Raquel Gardini Sanches Palasio

Bióloga, Doutoranda em Epidemiologia, Faculdade de Saúde Publica, USP

Título da tese: Padrão de distribuição da diversidade genética molecular e espacial de *Biomphalaria* e sua relação com a ocorrência da esquistossomose, região do médio Paranapanema, Estado de São Paulo.

Orientada pelo Prof Dr. Francisco Chiaravalloti Neto e pela pesquisadora da Sucen Dr. Roseli Tuan.

Meus exec

Proposta de Trabalho Final

Proposta A

Uma função irá organizar sequência(s) nucleotídica(s) e as informações destas em um arquivo *.txt* selecionado a partir de uma pasta do computador do usuário.

Objetos e argumentos de entrada e saída

As opções de entrada conteriam um vetor, com os códigos das sequências que queira selecionar e a pasta onde as mesmas estarão. Estas devem estar em formato *.fasta* com o mesmo nome ou parte do nome da sequência dada no vetor. E uma *.data.frame* com as informações da espécie, localidade, data da coleta e coordenadas nas colunas, uma delas com os mesmos códigos das sequências, não precisando ser do mesmo tamanho que o vetor.

A função retornaria um arquivo *.txt*, como abaixo:

“>766_Bg_Our [organism=Biomphalaria glabrata] cytochrome oxidase subunit I (COI) gene, partial sequence; mitochondrial [Specimen Voucher=LBMSU766] [country=Brazil: Sao Paulo: Ourinhos: Corrego Christoni] [collection date= jun, 2015] [Lat-Lon=-22.967361,-49.874194]” TTATCATTATTGATTCGTTTGGAGCTTGGTACTACTC….

“>788_Bg_Our [organ….”

https://www.ncbi.nlm.nih.gov/WebSub/html/help/fasta.html

Para isso pretendo usar o pacote pegas que tem a função read.dna para ler os arquivo fastas. A função list.file e grep, para encontrar a pasta e selecionar o arquivo com parte do nome. Pensei em dar uma opção de qual gene irá aparecer no arquivo final se a pessoa não escrever nada (seria gene=“COI” : cytochrome oxidase subunit I (COI) gene, partial sequence; mitochondrial) ou na entrada terá que ser escrito.

Uso da função

O arquivo que a função retornará poderá ser utilizado para ajudar a depositar as sequências em um banco de dados como o NCBI-GenBank. Pois neste formato o banco de dados lê automaticamente, e você não precisará adicionar as informações uma a uma, poupando tempo. E mesmo para arrumar os arquivos *fasta* irá ajudar, bastando colocar todas as sequências em uma pasta e um arquivo com todas essas informações. E com essa função é possível selecionar apenas algumas sequências de interesse de forma mais simples.

Proposta B

Uma função irá procurar a temperatura ou precipitação etc... de um data frame para outro. E fazer uma regressão linear. A função retornará um plot e os coeficientes da regressão.

Objetos e argumentos de entrada e saída

Para isso o usuário deve fornecer dois arquivos de entrada, ambos *data.frame*, não precisa ser de mesmo tamanho. Um com os dados observados com as datas das coletas nas colunas e outro *data.frame* com dados climáticos com datas e variáveis preditoras nas colunas. Ex. temperatura ou precipitação. O usuário deverá informar qual coluna e a variável que será estudada.

Para isso pretendo em usar a função merge, para selecionar os dados de uma planilha para outra.

Pensei em adicionar na função a espécie e com isso fazer um plot para cada espécie com a linha da regressão, coeficiente de cada e o p-valor, e retornando uma tabela com esses valores.

Uso da função

A possibilidade de fazer uma regressão com temperatura ou outra variável de um banco de dados externo confiável. Em casos que não foi feito a medição na hora da coleta. Também podem ser utilizadas outras variáveis preditoras como média de precipitação etc. Exemplo: Tenho os dados da quantidade de certa espécie coletada em determinada data. Com isso consigo achar a temperatura daquele dia ou do mês. Com dados previamente baixado de um site meteorológico. Para isso, não preciso encontrar uma a uma, mas a função me retornará todas de uma vez, e também fará uma regressão para ver se há relação entre as variáveis.

