Package ‘GenomicTools.fileHandler’

March 5, 2020

Type Package
Title File Handlers for Genomic Data Analysis
Version 0.1.5.9
Date 2020-03-05
Author Daniel Fischer
Maintainer Daniel Fischer <daniel.fischer@luke.fi>
Description A collection of I/O tools for handling the most commonly used genomic datafiles, like fasta/-q, bed, gff, gtf, ped/map and vcf.
Depends R (>= 3.3), data.table (>= 1.9.6)
Imports snpStats
Suggests knitr, rmarkdown
VignetteBuilder knitr
License GPL (>= 2)
Encoding UTF-8
LazyLoad yes
RoxygenNote 6.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2020-03-05 08:00:07 UTC

R topics documented:

GenomicTools.fileHandler-package ........................................ 2
example.bed .................................................................. 3
example.fasta ................................................................. 3
example.fastq ................................................................. 4
example.gff ................................................................. 4
example.gtf ................................................................. 5
example.ped ................................................................. 5
example.vcf ................................................. 6
example2.gtf.gz ........................................... 6
exportBed ................................................ 7
exportFA .................................................. 8
importBed ............................................... 9
importBlastTab ......................................... 10
importFA ............................................... 10
importFeatureCounts ................................. 11
importFQ .............................................. 12
importGFF .............................................. 13
importGFF3 ............................................ 14
importGTF .............................................. 15
importPED ............................................. 16
importSTARLog ........................................ 17
importVCF .............................................. 18
importXML .............................................. 19
plotTotalReads ......................................... 20
plotUniquelyMappedReads ............................ 20
prereadGTF ............................................. 21
print.bed ............................................. 22
print.fa ................................................ 22
print.featureCounts ................................. 23
print.fq ............................................... 23
print.gtf ............................................. 24
print.pedMap ......................................... 25
print.vcf ............................................. 25
summary.bed .......................................... 26
summary.fa ........................................... 26
summary.featureCounts .............................. 27
summary.fq ........................................... 28
summary.gtf .......................................... 28
summary.STARLog ..................................... 29

Index .......................... 30

GenomicTools.fileHandler-package

R Package To Handle Files From Genomic Data GenomicTools.fileHandler is a loose collection of I/O Functions Needed in Genomic Data Analysis

Description

Package: GenomicTools.fileHandler
Type: Package
Version: 0.1.5.9
Date: 2020-03-05
example.bed

License: GPL
LazyLoad: yes

Author(s)
Daniel Fischer
Maintainer: Daniel Fischer <daniel.fischer@luke.fi>

Example Gene Annotation in Bed-Format

Description
This file contains some example lines to represent a typical bed file that can be used to try the corresponding functions.

Format
A file with three column Chr, Start and End.

Details
The file is located in the /extdata folder of the package and is accessible after installation via
system.file("extdata","example.bed",package="GenomicTools.fileHandler")

example.fasta

Example Sequencing Reads in fasta-Format

Description
This file contains some example reads to represent a typical fasta file that can be used to try the corresponding functions.

Details
The file is located in the /extdata folder of the package and is accessible after installation via
system.file("extdata","example.fasta",package="GenomicTools.fileHandler")

Author(s)
Daniel Fischer
example.fastq

Example Sequencing Reads in fastq-Format

Description

This file contains some example reads to represent a typical fastq file that can be used to try the corresponding functions.

Details

The file is located in the /extdata folder of the package and is accessible after installation via

```r
system.file("extdata","example.fastq",package="GenomicTools.fileHandler")
```

Author(s)

Daniel Fischer

example.gff

Example Gene Annotation in gff-Format

Description

This file contains some example gene annotations to represent a typical gff file that can be used to try the corresponding functions.

Details

The file is located in the /extdata folder of the package and is accessible after installation via

```r
system.file("extdata","example.gff",package="GenomicTools.fileHandler")
```

Author(s)

Daniel Fischer
**Description**

This file contains some example gene annotations to represent a typical gtf file that can be used to try the corresponding functions.

