Help to convert R package into a Windows compatible.

<https://stackoverflow.com/questions/1685181/making-r-package-work-in-both-windows-and-linux>

<https://csgillespie.github.io/efficientR/efficient-set-up.html>

Problem to solve

1)

#build a fake species-by\_gene matrix

class(Seq.DF4[[1]])

SpbyGeneMat=rbind(as.matrix(Seq.DF4[[1]]), c("Titi\_titi",0, 2, 1), c("Toto\_toto", 0, 0, 4))

row.names(SpbyGeneMat)=SpbyGeneMat[,1]

SpbyGeneMat=as.data.frame(SpbyGeneMat)

SelGene.MaxSpCov(input = SpbyGeneMat, NBGene = c("co1", "12srrna"))

SelGene.MaxSpCov(input = SpbyGeneMat, NBGene = c("co1", "16srrna"))

SelGene.MaxSpCov(input = SpbyGeneMat, NBGene = c("co1"))

Error in `rownames<-`(x, value) :

attempt to set 'rownames' on an object with no dimensions

SelGene.MaxSpCov(input = SpbyGeneMat, NBGene = 1)

$Minimum\_Number\_of\_Gene\_With\_Full\_Species\_Coverage

[1] 2

$List\_Gene\_Name\_Full\_Species\_Coverage

[1] "co1" "12srrna"

$List\_Gene\_Name\_Selected\_Genes

genename

"co1"

$Species\_Coverage\_Selected\_Genes

[1] 1 ## problem should be 3

$Species\_Coverage\_Selected\_Genes\_percentage

[1] 33.33333 ### Problem should be 3

2)

SelBestSeq {regPhylo}

In the documentation need to include the reference for Antonelli et al. 2016

Also in the output parameter need to precise that the name of the file with the path can be provided but the folder must be created first.

3) example of the mafft function from phyloch R package.

mafft

function (x, y, add, method = "localpair", maxiterate = 1000,

op = 1.53, ep = 0, path, quiet)

{

os <- .Platform$OS

if (missing(quiet))

quiet <- TRUE

qut <- " "

if (missing(path))

path <- "/usr/local/bin/mafft"

fns <- c("mafftin.fas", "mafftpro.fas", "mafftout.fas")

unlink(fns[file.exists(fns)])

if (missing(y)) {

write.fas(x, fns[1])

call.mafft <- paste(path, " --", method, " --", "maxiterate ",

maxiterate, qut, "--op ", op, " --ep ", ep, " ",

fns[1], " > ", fns[3], sep = "")

}

else {

if (missing(add))

add <- "addprofile"

add <- match.arg(add, c("add", "addprofile"))

add <- paste("--", add, sep = "")

write.fas(x, fns[1])

write.fas(y, fns[2])

call.mafft <- paste(path, qut, add, fns[2], fns[1], ">",

fns[3])

}

if (!quiet)

message(call.mafft)

if (os == "unix") {

system(call.mafft, intern = FALSE)

res <- length(scan(fns[3], what = "c", quiet = TRUE))

if (res != 0)

res <- read.fas(fns[3])

}

else {

res <- system(call.mafft, intern = TRUE, ignore.stderr = FALSE)

if (length(grep("error|ERROR", res)) > 0) {

res <- 0

}

else {

res <- read.fas(fns[3])

}

}

unlink(fns[file.exists(fns)])

return(res)

}

<environment: namespace:phyloch>

The function is working well on linux if the path to the software is provided following the next example

require(phyloch)

require(ape)

data(woodmouse)

?mafft

mafft(woodmouse)

getwd()

testmafft=mafft(woodmouse, path = "C:/Users/deme/Documents/Programs/Mafft/mafft-7.409-win64-signed/mafft-win/mafft")

Another example using Gblocks based on the gblocks function in phyloch.

> gblocks

function (x, b1 = 0.5, b2 = b1, b3 = ncol(x), b4 = 2, b5 = "a",

exec)

{

if (b1 < 0.5 | b1 > 1)

stop("b1 not in [0.5, 1]")

if (b2 < b1 | b2 > 1)

stop("b2 not in [b1, 1]")

if (b3 < 0 | b4 > ncol(x))

stop("b3 not in [0, ", ncol(x), "]")

if (b4 < 2 | b4 > ncol(x))

stop("b4 not in [2, ", ncol(x), "]")

b5 <- match.arg(b5, c("a", "h", "n"))

rwd <- getwd()

if (missing(exec))

exec <- "/Applications/Gblocks\_0.91b"

setwd(exec)

ntax <- nrow(x)

b1 <- round(ntax \* b1) + 1

b2 <- round(ntax \* b2) + 1

cat("--- Executing Gblocks: ---")

