Package ‘biomartr’

December 2, 2023

**Title**  Genomic Data Retrieval

**Version**  1.0.7

**Description**  Perform large scale genomic data retrieval and functional annotation retrieval. This package aims to provide users with a standardized way to automate genome, proteome, 'RNA', coding sequence ('CDS'), 'GFF', and metagenome retrieval from 'NCBI RefSeq', 'NCBI Genbank', 'ENSEMBL', and 'UniProt' databases. Furthermore, an interface to the 'BioMart' database (Smedley et al. (2009) <doi:10.1186/1471-2164-10-22>) allows users to retrieve functional annotation for genomic loci. In addition, users can download entire databases such as 'NCBI RefSeq' (Pruitt et al. (2007) <doi:10.1093/nar/gkl842>), 'NCBI nr', 'NCBI nt', 'NCBI Genbank' (Benson et al. (2013) <doi:10.1093/nar/gks1195>), etc. with only one command.

**VignetteBuilder**  knitr

**NeedsCompilation**  yes

** Depends**  R (>= 3.1.1)

** Imports**  biomaRt, Biostrings, tibble, jsonlite, data.table (= 1.9.4), dplyr (= 0.3.0), readr (= 1.4.0), downloader (= 0.3), RCurl (= 1.95-4.5), XML (= 3.98-1.1), httr (= 0.6.1), stringr (= 0.6.2), purrr, R.utils, philentropy, withr, fs

** Suggests**  knitr (= 1.6), rmarkdown (= 0.3.3), devtools (= 1.6.1), testthat, seqinr, magrittr

** License**  GPL-2

** URL**  https://docs.ropensci.org/biomartr/, https://github.com/ropensci/biomartr

** BugReports**  https://github.com/ropensci/biomartr/issues

** RoxygenNote**  7.2.3

** Encoding**  UTF-8

**X-schema.org-keywords**  BioMart, genomic-data-retrieval, annotation-retrieval, database-retrieval, NCBI, ENSEMBL, biological-data-retrieval

**X-schema.org-applicationCategory**  Data Access
X-schema.org-isPartof `ropensci.org`

Author Hajk-Georg Drost [aut, cre] (<https://orcid.org/0000-0002-1567-306X>), Haakon Tjeldnes [aut, ctb]

Maintainer Hajk-Georg Drost <hajk-georg.drost@tuebingen.mpg.de>

Repository CRAN

Date/Publication 2023-12-02 17:40:07 UTC

R topics documented:

- biomartr-package
- biomart
- cachedir
- cachedir_set
- check_annotation_biomart
- download.database
- download.database.all
- ensembl_divisions
- get.ensembl.info
- getAssemblyStats
- getAttributes
- getBio
- getBioSet
- getCDS
- getCDSSet
- getCollection
- getCollectionSet
- getDatasets
- getENSEMBL
- getENSEMBL.gtf
- getENSEMBL.Seq
- getENSEMBLGENOMESInfo
- getENSMBSInfo
- getFilters
- getGenome
- getGENOMEREPORT
- getGenomeSet
- getGFF
- getGFFSet
- getGO
- getGroups
- getGTF
- getKingdomAssemblySummary
- getKingdoms
- getMarts
- getMetaGenomeAnnotations
- getMetaGenomes
Description

This package interacts with a suite of web Application Programming Interfaces and FTP sites to perform automated genomic data retrieval and annotation information retrieval.
About

To automate the retrieval process on a meta-genomic scale, this package provides useful interface functions for genomic sequence retrieval and functional annotation retrieval. The major aim of biomartr is to facilitate computational reproducibility and large-scale handling of genomic data for (meta-)genomic analyses.

In detail, biomartr aims to provide users with an easy to use framework to obtain genome, proteome, CDS, GFF (annotation), genome assembly quality, and metagenome project data. Furthermore, an interface to the Ensembl Biomart database allows users to retrieve functional annotation for genomic loci. Users can download entire databases such as

- NCBI RefSeq
- NCBI nr
- NCBI nt
- NCBI Genbank
- NCBI nt
- Ensembl
- Ensembl Genomes
- UniProt

Author(s)

Hajk-Georg Drost <hajk-georg.drost@tuebingen.mpg.de>

---

biomart  

Main BioMart Query Function

Description

This function takes a set of gene ids and the biomart specifications and performs a biomart query for the given set of gene ids.

Usage

biomart(genes, mart, dataset, attributes, filters, mute_citation = FALSE, ...)

Arguments

- **genes**: a character vector storing the gene ids of a organisms of interest to be queried against BioMart.
- **mart**: a character string specifying the mart to be used. Users can obtain available marts using `getMarts`.
- **dataset**: a character string specifying the dataset within the mart to be used, e.g. `dataset = "hsapiens_gene_ensembl"`. 
biomart

attributes a character vector specifying the attributes that shall be used, e.g. attributes = c("start_position","end_position","description").

filters a character vector specifying the filter (query key) for the BioMart query, e.g. filter = "ensembl_gene_id".

mute_citation logical value indicating whether citation message should be muted.

... additional parameters for the getBM function.

Details

This function is the main query function of the biomartR package.

It enables to fastly access annotations of a given gene set based on the biomart package implemented by Steffen Durinck et al.

Value

A data.table storing the initial query gene vector in the first column, the output gene vector in the second column, and all attributes in the following columns.

Author(s)

Hajk-Georg Drost

See Also

Other biomaRt: getAttributes(), getDatasets(), getMarts(), organismBM(), organismFilters()

Examples

### Not run:
# 1) select a mart
getMarts()

# we will select mart 'plants_mart' and search for available datasets
getDatasets(mart = "plants_mart")

# we choose dataset 'athaliana_eg_gene' and run biomart()
# using mart: 'plants_mart', dataset: "athaliana_eg_gene"
# attributes: c("start_position","end_position","description")
# for an example gene set of Arabidopsis thaliana:
# c("AT1G06090", "AT1G06100", "AT1G06110", "AT1G06120",
# "AT1G06130", "AT1G06200")

biomart(genes = c("AT1G06090", "AT1G06100",
                  "AT1G06110", "AT1G06120",
                  "AT1G06130", "AT1G06200"),
        mart = "plants_mart",
        dataset = "athaliana_eg_gene",
        attributes = c("start_position","end_position","description"),
        filters = "ensembl_gene_id")
cachedir

Description
Get directory to store back end files like kingdom summaries etc

Usage
cachedir(non_temp_cache = "~/biomartr_cache_dir.rds")

Arguments
non_temp_cache "~/biomartr_cache_dir.rds",

Value
reads the rds file, and returns the path for local cache, if not existing, use tempdir().

See Also
Other cachedir: cachedir_set()

Examples
cachedir()

cachedir_set

Description
Set directory to store back end files like kingdom summaries etc

Usage
cachedir_set(path)

Arguments
path the path to cache dir, example "~/Bio_data/biomartr_cache/"
check_annotation_biomartr

**Value**

invisible(NULL), only save the file to path location

**See Also**

Other cachedir: `cachedir()`

**Examples**

```r
# By default it is tempdir()
cachedir()
# cachedir_set("~/Bio_data/biomartr_cache/")
cachedir()
```

---

**check_annotation_biomartr**

*Check whether an annotation file contains outlier lines*

**Description**

Some annotation files include lines with character lengths greater than 65000. This causes problems when trying to import such annotation files into R using `import`. To overcome this issue, this function screens for such lines in a given annotation file and removes these lines so that `import` can handle the file.

**Usage**

```r
check_annotation_biomartr(annotation_file, remove_annotation_outliers = FALSE)
```

**Arguments**

- `annotation_file`
  
  a file path to the annotation file.

- `remove_annotation_outliers`
  
  shall outlier lines be removed from the input `annotation_file`? If yes, then the initial `annotation_file` will be overwritten and the removed outlier lines will be stored at `tempdir` for further exploration.

**Author(s)**

Hajk-Georg Drost
Examples

```r
## Not run:
# download an example annotation file from NCBI RefSeq
Ath_path <- biomartr::getGFF(organism = "Arabidopsis thaliana")
# run annotation file check on the downloaded file
biomartr::check_annotation_biomartr(Ath_path)
# several outlier lines were detected, thus we re-run the
# function using 'remove_annotation_outliers = TRUE'
# to remove the outliers and overwrite the file
biomartr::check_annotation_biomartr(Ath_path, remove_annotation_outliers = TRUE)

## End(Not run)
```

---

download.database  
**Download a NCBI Database to Your Local Hard Drive**

Description

This function allows users to download a database selected by `listDatabases` to their local hard drive.