> Oi Raquel,
>
> achei sua proposta A muito boa, e parece que vc já sabe como prosseguir, então manda bala. Não se esqueça de colocar no help um exemplo que rode, caso o exemplo precise de arquivos, coloque algum arquivo de exemplo acompanhando sua função pra gente conseguir conferir se

ela roda.

—-*Danilo G Muniz*

Trabalho Final - Proposta A

Função genbank

```
 genbank <- function(x, pasta, gene, primer, n, organism=NULL, voucher=NULL,
country=NULL, state=NULL, municipio=NULL, local=NULL, collection_date=NULL,
Lat=NULL, Lon=NULL)
{
  require(pegas) ### tem a função read.dna, precisa estar instalado antes
  library(pegas) ### carregar o pacote
  list<- list.files( pasta, pattern= primer) ## lista na minha pasta os
arquivos com uma certa palavra, que aqui e o primer.
  #### conferir se ha arquivo fasta com o nome de x
  for (i in 1:length(x))
  {
    seq <- grep (c(x[i]), list)
    if(length(seq)==0)
    {stop("Na pasta informada esta faltando arquivo com os valores de x= ",
x[i] )
    }
    else{
      for (i in 1:length(x))
      {
        seq <- list[grep(c(x[i]), list)]
        conferir <- seq[grep("fasta", seq)]
        if(length(conferir)==0)
        {stop("Na pasta informada esta faltando arquivo no formato fasta,
com os valores de x= ", x[i] )
        }
      }
    }
  }

  #### etapa para conferir se na minha pasta tem os aquivos que preciso para
a função
  arquivo <- list[grep("fasta", list)] ### lista os arquivos com fasta, no
aquivo nomeado list
  if(length(arquivo)<length(x))  ### se o tamanho do arquivo for menor que a
quantidade x a função para
  {
    falta <- length(x)-length(arquivo)
    stop("Na pasta informada esta faltando ", falta," arquivo fasta, tem ",
```

```
length(x), " de valor de entrada e ", length(list), " arquivos fasta") ### para a função caso esteja faltando fasta na pasta
  }
  ### começando a função
  results <- rep(NA,length(x)) ## arquivos vazios para rodar o for
  name <- rep(NA,length(x))
  dado <- rep(NA,length(x))
  for (i in 1:length(x))
  {
    seq <- list[grep(c(x[i]), list)] #### procurar no vetor list ( com a função grep), os valores de x
    fasta <- seq[grep("fasta", seq)] #### procurar no vetor seq, os valores com fasta no nome
    pop <-read.dna(fasta, format="fasta")  ### ler a sequencia em formato fasta com o nome selecionado
    y <- paste(pop, collapse = "") #### juntar todas as bases hidrogenadas em um único carácter
    results[i] <- y   #### colocar as sequencias no vetor results
    name[i] <- paste(">",x[i], collapse = "", sep="") ### adicionar o nome da sequencia, preciso do ">", para a saida ser um formato fasta, e mais o valor do x de cada sequencia
    dado<-   strwrap(paste(name, results)) ### juntar o nome e a sequencia, separando-os por um quebra linha por isso que usei a função strwrap, necessário para manter o formato fasta
    }
  ### gene COI
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) & is.null(Lon) & is.null(collection_date) & is.null(state) & is.null(country)& is.null(organism)) ### para dados faltantes
  {if(gene=="COI") ### selecionar o gene COI
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," cytochrome oxidase subunit I (COI) gene, partial sequence;mitochondrial", collapse = "", sep="") ### substituir o nome que tenha o mesmo valor de x, com mais informações, no caso apenas que e COI
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua área de trabalho com o nome COI_result, do gene COI") ### informar que foi salvo o arquivo com n sequencias
    return( write(dado, file="COI_result.txt")) } ### salvar um arquivo txt na pasta do Working Directory do R
  }
  #### Fiz o "if" para as possibilidade da dados faltantes e um com todas as informações que pode ter a função: especie, localidade, lat-long etc...., e depois fiz o mesmo para o gene 16s, e um com um outro gene que o usuário queira colocar, com suas possibilidades de informações de entrada.
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) & is.null(Lon) & is.null(collection_date) & is.null(state) & is.null(country)) ### para dados faltantes
```

```
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome COI_result, com a especie, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state)) ### para dados
faltantes
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=", country[n==x[i]],"]" , collapse = "",
sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome COI_result, com a especie e pais, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date)) ### para dados faltantes
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=",
country[n==x[i]],":",state[n==x[i]],"]" , collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome COI_result, com a espécie, pais e estado, do
gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon)) ### para dados faltantes
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=",
country[n==x[i]],":",state[n==x[i]],"]" , " [collection
date=",collection_date[n==x[i]], "]", collapse = "", sep="")
```

```
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome COI_result, com nome da especie, pais e data de
coleta, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher)& is.null(local) & is.null(Lat) & is.null(Lon)) ###
para dado faltante
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],
":", municipio[n==x[i]],"]" , " [collection date=",collection_date[n==x[i]],
