Package 'AnnotationBustR'

October 12, 2022

Version 1.3.0

Date 2020-09-23

Title Extract Subsequences from GenBank Annotations

Author Samuel R. Borstein <sam@borstein.com>, Brian O'Meara <bomeara@utk.edu>

Maintainer Samuel R. Borstein <sam@borstein.com>

Depends R (>= 3.4)

Imports ape (>= 4.1), seqinr (>= 3.3-6)

Description Extraction of subsequences into FASTA files from GenBank annotations where gene names may vary among accessions. Borstein & O'Meara (2018) <doi:10.7717/peerj.5179>.

License GPL (>= 2)

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://github.com/sborstein/AnnotationBustR, https://www.ncbi.nlm.nih.gov/nuccore, https://peerj.com/articles/5179/

BugReports https://github.com/sborstein/AnnotationBustR/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-24 08:10:13 UTC

R topics documented:

AnnotationBust 2
AnnotationBustR 4
cpDNAterms 5
FindLongestSeq 6

MergeSearchTerms 7
mtDNAterms 8
mtDNAtermsPlants 8
rDNAterms 9

Index **10**

`AnnotationBust` *Breaks up genbank sequences into their annotated components based on a set of search terms and writes each subsequence of interest to a FASTA for each accession number supplied.*

Description

Breaks up genbank sequences into their annotated components based on a set of search terms and writes each subsequence of interest to a FASTA for each accession number supplied.

Usage

```
AnnotationBust(
  Accessions,
  Terms,
  Duplicates = NULL,
  DuplicateInstances = NULL,
  TranslateSeqs = FALSE,
  TranslateCode = 1,
  DuplicateSpecies = FALSE,
  Prefix = NULL,
  TidyAccessions = TRUE
)
```

Arguments

`Accessions`	A vector of INSDC GenBank accession numbers. Note that refseq numbers are not supported at the moment (i.e. prefixes like XM_ and NC_) will not work.
`Terms`	A data frame of search terms. Search terms for animal mitogenomes, nuclear rRNA, chloroplast genomes, and plant mitogenomes are pre-made and can be loaded using the data()function. Additional terms can be addded using the MergeSearchTerms function.
`Duplicates`	A vector of loci names that have more than one copy. Default is NULL
`DuplicateInstances`	A numeric vector the length of Duplicates of the number of duplicates for each duplicated gene you wish to extract.Default is NULL
`TranslateSeqs`	Logical as to whether or not the sequence should be translated to the corresponding peptide sequence. Default is FALSE. Note that this only makes sense to list as TRUE for protein coding sequences.

`TranslateCode` Numerical representing the GenBank translation code for which sequences should be translated under. Default is 1. For all options see ?seqinr::getTrans. Some possible useful ones are: 2. Vertebrate Mitochondrial, 5. Invertebrate Mitochondrial, and 11. bacterial+plantplastid

`DuplicateSpecies` Logical. As to whether there are duplicate individuals per species. If TRUE, adds the accession number to the fasta file

`Prefix` Character If a prefix is specified, all output FASTA files written will begin with the prefix. Default is NULL.

`TidyAccessions` Logical as to whether the accession table should have a single row per species. If numerous accessions for a species occur, they will be seperated by a comma in the accession table. Default=TRUE.

Details

The AnnotationBust function takes a vector of accession numbers and a data frame of search terms and extracts subsequences from genomes or concatenated sequences. This function requires a steady internet connection. It writes files in the FASTA format to the working directory and returns an accession table. Files append, so use different prefixes between runs, otherwise they will be added to the current files in the working directory if the names are the same. AnnoitationBustR comes with pre-made search terms for metazoan mitogenomes, plant mitogenomes, chloroplast genomes, and rDNA that can be loaded using data(mtDNAterms), data(mtDNAtermsPlants),data(cpDNAterms), and data(rDNAterms) respectively. Search terms can be completely made by the user as long as they follow a similar format with three columns. The first, Locus, should contain the name of the locus that will also be used to name the files. We recommend following a similar naming convention to what we currently have in the pre-made data frames to ensure that files are named properly, characters like "-" or ".", and names starting with numbers should be avoided as it can cause errors with R. The second column, Type, contains the type of subsequence it is, with options being CDS, rRNA, tRNA, misc_RNA, Intro, Exon, and D_Loop. The last column, Name, consists of a possible name for the locus of interest as it might appear in an annotation. For numerous synonyms for the same locus, one should have each synonym as its own row.An additional fourth column is needed for extracting introns/exons. This column, called IntronExonNumber should contain the number of the intron to extract. If extracting both introns/exons and non-intron/exon sequences the fourth column should be NA for non-intron/exon sequence types. See the examples below and the vignette for detailed examples on extracting intron and exons. It is possible that some subsequences are not fully annotated on ACNUC and, therefore, are not extractable. These will return in the accession table as "type not fully Ann". It is also possible that the sequence has no annotations at all, for which it will return "No Ann. For". If the function returns "Acc. Not Found", the accession number supplied could not be found on NCBI. If "Not On ACNUC GenBank" is returned, the accession is not available through AcNUC. This may be due to ACNUC not being fully up to date. To see the last time ACNUC was updated, run seqinr::choosebank("genbank", infobank=T).