Usage

```r
download.database(db, path = "database")
```

Arguments

- `db`: a character string specifying the database that shall be downloaded (selected from `listDatabases`).
- `path`: a character string specifying the location (a folder) in which the corresponding database shall be stored. Default is `path = "database"`. In case this folder does not exist yet, it will be created.

Details

This function downloads large databases to your hard drive. For this purpose a folder named `database` (default) is created and the corresponding database then stored in this folder.

Value

File path to the downloaded database file.

Author(s)

Hajk-Georg Drost

See Also

`download.database.all`, `listDatabases`
**download.database.all**  
*Download all elements of an NCBI database*

---

**Description**

The `download.database` functions allows users to retrieve individual packages of a NCBI database. This function is designed to retrieve the entire database selected by the users (hence all packages corresponding to this database).

**Usage**

```r
download.database.all(db, path = NULL)
```

**Arguments**

- `db`  
a character string specifying the database that shall be downloaded (selected from `listDatabases`).

- `path`  
a character string specifying the location (a folder) in which the corresponding database shall be stored. In case this folder does not exist yet, it will be created.

**Value**

A character vector storing the file paths of the downloaded databases.

**Author(s)**

Hajk-Georg Drost

**See Also**

`download.database, listNCBIDatabases`
Examples

```r
## Not run:
# search for available NCBI databases
listNCBIDatabases(db = "all")
# choose database NCBI nr and download complete database
download.database.all(db = "nr", path = "nr")
## End(Not run)
```

**ensembly_divisions**  
List all available ENSEMBL divisions

Description

Retrieve a list of available databases on ENSEMBL for which `get.ensembl.info` can be retrieved.

Usage

```r
ensembly_divisions()
```

Author(s)

Hajk-Georg Drost

Examples

```r
ensembly_divisions()
```

**get.ensembl.info**  
Helper function to retrieve species information from the ENSEMBL API

Description

This function interfaces with the ENSEMBL API (https://rest.ensembl.org/info/species?content-type=application/json) and internally stores the output to use this information for subsequent retrieval function calls.

Usage

```r
get.ensembl.info(update = FALSE, division)
```

Arguments

- `update` logical, default FALSE. If TRUE, force re-download of info.
- `division` the ENSEMBL database (division) for which information shall be retrieved (available options can be obtained with `ensembly_divisions`).
getAssemblyStats

Author(s)
Hajk-Georg Drost

See Also
ensembl_divisions, getKingdomAssemblySummary, getENSEMBLInfo

Examples

## Not run:
# Look at available ENSEMBL division options
ensembl_divisions()
# Retrieve available information for EnsemblVertebrates
example <- get.ensembl.info(division = "EnsemblVertebrates")
example
# Update information file stored in the tempdir() folder.
example_update <- get.ensembl.info(division = "EnsemblVertebrates", update = TRUE)
example_update

## End(Not run)

getAssemblyStats	Genome Assembly Stats Retrieval

Description

Main genome assembly stats retrieval function for an organism of interest. By specifying the scientific name of an organism of interest the corresponding genome assembly stats file storing the assembly statistics of the organism of interest can be downloaded and stored locally. Genome assembly stats files can be retrieved from several databases.

Usage

getAssemblyStats(
  db = "refseq",
  organism,
  reference = FALSE,
  skip_bacteria = TRUE,
  release = NULL,
  type = "download",
  path = file.path("_ncbi_downloads", "genomeassembly_stats"),
  mute_citation = FALSE
)
getAssemblyStats

Arguments

- **db**
  - A character string specifying the database from which the genome shall be retrieved:
    - `db = "refseq"`
    - `db = "genbank"`
    - `db = "ensembl"`

- **organism**
  - A character string specifying the scientific name of the organism of interest, e.g. `organism = "Homo sapiens"`.

- **reference**
  - A logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

- **skip_bacteria**
  - Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they need to actively specify `skip_bacteria = FALSE`. When `skip_bacteria = FALSE` is set then the bacterial summary file will be downloaded.

- **release**
  - Most recent database version is used. `release = 75` would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

- **type**
  - Shall only the file be retrieved (default) `type = "download"` or should the corresponding file be downloaded and subsequently be imported `type = "import"`.

- **path**
  - A character string specifying the location (a folder) in which the corresponding file shall be stored. Default is `path = file.path("_ncbi_downloads","genomeassembly_stats")`.

- **mute_citation**
  - Logical value indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:


To retrieve available scientific names of organisms and creates a directory `'_ncbi_downloads/genomeassembly_stats'` to store the Genome Assembly Stats of interest as text file for future processing. In case the corresponding fasta file already exists within the `'_ncbi_downloads/genomeassembly_stats'` folder and is accessible within the workspace, no download process will be performed.


Value

- File path to downloaded genome assembly stats file.

Author(s)

- Hajk-Georg Drost
getAttributes

Retrieve All Available Attributes for a Specific Dataset

Description

This function queries the BioMart Interface and returns a table storing all available attributes for a specific dataset.

Usage

getAttributes(mart, dataset, mute_citation = FALSE)

Arguments

- **mart**: a character string specifying the database (mart) for which datasets shall be listed.
- **dataset**: a character string specifying the dataset for which attributes shall be listed.
- **mute_citation**: logical value indicating whether citation message should be muted.

Author(s)

Hajk-Georg Drost

See Also

getGenome, getProteome, getCDs, getGFF, getRNA, getCollection, meta.retrieval, read_assemblystats

Examples

```r
## Not run:
# download the genome assembly stats file of Saccharomyces cerevisiae
# from NCBI RefSeq
# and store the corresponding genome file in
# '_ncbi_downloads/genomeassembly_stats'
file_path <- getAssemblyStats(db = "refseq",
    organism = "Saccharomyces cerevisiae",
    path = file.path("_ncbi_downloads","genomeassembly_stats"))
# import the raw file as it is downloaded
Scerevisiae.stats <- read_assemblystats(file_path, type = "raw")

# download the genome assembly stats file of Saccharomyces cerevisiae
# from NCBI RefSeq
# and import overall statistics of the genome assembly
Scerevisiae.stats.import <- getAssemblyStats(db = "refseq",
    organism = "Saccharomyces cerevisiae",
    type = "import",
    path = file.path("_ncbi_downloads","genomeassembly_stats"))

## End(Not run)
```
See Also

Other biomaRt: `biomart()`, `getDatasets()`, `getMarts()`, `organismBM()`, `organismFilters()`

Examples

```r
## Not run:
# search for available datasets
getMarts()

# choose database (mart): ENSEMBL_MART_ENSEMBL
# and get a table of all available datasets from this BioMart database
head(getDatasets(mart = "ENSEMBL_MART_ENSEMBL"), 10)

# choose dataset: "hsapiens_gene_ensembl"
head(getAttributes(mart = "ENSEMBL_MART_ENSEMBL",
                   dataset = "hsapiens_gene_ensembl"), 5)

## End(Not run)
```

---

**getBio**

A wrapper to all bio getters, selected with 'type' argument

Description

A wrapper to all bio getters, selected with 'type' argument

Usage

```r
getBio(
  db = "refseq",
  organism,
  type,
  reference = FALSE,
  release = NULL,
  gunzip = FALSE,
  update = FALSE,
  skip_bacteria = TRUE,
  path = paste0("set_", toupper(type)),
  remove_annotation_outliers = FALSE,
  analyse_genome = FALSE,
  assembly_type = "toplevel",
  format = "gff3",
  mute_citation = FALSE
)
```
Arguments

**db** a character string specifying the database from which the genome shall be retrieved:
- db = "refseq"
- db = "genbank"
- db = "ensembl"

**organism** Organism selector id, there are three options to characterize an organism:
- by scientific name: e.g. organism = "Homo sapiens"
- by database specific accession identifier: e.g. organism = "GCF_000001405.37" (= NCBI RefSeq identifier for Homo sapiens)
- by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606" (= taxid of Homo sapiens)

**type** biological sequence type. (alternatives are: genome, gff, cds, rna, proteome, assembly_stats, repeat_masker, collection (all the others))

**reference** a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

**release** a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

**gunzip** a logical, indicating whether or not files should be unzipped.

**update** logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc)

**skip_bacteria** Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

**path** character, default location is paste0("set_", toupper(type))

**remove_annotation_outliers** shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at tempdir for further exploration.

