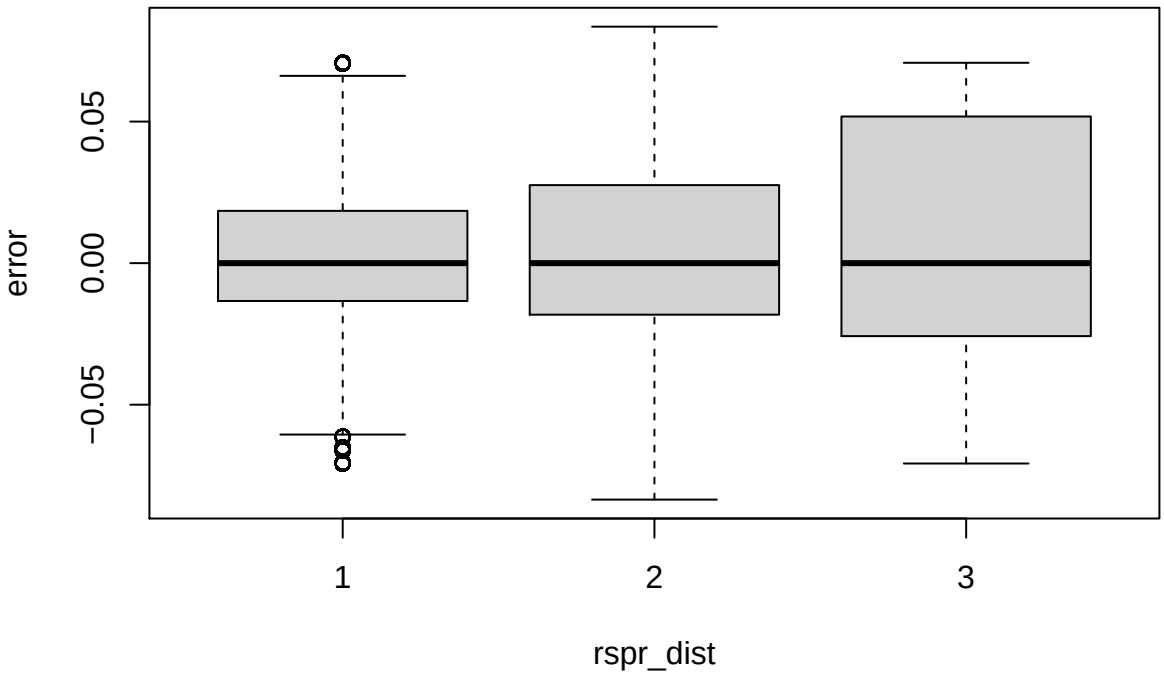
```
## [619] 0.01865385 0.04428220 0.04428220 0.03245262 0.04428220 0.04428220
## [625] 0.04428220 0.01865385 0.04428220 0.01307692 0.04428220 0.04428220
## [631] 0.04428220 0.01865385 0.01307692 0.04428220 0.03245262 0.04428220
## [637] 0.04428220 0.04428220 0.04428220 0.01865385 0.01865385 0.00000000
## [643] 0.03245262 0.01865385 0.01865385 0.01865385 0.01865385 0.03245262
## [649] 0.01307692 0.00000000 0.00000000 0.01307692 0.01307692 0.01307692
## [655] 0.00000000 0.00000000 0.01865385 0.04428220 0.03245262 0.01307692
## [661] 0.04428220 0.04428220 0.04428220 0.01865385 0.04428220 0.04428220
## [667] 0.03245262 0.01865385 0.01307692 0.04428220 0.04428220 0.04428220
## [673] 0.04428220 0.04428220 0.04428220 0.04428220 0.01865385 0.01307692
## [679] 0.04428220 0.04428220 0.04428220 0.03245262 0.04428220 0.04428220
## [685] 0.04428220 0.03245262 0.01865385 0.04428220 0.04428220 0.04428220
## [691] 0.01307692 0.04428220 0.04428220 0.01865385 0.01865385 0.00000000
## [697] 0.03245262 0.01865385 0.01865385 0.01865385 0.01865385 0.03245262
## [703] 0.01865385 0.04428220 0.01307692 0.04428220 0.03245262 0.04428220
## [709] 0.04428220 0.04428220 0.04428220 0.00000000 0.01307692 0.00000000
## [715] 0.01307692 0.01307692 0.01307692 0.00000000 0.00000000 0.01865385
## [721] 0.03245262 0.04428220 0.01307692 0.04428220 0.04428220 0.04428220
## [727] 0.01865385 0.04428220 0.04428220 0.01865385 0.03245262 0.01307692
## [733] 0.04428220 0.04428220 0.04428220 0.04428220 0.04428220 0.04428220
## [739] 0.01865385 0.04428220 0.01307692 0.04428220 0.04428220 0.04428220
## [745] 0.03245262 0.04428220 0.04428220 0.03245262 0.04428220 0.01865385
## [751] 0.04428220 0.04428220 0.04428220 0.01307692 0.04428220 0.04428220
## [757] 0.01307692 0.00000000 0.00000000 0.01307692 0.00000000 0.00000000
## [763] 0.01307692 0.01307692 0.01865385 0.00000000 0.01307692 0.00000000
## [769] 0.01307692 0.00000000 0.00000000 0.01307692 0.01307692 0.01865385
## [775] 0.00000000 0.00000000 0.01307692 0.00000000 0.00000000 0.01307692
## [781] 0.01307692 0.01307692 0.01865385 0.01307692 0.01307692 0.00000000
## [787] 