For a more detailed walkthrough on using AnnotationBust you can access the vignette with vignette("AnnotationBustR).

Value

Writes a fasta file(s) to the current working directory selected for each unique subsequence of interest in Terms containing all the accession numbers the subsequence was found in.

An accesion table of class data.frame.

References

Borstein, Samuel R., and Brian C. O'Meara. "AnnotationBustR: An R package to extract subsequences from GenBank annotations." PeerJ Preprints 5 (2017): e2920v1.

Examples

```
## Not run:
#Create vector of three NCBI accessions of rDNA toget subsequences of and load rDNA terms.
ncbi.accessions<-c("FJ706295","FJ706343","FJ706292")
data(rDNAterms)#load rDNA search terms from AnnotationBustR
#Run AnnotationBustR and write files to working directory
my.sequences<-AnnotationBust(ncbi.accessions, rDNAterms, DuplicateSpecies=TRUE,
Prefix="Example1")
my.sequences#Return the accession table for each species.

###Example With matK CDS and trnK Introns/Exons##
#Subset out matK from cpDNAterms
cds.terms<-subset(cpDNAterms,cpDNAterms$Locus=="matK")
#Create a vecotr of NA so we can merge with the search terms for introns and exons
cds.terms<-cbind(cds.terms,(rep(NA,length(cds.terms$Locus))))
colnames(cds.terms)[4]<-"IntronExonNumber"

#Prepare a search term table for the intron and exons to remove
#We can start with the cpDNAterms for trnK
IntronExon.terms<-subset(cpDNAterms,cpDNAterms$Locus=="trnK")
#As we want to go for two exons, we will want the synonyms repeated as we are doing and intron
#and an exon
IntronExon.terms<-rbind(IntronExon.terms,IntronExon.terms)#duplicate the terms
#rep the sequence type we want to extract
IntronExon.terms$Type<-rep(c("Intron","Intron","Exon","Exon"))
IntronExon.terms$Locus<-rep(c("trnK_Intron","trnK_Exon2"),each=2)
IntronExon.terms<-cbind(IntronExon.terms,rep(c(1,1,2,2)))#Add intron/exon number info
#change column name for number info for IntronExon name
colnames(IntronExon.terms)[4]<-"IntronExonNumber"
#We can then merge everything together with MergeSearchTerms terms
IntronExonExampleTerms<-MergeSearchTerms(IntronExon.terms,cds.terms)

#Run AnnotationBust
IntronExon.example<-AnnotationBust(Accessions=c("KX687911.1", "KX687910.1"),
Terms=IntronExonExampleTerms, Prefix="DemoIntronExon")

## End(Not run)
```

AnnotationBustR *An R package to extract sub-sequences from GenBank annotations under different synonyms*

Description

An R package to extract sub-sequences from GenBank annotations under different synonyms.

Details

Package: AnnotationBustR

Type: Package

Title: An R package to extract sub-sequences from GenBank annotations under different synonyms

Version: 1.1

Date: 2017-8-15

License: GPL (>= 2)

This package allows users to quickly extract sub-sequences from GenBank accession numbers that may be annotated under different synonyms. It writes these sub-sequences to FASTA files and creates a corresponding accession table. The package comes with pre-made search terms with synonyms. A vignette going over the basic functions and how to use them can be accessed with vignette("AnnotationBustR-vignette").

Author(s)

Samuel Borstein, Brian O'Meara. Maintainer: Samuel Borstein <sborstei@vols.utk.edu>

See Also

AnnotationBust,cpDNAterms,FindLongestSeq,MergeSearchTerms,mtDNAterms,rDNAterms

---

`cpDNAterms` *Chloroplast DNA (cpDNA) Search Terms*

Description

A data frame containing search terms for Chloroplast loci. Can be subset for loci of interest. Columns are as follows and users should follow the column format if they wish to add search terms using the MergeSearchTerms function:

Usage

```
cpDNAterms
```

Format

A data frame of of 364 rows and 3 columns

- Locus: Locus name, FASTA files will be written with this name
- Type: Type of subsequence, either CDS,tRNA,rRNA, or misc_RNA
- Name:Name of synonym for a locus to search for

See Also

MergeSearchTerms

FindLongestSeq *Find the longest sequence for each species from a list of GenBank accession numbers.*

Description

Find the longest sequence for each species from a list of GenBank accession numbers.

Usage

```
FindLongestSeq(Accessions)
```

Arguments

Accessions	A vector of GenBank accession numbers.

Details

For a set of GenBank accession numbers, this will return the longest sequence for in the set for species.