**analyse_genome** logical, default FALSE. If TRUE, get general genome statistics like gc content etc. For more details, see ?summary_genome

**assembly_type** character, default c("primary_assembly", "toplevel"). Used for ensembl only, specifies the genome assembly type. Searches for both primary and toplevel, and if both are found, uses the first by order (so primary is prioritized by default). The Primary assembly should usually be used if it exists. The "primary assembly" contains all the top-level sequence regions, excluding alternative haplotypes and patches. If the primary assembly file is not present for a species
getBioSet

Generic Bio data set extractor

Description

Usually you want to use one of the specific set extractors

Usage

getBioSet(
  db = "refseq",
  organisms,
  set_type,
  reference = FALSE,
  release = NULL,
  gunzip = TRUE,
  update = FALSE,
  skip_bacteria = TRUE,
)

(only defined for standard model organisms), that indicates that there were no haplotype/patch regions, and in such cases, the 'toplevel file is used. For more details see: ensembl tutorial

format "gff3", alternative "gtf" for ensembl.
mute_citation logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the overview.txt file from NCBI:


and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes'. In case the corresponding fasta file already exists within the '_ncbi_downloads/genomes' folder and is accessible within the workspace, no download process will be performed. For other file types the same rule applies.

Value

File path to downloaded genome.

Author(s)

Hajk-Georg Drost

See Also

Other getBio: getCDs(), getCollection(), getGFF(), getGenome(), getProteome(), getRNA()
getBioSet

path = paste0("set_", toupper(set_type)),
remove.annotation.outliers = FALSE,
assembly_type = "toplevel",
format = "gff3",
mute.citation = FALSE
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
- db = "refseq"
- db = "genbank"
- db = "ensembl"

organisms a character vector storing the names of the organisms than shall be retrieved. There are three available options to characterize an organism:

set_type the biological sequence type that shall be retrieved. Available options are
- set_type = "genome"
- set_type = "proteome"
- set_type = "cds"
- set_type = "gff"
- set_type = "rna"
- set_type = "assembly.stats"
- set_type = "repeat.masker"
- set_type = "collection" (all the others)

reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

gunzip a logical, indicating whether or not files should be unzipped.

update logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc).

skip.bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip.bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

path character, default location is paste0("set_", toupper(set_type))
remove_annotation_outliers
shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at tempdir for further exploration.

assembly_type
character, default c("primary_assembly", "toplevel"). Used for ensembl only, specifies the genome assembly type. Searches for both primary and toplevel, and if both are found, uses the first by order (so primary is prioritized by default). The Primary assembly should usually be used if it exists. The "primary assembly" contains all the top-level sequence regions, excluding alternative haplotypes and patches. If the primary assembly file is not present for a species (only defined for standard model organisms), that indicates that there were no haplotype/patch regions, and in such cases, the 'toplevel file is used. For more details see: ensembl tutorial

format
"gff3", alternative "gtf" for ensembl.
mute_citation
logical, default FALSE, indicating whether citation message should be muted.

Details
Internally this function loads the the overview.txt file from NCBI:
and creates a directory 'set_CDSs' to store the CDSs of interest as fasta files for future processing. In case the corresponding fasta file already exists within the 'set_CDSs' folder and is accessible within the workspace, no download process will be performed.

Value
File path to downloaded genomes (names are identifiers: 'new' (file was downloaded now), 'old' files did already exist)

Author(s)
Hajk-Georg Drost

See Also
getBio
Other getBioSet: getCDSSet(), getCollectionsSet(), getGFFSet(), getGenomeSet(), getProteomeSet(), getRNASet()

Examples
## Not run:
getBioSet("refseq", organisms = c("Arabidopsis thaliana",
                             "Arabidopsis lyrata",
                             "Capsella rubella"),
set_type = "cds")

## End(Not run)
getCDS

Coding Sequence Retrieval

Description

Main retrieval function for coding sequences (CDS) of an organism of interest. By specifying the scientific name of an organism of interest the corresponding fasta-file storing the CDS information for the organism of interest can be downloaded and stored locally. CDS files can be retrieved from several databases.

Usage

getCDS(
  db = "refseq",
  organism,
  reference = FALSE,
  skip_bacteria = TRUE,
  release = NULL,
  gunzip = FALSE,
  path = file.path("_ncbi_downloads", "CDS"),
  mute_citation = FALSE
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
  • db = "ensembl"

organism Organism selector id, there are three options to characterize an organism:
  • by scientific name: e.g. organism = "Homo sapiens"
  • by database specific accession identifier: e.g. organism = "GCF_000001405.37"
  (= NCBI RefSeq identifier for Homo sapiens)
  • by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
  (= taxid of Homo sapiens)

reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.
getCDS

release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

gunzip a logical, indicating whether or not files should be unzipped.

path a character string specifying the location (a folder) in which the corresponding CDS file shall be stored. Default is path = file.path("_ncbi_downloads","CDS").

mute_citation logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:
and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes'. In case the corresponding fasta file already exists within the '_ncbi_downloads/genomes' folder and is accessible within the workspace, no download process will be performed. For other file types the same rule applies.

Value

File path to downloaded genome.

Author(s)

Hajk-Georg Drost

See Also

Other getBio: getBio(), getCollection(), getGFF(), getGenome(), getProteome(), getRNA()
Other cds: getCDSSet(), read_cds()

Examples

## Not run:
# download the genome of Arabidopsis thaliana from refseq
# and store the corresponding genome CDS file in '_ncbi_downloads/CDS'
file_path <- getCDS( db = "refseq",
                   organism = "Arabidopsis thaliana",
                   path = file.path("_ncbi_downloads","CDS"))

Ath_CDS <- read_cds(file_path, format = "fasta")

## End(Not run)
getCDSSet

CDS retrieval of multiple species

Description

Main CDS retrieval function for a set of organism of interest. By specifying the scientific names of the organisms of interest the corresponding fasta-files storing the CDS of the organisms of interest will be downloaded and stored locally. CDS files can be retrieved from several databases.

Usage

getCDSSet(
  db = "refseq", 
  organisms, 
  reference = FALSE, 
  release = NULL, 
  gunzip = TRUE, 
  update = FALSE, 
  path = "set_CDS"
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
  • db = "ensembl"

organisms a character vector storing the names of the organisms than shall be retrieved. There are three available options to characterize an organism:

reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

gunzip a logical, indicating whether or not files should be unzipped.

update logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc)

path character, default location is paste0("set_", toupper(set_type))
getCollection

Details

Internally this function loads the overview.txt file from NCBI:

and creates a directory ‘set_CDSs’ to store the CDSs of interest as fasta files for future processing.
In case the corresponding fasta file already exists within the ‘set_CDSs’ folder and is accessible
within the workspace, no download process will be performed.

Value

File path to downloaded genomes (names are identifiers: 'new' (file was downloaded now), 'old'
files did already exist)

Author(s)

Hajk-Georg Drost

See Also

Other getBioSet: getBioSet(), getCollectionSet(), getGFFSet(), getGenomeSet(), getProteomeSet(),
getRNASet()

Other cds: getCDS(), read_cds()

Examples

## Not run:
getBioSet("refseq", organisms = c("Arabidopsis thaliana",
                               "Arabidopsis lyrata",
                               "Capsella rubella"),
       set_type = "cds")

## End(Not run)

---

getCollection

Retrieve a Collection: Genome, Proteome, CDS, RNA, GFF, Repeat
Masker, AssemblyStats

Description

Main collection retrieval function for an organism of interest. By specifying the scientific name of
an organism of interest a collection consisting of the genome file, proteome file, CDS file, RNA file,
GFF file, Repeat Masker file, AssemblyStats file of the organism of interest can be downloaded and
stored locally. Collections can be retrieved from several databases.
Usage

getCodeion(
  db = "refseq",
  organism,
  reference = TRUE,
  skip_bacteria = TRUE,
  release = NULL,
  assembly_type = "toplevel",
  analyse_genome = FALSE,
  remove_annotation_outliers = FALSE,
  gunzip = FALSE,
  path = file.path("_db_downloads", "collections"),
  mute_citation = FALSE
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
  • db = "ensembl"
organism Organism selector id, there are three options to characterize an organism:
  • by scientific name: e.g. organism = "Homo sapiens"
  • by database specific accession identifier: e.g. organism = "GCF_000001405.37"
    (= NCBI RefSeq identifier for Homo sapiens)
  • by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
    (= taxid of Homo sapiens)
reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.
skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.
release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.
assembly_type character, default c("primary_assembly", "toplevel"). Used for ensembl only, specifies the genome assembly type. Searches for both primary and toplevel, and if both are found, uses the first by order (so primary is prioritized by default). The Primary assembly should usually be used if it exists. The "primary assembly" contains all the top-level sequence regions, excluding alternative haplotypes and patches. If the primary assembly file is not present for a species
getCollection

(only defined for standard model organisms), that indicates that there were no haplotype/patch regions, and in such cases, the 'toplevel file is used. For more details see: ensembl tutorial

analyse_genome logical, default FALSE. If TRUE, get general genome statistics like gc content etc. For more details, see ?summary_genome

remove_annotation_outliers shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at tempdir for further exploration.

gunzip a logical, indicating whether or not files should be unzipped.

path a character string specifying the location (a folder) in which the corresponding collection shall be stored. Default is path = file.path("_db_downloads","collections").

mute_citation logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:

and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes'. In case the corresponding fasta file already exists within the '_ncbi_downloads/genomes' folder and is accessible within the workspace, no download process will be performed. For other file types the same rule applies.

