Package 'hoardeR'

June 30, 2015

Type Package

Title Information Retrieval for Genetic Datasets

Version 0.1

Date 2015-06-29

Author Daniel Fischer [aut, cre],
Anu Sironen [aut]

Maintainer Daniel Fischer <daniel.fischer@luke.fi>

Depends R (>= 3.2.0)

Imports httr (>= 0.2), XML (>= 3.98-1.1), stringr (>= 0.6.2), MASS (>= 7.3-31), R.utils (>= 1.32.4), stats (>= 3.2.0), utils (>= 3.2.0), graphics (>= 3.2.0)

Description Information retrieval from National Center for Biotechnology Information (NCBI) databases, with main focus on identifying genes in unannotated organisms via Blast similarity search in annotated organisms.

License GPL (>= 2)

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-30 17:49:03

R topics documented:

`hoardeR-package` *Collect and Retrieve Annotation Data for Various Genetic Data.*

Description

The hoardeR package is designed for collecting, retrieving and transforming data from various sources. The current main focus is on setting up a connection to the NCBI Blast service. Also, the gene information for Ensemble Genes can be retrieved from NCBI. Methods for visualizing the results are currently under development. The latest 'night-build' of the package can be retrieved from

https://github.com/fischuu/hoardeR

Details

Package:	hoardeR
Type:	Package
Version:	0.1
Date:	2015-06-29
License:	GPL
LazyLoad:	yes

Author(s)

Daniel Fischer, Anu Sironen

Maintainer: Daniel Fischer <daniel.fischer@luke.fi>

blastSeq *Sending Genomic Sequences to NCBI Blast service*

Description

This function sends genomic sequences to the NCBI Blast service.

Usage

```
blastSeq(seq, n_blast=20, delay_req=3, delay_rid=60, email=NULL,
         xmlFolder=NULL,logFolder=NULL, keepInMemory=TRUE,
         database="chromosome", verbose=TRUE, createLog=FALSE)
```

Arguments

seq	The fasta sequence that should be blasted.
n_blast	Amount of parallel blast requests, in case seq is a vector.
delay_req	Seconds between the single Blast requests.
delay_rid	Seconds between the single result requests.
email	User email, required information from NCBI (String).
xmlFolder	Path to the result folder.
logFolder	Path to the log folder.
keepInMemory	Logical, shall the results be kept in the memory.
database	The NCBI database to use.
verbose	Shall the program give extensive feedback.
createLog	Create log files, needed for continuing a crashed program.

Details

This function sends fasta sequences to the NCBI blast service. The defaults for the delays are required by NCBI and must not be smaller than the default values. Also, NCBI asks the user to provide an email address.

The input seq can be a vector of strings. In that case the sequences are one after another processed. The option n_blast sets then the upper threshold of how many blast requests are send to the NCBI Blast service at a time and kept running there parallel. It is here in the users obligation not to misuse the service with too many parallel requests.

The xmlFolder parameter specifies the folder to where the XML results will be stored. In case the folder does not exist, R will create it.

For larger projects this option is advisable, as large projects can easily flood the memory.

In case the option keepInMemory is set to TRUE the Blast results will be kept in memory, otherwise they will be just written to the HDD, given the xmlFolder. Especially if many sequences are send to NCBI it is recommended not to keep the result in the memory.

If log files should be written (`createLog=TRUE`) a log path should be given in `logPath`. However, if a xmlPath is given and the option `createLog=TRUE` is set, then the log folder will be automatically created in the parental folder of the `xmlFolder` and is called `hoardeRlogs`. Setting the option `createLog=TRUE` is required to continue crashed blast runs.

Value

An xml file that contains the the NCBI result.

Author(s)

Daniel Fischer

Examples

```
## Not run:
blastSeq("ACGTGCATCGACTAGCTACGACTACGACTATC")

## End(Not run)
```

`getAssemblies` *Extracting Assemblies.*

Description

This function extracts the assemblies from an xml object.

Usage

```
getAssemblies(xml)
```

Arguments

`xml`	An xml object.

