Package 'BSgenomeForge'

November 4, 2024

Title Forge your own BSgenome data package

Description A set of tools to forge BSgenome data packages. Supersedes the old seed-based tools from the BSgenome software package. This package allows the user to create a BSgenome data package in one function call, simplifying the old seed-based process.

biocViews Infrastructure, DataRepresentation, GenomeAssembly, Annotation, GenomeAnnotation, Sequencing, Alignment, DataImport, SequenceMatching

URL https://bioconductor.org/packages/BSgenomeForge

BugReports https://github.com/Bioconductor/BSgenomeForge/issues

Version 1.7.0

License Artistic-2.0

Encoding UTF-8

Depends R (>= 4.3.0), methods, BiocGenerics, IRanges, GenomeInfoDb (>= 1.33.17), Biostrings, BSgenome

Imports utils, stats, Biobase, S4Vectors, GenomicRanges, BiocIO, rtracklayer

Suggests GenomicFeatures, Rsamtools, testthat, knitr, rmarkdown, BiocStyle, devtools, BSgenome.Celegans.UCSC.ce2

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/BSgenomeForge

git_branch devel

git_last_commit b850d52

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-11-04

Author Hervé Pagès [aut, cre], Atuhurira Kirabo Kakopo [aut], Emmanuel Chigozie Elendu [ctb], Prisca Chidimma Maduka [ctb]

Maintainer Hervé Pagès <hpages.on.github@gmail.com>

Contents

`BSgenomeForge-package` *The BSgenomeForge package*

Description

A package that simplifies the process of forging a BSgenome data package, by allowing the user to use one function to create the package.

Details

BSgenomeForge provides two major functions, the `forgeBSgenomeDataPkgFromNCBI` function and `forgeBSgenomeDataPkgFromUCSC` function which allow one to forge a BSgenome data package from a NCBI assembly or UCSC genome respectively.

For an overview of the functionality provided by the package, please see the vignette: `vignette("QuickBSgenomeForge", package="BSgenomeForge")`

Author(s)

Atuhurira Kirabo Kakopo, Hervé Pagès

Maintainer: Hervé Pagès

See Also

- The `forgeBSgenomeDataPkgFromNCBI` function for creating a BSgenome data package from a NCBI assembly.
- The `forgeBSgenomeDataPkgFromUCSC` function for creating a BSgenome data package from a UCSC genome.

Examples

```
## ---------------------------------------------------------------------
## EXAMPLE 1
## ---------------------------------------------------------------------

## Create a BSgenome data package for NCBI assembly GCF_000857545.1
## (organism Torque teno virus 1):
forgeBSgenomeDataPkgFromNCBI(assembly_accession="GCF_000857545.1",
                             pkg_maintainer="Jane Doe <janedoe@gmail.com>",
                             organism="Torque teno virus 1",
                             circ_seqs="NC_002076.2",
                             destdir=tempdir())

## ---------------------------------------------------------------------
## EXAMPLE 2
## ---------------------------------------------------------------------

## Create a BSgenome data package for UCSC genome wuhCor1 (SARS-CoV-2
## assembly, see https://genome.ucsc.edu/cgi-bin/hgGateway?db=wuhCor1):
forgeBSgenomeDataPkgFromUCSC(
    genome="wuhCor1",
    organism="Severe acute respiratory syndrome coronavirus 2",
    pkg_maintainer="Jane Doe <janedoe@gmail.com>",
    destdir=tempdir()
)
```

AdvancedBSgenomeForge *Advanced BSgenomeForge functions*

Description

A set of functions for making a BSgenome data package from any genome assembly not covered by forgeBSgenomeDataPkgFromNCBI() or forgeBSgenomeDataPkgFromUCSC().