Value

File path to downloaded genome.

Author(s)

Hajk-Georg Drost

See Also

Other getBio: getBio(), getCDS(), getGFF(), getGenome(), getProteome(), getRNA()
Other collection: getCollectionSet()

Examples

## Not run:
# download the collection of Homo sapiens from refseq
# and store the corresponding genome file in '_ncbi_downloads/collection'
Hsap_collection <- getCollection( db = "refseq",
       organism = "Homo sapiens",
       path = file.path("_db_downloads","collections"))
# download the collection of Homo sapiens from genbank
# and store the corresponding genome file in '_ncbi_downloads/collection'
Hsap_collection <- getCollection( db = "genbank",
organism = "Homo sapiens",
path = file.path("_db_downloads","collections"))
# download the collection of Homo sapiens from ensembl
# and store the corresponding genome file in 'ncbi_downloads/collection'
Hsap_collection <- getCollection( db = "ensembl",
organism = "Homo sapiens",
path = file.path("_db_downloads","collections"))

## End(Not run)

---

**getCollectionSet**

*Retrieve a Collection: Genome, Proteome, CDS, RNA, GFF, Repeat Masker, AssemblyStats of multiple species*

**Description**

Main collection retrieval function for an organism of interest. By specifying the scientific name of
an organism of interest a collection consisting of the genome file, proteome file, CDS file, RNA file,
GFF file, Repeat Masker file, AssemblyStats file of the organism of interest can be downloaded and
stored locally. Collections can be retrieved from several databases.

**Usage**

```r
getCollectionSet(
  db = "refseq",
  organisms,
  reference = FALSE,
  release = NULL,
  skip_bacteria = TRUE,
  gunzip = TRUE,
  update = FALSE,
  remove_annotation_outliers = TRUE,
  path = "set_collections",
  mute_citation = FALSE
)
```

**Arguments**

- `db` a character string specifying the database from which the genome shall be retrieved:
  - `db = "refseq"
  - `db = "genbank"
  - `db = "ensembl"

- `organisms` a character vector storing the names of the organisms than shall be retrieved.

There are three available options to characterize an organism:
**reference**
a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

**release**
a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

**skip_bacteria**
Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

**gunzip**
a logical, indicating whether or not files should be unzipped.

**update**
logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc)

**remove_annotation_outliers**
shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at tempdir for further exploration.

**path**
a character string specifying the location (a folder) in which the corresponding collection shall be stored. Default is path = file.path("_db_downloads", "collections").

**mute_citation**
logical, default FALSE, indicating whether citation message should be muted.

**Details**
Internally this function loads the the overview.txt file from NCBI:
and creates a directory ‘set_CDSs’ to store the CDSs of interest as fasta files for future processing. In case the corresponding fasta file already exists within the ‘set_CDSs’ folder and is accessible within the workspace, no download process will be performed.

**Value**
File path to downloaded genomes (names are identifiers: ‘new’ (file was downloaded now), ’old’ files did already exist)

**Author(s)**
Hajk-Georg Drost

**See Also**
Other getBioSet: getBioSet(), getCDSSet(), getGFFSet(), getGenomeSet(), getProteomeSet(), getRNASet()
Other collection: getCollection()
getDatasets

Examples

## Not run:
getBioSet("refseq", organisms = c("Arabidopsis thaliana",
    "Arabidopsis lyrata",
    "Capsella rubella"),
    set_type = "cds")

## End(Not run)

getDatasets
Retrieve All Available Datasets for a BioMart Database

Description

This function queries the BioMart API and returns a table storing all available datasets for a selected BioMart databases.

Usage

getDatasets(mart, mute_citation = FALSE)

Arguments

mart a character string specifying the database (mart) for which datasets shall be listed.
mute_citation logical value indicating whether citation message should be muted.

Author(s)

Hajk-Georg Drost

See Also

Other biomaRt: biomart(), getAttributes(), getMarts(), organismBM(), organismFilters()

Examples

## Not run:
# search for available datasets
# getMarts()
# choose database: "ENSEML_MART_ENSEMBL"
head(getDatasets("ENSEML_MART_ENSEMBL"), 10)

## End(Not run)
getENSEMBL  

Description

Backend function for retrieving files sequence and annotation files from the ENSEMBL ftp server

Usage

getENSEMBL(
  organism,
  type = "dna",
  id.type = "toplevel",
  release = NULL,
  path,
  format
)

Arguments

organism  Organism selector id, there are three options to characterize an organism:
  • by scientific name: e.g. organism = "Homo sapiens"
  • by database specific accession identifier: e.g. organism = "GCF_000001405.37"
    (= NCBI RefSeq identifier for Homo sapiens)
  • by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
    (= taxid of Homo sapiens)

type character, biological sequence type (e.g. "dna", "cds")
id.type a character, default "toplevel". id type of assembly, either "toplevel" or "primary_assembly" for genomes. Can be other strings, for non genome objects.
release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.
path location where file shall be stored.
format "gff3", alternative "gtf" for ensembl.

Value

either a character path to downloaded file, or a logical FALSE, specifying failure.

Author(s)

Hajk-Georg Drost
getENSEMBL.gtf

Helper function for retrieving gtf files from ENSEMBL

Description

This function downloads gff files of query organisms from ENSEMBL.

Usage

getENSEMBL.gtf(organism, type = "dna", path, release = NULL)

Arguments

organism  Organism selector id, there are three options to characterize an organism:
• by scientific name: e.g. organism = "Homo sapiens"
• by database specific accession identifier: e.g. organism = "GCF_000001405.37"
   (= NCBI RefSeq identifier for Homo sapiens)
• by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
   (= taxid of Homo sapiens)
type  character, biological sequence type (e.g. "dna", "cds")
path  location where file shall be stored.
release  a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

Value

character filepath to download file, returns FALSE if failed.

Author(s)

Hajk-Georg Drost
getENSEMBL.Seq

Helper function for retrieving biological sequence files from ENSEMBL

Description

This function downloads gff files of query organisms from ENSEMBL.

Usage

getENSEMBL.Seq(
  organism,
  type = "dna",
  id.type = "toplevel",
  release = NULL,
  path
)

Arguments

organism Organism selector id, there are three options to characterize an organism:
  • by scientific name: e.g. organism = "Homo sapiens"
  • by database specific accession identifier: e.g. organism = "GCF_000001405.37"
    (= NCBI RefSeq identifier for Homo sapiens)
  • by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
    (= taxid of Homo sapiens)

type character, biological sequence type (e.g. "dna", "cds")
id.type a character, default "toplevel". id type of assembly, either "toplevel" or "primary_assembly" for genomes. Can be other strings, for non genome objects.
release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.
path location where file shall be stored.

Value

either a character path to downloaded file, or a logical FALSE, specifying failure.

Author(s)

Hajk-Georg Drost
getENSEMBLGENOMESInfo  Retrieve ENSEMBLGENOMES info file

Description
Retrieve species and genome information from http://rest.ensemblgenomes.org/info/species?content-type=application/json/.

Usage
getENSEMBLGENOMESInfo()

Author(s)
Hajk-Georg Drost

Examples
## Not run:
info.file <- getENSEMBLGENOMESInfo()
info.file
## End(Not run)

getENSEMBLInfo  Retrieve ENSEMBL info file

Description
Retrieve species and genome information from http://rest.ensembl.org/info/species?content-type=application/json/.