Details

This function extracts the information from an imported xml object.

Value

A matrix.

Author(s)

Daniel Fischer

Examples

```
## Not run:
getAssemblies(xml)

## End(Not run)
```

`getEnsgInfo` *Retrieve Gene Information From The NCBI Database.*

Description

This function retrieves for a given Ensemble Number the corresponding information from the NCBI database.

Usage

```
getEnsgInfo(ensg)
```

Arguments

`ensg`	Ensemble ID.

Details

This function retrieves for a given Ensemble Number the corresponding information from the NCBI database. The object `ensg` can also be a vector of Ensemble IDs.

Value

A matrix with information.

Author(s)

Daniel Fischer

Examples

```
## Not run:
ensg <- c("ENSG00000174482", "ENSG00000113494")
getEnsgInfo(ensg)

## End(Not run)
```

getGeneLocation *Extracting Gene Locations.*

Description

This function extracts the gene locations from an imported gtf or gff3 file.

Usage

```
getGeneLocation(gtf)
```

Arguments

gtf	An imported gtf object.

Details

This function extracts the information from an imported gtf object.

Value

A matrix.

Author(s)

Daniel Fischer

Examples

```
## Not run:
getGeneLocation(gtf)

## End(Not run)
```

getGeneSeq *Extracting a gene sequence from NCBI database.*

Description

This function retrieves a gene sequence from the NCBI database.

Usage

```
getGeneSeq(chr, start, end, organism)
```

Arguments

chr	Chromosome number, numeric/string
start	Start position, numeric
end	End position, numeric
organism	Name of the organism, string

Details

Extracting a gene sequence from NCBI database.

Value

A string that contains the genomic sequence.

Author(s)

Daniel Fischer

Examples

```
## Not run:
# Extracting for Sus Scrofa, build version 3:
getGeneSeq(1,2134,14532,"susScr3")

## End(Not run)
```

---

importBlastTab *Import a Tab Delimited Blast Output File*

Description

This function imports a tab delimited blast output.

Usage

```
importBlastTab(file)
```

Arguments

file	File name of the file.

Details

This function imports a tab delimited blast output file, currently the same as `read.table`

Value

A data frame containing the columns of the file.

Author(s)

Daniel Fischer

`importFA` *Importing a Fasta File.*

Description

This function imports a standard fasta file.

Usage

```
importFA(file)
```

Arguments

`file`	Specifies the filename/path.

Details

This function imports a standard fasta file. It assumes that label and sequence lines are alternating, meaning in the odd lines is the sequence name given, starting with > and in the even rows are the corresponding sequences.

Value

An object of class `fa` containing the sequences. The names coorespond to the sequence names given in the fasta file.

Author(s)

Daniel Fischer

See Also

`print.fa`, `summary.fa`

Examples

```
## Not run:
importFA(file="myFasta.fa")

## End(Not run)
```

importGFF3 *Import a GFF3 File*

Description

This function imports a gff3 file.

Usage

```
importGFF3(gff)
```

Arguments

gff	File name of the gff3 file

Details

This function imports a gff file and splits the last column which is usually tricky to handle as the order of the variables is not always the same.

Value

A data frame containing the columns of the gtf file, including the splitted last column.

Author(s)

Daniel Fischer

importGTF *Import a GTF File*

Description

This function imports a gtf file.

Usage

```
importGTF(gtf, skip = 0, nrow = -1)
```

Arguments

gtf	File name of the gtf file.
skip	Rows to skip from the top.
nrow	Total amount of rows read.

Details

This function imports a gtf file and splits the column 9 which is usually tricky to handle as the order of the variables is not always the same.

Value

A data frame containing the columns of the gtf file, including the splitted last column.

Author(s)

Daniel Fischer

`importPedMap` *Import a ped/map File Pair*

Description

This function imports a ped/map file pair.