Usage

```
## Main functions:

forgeBSgenomeDataPkg(x, seqs_srcdir=".", destdir=".", replace=FALSE, verbose=TRUE)

forgeMaskedBSgenomeDataPkg(x, masks_srcdir=".", destdir=".", verbose=TRUE)

## Low-level helpers:

forgeSeqlengthsRdsFile(seqnames, prefix="", suffix=".fa",
                       seqs_srcdir=".", seqs_destdir=".",
                       genome=NA_character_, verbose=TRUE)
```

```
forgeSeqlengthsRdaFile(seqnames, prefix="", suffix=".fa",
                       seqs_srcdir=".", seqs_destdir=".",
                       genome=NA_character_, verbose=TRUE)

forgeSeqFiles(provider, genome,
              seqnames, mseqnames=NULL,
              seqfile_name=NA, prefix="", suffix=".fa",
              seqs_srcdir=".", seqs_destdir=".",
              ondisk_seq_format=c("2bit", "rds", "rda", "fa.rz", "fa"),
              verbose=TRUE)

forgeMasksFiles(seqnames, nmask_per_seq,
                seqs_destdir=".",
                ondisk_seq_format=c("2bit", "rda", "fa.rz", "fa"),
                masks_srcdir=".", masks_destdir=".",
                AGAPSfiles_type="gap", AGAPSfiles_name=NA,
                AGAPSfiles_prefix="", AGAPSfiles_suffix="_gap.txt",
                RMfiles_name=NA, RMfiles_prefix="", RMfiles_suffix=".fa.out",
                TRFfiles_name=NA, TRFfiles_prefix="", TRFfiles_suffix=".bed",
                verbose=TRUE)
```

Arguments

x	For `forgeBSgenomeDataPkg()`: A BSgenomeDataPkgSeed object or the name of a BSgenome data package seed file. For `forgeMaskedBSgenomeDataPkg()`: A MaskedBSgenomeDataPkgSeed object or the name of a masked BSgenome data package seed file. See the "Advanced BSgenomeForge usage" vignette in this package for more information.
seqs_srcdir, masks_srcdir	Single strings indicating the path to the source directories i.e. to the directories containing the source data files. Only read access to these directories is needed. See the "Advanced BSgenomeForge usage" vignette in this package for more information.
destdir	A single string indicating the path to the directory where the source tree of the target package should be created. This directory must already exist. See the "Advanced BSgenomeForge usage" vignette in this package for more information.
replace	`TRUE` or `FALSE`. When set to TRUE, `replace` replaces the package directory if it already exists.
verbose	`TRUE` or `FALSE`.
provider	The provider of the *sequence data files* e.g. `"UCSC"`, `"NCBI"`, `"BDGP"`, `"FlyBase"`, etc...
genome	The name of the genome. Typically the name of an NCBI assembly (e.g. `"GRCh38.p12"`, `"WBcel235"`, `"TAIR10.1"`, `"ARS-UCD1.2"`, etc...) or UCSC genome (e.g. `"hg38"`, `"bosTau9"`, `"galGal6"`, `"ce11"`, etc...).

`seqnames`, `mseqnames`
: A character vector containing the names of the single (for `seqnames`) and multiple (for `mseqnames`) sequences to forge. See the "Advanced BSgenomeForge usage" vignette in this package for more information.

`seqfile_name`, `prefix`, `suffix`
: See the "Advanced BSgenomeForge usage" vignette in this package for more information, in particular the description of the `seqfile_name`, `seqfiles_prefix` and `seqfiles_suffix` fields of a BSgenome data package seed file.

`seqs_destdir`, `masks_destdir`
: During the forging process the source data files are converted into serialized Biostrings objects. `seqs_destdir` and `masks_destdir` must be single strings indicating the path to the directories where these serialized objects should be saved. These directories must already exist.

  Both `forgeSeqlengthsRdsFile` and `forgeSeqlengthsRdaFile` will produce a single `.rds` or `.rda` file. Both `forgeSeqFiles` and `forgeMasksFiles` will produce one file per sequence (all files being either `.rds` or `.rda` files).

`ondisk_seq_format`
: Specifies how the single sequences should be stored in the forged package. Can be `"2bit"`, `"rds"`, `"rda"`, `"fa.rz"`, or `"fa"`. If `"2bit"` (the default), then all the single sequences are stored in a single twoBit file. If `"rds"` or `"rda"`, then each single sequence is stored in a separated serialized XString derivative (one per single sequence). If `"fa.rz"` or `"fa"`, then all the single sequences are stored in a single FASTA file (compressed in the RAZip format if `"fa.rz"`).