"]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, com o nome COI_result, com nome da especie, pais,
município e data de coleta, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)) ### para dados
faltantes
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=",
country[n==x[i]],":",state[n==x[i]],"]" , " [collection
date=",collection_date[n==x[i]], "]", " [Lat-Lon=", Lat[n==x[i]], ",",
Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher, município e local, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher) & is.null(local)) ### para dados faltantes
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],
":", municipio[n==x[i]],"]" , " [collection date=",collection_date[n==x[i]],
"]", " [Lat-Lon=", Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "",
sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
```

```
área de trabalho, sem Voucher e local, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher) & is.null(Lat) & is.null(Lon)) ### para dado faltante
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],
":", municipio[n==x[i]],":", local[n==x[i]],"]" , " [collection
date=",collection_date[n==x[i]], "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher e latitude e longitude, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  if(is.null(voucher)) ### para dados faltantes
  {if(gene=="COI")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],
":", municipio[n==x[i]], ":", local[n==x[i]], "]" , " [collection
date=",collection_date[n==x[i]], "]", " [Lat-Lon=", Lat[n==x[i]], ",",
Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher, do gene COI")
    return( write(dado, file="COI_result.txt")) }
  }
  #### PARA TODOS OS CAMPOS
  else
  {
    if(gene=="COI")
    {for (i in 1:length(x))
    {
      dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," cytochrome oxidase subunit I (COI) gene, partial
sequence;mitochondrial"," [Specimen Voucher=", voucher[n==x[i]], "],"
[country=", country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]],
":", local[n==x[i]], "]" , " [collection date=",collection_date[n==x[i]],
"]", " [Lat-Lon=", Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "",
sep="")
    }
      cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, do gene COI")
      return( write(dado, file="COI_result.txt")) }
  }
  #### 16S ####
```

```
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state) & is.null(country)&
is.null(organism)) ### para dados faltantes
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," 16S ribosomal RNA gene,
partial sequence; mitochondrial", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome 16S_result, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state) & is.null(country))
### para dados faltantes
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome 16S_result, com a especie, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state)) ### para dados
faltantes
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],"]" , collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
area de trabalho com o nome 16S_result, com a especie e pais, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date)) ### para dados faltantes
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],"]" ,
collapse = "", sep="")
```

```
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome 16S_result, com a especie, pais e estado, do
gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon)) ### para dado faltante
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],"]" , "
[collection date=",collection_date[n==x[i]], "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome 16S_result, com nome da especie, pais e data de
coleta, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher)& is.null(local) & is.null(Lat) & is.null(Lon)) ###
para dado faltante
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]], ":",
municipio[n==x[i]],"]" , " [collection date=",collection_date[n==x[i]], "]",
collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, com o nome 16S_result, com nome da especie, pais,
municipio e data de coleta, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)) ### para dados
faltantes
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]],"]" , "
[collection date=",collection_date[n==x[i]], "]", " [Lat-Lon=",
Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher, município e local, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
```

```
  }
  if(is.null(voucher) & is.null(local)) ### para dado faltante
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]], ":",
municipio[n==x[i]],"]" , " [collection date=",collection_date[n==x[i]], "]",
" [Lat-Lon=", Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher e local, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher) & is.null(Lat) & is.null(Lon)) ### para dados
faltantes
  {if(gene=="16S")
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial", " [country=", country[n==x[i]],":",state[n==x[i]], ":",
municipio[n==x[i]],":", local[n==x[i]],"]" , " [collection
date=",collection_date[n==x[i]], "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher e latitude e longitude, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  }
  if(is.null(voucher))
  {
    if(gene=="16S")
    {for (i in 1:length(x))
    {
      dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial"," [country=", country[n==x[i]],":",state[n==x[i]], ":",
municipio[n==x[i]], ":", local[n==x[i]], "]" , " [collection
date=",collection_date[n==x[i]], "]", " [Lat-Lon=", Lat[n==x[i]], ",",
Lon[n==x[i]],  "]", collapse = "", sep="")
    }
      cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher do gene 16S")
      return( write(dado, file="16S_result.txt")) }
  }
  ### tudo
  if(gene=="16S")
  {for (i in 1:length(x))
  {
```

```
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]"," 16S ribosomal RNA gene, partial sequence;
mitochondrial"," [Specimen Voucher=", voucher[n==x[i]], "]"," [country=",
country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]], ":",
local[n==x[i]], "]" , " [collection date=",collection_date[n==x[i]], "]", "
[Lat-Lon=", Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, do gene 16S")
    return( write(dado, file="16S_result.txt")) }
  #### outro gene
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state) & is.null(country)
& is.null(organism)) ### para dado faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i],gene, collapse = "",
sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, do gene", gene)
    return( write(dado, file="result.txt"))  }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state) & is.null(country))
### para dado faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com o nome result, com a especie, do", gene)
    return( write(dado, file="result.txt")) }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date) & is.null(state)) ### para dados
faltantes
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=", country[n==x[i]],"]" , collapse
= "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com a especie e pais, do gene", gene )
    return( write(dado, file="result.txt")) }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon) & is.null(collection_date)) ### para dados faltantes
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=",
```

```
country[n==x[i]],":",state[n==x[i]],"]" , collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com a especie, pais e estado, do gene", gene )
    return( write(dado, file="result.txt")) }
  if(is.null(voucher) & is.null(municipio) & is.null(local)& is.null(Lat) &
is.null(Lon)) ### para dado faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=",
country[n==x[i]],":",state[n==x[i]],"]" , " [collection
date=",collection_date[n==x[i]], "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com nome da especie, pais e data de coleta, do gene", gene
)
    return( write(dado, file="result.txt")) }
  if(is.null(voucher)& is.null(local) & is.null(Lat) & is.null(Lon)) ###
para dado faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=",
country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]],"]" , "
[collection date=",collection_date[n==x[i]], "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho com nome da especie, pais, município e data de coleta, do
gene", gene )
    return( write(dado, file="result.txt")) }
  if(is.null(voucher) & is.null(municipio) & is.null(local)) ### para dado
faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=",
country[n==x[i]],":",state[n==x[i]],"]" , " [collection
date=",collection_date[n==x[i]], "]", " [Lat-Lon=", Lat[n==x[i]], ",",
Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher, município e local, do gene", gene )
    return( write(dado, file="result.txt")) }
  if(is.null(voucher) & is.null(local)) ### para dado faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=",
country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]],"]" , "
```

```
[collection date=",collection_date[n==x[i]], "]", " [Lat-Lon=",
Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
area de trabalho, sem Voucher e local, do gene", gene)
    return( write(dado, file="result.txt")) }
  if(is.null(voucher) & is.null(Lat) & is.null(Lon)) ### para dado faltante
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene , " [country=",
country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]],":",
local[n==x[i]],"]" , " [collection date=",collection_date[n==x[i]], "]",
collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher e latitude e longitude, do gene", gene)
    return( write(dado, file="result.txt")) }
  if(is.null(voucher))
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene ," [country=",
country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]], ":",
local[n==x[i]], "]" , " [collection date=",collection_date[n==x[i]], "]", "
[Lat-Lon=", Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "", sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, sem Voucher do gene", gene)
    return( write(dado, file="result.txt")) }
  ### todas as informações disponíveis em outro gene, informado pelo usuário
  else
  {for (i in 1:length(x))
  {
    dado[grep(c(x[i]), dado)] <-  paste(">",x[i]," [organism=",
organism[n==x[i]], "]",gene ," [Specimen Voucher=", voucher[n==x[i]], "],"
[country=", country[n==x[i]],":",state[n==x[i]], ":", municipio[n==x[i]],
":", local[n==x[i]], "]" , " [collection date=",collection_date[n==x[i]],
"]", " [Lat-Lon=", Lat[n==x[i]], ",", Lon[n==x[i]],  "]", collapse = "",
sep="")
  }
    cat("Foi salvo um arquivo txt com ", length(x), " sequencias, na sua
área de trabalho, do gene", gene)
    return( write(dado, file="result.txt")) }
}
```