Usage
gitENSEMBLInfo(update = FALSE, divisions = ensembl_divisions())

Arguments
update logical, default FALSE. If TRUE, update cached list, if FALSE use existing cache (if it exists). For cache location see cachedir().
divisions character, name of divisions to check, default is all from ensembl_divisions(). If NULL, also all is used.

Value
a tibble table storing info for all available ENSEMBL divisions.
getFilters

Retrieve All Available Filters for a Specific Dataset

Description

This function queries the BioMart API and returns a table storing all available filters for a specific dataset.

Usage

getFilters(mart, dataset, mute_citation = FALSE)

Arguments

mart a character string specifying the database (mart) for which datasets shall be listed.
dataset a character string specifying the dataset for which filters shall be listed.
mute_citation logical value indicating whether citation message should be muted.

Author(s)

Hajk-Georg Drost

See Also

getMarts, getDatasets, getAttributes, organismBM, organismFilters, organismAttributes
getGenome

## Examples

```r
# Not run:
# search for available datasets
# getMarts()
# choose database (mart): "ENSEMBL_MART_ENSEMBL"
# head(getDatasets(mart = "ENSEMBL_MART_ENSEMBL"), 10)
# choose dataset: "hsapiens_gene_ensembl"
head(getFilters(mart = "ENSEMBL_MART_ENSEMBL",
             dataset = "hsapiens_gene_ensembl"), 5)
```

## Description

Main genome retrieval function for an organism of interest. By specifying the scientific name of an organism of interest the corresponding fasta-file storing the genome of the organism of interest can be downloaded and stored locally. Genome files can be retrieved from several databases. In addition, the genome summary statistics for the retrieved species is stored locally to provide users with insights regarding the genome assembly quality (see `summary_genome` for details). This is useful when comparing genomes with large difference in genome assembly qualities.

## Usage

```r
getGenome(
    db = "refseq",
    organism,
    reference = FALSE,
    skip_bacteria = TRUE,
    release = NULL,
    gunzip = FALSE,
    path = file.path("_ncbi_downloads", "genomes"),
    assembly_type = "toplevel",
    mute_citation = FALSE,
    analyse_genome = FALSE
)
```

## Arguments

- `db` a character string specifying the database from which the genome shall be retrieved:
  - `db = "refseq"`
  - `db = "genbank"`
  - `db = "ensembl"`
- `organism` Organism selector id, there are three options to characterize an organism:
getGenome

- by scientific name: e.g. organism = "Homo sapiens"
- by database specific accession identifier: e.g. organism = "GCF_000001405.37" (= NCBI RefSeq identifier for Homo sapiens)
- by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606" (= taxid of Homo sapiens)

reference
a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

skip_bacteria
Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

release
a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

gunzip
a logical, indicating whether or not files should be unzipped.

path
character, default location is paste0("set_", toupper(type))

assembly_type
character, default c("primary_assembly", "toplevel"). Used for ensembl only, specifies the genome assembly type. Searches for both primary and toplevel, and if both are found, uses the first by order (so primary is prioritized by default). The Primary assembly should usually be used if it exists. The "primary assembly" contains all the top-level sequence regions, excluding alternative haplotypes and patches. If the primary assembly file is not present for a species (only defined for standard model organisms), that indicates that there were no haplotype/patch regions, and in such cases, the ’toplevel file is used. For more details see: ensembl tutorial

mute_citation
logical, default FALSE, indicating whether citation message should be muted.

analyse_genome
logical, default FALSE. If TRUE, get general genome statistics like gc content etc. For more details, see ?summary_genome

Details

Internally this function loads the the overview.txt file from NCBI:
and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes’.
In case the corresponding fasta file already exists within the ’_ncbi_downloads/genomes’ folder and is accessible within the workspace, no download process will be performed. For other file types the same rule applies.

Value

File path to downloaded genome.
getGENOMEREPORT

Author(s)

Hajk-Georg Drost

See Also

Other getBio: getBio(), getCDS(), getCollection(), getGFF(), getProteome(), getRNA()
Other genome: getGenomeSet(), read_genome()

getGENOMEREPORT

Retrieve NCBI GENOME_REPORTS file

Description


Usage

getGENOMEREPORT(
  local_file = file.path(cachedir(), "_ncbi_downloads", "overview.txt")
)

Arguments

local_file character, file path, default: file.path(cachedir(), "_ncbi_downloads", "overview.txt")

Value

a tibble object with the report

Author(s)

Hajk-Georg Drost

Examples

## Not run:
report <- getGENOMEREPORT()
report

## End(Not run)
getGenomeSet (Genome Retrieval of multiple species)

Description
Main genome retrieval function for a set of organism of interest. By specifying the scientific names of the organisms of interest the corresponding fasta-files storing the genome of the organisms of interest will be downloaded and stored locally. Genome files can be retrieved from several databases.

Usage
getGenomeSet(
  db = "refseq", 
  organisms, 
  reference = FALSE, 
  release = NULL, 
  skip_bacteria = TRUE, 
  gunzip = TRUE, 
  update = FALSE, 
  path = "set_genomes", 
  assembly_type = "toplevel", 
  mute_citation = FALSE
)

Arguments

  db      a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
  • db = "ensembl"

  organisms a character vector storing the names of the organisms than shall be retrieved. There are three available options to characterize an organism:

  reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

  release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

  skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.
getGenomeSet

```
  gunzip a logical, indicating whether or not files should be unzipped.
  update logical, default FALSE. Updated backend cached files needed. Usually keep this
  false, to make ut run much faster. Only set to TRUE, if you believe you cache is
  outdated (Species only exist in newest release etc)
  path a character string specifying the location (a folder) in which the corresponding
  genomes shall be stored. Default is path = "set_genomes".
  assembly_type character, default c("primary_assembly", "toplevel"). Used for ensembl only,
  specifies the genome assembly type. Searches for both primary and toplevel,
  and if both are found, uses the first by order (so primary is prioritized by de-
  fault). The Primary assembly should usually be used if it exists. The "primary
  assembly" contains all the top-level sequence regions, excluding alternative hap-
  lotypes and patches. If the primary assembly file is not present for a species
  (only defined for standard model organisms), that indicates that there were no
  haplotype/patch regions, and in such cases, the 'toplevel file is used. For more
  details see: ensembl tutorial
  mute_citation logical, default FALSE, indicating whether citation message should be muted.
```

**Details**

Internally this function loads the the overview.txt file from NCBI:


and creates a directory 'set_CDSs' to store the CDSs of interest as fasta files for future processing. In case the corresponding fasta file already exists within the 'set_CDSs' folder and is accessible within the workspace, no download process will be performed.

**Value**

File path to downloaded genomes (names are identifiers: 'new' (file was downloaded now), 'old' files did already exist)

**Author(s)**

Hajk-Georg Drost

**See Also**

Other getBioSet: `getBioSet()`, `getCDSSet()`, `getCollectionSet()`, `getGFFSet()`, `getProteomeSet()`, `getRNASet()`

Other genome: `getGenome()`, `read_genome()`

**Examples**

```r
## Not run:
getBioSet("refseq", organisms = c("Arabidopsis thaliana",
                            "Arabidopsis lyrata",
                            "Capsella rubella")
```

getGFF

set_type = "cds"

## End(Not run)

---

**getGFF**  
*Genome Annotation Retrieval (GFF3)*

**Description**

Main retrieval function for GFF files of an organism of interest. By specifying the scientific name of an organism of interest the corresponding gff file storing the annotation for the organism of interest can be downloaded and stored locally. GFF files can be retrieved from several databases.

**Usage**

```r
getGFF(
  db = "refseq",
  organism,
  reference = FALSE,
  skip_bacteria = TRUE,
  release = NULL,
  gunzip = FALSE,
  remove_annotation_outliers = FALSE,
  path = file.path("_ncbi_downloads", "annotation"),
  mute_citation = FALSE,
  format = "gff3"
)
```

**Arguments**

- `db`  
  A character string specifying the database from which the genome shall be retrieved:
  - `db = "refseq"`
  - `db = "genbank"`
  - `db = "ensembl"`

- `organism`  
  Organism selector id, there are three options to characterize an organism:
  - by scientific name: e.g. `organism = "Homo sapiens"`
  - by database specific accession identifier: e.g. `organism = "GCF_000001405.37"`  
    (= NCBI RefSeq identifier for *Homo sapiens*)
  - by taxonomic identifier from NCBI Taxonomy: e.g. `organism = "9606"`  
    (= taxid of *Homo sapiens*)

- `reference`  
  A logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.
skip_bacteria  Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify `skip_bacteria = FALSE`. When `skip_bacteria = FALSE` is set then the bacterial summary file will be downloaded.

release  a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

gunzip  a logical, indicating whether or not files should be unzipped.

remove_annotation_outliers  shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at `tempdir` for further exploration.

path  a character string specifying the location (a folder) in which the corresponding annotation file shall be stored. Default is path = file.path("_ncbi_downloads","annotation").

mute_citation  logical, default FALSE, indicating whether citation message should be muted.

format  "gff3", alternative "gtf" for ensembl.