`nmask_per_seq`
: A single integer indicating the desired number of masks per sequence. See the "Advanced BSgenomeForge usage" vignette in this package for more information.

`AGAPSfiles_type`, `AGAPSfiles_name`, `AGAPSfiles_prefix`, `AGAPSfiles_suffix`, `RMfiles_name`, `RMfiles_prefix`, `RMfiles_suffix`, `TRFfiles_name`, `TRFfiles_prefix`, `TRFfiles_suffix`
: These arguments are named accordingly to the corresponding fields of a BSgenome data package seed file. See the "Advanced BSgenomeForge usage" vignette in this package for more information.

Details

These functions are intended for Bioconductor users who want to make a new BSgenome data package, not for regular users of these packages. See the "Advanced BSgenomeForge usage" vignette in this package (`vignette("AdvancedBSgenomeForge")`) for an extensive coverage of this topic.

Author(s)

H. Pagès

See Also

- `forgeBSgenomeDataPkgFromNCBI` and `forgeBSgenomeDataPkgFromUCSC` in the **BSgenomeForge** package.
- `available.genomes` to find BSgenome data packages available in Bioconductor.
- BSgenome objects.

Examples

```
seqs_srcdir <- system.file("extdata", package="BSgenome")
seqnames <- c("chrX", "chrM")

## Forge .2bit sequence files:
forgeSeqFiles("UCSC", "ce2",
              seqnames, prefix="ce2", suffix=".fa.gz",
              seqs_srcdir=seqs_srcdir,
              seqs_destdir=tempdir(), ondisk_seq_format="2bit")

## Forge .rds sequence files:
forgeSeqFiles("UCSC", "ce2",
              seqnames, prefix="ce2", suffix=".fa.gz",
              seqs_srcdir=seqs_srcdir,
              seqs_destdir=tempdir(), ondisk_seq_format="rds")

## Sanity checks:
library(BSgenome.Celegans.UCSC.ce2)
genome <- BSgenome.Celegans.UCSC.ce2

ce2_sequences <- import(file.path(tempdir(), "single_sequences.2bit"))
ce2_sequences0 <- DNAStringSet(list(chrX=genome$chrX, chrM=genome$chrM))
stopifnot(identical(names(ce2_sequences0), names(ce2_sequences)),
          all(ce2_sequences0 == ce2_sequences))

chrX <- readRDS(file.path(tempdir(), "chrX.rds"))
stopifnot(genome$chrX == chrX)
chrM <- readRDS(file.path(tempdir(), "chrM.rds"))
stopifnot(genome$chrM == chrM)
```

---

downloadGenomicSequencesFromNCBI

Download genomic sequences from NCBI

Description

A utility function to download the compressed FASTA file that contains the genomic sequences of a given NCBI assembly.

Usage

```
downloadGenomicSequencesFromNCBI(assembly_accession, assembly_name=NA,
                                 destdir=".", method, quiet=FALSE)
```

Arguments

`assembly_accession`

A single string containing a GenBank assembly accession (e.g. `"GCA_000001405.15"`) or a RefSeq assembly accession (e.g. `"GCF_000001405.26"`).

assembly_name	A single string or NA.
destdir	A single string containing the path to the directory where the compressed FASTA file is to be downloaded. This directory must already exist. Note that, by default, the file will be downloaded to the current directory ("..").
method, quiet	Passed to the internal call to download.file(). See ?download.file in the **utils** package for more information.

Details

This function is intended for Bioconductor users who want to download the compressed FASTA file from NCBI for a given assembly specified by the assembly_accession argument.

Value

The path to the downloaded file as an invisible string.

Author(s)

Prisca Chidimma Maduka

See Also

- The download.file function in the **utils** package that downloadGenomicSequencesFromNCBI uses internally to download the compressed FASTA file.
- The downloadGenomicSequencesFromUCSC function to download genomic sequences from UCSC.