Help da função genbank

```
genbank                 package:unknown                 R Documentation
```

```
  Função ira a organizar a(s) sequencia(s) nucleotídica(s) de uma pasta do
usuário e adicionara as informações como especie, localidade, data da coleta
e coordenadas em um arquivo txt.
  Description:
    O arquivo que a função retorna estará com o formato para depositar as
sequências em um banco de dados como o NCBI, GenBank. A função seleciona a
pasta do usuário e encontra os arquivos com o nome do vetor que o usuário
forneceu de entrada, com os códigos das sequencia(s) que queira selecionar.
Esta(s) devem estar em formato fasta com o mesmo nome ou parte do nome da
sequencia dada no vetor. E uma data.frame com as informações da especie,
localidade, data da coleta e coordenadas nas colunas, com uma coluna com os
mesmo códigos da sequencia, estes não precisam ser do mesmo tamanho que o
vetor.
  Usage:
  genbank <- function(x, pasta, gene, primer, n, organism=NULL,
voucher=NULL, country=NULL, state=NULL, municipio=NULL, local=NULL,
collection_date=NULL, Lat=NULL, Lon=NULL)

  Arguments:
    x                vetor com o códigos das sequencias (Ver detalhe)
    pasta            pasta onde estão as sequências fasta
    gene             gene da sequencias, como padrão pode ser COI, 16S, mas
pode ser qualquer outro
    primer           fowrd ou o reverso (Ver detalhe)
    n                coluna do data.frame, que corresponde o mesmo numero de
x.
    organism         coluna do data.frame como a especie
    voucher          coluna do data.frame como a identificação do voucher
    country          coluna do data.frame como o pais de coleta do organismo
    state            coluna do data.frame como o estado de coleta do
organismo
    municipio        coluna do data.frame como o município de coleta do
organismo
    local            coluna do data.frame como o local de coleta do
organismo (Ver detalhes)
    collection_date  coluna do data.frame como a data de coleta do organismo
(Ver detalhes)
    Lat              coluna do data.frame como a latitude
    Lon              coluna do data.frame como a longitude
  Details:
  o valor de x tem que corresponder o valor de n da coluna do data.frame
  primer tem que estar no nome da sequência fasta da pasta fornecida pelo
usuário. Ex: 346_COI_HCO, primer=HCO do gene COI.
  o valor de n e uma coluna do data.frame que já deve esta aberto na área de
trabalho do r.
  Local pode ser algo mais especifico, um córrego, rio, rua etc....
  Necessário estar instalado o pacote pegas.
  Value:
  O vetor x não precisa ter o mesmo tamanho que o n do data.frame.
  Se o gene=COI
```

```
  comp1 :   genbank(x,"C:/Curso R/seq", gene="COI", primer= "HCO", n=z$DNA),
retorna uma arquivo txt com "cytochrome oxidase subunit I (COI) gene,
partial sequence;mitochondrial"