Usage

```
importPedMap(ped, map=NULL, pedSep="\t", pedHeader=FALSE, genoSep=" ",
             mapSep="\t", mapHeader=FALSE, na.value="0")
```

Arguments

`ped`	File name of the ped file.
`map`	File name of the map file, optional, see details.
`pedSep`	Column separator in the ped file.
`pedHeader`	Logical, ped file contains header.
`genoSep`	Separator for Genotype, see details.
`mapSep`	Column separator in the map file.
`mapHeader`	Logical, map file contains header.
`na.value`	Character, encoding of missing values.

Details

This function imports a ped/map file pair. For that it is sufficient to provide the file name of the ped file, if the map file has the same name, but just the .map ending (e.g. myFile.ped and myFile.map). Also, the file suffix .ped can be ommited.

The `genoSep` option provides the separator between the Alleles within one Genotype, e.g 'A A' (`genoSep=" "`) or 'A/A' (`genoSep="/"`).

Value

A list of type pedMap with the three list items:

map	Matrix with the Genotype Map information.
fam	Matrix with the family information.
geno	Matrix with the genotype information.

Author(s)

Daniel Fischer

---

importXML *Import a XML File*

Description

This function imports a xml file produced from blastSeq.

Usage

```
importXML(fa, folder, idTH = 0.8, verbose=TRUE)
```

Arguments

fa	Sequence names.
folder	Folder , where the xml files are stored.
idTH	Identity threshold, see details.
verbose	Logical, function give status messages.

Details

This function imports a xml files produced from the blastSeq function. The idTh options sets the limit, what the minimum id threshold is until a hit will be taken into the result data frame.

Value

A data frame containing the results.

Author(s)

Daniel Fischer

print.ensgInfo *Print an ensgInfo Object*

Description

Prints an ensgInfo object.

Usage

```
 ## S3 method for class 'ensgInfo'
print(x, full=FALSE, ...)
```

Arguments

x	Object of class ensgInfo.
full	Logical, shall the full information be plotted.
...	Additional arguments.

Details

The print function displays an ensgInfo object. By default just the Ensembl ID and the corresponding gene name is plotted. Setting the option full=TRUE provides further information.

Author(s)

Daniel Fischer

print.fa *Print an fa Object*

Description

Prints an fa object.

Usage

```
 ## S3 method for class 'fa'
print(x, n=2, seq.out=50, ...)
```

Arguments

x	Object of class fa.
n	Amount of elements to be displayed, numeric.
seq.out	Length of each element to be displayed, numeric..
...	Additional parameters.

Details

The print function displays an `fa` object. By default just the first two elements with their first 50 bases are displayed. To display the full sequence, set `seq.out=NULL`.

Author(s)

Daniel Fischer

`print.pedMap` *Print an pedMap Object*

Description

Prints an `pedMap` object.

Usage

```
 ## S3 method for class 'pedMap'
print(x, nrow=5, ncol=10, ...)
```

Arguments

`x`	Object of class `pedMap`.
`nrow`	Amount of rows to be displayed, numeric.
`ncol`	Amount of cols to be displayed, numeric.
`...`	Additional parameters.

Details

The print function displays an `pedMap` object. By default just the first 5 rows of each list item and the first 10 columns of the `geno` matrix are displayed.

Author(s)

Daniel Fischer

print.xmlImport *Print an xmlImport Object*

Description

Prints an xmlImport object.

Usage

```
 ## S3 method for class 'xmlImport'
print(x, n=2, ...)
```

Arguments

x	Object of class xmlImport.
n	Amount of elements to be displayed, numeric.
...	Additional parameters.

Details

The print function displays an xmlImport object. By default just the first two elements are displayed.

Author(s)

Daniel Fischer

species *Species Name for Blast Search*

Description

Standardized species names for blast search.

Usage

```
species
```

Format

This vector contains 134 species names.

Note

Note, the names have been extracted on 1.6.2014 from the NCBI server.

speciesFigure *Showing Quantities of Different Species.*

Description

This function visualizes the different quantities of blast machtes.

Usage

```
speciesFigure(xml, species=NULL, type="chr", n=2:11, plot=TRUE)
```

Arguments

xml	An xml object.
species	A vector with species names.
type	The type of plot.
n	Ranks to be plotted
plot	Logical, shall the plot be plotted

Details

This function plots the frequency barplot of the blast results, divided for each species. The species of interest can be provided with the species object.

