Package 'AnnotationBustR'

August 31, 2026

Version 2.0

Date 2026-08-30

Title Extract Subsequences from GenBank Annotations

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

Depends R (>= 4.5)

Imports Biostrings (>= 2.78.0), BSgenome (>= 1.78.0), dplyr (>= 1.2.1), GenomicFeatures (>= 1.62.0), GenomicRanges (>= 1.62.0), magrittr (>= 2.0.5), rentrez (>= 1.2.4), stringr (>= 1.6.0)

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 (>= 3)

LazyData true

Suggests knitr, rmarkdown, seqinr, testthat

VignetteBuilder knitr

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

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

Encoding UTF-8

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Samuel R. Borstein [aut, cre] (ORCID: <https://orcid.org/0000-0002-7258-141X>), Brian O'Meara [aut] (ORCID: <https://orcid.org/0000-0002-0337-5997>)

Repository CRAN

Date/Publication 2026-08-31 19:00:19 UTC

Contents

`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 = "None",
  DuplicateSpecies = FALSE,
  Prefix = NULL,
  TidyAccessions = TRUE,
  Reference = TRUE,
  Verbose = TRUE
)
```

Arguments

`Accessions`	A vector of GenBank accession numbers.
`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 added using the MergeSearchTerms function or a user supplied data frame may be provided.

`Duplicates` A vector of the features which occur more than once in the sequence and for which the extraction of multiple copies is desired. Default is NULL. In the case more than one copy exists when set to NULL, only the first instance will be extracted.

`DuplicateInstances` A numeric vector the length of Duplicates of the number of duplicates for each duplicated feature you wish to extract. Only needs to be used if Duplicates are provided.Default is NULL (i.e. no duplicates).

`TranslateSeqs` Should coding sequences (cds) be translated to the corresponding peptide sequence? Options include, Only, None, or Both. Both returns both the DNA sequence and the corresponding peptide sequence. Default is FALSE.

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

`Prefix` Character. If a prefix is specified, all output FASTA files written will begin with the prefix followed by an underscore. Default is NULL (i.e. no prefix).

`TidyAccessions` Logical. Should the accession table have a single row per species? If numerous accessions for a species occur, they will be separated by a comma in the accession table. Default=TRUE.

`Reference` Logical. Should reference information be captured and added as a column to the accession table? Default is TRUE.

`Verbose` Logical. Should progress be printed to the screen. The current accession and species name will be printed to the screen.

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 connects directly to the NCBI database and requires a steady internet connection. The function writes files in the FASTA format to the current working directory and returns an accession table. Files append, so use different prefixes between runs, otherwise they will be added to current files in the working directory with the same name.

AnnotationBustR 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 if they follow a similar format with three columns. The first, Feature, should contain the name of the features as the user wishes it be displayed in their files as it is used to name the files and create the accession table. 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 (eg. CDS, exon, intron, rRNA, tRNA, misc_RNA, misc_feature, D_Loop). The last column, Name, consists of a possible synonym for the feature 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 desired intron or exon 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.

In the event that an accession is not found on NCBI, the message "Accession # not found on NBI".

For a more detailed walk-through 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.

Author(s)

Samuel R. Borstein, Brian C. O'Meara

References

Borstein SR, and O'Meara BC. 2018. AnnotationBustR: an R package to extract subsequences from GenBank annotations. PeerJ 6:e5179. 10.7717/peerj.5179.

Examples

```
ncbi.accessions <- c("FJ706295","FJ706343","FJ706292")
data(rDNAterms)#load rDNA search terms from AnnotationBustR
my.sequences <- AnnotationBust(Accessions = ncbi.accessions, rDNAterms, DuplicateSpecies=TRUE,
Prefix="Example1", Reference = TRUE)
my.sequences

###Example With matK CDS and addint introns/exons for trnK###
#Subset out matK from cpDNAterms
cds.terms <- subset(cpDNAterms,cpDNAterms$Feature=="matK")
#Create a vecotr of NA so we can merge with the search terms for introns and exons
cds.terms <- cbind(cds.terms,(rep(NA,length(cds.terms$Feature))))
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$Feature=="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$Type <- rep(c("intron","intron","exon","exon"))
IntronExon.terms$Feature <- 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")
```

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: 2.0

Date: 2026-6-16

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 <sam@borstein.com>

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 391 rows and 3 columns

- Feature: Feature name, FASTA files will be written with this name.
- Type: Type of feature, either CDS,tRNA,rRNA.
- Name: Name of synonym for a feature 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, BatchSize = 300)
```

Arguments

Accessions	A vector of GenBank accession numbers.
BatchSize	Numeric. If the number of accessions is over the number provided, requests will be sent in batches of this amount. This is necessary for the NCBI servers. If you receive an HTTP 414 error, try to reduce the size of the batch. Default is 300.

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 accessions for each species.
genbank.accessions<-c("KP978059.1","KP978060.1","JX516105.1","JX516111.1")

#returns the longest sequence respectively for the two species.
long.seq.result <- FindLongestSeq(genbank.accessions)
```

`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 annotation 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 extract
#In this example, add the trnK intron to the terms

###Example With matK CDS and addint introns/exons for trnK###
#Subset out matK from cpDNAterms
cds.terms <- subset(cpDNAterms,cpDNAterms$Feature=="matK")
#Create a vecotr of NA so we can merge with the search terms for introns and exons
cds.terms <- cbind(cds.terms,(rep(NA,length(cds.terms$Feature))))
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$Feature=="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$Type <- rep(c("intron","intron","exon","exon"))
IntronExon.terms$Feature <- 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)
```

`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 254 rows and 3 columns

- Feature: Feature name, FASTA files will be written with this name.
- Type: Type of feature, CDS,tRNA,rRNA,misc_feature, or D-loop.
- Name: Name of synonym for a feature 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

- Feature: Feature name, FASTA files will be written with this name.
- Type: Type of feature, either CDS,tRNA,rRNA.
- Name: Name of synonym for a feature 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 14 rows and 3 columns

- Feature: Feature name, FASTA files will be written with this name.
- Type: Type of feature, either rRNA or misc_RNA.
- Name: Name of synonym for a feature to search for.

See Also

MergeSearchTerms

Index