**Details**

The file is located in the /extdata folder of the package and is accessible after installation via:

```r
system.file("extdata","example.gtf",package="GenomicTools.fileHandler")
```

**Author(s)**

Daniel Fischer

---

**Description**

This file contains some example variants to represent a typical ped/map file pair that can be used to try the corresponding functions.

**Details**

The file is located in the /extdata folder of the package and is accessible after installation via:

```r
system.file("extdata","example.ped",package="GenomicTools.fileHandler")
```

**Author(s)**

Daniel Fischer
example.vcf  Example Variant data in vcf-Format

Description

This file contains some example variants to represent a typical vcf file that can be used to try the
corresponding functions.

Details

The file is located in the /extdata folder of the package and is accessible after installation via
system.file("extdata","example.vcf",package="GenomicTools.fileHandler")

Author(s)

Daniel Fischer

dexample2.gtf.gz  Example Gene Annotation in zipped gtf-Format

Description

This file contains some example gene annotations to represent a typical zipped gtf file that can be
used to try the corresponding functions.

Details

The file is located in the /extdata folder of the package and is accessible after installation via
system.file("extdata","example2.gtf.gz",package="GenomicTools.fileHandler")

Author(s)

Daniel Fischer
exportBed

Exporting a Bed File.

Description
This function exports a standard bed file.

Usage
exportBed(x, file = NULL, header = FALSE)

Arguments
x  data.frame
file  Character, specifies filename/path
header  Logical, shall a header be written

Details
This function exports a data.frame to a standard bed file. If no file name is given, the variable name will be used instead.

Value
A bed file

Author(s)
Daniel Fischer

Examples

```r
novelBed <- data.frame(Chr=c(11,18,3),
    Start=c(72554673, 62550696, 18148822),
    End=c(72555273, 62551296, 18149422),
    Gene=c("LOC1", "LOC2", "LOC3"))

# Create a temporary file to where the output of the function is stored
myfile <- file.path(tempdir(), "myLocs.bed")

exportBed(novelBed, file=myfile)
exportBed(novelBed, file=myfile, header=TRUE)
```
exportFA  

Exporting a Fasta File.

Description

This function exports a standard fasta file.

Usage

exportFA(fa, file = NULL)

Arguments

fa  
    fasta object

file  
    Character, specifies filename/path

Details

This function exports a fasta object to a standard fasta file. If no file name is given, the variable
name will be used instead.

Value

A fasta file

Author(s)

Daniel Fischer

Examples

# Define here the location on HDD for the example file
fpath <- system.file("extdata","example.fasta", package="GenomicTools.fileHandler")
# Import the example fasta file
fastaFile <- importFA(file=fpath)
newFasta <- fastaFile[1:5]

myfile <- file.path(tempdir(), "myLocs.fa")

exportFA(newFasta, file=myfile)
importBed

Importing a Bed File.

Description

This function imports a standard bed file

Usage

importBed(file, header = FALSE, sep = "\t")

Arguments

file Specifies the filename/path
header Logical, is a header present
sep Column separator

Details

This function imports a standard bed-file into a data.frame. It is basically a convenience wrapper around read.table. However, if no header lines is given, this function automatically assigns the column names, as they are given in the bed-specification on the Ensembl page here: https://www.ensembl.org/info/website/upload/bed.html

Value

A data.frame

Author(s)

Daniel Fischer

See Also

[exportBed], [read.table]

Examples

# Define here the location on HDD for the example file
fpath <- system.file("extdata","example.bed", package="GenomicTools.fileHandler")
# Import the example bed file
bedFile <- importBed(file=fpath)
importBlastTab  
Import a Tab Delimited Blast Output File

**Description**

This function imports a tab delimited blast output.

**Usage**

```r
importBlastTab(file)
```

**Arguments**

- `file`  
  Filename

**Details**

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

**Value**

A data.frame

**Author(s)**

Daniel Fischer

---

importFA  
Importing a Fasta File.