Value

A figure.

Author(s)

Daniel Fischer

Examples

```
## Not run:
  speciesFigure(xml, species=NULL, type="chr", n=2:11, plot=TRUE)

## End(Not run)
```

subDose *Rewrite the Dose File from a Beagle Output*

Description

This function takes a Dose Beagle output and rewrites the output.

Usage

```
subDose(file=NULL, vmmk=NULL, out=NULL, removeInsertions=TRUE, verbose=TRUE)
```

Arguments

file	Location of the original Beagle file (String).
vmmk	Location of the Variant Map Master key (String).
out	Name and location of the output file (String).
verbose	The function gives feedback.
removeInsertions	All Indels will be removed..

Details

This function takes a Beagle Dose file and rewrites the alleles from numerical to character, based on the information provided in a variant map master key.

Value

A rewritten beagle phased file.

Author(s)

Daniel Fischer

subGprobs *Rewrite the Gprobs File from a Beagle Output*

Description

This function takes a Gprobs Beagle output and rewrites the output.

Usage

```
subGprobs(file=NULL, vmmk=NULL, out=NULL, chunkSize=100000, removeInsertions=TRUE,
          verbose = TRUE, writeOut=TRUE)
```

Arguments

file	Location of the original Beagle file (String).
vmmk	Location of the Variant Map Master key (String).
out	Name and location of the output file (String).
chunkSize	For large Beagle files, the chunk size.
removeInsertions	All Indels will be removed.
verbose	The function gives feedback.
writeOut	Logical, write the output back to the HDD.

Details

This function takes a Beagle Gprobs file and rewrites the alleles from numerical to character, based on the information provided in a variant map master key. For larger files the function can process the rewriting in chunks in order to save memory.

Value

A rewritten beagle Gprobs file.

Author(s)

Daniel Fischer

---

subPhased *Rewrite the Phased File from a Beagle Output*

Description

This function takes a phased Beagle output and rewrites the output.

Usage

```
subPhased(file=NULL, vmmk = NULL, out=NULL, chunkSize=100000, verbose=TRUE,
          removeInsertions=TRUE)
```

Arguments

file	Location of the original Beagle file (String).
vmmk	Location of the Variant Map Master key (String).
out	Name and location of the output file (String).
chunkSize	For large Beagle files, the chunk size.
verbose	The function gives feedback.
removeInsertions	All Indels will be removed.

Details

This function takes a Beagle phased file and rewrites the alleles from numerical to character, based on the information provided in a variant map master key. For larger files the function can process the rewriting in chunks in order to save memory.

Value

A rewritten beagle phased file.

Author(s)

Daniel Fischer

`summary.ensgInfo` *Summarize an ensgInfo Object*

Description

Summarizes and prints an `ensgInfo` object in an informative way.

Usage

```
 ## S3 method for class 'ensgInfo'
summary(object, ...)
```

Arguments

`object`	Object of class `ensgInfo`.
`...`	Additional parameters.

Details

Summary for a `ensgInfo` object, providing the amount of different gene types in the query.

Author(s)

Daniel Fischer

summary.fa *Summarize an fa Object*

Description

Summarizes and prints an fa object in an informative way.

Usage

```
 ## S3 method for class 'fa'
summary(object, ...)
```

Arguments

object	Object of class fa.
...	Additional parameters.

Details

Summary for a fa object, providing the amount of sequences, the minimum and maximum length as well as the average length.

Author(s)

Daniel Fischer

summary.pedMap *Summarize an pedMap Object*

Description

Summarizes an pedMap object in an informative way.

Usage

```
 ## S3 method for class 'pedMap'
summary(object, ...)
```

Arguments

object	Object of class pedMap.
...	Additional parameters.

Details

Summary for a pedMap object, providing the dimensions of the map, the fam and the geno matrix as well as the total amount of Allele A/A, A/B, B/B as well as the amount of missing data and the monomorhic locations.

Author(s)

Daniel Fischer

Index