0.01307692 0.00000000 0.00000000 0.00000000 0.01307692 0.01865385
## [793] 0.00000000 0.00000000 0.00000000 0.00000000 0.01307692 0.01307692
## [799] 0.01307692 0.01307692 0.01865385 0.00000000 0.00000000 0.01307692
## [805] 0.00000000 0.01307692 0.01307692 0.00000000 0.01307692 0.01865385
## [811] 0.01865385 0.01865385 0.01865385 0.01865385 0.01865385 0.01865385
## [817] 0.00000000 0.03245262 0.03245262 0.04428220 0.03245262 0.01307692
## [823] 0.04428220 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220
## [829] 0.03245262 0.04428220 0.01307692 0.04428220 0.01865385 0.04428220
## [835] 0.04428220 0.04428220 0.04428220 0.01307692 0.01307692 0.00000000
## [841] 0.01307692 0.00000000 0.01307692 0.00000000 0.00000000 0.01865385
## [847] 0.04428220 0.04428220 0.01307692 0.04428220 0.01865385 0.04428220
## [853] 0.03245262 0.04428220 0.04428220 0.01865385 0.01865385 0.00000000
## [859] 0.01865385 0.01865385 0.03245262 0.01865385 0.01865385 0.03245262
## [865] 0.04428220 0.04428220 0.01307692 0.04428220 0.03245262 0.04428220
## [871] 0.01865385 0.04428220 0.04428220 0.04428220 0.04428220 0.01865385
## [877] 0.04428220 0.03245262 0.04428220 0.01307692 0.04428220 0.04428220
## [883] 0.04428220 0.03245262 0.01865385 0.04428220 0.01307692 0.04428220
## [889] 0.04428220 0.04428220 0.04428220 0.03245262 0.04428220 0.01865385
## [895] 0.04428220 0.01307692 0.04428220 0.04428220 0.04428220 0.04428220
## [901] 0.01865385 0.01865385 0.01865385 0.01865385 0.00000000 0.03245262
## [907] 0.01865385 0.01865385 0.03245262 0.04428220 0.04428220 0.01865385
## [913] 0.04428220 0.01307692 0.04428220 0.03245262 0.04428220 0.04428220
## [919] 0.01307692 0.01307692 0.00000000 0.01307692 0.00000000 0.01307692
## [925] 0.00000000 0.00000000 0.01865385 0.04428220 0.04428220 0.03245262
## [931] 0.04428220 0.01307692 0.04428220 0.01865385 0.04428220 0.04428220
## [937] 0.04428220 0.04428220 0.03245262 0.04428220 0.01865385 0.04428220
```

```
## [943] 0.01307692 0.04428220 0.04428220
```

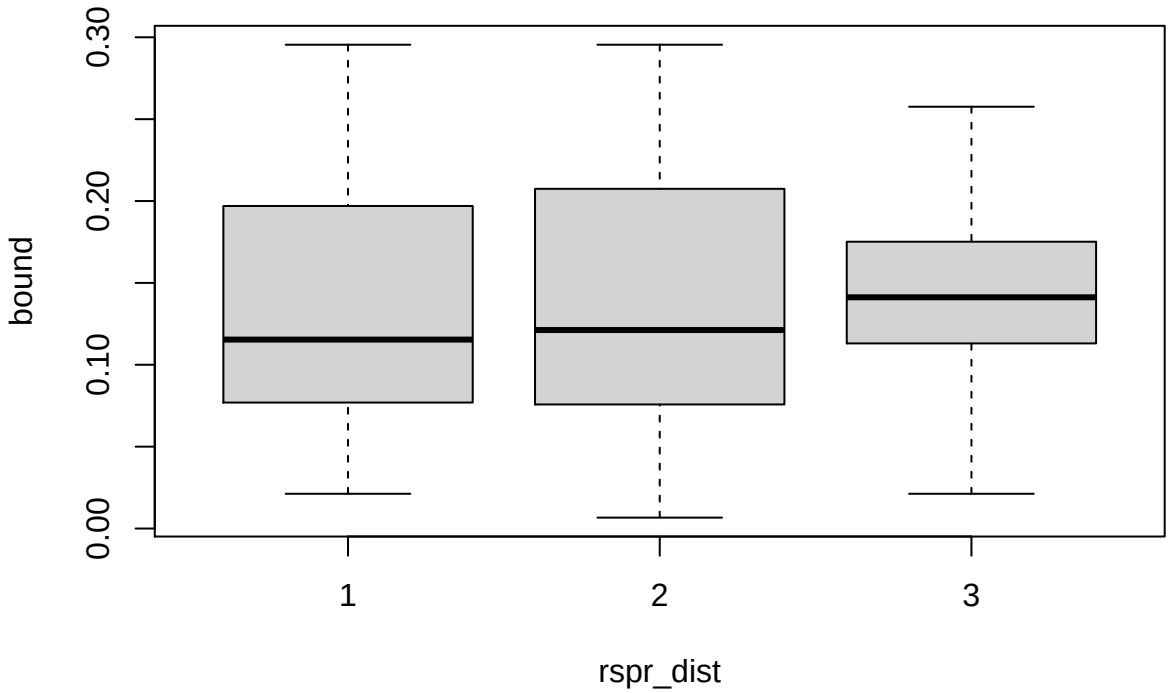
```
#####

complete.Splits <- parallel::mclapply(all.trees,
                                      get_clades, mc.cores = 12)

