CyclicPeptidedNdS

Imogen Dumville

21/02/2020

Ortho\_long files are split into three columns; ortholog id, gene code and strain code.

## V1 V2 V3  
## 1 og\_0028 CCE32443 CCE27021  
## 2 og\_0063 CCE29570 CCE27021  
## 3 og\_0161 CCE27021 CCE27021  
## 4 og\_0168 CCE32028 CCE27021  
## 5 og\_0178 CCE32757 CCE27021  
## 6 og\_0237 CCE28302 CCE27021

This file is subsetted with the aid of a file containing orthologs pertaining to a group of genes. In this case, orthologs relating to cyclic peptide production.

pa <- read.delim("pa.csv", header=FALSE, stringsAsFactors=FALSE) #pa.csv  
to\_keep <- unique(pa$V1)  
CP\_orthos <- subset(ortho\_long, V1 %in% to\_keep)  
  
write.table(CP\_orthos, file = "subsetCP.tsv", row.names = FALSE, col.names = FALSE, quote = FALSE)

## og\_0076.EamaE4668\_003533.T1.EamaE4668  
## 1 og\_0157 EamaE4668\_001631-T1 EamaE4668  
## 2 og\_1092 EamaE4668\_006529-T1 EamaE4668  
## 3 og\_1881 EamaE4668\_006771-T1 EamaE4668  
## 4 og\_3338 EamaE4668\_006651-T1 EamaE4668  
## 5 og\_3726 EamaE4668\_001457-T1 EamaE4668  
## 6 og\_3865 EamaE4668\_002715-T1 EamaE4668

This subset file is then used in shell. Utilising two python scripts to isolate proteins and nucleotide files (geneorthologs.py & protorthologs.py in master.sh), dNdS.tsv is created.

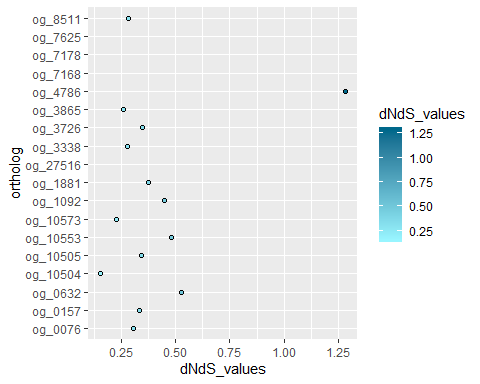
dNdS <- read.csv(file = "dNdS.tsv")  
head(dNdS)

## og\_0076.0.306  
## 1 og\_0157 0.335  
## 2 og\_0632 0.526  
## 3 og\_10504 0.155  
## 4 og\_10505 0.343  
## 5 og\_10553 0.481  
## 6 og\_10573 0.229

dNdS above 0.4 or alignments with more than 15 strains are further analysed. Orthlog sequence files (as isolated by the subset file) are uploaded and anlysed in DataMonkey Meme to observe selection processes. Low p values are further analysed.InterProScan is used to observe domains. Results are observed in a qualitative fashion in dNdSIII.csv.

dNdS results are plotted, after empty dNdS values removed (dNdS.tsv converted to csv) (dNdsplotting.R)

dNdS <- read.csv("dNdS.csv", header=FALSE, stringsAsFactors=FALSE)  
dNdS <- dNdS[(2:19), c(1,2)]  
colnames(dNdS) <- c("ortholog", "dNdS\_values")  
dNdS$dNdS\_values <- as.numeric(as.character(dNdS$dNdS\_values))  
library(ggplot2)  
plot <- ggplot(dNdS, aes(dNdS\_values, ortholog)) + geom\_point(aes(fill=dNdS\_values), colour="black", pch=21)  
plot + scale\_fill\_continuous(low = "cadetblue1", high = "deepskyblue4")



For ancestral state reconstruction, an ASTRAL - produced strain-containing tree must have branch lengths appended to it. To create branchlengths, a subset of all orthologs containing all 24 strains is produced. This is produced in list form

library(ape)  
library(stringr)  
  
all\_filepath\_trees <- list.files("TreeswCpur/", full.names = TRUE)  
all\_full\_trees <- read.tree("all\_full\_trees.phy")   
  
files = which(Ntip(all\_full\_trees) == 23) #total number of strains  
filepaths <- all\_filepath\_trees[files]  
og\_nums\_full <- str\_extract(string = filepaths, pattern = "og\_\\d+")  
  
  
ortho\_full\_rows <- lapply(og\_nums\_full, function(x) which(ortho\_long$V1 == (x)))  
rownums <- unlist(ortho\_full\_rows)  
subsetorthos <- ortho\_long[(rownums),]  
  
full <- unique(subsetorthos$V1)  
write.table(full, file = "subsetfull.tsv", row.names = FALSE, col.names = FALSE, quote = FALSE)  
head(full)

## [1] og\_0001 og\_0004 og\_0005 og\_0011 og\_0016 og\_0017  
## 30793 Levels: og\_0001 og\_0002 og\_0003 og\_0004 og\_0005 og\_0006 og\_0007 ... og\_9999

concatenate.py is then ran to produce a RAxML tree with branch lengths. This is then reimported back into R and ancestral state reconstruction can be carried out. In this example, the presence of og\_10573 is analysed

library(tidyr)  
branchtree <- read.tree(file = "RAxML\_result.fullbranchlengthstree") #produced by concatenate.py  
rooted\_branchtree <- root(branchtree, "CCE27021")  
droppedtip <- drop.tip(rooted\_branchtree, "CCE27021")  
plot(droppedtip)  
axisPhylo() #observe tree  
  
  
pa <- read.delim("pa.csv", header=FALSE, stringsAsFactors=FALSE) #pa.csv  
#CSV describing presence of orthologs  
  
paspread <- spread(pa, V1, V3)  
rownames(paspread) <- paspread$V2  
paspread$V2 <- NULL  
  
og\_10573 <- paspread[,'og\_10573', drop = FALSE] #test og  
matrixog <- as.matrix(og\_10573)[,1]  
  
chronos\_relative <- makeChronosCalib(droppedtip, node = "root", age.min = 1, age.max = 1, interactive = FALSE, soft.bounds = FALSE)   
chronos\_relative #pretending to add chronos / how to add to tree?

## node age.min age.max soft.bounds  
## 1 24 1 1 FALSE

Q <- matrix( c(0,1,2,0) , nrow=2)  
ancestral <- ace(matrixog, droppedtip, type = "discrete", model = Q)  
#had to produce matrix for all rates different model manually as ARD had errors  
margancesstates <- round(ancestral$lik.anc,2)  
  
nodelabels(node=1:droppedtip$Nnode+Ntip(droppedtip), pie=ancestral$lik.anc,cex=0.5)  
  
legend("bottomright", legend = c("Present", "Absent"), fill = c("Cyan", "Red"), cex = 0.75)  
title(main = "Ancestral presence of cyclic peptide og\_10573", xlab = "Branchlength")