Value

A list of genbank accessions numbers for the longest sequence for each taxon in a list of accession numbers.

Examples

```
#a vector of 4 genbank accessions, there are two for each species.
genbank.accessions<-c("KP978059.1","KP978060.1","JX516105.1","JX516111.1")
## Not run:
#returns the longest sequence respectively for the two species.
long.seq.result<-FindLongestSeq(genbank.accessions)

## End(Not run)
```

MergeSearchTerms *Merging of two tables containing search terms to expand search term database for the AnnotationBust function.*

Description

This function merges two data frames with search terms. This allows users to easily add search terms to data frames (either their own or ones included in this package using data() as GenBank annotations for the same genes may vary in gene name.

Usage

```
MergeSearchTerms(..., SortGenes = FALSE)
```

Arguments

...	the data frames of search terms you want to combine into a single data frame The Data frame(s) should have stringsAsFactors=FALSE listed if you want to sort them.
SortGenes	Should the final data frame be sorted by gene name? Default is FALSE.

Value

A new merged data frame with all the search terms combined from the lists supplied. If sort.gene=TRUE, genes will be sorted by name.

Examples

```
#load the list of search terms for mitochondrial genes
data(mtDNAterms)

#Make a data.frame of new terms to add.
#This is a dummy example for a non-real annoation of COI, but lets pretend it is real.
add.name<-data.frame("COI","CDS", "CX1")

# make the column names the same for combination.
colnames(add.name)<-colnames(mtDNAterms)

#Run the merge search term function without sorting based on gene name.
new.terms<-MergeSearchTerms(add.name, mtDNAterms, SortGenes=FALSE)

#Run the merge search term function with sorting based on gene name.
new.terms<-MergeSearchTerms(add.name, mtDNAterms, SortGenes=TRUE)

#Merge search terms and create an additional column for introns and/or exons to
#In this example, add the trnK intron to the terms
#create empty IntornExonNumber column for non-intron/exons
cp.terms<-cbind(cpDNAterms,rep(NA,length(cpDNAterms$Name)))
colnames(cp.terms)[4]<-"IntronExonNumber"#Name the column IntronExonNumber
```

```
trnK.intron.terms<-subset(cpDNAterms,cpDNAterms$Locus=="trnK")#subset trnK
#Create a vector of 1's the same length as the number of rows for trnK
trnK.terms<-cbind(trnK.intron.terms,rep(1,length(trnK.intron.terms$Name)))
colnames(trnK.terms)[4]<-"IntronExonNumber"#Name the column IntronExonNumber
#Use MergeSearchTerms to merge the modified cpDNAterms and new intron terms
all.terms<-MergeSearchTerms(cp.terms,trnK.terms)
```

`mtDNAterms` *Mitochondrial DNA Search Terms for Animals*

Description

A data frame containing search terms for animal mitochondrial loci. Can be subset for loci of interest. Columns are as follows and users should follow the column format if they wish to add search terms using the MergeSearchTerms function:

Usage

```
mtDNAterms
```

Format

A data frame of of 253 rows and 3 columns

- Locus: Locus name, FASTA files will be written with this name
- Type: Type of subsequence, either CDS,tRNA,rRNA,misc_RNA, or D-loop
- Name:Name of synonym for a locus to search for

See Also

`MergeSearchTerms`

`mtDNAtermsPlants` *Mitochondrial DNA Search Terms for Plants*

Description

A data frame containing search terms for plant mitochondrial loci. Can be subset for loci of interest. Columns are as follows and users should follow the column format if they wish to add search terms using the MergeSearchTerms function:

Usage

```
mtDNAtermsPlants
```

Format

A data frame of of 248 rows and 3 columns

- Locus: Locus name, FASTA files will be written with this name
- Type: Type of subsequence, either CDS,tRNA,rRNA,misc_RNA, or D-loop
- Name:Name of synonym for a locus to search for

See Also

MergeSearchTerms

---

rDNAterms *Ribosomal DNA (rDNA) Search Terms*

Description

A data frame containing search terms for ribosomal RNA loci. Can be subset for loci of interest. Columns are as follows and users should follow the column format if they wish to add search terms using the MergeSearchTerms function:

Usage

```
rDNAterms
```

Format

A data frame of of 7 rows and 3 columns

- Locus: Locus name, FASTA files will be written with this name
- Type: Type of subsequence, either rRNA or misc_RNA
- Name:Name of synonym for a locus to search for

See Also

MergeSearchTerms

Index

∗ **datasets**
cpDNAterms, 5
mtDNAterms, 8
mtDNAtermsPlants, 8
rDNAterms, 9

AnnotationBust, 2, *5*
AnnotationBustR, 4

cpDNAterms, *5*, 5

FindLongestSeq, *5*, 6

MergeSearchTerms, *5, 6*, 7, *8, 9*
mtDNAterms, *5*, 8
mtDNAtermsPlants, 8

rDNAterms, *5*, 9