**Description**

This function imports a standard fasta file

**Usage**

```r
importFA(file)
```

**Arguments**

- `file`  
  Specifies the filename/path
importFeatureCounts

Details
This function imports a standard fasta file. Hereby, it does not matter if the identifier and sequence are alternating or not, as the rows starting with ‘>’ are used as identifier.
The example file was downloaded from here and was then further truncated respective transformed to fasta format:
ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/HG00096/sequence_read/

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

Author(s)
Daniel Fischer

See Also
print.fa, summary.fa

Examples

```r
# Define here the location on HDD for the example file
fpath <- system.file("extdata", "example.fasta", package="GenomicTools.fileHandler")
# Import the example fasta file
fastaFile <- importFA(file=fpath)
```

---

importFeatureCounts Import from FeatureCounts

Description
This function imports the output from FeatureCounts

Usage

```r
importFeatureCounts(file, skip = 0, headerLine = 2)
```

Arguments

- `file` Character, file name
- `skip` Number of lines to skip from txt file
- `headerLine` Linenumber that contains the header information
importFQ

Details
FeatureCounts produces two files, the txt that contain the expression values and then the summary that contains all the information about the mapping statistics. This function imports both and stores them in a corresponding list.

Value
A list with expValues, geneInfo and summary

Author(s)
Daniel Fischer

Examples

# Define here the location on HDD for the example file
fpath <- system.file("extdata","featureCountsExample.txt", package="GenomicTools.fileHandler")
# Import the example featureCounts file
fcFile <- importFeatureCounts(file=fpath)

importFQ

Importing a Fastq File.

Description
This function imports a standard fastq file

Usage
importFQ(file)

Arguments
file Specifies the filename/path

Details
This function imports a standard fastq file that consists out of blocks of four lines per entry

Value
An object of class fq containing the sequences and the quality measure. The names correspond to the sequence names given in the fasta file.

Author(s)
Daniel Fischer
importGFF

See Also

print.fq, summary.fq

Examples

# Define here the location on HDD for the example file
fpath <- system.file("extdata","example.fastq", package="GenomicTools.fileHandler")
# Import the example fastq file
fastqFile <- importFQ(file=fpath)

importGFF

Description

Import a GFF file

Usage

importGFF(file, skip = "auto", nrow = -1, use.data.table = TRUE,
level = "gene", features = NULL, num.features = c("FPKM", "TPM"),
print.features = FALSE, merge.feature = NULL, merge.all = TRUE,
class.names = NULL, verbose = TRUE)

Arguments

file file or folder
skip numeric, lines to skip
nrow numeric, lines to read
use.data.table logical
level Character, read level, default: "gene"
features features to import
num.features names of the numeric features
print.features Logical, print available features
merge.feature Character, merge multiple samples to dataset
merge.all Logical, shall all samples be merged together
class.names Definition of class name sin V9
verbose Logical, verbose function output

Details

This function imports a standard gff file.
importGFF3

Value
A gff object

Author(s)
Daniel Fischer

Examples

# Define here the location on HDD for the example file
fpath <- system.file("extdata","example.gff", package="GenomicTools.fileHandler")
# Import the example gff file
importGFF(fpath)

Description
Import a GFF3 file

Usage
importGFF3(gff, chromosomes)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gff</td>
<td>file or folder</td>
</tr>
<tr>
<td>chromosomes</td>
<td>The chromosome to import</td>
</tr>
</tbody>
</table>

Details
This function imports a standard gff3 file.

Value
A gff object

Author(s)
Daniel Fischer
importGTF

Import a GTF File

Description

This function imports a gtf file.

