Roberta Graboski

Meu nome é Roberta Graboski, sou aluna de doutorado do IBUSP. Desenvolvo minha pesquisa no museu de zoologia da USP. Meu projeto é filogenia molecular de Amphisbaenia e evolução das formas da cabeça neste grupo utilizando morfometría geométrica.

exec

Trabalho Final

Plano A

O GenBank é um banco de dados de sequências de DNA e de aminoácidos do Centro de Informações Biotecnológicas do Estados Unidos (http://www.ncbi.nlm.nih.gov). Este banco de dados possui informações genéticas de milhares de seres vivos e atualmente e é uma ferramenta importante para quem trabalha com filogenias moleculares. Utilizando as sequências depositadas neste banco, é possível gerar matrizes com grande quantidade de dados. Entretanto, necessitamos colocar todas as informações das sequências que baixamos, tais como: número de acesso do GenBank, organismo, quantos pares de bases tem a sequência, etc. Todas essas informações geralmente são colocadas manualmente em uma tabela. O objetivo da minha função seria baixar através do R as sequências do GenBank, realizando uma busca por organismo (o pacote “ape” ou “rentrez” irão auxiliar nesta etapa) e gerar um data frame com as informações contidas em cada sequência que baixamos.

Plano B

As árvores filogenéticas são representação gráficas das relações evolutivas entre os organismos. Para construir uma árvore filogenética, podemos utilizar matrizes de dados moleculares com diversos genes, que posteriormente são concatenados para formar uma matriz única. Quando olhamos uma filogenia, geralmente não sabemos o quão completa está se encontra. Por exemplo, um táxon pode possuir 1 gene sequenciado, enquanto que outro táxon possui 7 genes sequenciados . Entretanto, para saber esta informação temos que retornar a matriz original e ver individuo por individuo quantos genes temos para cada organismo. Está não é uma maneira clara de visualizarmos os dados. O objetivo da minha função é plotar ao lado de cada táxon da filogenia a quantidade de genes que este possui sequenciado, utilizando uma legenda de cores. Para isso o pacote “ape” irá me auxiliar.

Código do Plano B

```
library(ape)
library(geiger)
z<-function(x,y)
{
    x<-read.tree(file.choose()) #busca do arquivo com a arvore
    if(class(x)=="phylo") # testa se o arquivo e da classe phylo
        {
        cat("\nYour tree file was loaded\n")
        t = ladderize(x) # laderiza a arvore
        }
    else
        {
        cat("\nYour file is not a tree in newick format\n")
        }
    y<-read.csv(file.choose()) #busca o arquivo com a lista de genes
    if(class(y)=="data.frame")
        {
        cat("\nYour csv file was loaded\n")
        l<-y
        l.data = data.frame(l[,2]) #transforma as colunas em data.frame
        rownames(l.data) = l[,1] #adiciona o nome das linhas do data.frame
baseado nos dados da lista
        N_genes = l.data[match(t$tip.label,rownames(l.data)),] #cria objeto
para fazer o match dos nomes da lista e dos terminais da arvore
        attach(l.data)#faz o data.frame ser default
        name.check(t, l.data) -> check_data #avalia o overlap dos dados
        if (check_data=="OK")
            {
            cat("\nThe match between your table and tree is OK\n")
            }
        else
            {
            cat("\nThe match between your table and tree is NOT OK\n")
            }
        }
    else
        {
        cat("\nYour file with the list of genes is not a csv file\n")
        }
    t_label <- character(length(t$tip.label)) #cria a lista de cores
    names(t_label) <- names(N_genes)
    t_label[N_genes==1] <- "red"
    t_label[N_genes==2] <- "orange"
    t_label[N_genes==3] <- "yellow"
    t_label[N_genes==4] <- "light green"
    t_label[N_genes==5] <- "green"
```

```
    t_label[N_genes==6] <- "dark green"
    names(t_label) <- row.names(l.data)
    pdf(file.choose(new=TRUE))# salva pdf com a arvore escolhendo o nome
    plot.phylo(t, no.margin=TRUE, x.lim=length(t$tip.label), edge.width=0.1,
cex=0.8, label.offset=(length(t$tip.label)/10))# estes valores podem ser
alterados visando um grafico melhor
    points(rep((length(t$tip.label)-(length(t$tip.label)/10)),
length(t$tip.label)), 1:length(t$tip.label), pch=22,
bg=t_label[t$tip.label], cex=(1), lwd=1)# estes valores podem ser alterados
visando um grafico melhor
    dev.off()
}
```

Help da função z

```
z.r            package:                R Documentation



Description:

     Função para plotar ao lado de cada táxon da filogenia a quantidade de
genes que este possui sequenciado utilizando uma legenda de cores. Salva um
arquivo em pdf.

Usage:

     z(x,y)


Arguments:

x: arvore filogenetica em formato Newick (parentetico)
y: lista com o número de genes em formato csv


Details:

A função permite selecionar os arquivos por uma janela gráfica.
A função permite selecionar o nome do arquivo em pdf que será salvo.

Warning:
    Dependendo do tamanho da filogenia, alterações nos parâmetros gráficos
podem ser necessárias

Author:

    Graboski, R(roberta.graboski@gmail.com)
```

```
References:


Examples:

# Carregue a função e digite exatamente a linha abaixo, selecionando os
arquivos na ordem: 1-arvore, 2-lista#
z(x,y)
```

Plano C

Ainda não pensei..:(

Roberta, sua proposta A está ok, precisa só definir bem o que vai no argumento ou argumentos da função (é só o organismo? não precisa definir o gene ou outras informações?). Você parece já ter visto os pacotes necessários, só se certifique que você consegue importar todas essas informações que você quer direto do GenBank (eu dei uma olhada e não achei essa opção).

Achei a proposta B bem interessante! Se der para manipular a saída da árvore filogenética produzida usando esse pacote ape, acho que fica bem legal (mas tem que verificar isso antes). De novo, precisa definir melhor os dados de entrada.

Reescreva as propostas incluindo os dados de entrada. Se quiser algumas dicas, veja Trabalho final.

—- *Sheina*