Examples

```
## Download the compressed FASTA file for NCBI assembly ASM972954v1 (see
## https://www.ncbi.nlm.nih.gov/assembly/GCF_009729545.1/):
downloadGenomicSequencesFromNCBI("GCF_009729545.1")

## Use the 'destdir' argument to specify the directory where to
## download the file:
downloadGenomicSequencesFromNCBI("GCF_009729545.1", destdir=tempdir())

## Download and import the file in R as a DNAStringSet object:
filepath <- downloadGenomicSequencesFromNCBI("GCF_009729545.1",
                                             destdir=tempdir())
genomic_sequences <- readDNAStringSet(filepath)
genomic_sequences
```

`downloadGenomicSequencesFromUCSC` *Download genomic sequences from UCSC*

Description

A utility function to download the 2bit file that contains the genomic sequences of a given UCSC genome.

Usage

```
downloadGenomicSequencesFromUCSC(
            genome,
            goldenPath.url=getOption("UCSC.goldenPath.url"),
            destdir=".", method, quiet=FALSE)
```

Arguments

`genome`	This is the name of the UCSC genome sequence to be downloaded. It is used to form the download URL.
`goldenPath.url`	A string set to `getOption("UCSC.goldenPath.url")` by default. `getOption("UCSC.goldenPath.url"` returns the goldenPath URL, `http://hgdownload.cse.ucsc.edu/goldenPath`.
`destdir`	A single string containing the path to the directory where the 2bit file is to be downloaded. This directory must already exist. Note that, by default, the file will be downloaded to the current directory (`"."`).
`method, quiet`	Passed to the internal call to `download.file()`. See `?download.file` in the **utils** package for more information.

Details

This function is intended for Bioconductor users who want to download the 2bit genomic sequence file of a UCSC genome specified by the `genome` argument.

Value

The path to the downloaded file as an invisible string.

Author(s)

Emmanuel Chigozie Elendu (Simplecodez)

See Also

- The `download.file` function in the **utils** package that `downloadGenomicSequencesFromUCSC` uses internally to download the 2bit file.
- The `downloadGenomicSequencesFromNCBI` function to download genomic sequences from NCBI.

Examples

```
## Download the 2bit file for UCSC genome sacCer1:
downloadGenomicSequencesFromUCSC("sacCer1")

## Use the 'destdir' argument to specify the directory where to
## download the file:
downloadGenomicSequencesFromUCSC("sacCer1", destdir=tempdir())

## Download and import the file in R as a DNAStringSet object:
filepath <- downloadGenomicSequencesFromUCSC("sacCer1", destdir=tempdir())
genomic_sequences <- import(filepath)
genomic_sequences
```

`fastaTo2bit` *Convert files from FASTA to 2bit*

Description

`fastaTo2bit` is a utility function to convert a FASTA file to the 2bit format.

Usage

```
fastaTo2bit(origfile, destfile, assembly_accession=NA)
```

Arguments

`origfile`	A single string containing the path to the FASTA file (possibly compressed) to read, e.g. `"felCat9.fa"`, `"felCat9.fa.gz"`, or `"path/to/felCat9.fa.gz"`.
`destfile`	A single string containing the path to the 2bit file to be written, e.g. `"felCat9.2bit"` or `"path/to/felCat9.2bit"`.
`assembly_accession`	A single string containing a GenBank assembly accession (e.g. `"GCA_009729545.1"`) or a RefSeq assembly accession (e.g. `"GCF_009729545.1"`). When specified, this uses `getChromInfoFromNCBI` to get chromosome information for the NCBI assembly, which is matched against the corresponding information in the FASTA file, consequently reordering its sequences. The sequences are then renamed from their GenBank or RefSeq accession assembly names, to their corresponding sequence names. If missing, the function does not perform sequence reordering or renaming.

Details

This function is intended for Bioconductor users who want to convert a FASTA file to the 2bit format.

Value

An invisible NULL.

Author(s)

Atuhurira Kirabo Kakopo and Hervé Pagès

See Also

- The readDNAStringSet function in the **Biostrings** package that fastaTo2bit uses internally to import the FASTA file.
- The export.2bit function in the **rtracklayer** package that fastaTo2bit uses internally to export the 2bit file.
- The getChromInfoFromNCBI function in the **GenomeInfoDb** package that fastaTo2bit uses internally to get chromosome information for the specified NCBI assembly.
- The downloadGenomicSequencesFromNCBI function that downloads genomic sequences from NCBI.