Usage

importGTF(file, skip = "auto", nrow = -1, use.data.table = TRUE,
level = "gene", features = NULL, num.features = c("FPKM", "TPM"),
print.features = FALSE, merge.feature = NULL, merge.all = TRUE,
class.names = NULL, verbose = TRUE)

Arguments

file file or folder
skip numeric, lines to skip
nrow numeric, lines to read
use.data.table logical
level Character, read level, default: "gene"
features features to import
num.features names of the numeric features
print.features Logical, print available features
merge.feature Character, merge multiple samples to dataset
merge.all Logical, shall all samples be merged
class.names Vector with class names
verbose Logical, verbose function output

Details

This function imports a gtf file. The features names to be imported are defined in features, several features are then provided as vector. A list of available feature can be printed, by setting print.features=TRUE.

The skip option allows to skip a given number of rows, the default is, however, auto. In that case, all rows that start with the # symbol are skipped.

In case a set of expression values given in gtf format should be imported and to be merged into a single data table, the feature that should be used for merging can be provided to the merge.feature option. In that case the function expects a folder in file and it will import all gtf located in that folder and merges them according to the merge.feature option. With the option class.names a vector of prefixes for the merged features can be provided. If this is kept empty, then the filenames of the gtf will be used instead (without gtf extension).

By default the function imprints all features in column 9 as string character. However, for common labels (FPKM and TPM) the class type is set automatically to numeric. Additional numerical feature names can be defined with the num.feature option.
importPED

Value

A gtf object

Author(s)

Daniel Fischer

Examples

```r
# Define here the location on HDD for the example file
fpath <- system.file("extdata","example.gtf", package="GenomicTools.fileHandler")
# Same file, but this time as gzipped version
fpath.gz <- system.file("extdata","example2.gtf.gz", package="GenomicTools.fileHandler")

# Import the example gtf file
importGTF(fpath, level="transcript", features=c("gene_id","FPKM"))

## Not run:
# For the current you need to have zcat installed (should be standard on a Linux system)
importGTF(fpath.gz, level="transcript", features=c("gene_id","FPKM"))

## End(Not run)
```

importPED

**Description**

Import a PED/MAP file pair

**Usage**

```r
importPED(file, n, snps = NULL, which, split = "\t| +", sep = ".",
na.strings = "0", lex.order = FALSE, verbose = TRUE)
```

**Arguments**

- `file`  ped filename
- `n`     Number of samples to read
- `snps`  map filename
- `which` Names of SNPS to import
- `split` Columns separator in ped file
- `sep`   Character that separates Alleles
- `na.strings` Definition for missing values
- `lex.order` Logical, lexicographical order
- `verbose` Logical, verbose output
Details

This function is to a large extend taken from snpStat::read.pedmap, but here is internally the data.table::fread function used that resulted in much faster file processing.

To import the data, the ped file can be provided to the file option and the map file to the snps option. If no option is given to snps and the file option is provided without any file extension, then the ped/map extension are automatically added.

Value

a pedmap object

Author(s)

Daniel Fischer

Examples

# Define here the location on HDD for the example file
pedPath <- system.file("extdata","example.ped", package="GenomicTools.fileHandler")
mapPath <- system.file("extdata","example.map", package="GenomicTools.fileHandler")
# Import the example ped/map files
importPED(file=pedPath, snps=mapPath)
importVCF

Details
This function imports the Log file from STAR

Value
a data frame

Author(s)
Daniel Fischer

importVCF

Description
Import a VCF function

Usage
importVCF(file, na.seq = "./.")

Arguments
file The file name
na.seq The missing value definition

Details
This function imports a VCF file.
In case the logical flag 'phased' is set to TRUE then the genotypes are expected to be in the format 0/0, otherwise they are expected to be like 0/1.
The example file was downloaded from here:

Value
A vcf object

Author(s)
Daniel Fischer
importXML

Examples

# Define here the location on HDD for the example file
fpath <- system.file("extdata","example.vcf", package="GenomicTools.fileHandler")
# Import the example vcf file
importVCF(fpath)

Description

Import an Blast XML file

Usage

importXML(folder, seqNames = NULL, which = NULL, idTH = 0.8,
verbose = TRUE)

Arguments

folder            Character, folder path
seqNames          Names of sequences
which             Which sequences to import
idTH              Use the threshold as cut-off
verbose           Logical, verbose output