Details

Internally this function loads the the overview.txt file from NCBI:


and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes'. In case the corresponding fasta file already exists within the '_ncbi_downloads/genomes' folder and is accessible within the workspace, no download process will be performed. For other file types the same rule applies.

Value

File path to downloaded genome.

Author(s)

Hajk-Georg Drost

See Also

Other getBio: getBio(), getCDS(), getCollections(), getGenome(), getProteome(), getRNA()
Other gff: getGFFSet(), read_gff()
Examples

```r
## Not run:
# download the annotation of Arabidopsis thaliana from refseq
# and store the corresponding genome file in '_ncbi_downloads/annotation'
Athal_gff <- getGFF(db = "refseq",
    organism = "Arabidopsis thaliana",
    path = file.path("_ncbi_downloads","annotation"),
    remove_annotation_outliers = TRUE)
Athal_gff_import <- read_gff(Athal_gff)

# download the genome of Arabidopsis thaliana from genbank
# and store the corresponding genome file in '_ncbi_downloads/annotation'
Athal_gff <- getGFF(db = "genbank",
    organism = "Arabidopsis thaliana",
    path = file.path("_ncbi_downloads","annotation"),
    remove_annotation_outliers = TRUE)
Athal_gff_import <- read_gff(Athal_gff)

# download the genome of Homo sapiens from ensembl
# and store the corresponding genome file in '_ncbi_downloads/annotation'
Hsap_gff <- getGFF(db = "ensembl",
    organism = "Homo sapiens",
    path = file.path("_ncbi_downloads","annotation"),
    remove_annotation_outliers = TRUE)
Hsap_gff_import <- read_gff(Hsap_gff)

## End(Not run)
```

---

getGFFSet

GFF retrieval of multiple species

Description

Main GFF retrieval function for a set of organism of interest. By specifying the scientific names of the organisms of interest the corresponding fasta-files storing the GFF of the organisms of interest will be downloaded and stored locally. GFF files can be retrieved from several databases.

Usage

```r
getGFFSet(
    db = "refseq",
    organisms,
    reference = FALSE,
    release = NULL,
    skip_bacteria = TRUE,
    gunzip = TRUE,
```
```r
getGFFSet

remove_annotation_outliers = FALSE,
update = FALSE,
format = "gff3",
path = "set_GFF",
mute_citation = FALSE
)

Arguments

db  a character string specifying the database from which the genome shall be retrieved:
   • db = "refseq"
   • db = "genbank"
   • db = "ensembl"

organisms a character vector storing the names of the organisms than shall be retrieved. There are three available options to characterize an organism:

reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

gunzip a logical, indicating whether or not files should be unzipped.

remove_annotation_outliers shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at tempdir for further exploration.

update logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc)

format "gff3", alternative "gtf" for ensembl.

path character, default location is paste0("set_", toupper(set_type))

mute_citation logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:
```
and creates a directory 'set_CDSs' to store the CDSs of interest as fasta files for future processing. In case the corresponding fasta file already exists within the 'set_CDSs' folder and is accessible within the workspace, no download process will be performed.

Value

File path to downloaded genomes (names are identifiers: 'new' (file was downloaded now), 'old' files did already exist)

Author(s)

Hajk-Georg Drost

See Also

Other getBioSet: `getBioSet()`, `getCDSSet()`, `getCollectionSet()`, `getGenomeSet()`, `getProteomeSet()`, `getRNASet()`

Other gff: `getGFF()`, `read_gff()`

Examples

## Not run:
getBioSet("refseq", organisms = c("Arabidopsis thaliana", "Arabidopsis lyrata", "Capsella rubella"), set_type = "cds")

## End(Not run)

---

getGO

Gene Ontology Query

Description

This function takes a gene id as character vector from a given query organism and returns the corresponding GO terms and additional GO information.

Usage

getGO(organism, genes, filters, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>organism</td>
<td>a character string specifying the scientific name of a query organism.</td>
</tr>
<tr>
<td>genes</td>
<td>a character vector storing the gene ids of a organisms of interest to be queried against Ensembl Biomart.</td>
</tr>
<tr>
<td>filters</td>
<td>a character vector specifying the filter (query key) for the Ensembl Biomart query, e.g. filter = &quot;ensembl_gene_id&quot;.</td>
</tr>
<tr>
<td>...</td>
<td>additional parameters that can be passed to the <code>biomart</code> function.</td>
</tr>
</tbody>
</table>
Details

This function takes the scientific name of a query organism, a set of genes for which GO terms and additional information shall be retrieved, and a filter argument that specifies the attribute for the query genes.

Author(s)

Hajk-Georg Drost

See Also

biomart, organismFilters, organismBM, getBM, getMarts, getDatasets, getFilters

Examples

```r
## Not run:
GO_tbl <- getGO(organism = "Arabidopsis thaliana",
genesis = c("AT1G06090", "AT1G06100"),
filters = "ensembl_gene_id")

# look at the result
head(GO_tbl)
## End(Not run)
```

---

getGroups

Retrieve available groups for a kingdom of life (only available for NCBI RefSeq and NCBI Genbank)

Description

A short list of available groups for a kingdom of life.

Usage

groups(db = "refseq", kingdom)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
  Default is db = "refseq".

kingdom a character string specifying for which kingdom of life groups shall be retrieved. See getKingdoms for details.
getGTF

Genome Annotation Retrieval (GTF)

Description

Main retrieval function for GTF files of an organism of interest. By specifying the scientific name of an organism of interest the corresponding GTF file storing the annotation for the organism of interest can be downloaded and stored locally. GTF files can be retrieved from several databases.

Usage

```r
getGTF(
  db = "ensembl",
  organism,
  remove_annotation_outliers = FALSE,
  path = file.path("ensembl", "annotation"),
  release = NULL,
  mute_citation = FALSE
)
```

Arguments

- `db` a character string specifying the database from which the genome shall be retrieved:
  - `db = "refseq"`
  - `db = "genbank"`
getGTF

organism

Organism selector id, there are three options to characterize an organism:

• by scientific name: e.g. organism = "Homo sapiens"
• by database specific accession identifier: e.g. organism = "GCF_000001405.37" (= NCBI RefSeq identifier for Homo sapiens)
• by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606" (= taxid of Homo sapiens)

remove_annotation_outliers

shall outlier lines be removed from the input annotation_file? If yes, then the initial annotation_file will be overwritten and the removed outlier lines will be stored at tempdir for further exploration.

path

a character string specifying the location (a folder) in which the corresponding annotation file shall be stored. Default is path = file.path("_ncbi_downloads","annotation").

release

a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

mute_citation

logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:


and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes'. In case the corresponding fasta file already exists within the '_ncbi_downloads/genomes' folder and is accessible within the workspace, no download process will be performed. For other file types the same rule applies.

Value

File path to downloaded genome.

Author(s)

Hajk-Georg Drost

See Also

Other getBio: getBio(), getCDS(), getCollection(), getGenome(), getProteome(), getRNA()
Other gff: getGFFSet(), read_gff()
getKingdomAssemblySummary

Retrieve and summarise the assembly_summary.txt files from NCBI for all kingdoms

Usage

getKingdomAssemblySummary(
  db,
  skip_bacteria = TRUE,
  file = assemblies_info_path(db)
)

Arguments

db database name. E.g. refseq or genbank.
skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they need to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.
file path, local path to total summary file, default is in tmp folder.

Author(s)

Hajk-Georg Drost
getKingdoms

Retrieve available kingdoms of life

Description

A short list of available kingdoms of life

Usage

getKingdoms(db = "refseq")

Arguments

db a character string specifying the database from which the genome shall be retrieved: db = "refseq", db = "genbank", db = "ensembl", db = "ensemblgenomes". Default is db = "refseq".

Author(s)

Hajk-Georg Drost

See Also

meta.retrieval, getGenome, getProteome, getCDS, getGroups

Examples

# retrieve kingdoms available from refseq
getKingdoms(db = "refseq")

# retrieve kingdoms available from genbank
getKingdoms(db = "genbank")
getMarts

Retrieve information about available Ensembl Biomart databases

Description

This function queries the Ensembl Biomart API and returns a table storing information about all available Ensembl Biomart databases.