Examples

```
## Most assemblies at NCBI can be accessed using either their GenBank
## or RefSeq assembly accession. For example assembly ASM972954v1 (for
## Acidianus infernus) can be accessed either with GCA_009729545.1
## (GenBank assembly accession) or GCF_009729545.1 (RefSeq assembly
## accession).
## See https://www.ncbi.nlm.nih.gov/assembly/GCA_009729545.1
## or https://www.ncbi.nlm.nih.gov/assembly/GCF_009729545.1 for
## the landing page of this assembly.

## ---------------------------------------------------------------------
## USING FASTA FILE FROM **GenBank** ASSEMBLY
## ---------------------------------------------------------------------

## Download the FASTA file containing the genomic sequences for
## the ASM972954v1 assembly to the tempdir() folder:
fasta_path <- downloadGenomicSequencesFromNCBI("GCA_009729545.1",
                                               destdir=tempdir())

## Use fastaTo2bit() to convert the file to 2bit. We're using the
## function in its simplest form here so there won't be any sequence
## renaming or reordering:
twobit_path1 <- tempfile(fileext=".2bit")
fastaTo2bit(fasta_path, twobit_path1)

## Take a look at the sequence names in the resulting 2bit file:
seqlevels(rtracklayer::TwoBitFile(twobit_path1))

## Note that seqlevels(rtracklayer::TwoBitFile(.)) is equivalent to
## names(rtracklayer::import.2bit(.)) but a lot more efficient because
## it doesn't load the sequences.

## Use fastaTo2bit() again to convert the file to 2bit. However
## this time we want the function to rename and reorder the
## sequences as in getChromInfoFromNCBI("GCA_009729545.1"), so
```

```
## we set 'assembly_accession' to "GCA_009729545.1" in the call
## to fastaTo2bit():
twobit_path2 <- tempfile(fileext=".2bit")
fastaTo2bit(fasta_path, twobit_path2, assembly_accession="GCA_009729545.1")

## Take a look at the sequence names in the resulting 2bit file:
seqlevels(rtracklayer::TwoBitFile(twobit_path2))

## ---------------------------------------------------------------------
## USING FASTA FILE FROM **RefSeq** ASSEMBLY
## ---------------------------------------------------------------------

## Same as above but using GCF_009729545.1 instead of GCA_009729545.1

fasta_path <- downloadGenomicSequencesFromNCBI("GCF_009729545.1",
                                               destdir=tempdir())

twobit_path1 <- tempfile(fileext=".2bit")
fastaTo2bit(fasta_path, twobit_path1)
seqlevels(rtracklayer::TwoBitFile(twobit_path1))

twobit_path2 <- tempfile(fileext=".2bit")
fastaTo2bit(fasta_path, twobit_path2, assembly_accession="GCF_009729545.1")
seqlevels(rtracklayer::TwoBitFile(twobit_path2))

## ---------------------------------------------------------------------
## USING A FASTA FILE WITH IUPAC AMBIGUITY LETTERS
## ---------------------------------------------------------------------

## Download the FASTA file containing the genomic sequences for
## for Escherichia coli assembly ASM1484v1 (see
## https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000014845.1/):
fasta_path <- downloadGenomicSequencesFromNCBI("GCF_000014845.1",
                                               destdir=tempdir())

## The DNA sequences in this file contain IUPAC ambiguity letters
## not supported by the 2bit format, so fastaTo2bit() will replace
## them with N's and issue a warning:
twobit_path <- tempfile(fileext=".2bit")
fastaTo2bit(fasta_path, twobit_path)  # warning!

## Use suppressWarnings() to suppress the warning:
suppressWarnings(fastaTo2bit(fasta_path, twobit_path))
```

`forgeBSgenomeDataPkgFRomNCBI`

Create a BSgenome data package from an NCBI assembly

Description

The `forgeBSgenomeDataPkgFRomNCBI` function allows the user to create a BSgenome data package from an NCBI assembly.