Details

This function imports XML files as provided as Blast output, it is mainly aimed to import the output of the hoardeR package

Value

An XML object

Author(s)

Daniel Fischer
plotTotalReads

Description
Plot the total reads

Usage
plotTotalReads(STARLog)

Arguments
STARLog A STARLog object

Details
This function plots the total reads from a STARlog object
Part of the diagnostic plot series for of the STARLog. The function accepts also a list of STARLogs and creates then comparative boxplots

Value
A plot

Author(s)
Daniel Fischer

plotUniquelyMappedReads

Description
Plot the uniquely mapped reads

Usage
plotUniquelyMappedReads(STARLog)

Arguments
STARLog A STARLog object
prereadGTF

Details
This function plots the percentage of uniquely reads from a STARlog object
Part of the diagnostic plot series for the STARLog. The function accepts also a list of STARLogs
and creates then comparative boxplots

Value
A plot

Author(s)
Daniel Fischer

Description
Preread a gtf file and prints features of it for importing it.

Usage
prereadGTF(file, nrow = 1000, skip = "auto")

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Filename</td>
</tr>
<tr>
<td>nrow</td>
<td>Number of rows to read</td>
</tr>
<tr>
<td>skip</td>
<td>Rows to skip from top</td>
</tr>
</tbody>
</table>

Details
This function reads in a gtf file and prints its features for the import step.
By default this function only imports the first 1000 rows, in case all rows should be imported set
nrow=-1.
The number to skip in the beginning can be adjusted by the skip option. The default is here auto
so that the function can identify the correct amount of header rows. Hence, this option should be
changed only, if there is a good reason.

Value
A list of available features

Author(s)
Daniel Fischer
print.bed

Print a bed Object

Description
Prints a bed object.

Usage
```r
## S3 method for class 'bed'
print(x, n = 6, ...)
```

Arguments
- `x`: Object of class bed.
- `n`: Number of lines to print
- `...`: Additional parameters

Details
The print function displays a bed object

Author(s)
Daniel Fischer

print.fa

Print a fa Object

Description
Prints a fa object.

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

Arguments
- `x`: Object of class fa.
- `n`: Number of sequences to display
- `seq.out`: Length of the subsequence to display
- `...`: Additional parameters
print.featureCounts  

Details  
The print function displays a fa object  

Author(s)  
Daniel Fischer  

print.featureCounts  

Description  
Prints an featureCounts object.  

Usage  
```r  
## S3 method for class 'featureCounts'  
print(x, ...)  
```

Arguments  
- `x` Object of class featureCounts.  
- `...` Additional parameters  

Details  
The print function displays a featureCounts object  

Author(s)  
Daniel Fischer  

print.fq  

Description  
Prints a fq object.  

Usage  
```r  
## S3 method for class 'fq'  
print(x, n = 2, seq.out = 50, print.qual = TRUE, ...)  
```
## Arguments

- `x`  
  Object of class `fq`.

- `n`  
  Number of sequences to display

- `seq.out`  
  Length of the subsequence to display

- `print.qual`  
  Logical, shall the quality measures also be printed

- `...`  
  Additional parameters

## Details

The print function displays a fa object

## Author(s)

Daniel Fischer
print.pedMap  Print a pedMap Object

Description

Prints an pedMap object.

Usage

```r
## S3 method for class 'pedMap'
print(x, n = 6, m = 6, ...)  
```

Arguments

- `x`: Object of class pedMap.
- `n`: Number of samples to display
- `m`: Number of columns to display
- `...`: Additional parameters

Details

The print function displays a pedMap object

Author(s)

Daniel Fischer

print.vcf  Print a vcf Object

Description

Prints an vcf object.

Usage

```r
## S3 method for class 'vcf'
print(x, n = 6, m = 6, fullHeader = FALSE, ...)  
```

Arguments

- `x`: Object of class vcf.
- `n`: Number of samples to display
- `m`: Number of columns to display
- `fullHeader`: Logical, shall the whole header be printed
- `...`: Additional parameters
Details

The print function displays a vcf object

Author(s)

Daniel Fischer

summary.bed  Summary of a bed Object

Description

Summarizes a bed object.