complete.indexes <- parallel::mclapply(complete.Splits,
                                       function(x) charmatch(x, all.splits),
                                       mc.cores = 12)

complete.clade.matrix <- do.call(rbind, parallel::mclapply(complete.indexes,
                                                           clade_indicators,
                                                           L = J,
                                                           mc.cores = 12))
colnames(complete.clade.matrix) <- all.splits
colMeans(complete.clade.matrix)
```

```
##          1          2         12          3         13         23        123
## 1.00000000 1.00000000 0.11111111 1.00000000 0.11111111 0.11111111 0.04761905
##          4         14         24        124         34        134        234
## 1.00000000 0.11111111 0.11111111 0.04761905 0.11111111 0.04761905 0.04761905
##       1234          5         15         25        125         35        135
## 0.04761905 1.00000000 0.11111111 0.11111111 0.04761905 0.11111111 0.04761905
##        235       1235         45        145        245       1245        345
## 0.04761905 0.04761905 0.11111111 0.04761905 0.04761905 0.04761905 0.04761905
##       1345       2345      12345          6         16         26        126
## 0.04761905 0.04761905 0.11111111 1.00000000 0.11111111 0.11111111 0.04761905
##         36        136        236       1236         46        146        246
## 0.11111111 0.04761905 0.04761905 0.04761905 0.11111111 0.04761905 0.04761905
##       1246        346       1346       2346      12346         56        156
## 0.04761905 0.04761905 0.04761905 0.04761905 0.11111111 0.11111111 0.04761905
##        256       1256        356       1356       2356      12356        456
## 0.04761905 0.04761905 0.04761905 0.04761905 0.04761905 0.11111111 0.04761905
##       1456       2456      12456       3456      13456      23456     123456
## 0.04761905 0.04761905 0.11111111 0.04761905 0.11111111 0.11111111 1.00000000
```

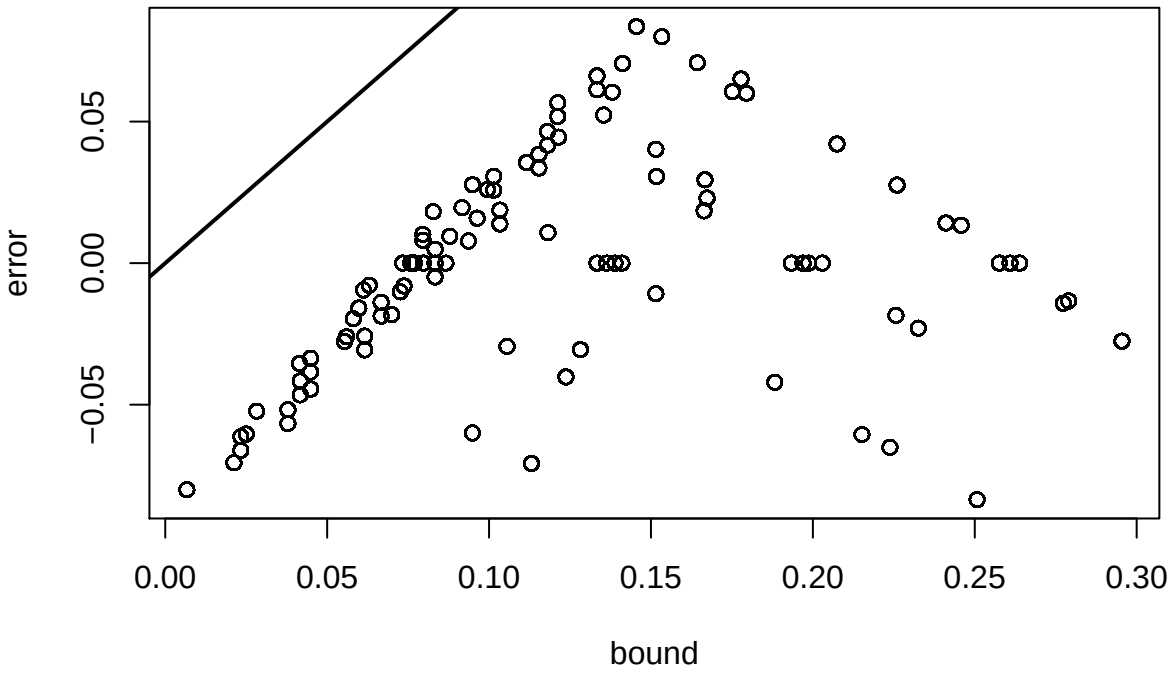
```
## Computing |AOc|
clade.sizes <- unlist(lapply(all.splits, get_csize))