Usage

```
forgeBSgenomeDataPkgFromNCBI(assembly_accession,
            pkg_maintainer, pkg_author=NA,
            pkg_version="1.0.0", pkg_license="Artistic-2.0",
            organism=NULL, circ_seqs=NULL, destdir=".")
```

Arguments

`assembly_accession`
: A single string containing a GenBank assembly accession (e.g. `"GCA_009729545.1"`) or a RefSeq assembly accession (e.g. `"GCF_000857545.1"`). Alternatively, if the assembly is registered in the **GenomeInfoDb** package (see `?GenomeInfoDb::registered_NCBI_ass` the assembly name (e.g. `"mLoxAfr1.hap2"`) can be supplied instead of its GenBank or RefSeq accession.

`pkg_maintainer`
: A single string containing the name and email address of the package maintainer (e.g `"Jane Doe, <janedoe@gmail.com>"`).

`pkg_author`
: A single string containing the name of the package author. When unspecified, this takes the value of `pkg_maintainer`.

`pkg_version`
: The version of the package. Set to `"1.0.0"` by default.

`pkg_license`
: The license of the package. This must be the name of a software license used for free and open-source packages. Set to `"Artistic-2.0"` by default.

`organism`
: The full name of the organism e.g. `"Homo sapiens"`, `"Felis catus"`, `"Loxodonta africana"`, `"Acidianus infernus"`, etc... Only needs to be specified if the assembly is *not* registered in the **GenomeInfoDb** package (see `?GenomeInfoDb::registered_NCBI_assem`

`circ_seqs`
: NULL (the default), or a character vector containing the names of the circular sequences in the assembly. This only needs to be specified if the assembly is *not* registered in the **GenomeInfoDb** package (if the assembly is registered then its circular sequences are known so there's no need to specify `circ_seqs`).

  Notes:

  - You can use `registered_NCBI_assemblies()` to get the list of assemblies that are registered in the **GenomeInfoDb** package.
  - Only assembled molecules can be circular. To see the list of assembled molecules for a given assembly, call `getChromInfoFromNCBI(assembly_accession, assembled.molecules.only=TRUE)$SequenceName`.
  - If the assembly is not registered and does not have circular sequences, then `circ_seqs` must be set to `character(0)`.

`destdir`
: A single string containing the path to the directory where the BSgenome data package is to be created. This directory must already exist. Note that, by default, the package will be created in the current directory (`"."`).

Details

This function is intended for Bioconductor users who want to forge a BSgenome data package from an NCBI assembly. It typically makes use of the `downloadGenomicSequencesFromNCBI` utility function to download the compressed FASTA file that contains the genomic sequences of the assembly, and stores it in the working directory. However, if the file already exists in the working directory, then it is used and not downloaded again.

Value

The path to the created package as an invisible string.

Author(s)

Atuhurira Kirabo Kakopo

See Also

- The `registered_NCBI_assemblies` and `getChromInfoFromNCBI` functions defined in the **GenomeInfoDb** package.
- The `downloadGenomicSequencesFromNCBI` function that `forgeBSgenomeDataPkgFromNCBI` uses internally to download the genomic sequences from NCBI.
- The `fastaTo2bit` function that `forgeBSgenomeDataPkgFromNCBI` uses internally to convert the file downloaded by `downloadGenomicSequencesFromNCBI` from FASTA to 2bit.
- The `forgeBSgenomeDataPkgFromUCSC` function for creating a BSgenome data package from a UCSC genome.