Usage

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

Arguments

object Object of class bed.
...	Additional parameters

Details

The summary function displays an informative summary of a bed object

Author(s)

Daniel Fischer

summary.fa  Summary of a fa Object

Description

Summarizes a fa object.

Usage

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

summarize.featureCounts

Arguments

- **object**: Object of class `fa`.

- **...**: Additional parameters

Details

The `summary` function displays an informative summary of a `fa` object.

Author(s)

Daniel Fischer

---

**summary.featureCounts**  *Summary of a featureCounts Object*

Description

Summarizes a `featureCounts` object.

Usage

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

Arguments

- **object**: Object of class `featureCounts`.

- **...**: Additional parameters

Details

The `summary` function displays an informative summary of a `featureCounts` object.

Author(s)

Daniel Fischer
## Summary of a fq Object

### Description
Summarizes a fq object.

### Usage
```r
## S3 method for class 'fq'
summary(object, ...)
```

### Arguments
- `object`: Object of class fq.
- `...`: Additional parameters

### Details
The summary function displays an informative summary of a fq object.

### Author(s)
Daniel Fischer

## Summary of a gtf Object

### Description
Summarizes a gtf object.

### Usage
```r
## S3 method for class 'gtf'
summary(object, ...)
```

### Arguments
- `object`: Object of class gtf.
- `...`: Additional parameters

### Details
The summary function displays an informative summary of a gtf object.

### Note
This documentation is part of the Bioconductor project.
Description

Summarizes a STARLog object.

Usage

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

Arguments

- `object` Object of class STARLog.
- `...` Additional parameters

Details

The summary function displays an informative summary of a STARLog object.

Author(s)

Daniel Fischer
Index

*Topic data
  example.bed, 3
  example.fasta, 3
  example.fastq, 4
  example.gff, 4
  example.gtf, 5
  example.ped, 5
  example.vcf, 6
  example2.gtf.gz, 6

*Topic methods
  print.bed, 22
  print.fa, 22
  print.featureCounts, 23
  print.fq, 23
  print.gtf, 24
  print.pedMap, 25
  print.vcf, 25
  summary.bed, 26
  summary.fa, 26
  summary.featureCounts, 27
  summary.fq, 28
  summary.gtf, 28
  summary.STARLog, 29

*Topic print
  print.bed, 22
  print.fa, 22
  print.featureCounts, 23
  print.fq, 23
  print.gtf, 24
  print.pedMap, 25
  print.vcf, 25

*Topic summary
  summary.bed, 26
  summary.fa, 26
  summary.featureCounts, 27
  summary.fq, 28
  summary.gtf, 28
  summary.STARLog, 29

example.bed, 3
duplicate
example.fasta, 3
duplicate
example.fastq, 4
duplicate
example.gff, 4
duplicate
example.gtf, 5
duplicate
example.ped, 5
duplicate
example.vcf, 6
duplicate
example2.gtf.gz, 6
exportBed, 7
exportFA, 8
GenomicTools.fileHandler-package, 2
importBed, 9
importBlastTab, 10
importFA, 10
importFeatureCounts, 11
importFQ, 12
importGFF, 13
importGFF3, 14
importGTF, 15
importPED, 16
importSTARLog, 17
importVCF, 18
importXML, 19
plotTotalReads, 20
plotUniquelyMappedReads, 20
prereadGTF, 21
print.bed, 22
print.fa, 22
print.featureCounts, 23
print.fq, 23
print.gtf, 24
print.pedMap, 25
print.vcf, 25
summary.bed, 26
summary.fa, 26
summary.featureCounts, 27
summary.fq, 28
summary.gtf, 28
summary.STARLog, 29

example.bed, 3
INDEX

summary.gtf, 28
summary.STARlog, 29