cat("\nMinimum number of sequences for a conserved position:",

b1)

cat("\nMinimum number of sequences for a flank position:",

b2)

cat("\nMaximum number of contiguous nonconserved positions:",

b3)

cat("\nMinimum length of a block:", b4)

cat("\nAllowed gap positions:", b5)

write.fas(x, "R2GBLOCK.fas")

system(paste("./Gblocks R2GBLOCK.fas -t=d", " -b1=", b1,

" -b2=", b2, " -b3=", b3, " -b4=", b4, " -b5=", b5, sep = ""),

show.output.on.console = FALSE)

out <- read.fas("R2GBLOCK.fas-gb")

unlink(list.files(pattern = "R2GBLOCK"))

setwd(rwd)

out

}

<environment: namespace:phyloch>

Working on windows with the following command.

testgblock=gblocks(testmafft, exec = "C:/Users/deme/Documents/Programs/Gblocks/Gblocks\_Windows\_0.91b/Gblocks\_0.91b")

Another example using prank function in the phyloch package

function (x, outfile = "PRANK", guidetree = NULL, gaprate = 0.025,

gapext = 0.75, path)

{

rwd <- getwd()

if (class(guidetree) == "phylo") {

missingseqs <- which(!guidetree$tip.label %in% x$nam)

if (length(missingseqs) > 0)

guidetree <- drop.tip(guidetree, missingseqs)

missingtips <- which(!x$nam %in% guidetree$tip.label)

if (length(missingtips) > 0) {

txt <- paste(x$nam[missingtips], "not contained in guide tree")

stop(txt)

}

if (!all(guidetree$tip.label %in% x$nam))

stop("Guidetree does not match sequences")

write.tree(guidetree, "prank\_guidetree.tre")

gtfile <- paste(rwd, "/prank\_guidetree.tre", sep = "")

id <- match(guidetree$tip.label, x$nam)

x$nam <- x$nam[id]

x$seq <- x$seq[id]

}

write.fas(x, "prankinput.fas", interleave = FALSE)

infile <- paste("'", rwd, "/prankinput.fas'", sep = "")

if (class(guidetree) == "phylo") {

call.prank <- paste(path, " -F -d=", infile, " -t=",

gtfile, " -o=", outfile, " -gaprate=", gaprate, " -gapext=",

gapext, sep = "")

}

else {

call.prank <- paste(path, " -F -d=", infile, " -o=",

outfile, " -gaprate=", gaprate, " -gapext=", gapext,

sep = "")

}

system(call.prank)

if (class(guidetree) == "phylo")

fn <- paste(outfile, "1.fas", sep = ".")

else fn <- paste(outfile, "2.fas", sep = ".")

read.fas(fn)

}

<environment: namespace:phyloch>

4) Problem with Align.Concat

The second time you run the command in the same folder and the Concat files have already been created the function throw an error message (I guess the function also try to include the nexus file).

# To run the example it might be better to copy the input alignment files

# provided by the package to a temporary directory created into the

# current working directory.

src.dir = system.file("extdata/multi.align/ForConcat", package = "regPhylo")

dir.create("TempDir.ForConcat")

# Set up the path of the TempDir folder.

dest.dir = paste(getwd(), "/TempDir.ForConcat", sep="")

file.names <- dir(src.dir)

# Copy all the files stored in regPhylo/extdata/multi.align/ForConcat"

# into a temporary folder.

sapply(file.names, function(x) {

file.copy(from = paste(src.dir, x, sep = "/"),

to = paste(dest.dir, x, sep = "/"),

overwrite = FALSE) })

# Run the function to build the supermatrix.

Align.Concat(input = "TempDir.ForConcat", Sp.List = NULL, outputConcat = NULL)

# Run the function to build a supermatrix including two species without DNA.

Align.Concat(input = "TempDir.ForConcat",

Sp.List = c("Titi\_titi", "Toto\_toto"),

outputConcat = "TempDir.ForConcat/Concat\_2spNoDNA")

Error in seqinr::read.fasta(listAl[i], as.string = TRUE) :

no line starting with a > character found

In addition: Warning message:

In readLines(file) :

incomplete final line found on 'TempDir.ForConcat/Concat\_2spNoDNA.nex'

#1 after the following line of code

listAl = paste(input, list.files(input), sep = "/")

We need to select only the fasta file with the .fas extension.

5) Problem in the example file for the function

It was

ClassifDF = read.csv("TempDir.CladeAge/Classif16sp.csv", header = TRUE)

Need to be

ClassifDF = read.csv("TempDir.TopoConstraints2/Classif16sp.csv", header = TRUE)