Examples

```
## ---------------------------------------------------------------------
## EXAMPLE 1
## ---------------------------------------------------------------------

## Create a BSgenome data package for NCBI assembly GCA_009729545.1
## (organism Acidianus infernus):
forgeBSgenomeDataPkgFromNCBI(assembly_accession="GCA_009729545.1",
                             pkg_maintainer="Jane Doe <janedoe@gmail.com>",
                             organism="Acidianus infernus",
                             destdir=tempdir())

## ---------------------------------------------------------------------
## EXAMPLE 2
## ---------------------------------------------------------------------

## Create a BSgenome data package for NCBI assembly GCF_000857545.1
## (organism Torque teno virus 1):
forgeBSgenomeDataPkgFromNCBI(assembly_accession="GCF_000857545.1",
                             pkg_maintainer="Jane Doe <janedoe@gmail.com>",
                             organism="Torque teno virus 1",
                             circ_seqs="NC_002076.2",
                             destdir=tempdir())
```

forgeBSgenomeDataPkgFromTwobitFile
Create a BSgenome data package from a 2bit file

Description

The forgeBSgenomeDataPkgFromTwobitFile function allows the user to create a BSgenome data package from a 2bit file.

Usage

```
forgeBSgenomeDataPkgFromTwobitFile(filepath, organism, provider, genome,
             pkg_maintainer, pkg_author=NA,
             pkg_version="1.0.0", pkg_license="Artistic-2.0",
             seqnames=NULL,
             circ_seqs=NULL,
             destdir=".")
```

Arguments

filepath	A single string containing a path to a 2bit file.
organism	The full name of the organism e.g. "Escherichia coli", "Mus musculus", "Saccharomyces cerevisiae", etc...
provider	The provider of the sequence data stored in the 2bit file e.g. "NCBI", "UCSC", "BDGP", "FlyBase", "WormBase", etc... Will be used to make part 3 of the package name.
genome	A single string specifying the name of the genome assembly e.g. "ASM1484v1", "mm39", or "sacCer3". Will be used to make part 4 of the package name.
pkg_maintainer	A single string containing the name and email address of the package maintainer (e.g "Jane Doe, <janedoe@gmail.com>").
pkg_author	A single string containing the name of the package author. When unspecified, this takes the value of pkg_maintainer.
pkg_version	The version of the package. Set to "1.0.0" by default.
pkg_license	The license of the package. This must be the name of a software license used for free and open-source packages. Set to "Artistic-2.0" by default.
seqnames	NULL (the default), or a character vector containing a subset of the sequence names stored in the 2bit file. Use this to select and/or reorder the sequences that will go in the BSgenome data package. By default (i.e. when seqnames is NULL), all the sequences in the file are selected and put in the BSgenome data package in the order that they appear in the 2bit file. Use seqlevels(rtracklayer::TwoBitFile(filepath)) to see the sequence names stored in the 2bit file.

`circ_seqs` NULL (the default), or a character vector providing the names of the sequences stored in the 2bit file that are known to be circular. Note that if `seqnames` is supplied then `circ_seqs` must be a subset of it.

Set to `circ_seqs` to `character(0)` if none of the sequences in the 2bit file (or in `seqnames`) is known to be circular.

By default (i.e. if `circ_seqs` is NULL), then the circular sequences are guessed based on their names. Because this is not really reliable, a warning is issued.

`destdir` A single string containing the path to the directory where the BSgenome data package is to be created. This directory must already exist. Note that, by default, the package will be created in the current directory (`"."`).

Details

This function is intended for Bioconductor users who want to forge a BSgenome data package from a 2bit file.

Value

The path to the created package as an invisible string.

Author(s)

Hervé Pagès

See Also

- The `forgeBSgenomeDataPkgFromNCBI` function for creating a BSgenome data package from an NCBI assembly (similar to `forgeBSgenomeDataPkgFromTwobitFile` but slightly more convenient).
- The `forgeBSgenomeDataPkgFromUCSC` function for creating a BSgenome data package from a UCSC genome (similar to `forgeBSgenomeDataPkgFromTwobitFile` but slightly more convenient).
- The `fastaTo2bit` function to convert a FASTA file to the 2bit format.