  Se o gene=16S
  comp2 :  genbank(x,"C:/Curso R/seq", gene="16S", primer= "AR", n=z$DNA),
retorna uma arquivo txt com "16S ribosomal RNA gene, partial sequence;
mitochondrial"
  Se o gene= a outro fornecido pelo usuário.
  comp3 :  genbank(x2,"C:/Curso R/seq", gene=" ribosomal RNA gene, partial
sequence; internal transcribed spacer 2 (ITS2)",primer= "ITTS2F")
  retorna uma arquivo txt com " ribosomal RNA gene, partial sequence;
internal transcribed spacer 2 (ITS2)"



  Warning:
  Se não tiver na pasta um arquivo fasta com valor de x, a função para e
retorna um mensagem do de erro com o valor
  que esta faltando.
  se estiver faltando um arquivo fasta na pasta, a função para e da um aviso
(Na pasta informada esta faltando n
  arquivo fasta, tem n de valor de entrada e n arquivos fasta).

  Author(s):
  Raquel Gardini Sanches Palasio
  raquelpalasio@usp.br
  References:
  Paradis E. 2010. pegas: an R package for population genetics with an
integrated–modular approach.
  Bioinformatics 26: 419–420. doi:10.1093/bioinformatics/btp696.
  https://www.ncbi.nlm.nih.gov/WebSub/html/help/fasta.html
  See Also:
  pegas, read.dna, list.files
  Examples:
  # carregar antes
  x <- c(346, 491, 492, 493)
  x2 <- c(650, 651, 653, 655, 658, 659)
  z <- read.csv("exemplo.csv", header = T, sep= ";", dec = ".")### Esta no
arquivo zip
  setwd("C:/Users/Raquel/Desktop/exemplo") # arquivo zip tem que ser
descompactado dentro da pasta com nome exemplo
  ### Com o Gene COI com todas as variáveis
  genbank(x,"C:/Users/Raquel/Desktop/exemplo", gene="COI", primer= "HCO",
n=z$DNA, organism=z$Especie, voucher=z$N_LBMSU, country=z$Pais,
state=z$Estado, municipio=z$Municipio, local=z$Local.de.coleta,
collection_date=z$Data.de.coleta, Lat=z$Latitude, Lon=z$Longitude)
  ### Com o Gene 16S com todas as variáveis
  genbank(x2,"C:/Users/Raquel/Desktop/exemplo", gene="16S", primer= "AR",
n=z$DNA, organism=z$Especie, voucher=z$N_LBMSU, country=z$Pais,
state=z$Estado, municipio=z$Municipio, local=z$Local.de.coleta,
collection_date=z$Data.de.coleta, Lat=z$Latitude, Lon=z$Longitude)
  ### Com outro gene com todas as variáveis, um valor de x
```

```
  genbank("650","C:/Users/Raquel/Desktop/exemplo", gene="ribosomal RNA gene,
partial sequence; internal transcribed spacer 2 (ITS2)", primer= "AR",
n=z$DNA, organism=z$Especie, voucher=z$N_LBMSU, country=z$Pais,
state=z$Estado, municipio=z$Municipio, local=z$Local.de.coleta,
collection_date=z$Data.de.coleta, Lat=z$Latitude, Lon=z$Longitude)
  ### apenas com a variável organism do Gene COI
  genbank(x,"C:/Users/Raquel/Desktop/exemplo", gene="COI", primer= "HCO",
n=z$DNA, organism=z$Especie)
```

Arquivos para rodar o exemplo

Pasta com exemplos Função genbankHelp da função

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:raquelpalasio:start

Last update: **2020/08/12 06:04**