nontrivial <- intersect(which(clade.sizes > 1), which(clade.sizes < n))
all.ntclades <- all.splits[nontrivial]
all.errors <- parallel::mclapply(all.ntclades,
                    function(clade) get_stuff(clade), mc.cores = 8)

all.errors.dt <- do.call(rbind, all.errors)

## Plotting
boxplot(error ~ rspr_dist, all.errors.dt)
```




```
boxplot(bound ~ rspr_dist, all.errors.dt)
```




```
plot(error~bound, all.errors.dt)
abline(a = 0, b = 1, lwd = 2)
```




```
range(all.errors.dt$error)
```

```
## [1] -0.08352285  0.08352285
```

```
range(all.errors.dt$bound)
```

```
## [1] 0.006666667 0.295454545
```

```
all.errors.dt[which.min(all.errors.dt$error), ]
```

```
## # A tibble: 1 x 14
##   index.x index.y ntaxa clade    Nx    Ny  AOcx  AOcy sum.x sum.y   error
##     <int>   <int> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1     125     187     6 1234     46    50    56    58 0.666 0.749 -0.0835
## # ... with 3 more variables: clade_size <int>, rspr_dist <dbl>, bound <dbl>
```

```
all.errors.dt[which.max(all.errors.dt$error), ]
```

```
## # A tibble: 1 x 14
##   index.x index.y ntaxa clade    Nx    Ny  AOcx  AOcy sum.x sum.y  error
##     <int>   <int> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1     187     377     6 1234     50    46    58    56 0.749 0.666 0.0835
## # ... with 3 more variables: clade_size <int>, rspr_dist <dbl>, bound <dbl>
```

```
all.errors.dt[which.min(all.errors.dt$bound), ]
```

```
## # A tibble: 1 x 14
##   index.x index.y ntaxa clade    Nx    Ny  AOcx  AOcy sum.x sum.y   error
##     <int>   <int> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1     787     797     6 12       50    50    22    26 0.434 0.514 -0.0800
## # ... with 3 more variables: clade_size <int>, rspr_dist <dbl>, bound <dbl>
```

```
all.errors.dt[which.max(all.errors.dt$bound), ]
```

```
## # A tibble: 1 x 14
##   index.x index.y ntaxa clade    Nx    Ny  A0cx  A0cy sum.x sum.y   error
##     <int>   <int> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1     108     153     6 12345    48    44    76    68 0.398 0.426 -0.0276
## # ... with 3 more variables: clade_size <int>, rspr_dist <dbl>, bound <dbl>
```