Examples

```
## Download the FASTA file containing the genomic sequences for
## for Escherichia coli assembly ASM1484v1 (see
## https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000014845.1/):
fasta_path <- downloadGenomicSequencesFromNCBI("GCF_000014845.1",
                                               destdir=tempdir())

## Use fastaTo2bit() to convert the file to 2bit:
twobit_path <- tempfile(fileext=".2bit")
fastaTo2bit(fasta_path, twobit_path)

## All the DNA sequences in Escherichia coli are circular:
circ_seqs <- seqlevels(rtracklayer::TwoBitFile(twobit_path))

## Note that seqlevels(rtracklayer::TwoBitFile(.)) is equivalent to
```

```
## names(rtracklayer::import.2bit(.)) but a lot more efficient because
## it doesn't load the sequences.

## Create a BSgenome data package from the 2bit file:
forgeBSgenomeDataPkgFromTwobitFile(
    filepath=twobit_path,
    organism="Escherichia coli",
    provider="NCBI",
    genome="ASM1484v1",
    pkg_maintainer="Jane Doe <janedoe@gmail.com>",
    circ_seqs=circ_seqs,
    destdir=tempdir()
)
```

`forgeBSgenomeDataPkgFromUCSC` *Create a BSgenome data package from a UCSC genome*

Description

The `forgeBSgenomeDataPkgFromUCSC` function allows the user to create a BSgenome data package from a UCSC genome.

Usage

```
forgeBSgenomeDataPkgFromUCSC(genome, organism,
             pkg_maintainer, pkg_author=NA,
             pkg_version="1.0.0", pkg_license="Artistic-2.0",
             circ_seqs=NULL,
             goldenPath.url=getOption("UCSC.goldenPath.url"),
             destdir=".")
```

Arguments

`genome`	A single string specifying the name of a UCSC genome (e.g. `"mm39"` or `"sacCer3"`).
`organism`	The full name of the organism e.g. `"Mus musculus"`, `"Saccharomyces cerevisiae"`, etc...
`pkg_maintainer`	A single string containing the name and email address of the package maintainer (e.g `"Jane Doe, <janedoe@gmail.com>"`).
`pkg_author`	A single string containing the name of the package author. When unspecified, this takes the value of `pkg_maintainer`.
`pkg_version`	The version of the package. Set to `"1.0.0"` by default.
`pkg_license`	The license of the package. This must be the name of a software license used for free and open-source packages. Set to `"Artistic-2.0"` by default.

`circ_seqs` NULL (the default), or a character vector containing the names of the circular sequences in the UCSC genome. This only needs to be specified if the genome is *not* registered in the **GenomeInfoDb** package (if the genome is registered then its circular sequences are known so there's no need to specify `circ_seqs`).

Notes:

- You can use `registered_UCSC_genomes()` to get the list of UCSC genomes that are registered in the **GenomeInfoDb** package.
- If the genome is not registered and does not have circular sequences, then `circ_seqs` must be set to `character(0)`.

`goldenPath.url` A single string specifying the URL to the UCSC goldenPath location where the genomic sequences and chromosome sizes are expected to be found.

`destdir` A single string containing the path to the directory where the BSgenome data package is to be created. This directory must already exist. Note that, by default, the package will be created in the current directory (`"."`).

Details

This function is intended for Bioconductor users who want to forge a BSgenome data package from a UCSC genome. It typically makes use of the `downloadGenomicSequencesFromUCSC` utility function to download the 2bit file that contains the genomic sequences of the genome, and stores it in the working directory. However, if the file already exists in the working directory, then it is used and not downloaded again.

Value

The path to the created package as an invisible string.

Author(s)

Hervé Pagès

See Also

- The `registered_UCSC_genomes` and `getChromInfoFromUCSC` functions defined in the **GenomeInfoDb** package.
- The `downloadGenomicSequencesFromUCSC` function that `forgeBSgenomeDataPkgFromUCSC` uses internally to download the genomic sequences from UCSC.
- The `forgeBSgenomeDataPkgFromNCBI` function for creating a BSgenome data package from an NCBI assembly.

Examples

```
## Create a BSgenome data package for UCSC genome wuhCor1 (SARS-CoV-2
## assembly, see https://genome.ucsc.edu/cgi-bin/hgGateway?db=wuhCor1):
forgeBSgenomeDataPkgFromUCSC(
    genome="wuhCor1",
    organism="Severe acute respiratory syndrome coronavirus 2",
    pkg_maintainer="Jane Doe <janedoe@gmail.com>",
```

```
    destdir=tempdir()
)
```

Index