Usage

getMarts(update = FALSE)

Arguments

update logical, default FALSE. If FALSE, use cached file if it exists. Set to TRUE to force new update

Author(s)

Hajk-Georg Drost

See Also

Other biomaRt: biomart(), getAttributes(), getDatasets(), organismBM(), organismFilters()

Examples

## Not run:
# get a table of all available databases from Ensembl Biomart
getMarts()

## End(Not run)

getMetaGenomeAnnotations

Retrieve annotation *.gff files for metagenomes from NCBI Genbank

Description

Retrieve available annotation *.gff files for metagenomes from NCBI Genbank. NCBI Genbank allows users to download entire metagenomes and their annotations of several metagenome projects. This function downloads available metagenomes that can then be downloaded via getMetaGenomes.
getMetaGenomes

Usage

getMetaGenomeAnnotations(
    name,
    path = file.path("_ncbi_downloads", "metagenome", "annotations")
)

Arguments

name metagenome name retrieved by listMetaGenomes.
path a character string specifying the location (a folder) in which the corresponding
metagenome annotations shall be stored. Default is path = file.path("_ncbi_downloads", "metagenome").

Author(s)

Hajk-Georg Drost

See Also

getMetaGenomes, listMetaGenomes, getGFF

Examples

## Not run:
# Frist, retrieve a list of available metagenomes
listMetaGenomes()

# Now, retrieve the 'human gut metagenome'
getMetaGenomeAnnotations(name = "human gut metagenome")
## End(Not run)

getMetaGenomes  Retrieve metagenomes from NCBI Genbank

Description

Retrieve available metagenomes from NCBI Genbank. NCBI Genbank allows users to download
entire metagenomes of several metagenome projects. This function downloads available metagenomes
that can then be downloaded via getMetaGenomes.

Usage

getMetaGenomes(name, path = file.path("_ncbi_downloads", "metagenome"))

Arguments

name metagenome name retrieved by listMetaGenomes.
path a character string specifying the location (a folder) in which the corresponding
metagenome shall be stored. Default is path = file.path("_ncbi_downloads", "metagenome").
getMetaGenomeSummary

Author(s)
Hajk-Georg Drost

See Also
getMeta GenomeAnnotations, listMetaGenomes

Examples

## Not run:
# Frist, retrieve a list of available metagenomes
listMetaGenomes()

# Now, retrieve the 'human gut metagenome'
getMetaGenomes(name = "human gut metagenome")

## End(Not run)

getMetaGenomeSummary Retrieve the assembly_summary.txt file from NCBI genbank metagenomes

Description
Retrieval function of the assembly_summary.txt file from NCBI genbank metagenomes. This file stores all available metagenome projects on NCBI Genbank.

Usage
getMetaGenomeSummary(
  local_file = file.path(cachedir(), "assembly_summary_metagenomes_genbank.txt")
)

Arguments
local_file where to store this backend file, default: file.path(cachedir(), "assembly_summary_metagenomes_genbank.txt")

Author(s)
Hajk-Georg Drost

See Also
getKingdomAssemblySummary, getSummaryFile
getProteome

Examples

## Not run:
meta.summary <- getMetaGenomeSummary()
meta.summary

## End(Not run)

getProteome  Proteome Retrieval

Description

Main proteome retrieval function for an organism of interest. By specifying the scientific name of
an organism of interest the corresponding fasta-file storing the proteome of the organism of interest
can be downloaded and stored locally. Proteome files can be retrieved from several databases.

Usage
getProteome(
  db = "refseq",
  organism,
  reference = TRUE,
  skip_bacteria = TRUE,
  release = NULL,
  gunzip = FALSE,
  update = TRUE,
  path = file.path("_ncbi_downloads", "proteomes"),
  mute_citation = FALSE
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  - db = "refseq"
  - db = "genbank"
  - db = "ensembl"

organism Organism selector id, there are three options to characterize an organism:
  - by scientific name: e.g. organism = "Homo sapiens"
  - by database specific accession identifier: e.g. organism = "GCF_000001405.37"
    (= NCBI RefSeq identifier for Homo sapiens)
  - by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
    (= taxid of Homo sapiens)

reference a logical value indicating whether or not a genome shall be downloaded if it isn’t
  marked in the database as either a reference genome or a representative genome.
skip_bacteria  Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify `skip_bacteria = FALSE`. When `skip_bacteria = FALSE` is set then the bacterial summary file will be downloaded.

release  a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is `release = NULL` meaning that the most recent database version is used. `release = 75` would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

gunzip  a logical, indicating whether or not files should be unzipped.

update  logical, default TRUE. (Uniprot only for now!) If species info file exists already, do not re download, makes it faster but the file can be old, i.e. no longer as complete as it could be.

path  a character string specifying the location (a folder) in which the corresponding proteome shall be stored. Default is `path = file.path("_ncbi_downloads","proteomes")`.

mute_citation  logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:


and creates a directory `'_ncbi_downloads/proteomes'` to store the proteome of interest as fasta file for future processing.

Value

File path to downloaded proteome.

Author(s)

Hajk-Georg Drost

See Also

Other getBio: `getBio()`, `getCDS()`, `getCollection()`, `getGFF()`, `getGenome()`, `getRNA()`

Other proteome: `getProteomeSet()`, `read_proteome()`

Examples

```r
## Not run:
# download the proteome of Arabidopsis thaliana from NCBI RefSeq
# and store the corresponding proteome file in '._ncbi_downloads/refseq/proteomes'
file_path <- getProteome( db = "refseq",
                        organism = "Arabidopsis thaliana",
                        path = file.path("_ncbi_downloads","refseq","proteomes") )
```
# import proteome into R session
Ath_proteome <- read_proteome(file_path, format = "fasta")

# download the proteome of Arabidopsis thaliana from NCBI Genbank
# and store the corresponding proteome file in `_ncbi_downloads/genbank/proteomes`
file_path <- getProteome( db = "genbank",
                        organism = "Arabidopsis thaliana",
                        path = file.path("_ncbi_downloads", "genbank", "proteomes") )

# import proteome into R session
Ath_proteome <- read_proteome(file_path, format = "fasta")

# and store the corresponding proteome file in `_downloads/uniprot/proteomes`
file_path <- getProteome( db = "uniprot",
                      organism = "Arabidopsis thaliana",
                      path = file.path("_downloads", "uniprot", "proteomes") )

# import proteome into R session
Ath_proteome <- read_proteome(file_path, format = "fasta")

# download the proteome of Arabidopsis thaliana from ENSEMBL
# and store the corresponding proteome file in `_downloads/ensembl/proteomes`
file_path <- getProteome( db = "ensembl",
                      organism = "Arabidopsis thaliana",
                      path = file.path("_downloads", "ensembl", "proteomes") )

# import proteome into R session
Ath_proteome <- read_proteome(file_path, format = "fasta")

## End(Not run)

---

**getProteomeSet**

Proteome retrieval of multiple species

### Description

Main proteome retrieval function for a set of organism of interest. By specifying the scientific names of the organisms of interest the corresponding fasta-files storing the proteome of the organisms of interest will be downloaded and stored locally. proteome files can be retrieved from several databases.

### Usage

```
getProteomeSet(
    db = "refseq",
    organisms,
    reference = FALSE,
    release = NULL,
    skip_bacteria = TRUE,
    gunzip = TRUE,
    update = FALSE,
    path = "set_proteomes",
)` `
mutecitation = FALSE

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
  • db = "ensembl"
organisms a character vector storing the names of the organisms than shall be retrieved. There are three available options to characterize an organism:
reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.
release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.
skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.
gunzip a logical, indicating whether or not files should be unzipped.
update logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc)
path a character string specifying the location (a folder) in which the corresponding proteomes shall be stored. Default is path = "set_proteomes".
mute_citation logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:
and creates a directory ’set_CDSs’ to store the CDSs of interest as fasta files for future processing. In case the corresponding fasta file already exists within the ’set_CDSs’ folder and is accessible within the workspace, no download process will be performed.

Value

File path to downloaded genomes (names are identifiers: ‘new’ (file was downloaded now), ‘old’ files did already exist)
getProteomeSet

Author(s)
Hajk-Georg Drost

See Also
Other getBioSet: getBioSet(), getCDSSet(), getCollectionSet(), getGFFSet(), getGenomeSet(), getRNASet()

Other proteome: getProteome(), read_proteome()

Examples
## Not run:
# download the proteomes of three different species at the same time
#### Database: NCBI RefSeq
file_paths <- getProteomeSet(db = "refseq", organisms = c("Arabidopsis thaliana",
              "Arabidopsis lyrata",
              "Capsella rubella"))

# look at file paths
file_paths

#### Database: NCBI Genbank
file_paths <- getProteomeSet(db = "genbank", organisms = c("Arabidopsis thaliana",
              "Arabidopsis lyrata",
              "Capsella rubella"))

# look at file paths
file_paths

# download the proteomes of three different species at the same time
#### Database: ENSEMBL
file_paths <- getProteomeSet(db = "ensembl", organisms = c("Homo sapiens",
              "Mus musculus",
              "Caenorhabditis elegans"))

# look at file paths
file_paths

# download the proteomes of three different species at the same time
#### Database: UniProt
file_paths <- getProteomeSet(db = "uniprot", organisms = c("Homo sapiens",
              "Mus musculus",
              "Caenorhabditis elegans"))

# look at file paths
file_paths

## End(Not run)
getReleasesretrieve available database releases or versions of ENSEMBL

Description
Retrieve available database releases or versions of ENSEMBL.

Usage
getReleases(db = "ensembl")

Arguments
db a character string specifying the database from which available release versions shall be retrieved:
  • db = "ensembl"

Author(s)
Hajk-Georg Drost

Examples
## Not run:
# retrieve available release versions of ENSEMBL
getReleases("ensembl")
## End(Not run)

getRepeatMaskerRepeat Masker Retrieval

Description
Main Repeat Masker output retrieval function for an organism of interest. By specifying the scientific name of an organism of interest the corresponding Repeat Masker file storing the genome of the organism of interest can be downloaded and stored locally. Repeat Masker files can be retrieved from several databases.
getRepeatMasker

Usage

getRepeatMasker(
  db = "refseq",
  organism, reference = FALSE,
  skip_bacteria = TRUE,
  release = NULL,
  gunzip = FALSE,
  path = file.path("_ncbi_downloads", "repeatmasker"),
  mute_citation = FALSE
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  • db = "refseq"
  • db = "genbank"
organism a character string specifying the scientific name of the organism of interest, e.g. organism = "Homo sapiens".
reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.
skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.
release most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.
gunzip a logical, indicating whether or not files should be unzipped.
path a character string specifying the location (a folder) in which the corresponding file shall be stored. Default is path = file.path("_ncbi_downloads","repeatmasker").
mute_citation logical value indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:
and creates a directory ’_ncbi_downloads/repeatmasker’ to store the files of interest as fasta file for future processing. In case the corresponding fasta file already exists within the ’_ncbi_downloads/repeatmasker’ folder and is accessible within the workspace, no download process will be performed.
Value

File path to downloaded Repeat Masker output file.

Author(s)

Hajk-Georg Drost

See Also

getGenome, getProteome, getCDS, getGFF, getRNA, getCollection, meta.retrieval, read_rm

Examples

## Not run:

# download the Repeat Masker output file of Homo sapiens from refseq
# and store the corresponding genome file in '_ncbi_downloads/genomes'
file_path <- getRepeatMasker(db = "refseq",
    organism = "Homo sapiens",
    path = file.path("_ncbi_downloads","repeatmasker"))

Hsap_repeatmasker <- read_rm(file_path)

## End(Not run)
Arguments

- **db**: a character string specifying the database from which the genome shall be retrieved:
  - `db = "refseq"`
  - `db = "genbank"`
  - `db = "ensembl"`

- **organism**: Organism selector id, there are three options to characterize an organism:
  - by scientific name: e.g. `organism = "Homo sapiens"`
  - by database specific accession identifier: e.g. `organism = "GCF_000001405.37"` (NCBI RefSeq identifier for Homo sapiens)
  - by taxonomic identifier from NCBI Taxonomy: e.g. `organism = "9606"` (taxid of Homo sapiens)

- **reference**: a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

- **skip_bacteria**: Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify `skip_bacteria = FALSE`.
  - When `skip_bacteria = FALSE` is set then the bacterial summary file will be downloaded.

- **release**: a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is `release = NULL` meaning that the most recent database version is used. `release = 75` would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

- **assembly_type**: character, default c("primary_assembly", "toplevel"). Used for ensembl only, specifies the genome assembly type. Searches for both primary and toplevel, and if both are found, uses the first by order (so primary is prioritized by default). The Primary assembly should usually be used if it exists. The "primary assembly" contains all the top-level sequence regions, excluding alternative haplotypes and patches. If the primary assembly file is not present for a species (only defined for standard model organisms), that indicates that there were no haplotype/patch regions, and in such cases, the 'toplevel file is used. For more details see: ensembl tutorial

- **path**: a character string specifying the location (a folder) in which the corresponding CDS file shall be stored. Default is `path = file.path("_ncbi_downloads", "RNA")`.

- **gunzip**: a logical, indicating whether or not files should be unzipped.

- **mute_citation**: logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:

and creates a directory relative to file type, if you get fasta genomes it will be _ncbi_downloads/genomes'.
In case the corresponding fasta file already exists within the '_ncbi_downloads/genomes' folder and
is accessible within the workspace, no download process will be performed. For other file types the
same rule applies.

Value
File path to downloaded genome.

Author(s)
Hajk-Georg Drost

See Also
Other getBio: getBio(), getCDS(), getCollection(), getGFF(), getGenome(), getProteome()
Other rna: getRNASet(), read_rna()

Examples
## Not run:
# download the RNA of Arabidopsis thaliana from refseq
# and store the corresponding RNA file in '_ncbi_downloads/RNA'
file_path <- getRNA( db = "refseq",
    organism = "Arabidopsis thaliana",
    path = file.path("_ncbi_downloads","RNA"))

Ath_RNA <- read_rna(file_path, format = "fasta")

## End(Not run)
getRNASet

gunzip = TRUE,
update = FALSE,
path = "set_RNAs",
mute_citation = FALSE
)

Arguments

db  a character string specifying the database from which the genome shall be retrieved:
   • db = "refseq"
   • db = "genbank"
   • db = "ensembl"

organisms a character vector storing the names of the organisms than shall be retrieved. There are three available options to characterize an organism:

reference a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome.

release a numeric, the database release version of ENSEMBL (db = "ensembl"). Default is release = NULL meaning that the most recent database version is used. release = 75 would for human would give the stable GRCh37 release in ensembl. Value must be > 46, since ensembl did not structure their data if the standard format before that.

skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

gunzip a logical, indicating whether or not files should be unzipped.

update logical, default FALSE. Updated backend cached files needed. Usually keep this false, to make ut run much faster. Only set to TRUE, if you believe you cache is outdated (Species only exist in newest release etc)

path a character string specifying the location (a folder) in which the corresponding RNAs shall be stored. Default is path = "set_RNAs".

mute_citation logical, default FALSE, indicating whether citation message should be muted.

Details

Internally this function loads the the overview.txt file from NCBI:


and creates a directory ’set_CDSs’ to store the CDSs of interest as fasta files for future processing. In case the corresponding fasta file already exists within the ’set_CDSs’ folder and is accessible within the workspace, no download process will be performed.
getValue

File path to downloaded genomes (names are identifiers: 'new' (file was downloaded now), 'old' files did already exist)

Author(s)
Hajk-Georg Drost

See Also
Other getBioSet: getBioSet(), getCDSSet(), getCollectionSet(), getGFFSet(), getGenomeSet(), getProteomeSet()
Other rna: getRNA(), read_rna()

Examples
## Not run:
getBioSet("refseq", organisms = c("Arabidopsis thaliana", "Arabidopsis lyrata", "Capsella rubella"), set_type = "cds")

## End(Not run)

getSummaryFile

Helper function to retrieve the assembly_summary.txt file from NCBI

Description
Retrieval function of the assembly_summary.txt file from NCBI.

Usage
getSummaryFile(db, kingdom, file = assemblies_info_path(db, kingdom))

Arguments
db
database name. E.g. refseq or genbank.
kingdom
kingdom for which assembly_summary.txt file shall be retrieved. See also getKingdoms.
file
path, local path to total summary file, default is in tmp folder.

Author(s)
Hajk-Georg Drost

See Also
getKingdomAssemblySummary, getMetaGenomeSummary
**getUniProtInfo**  
*Get uniprot info from organism*

### Description
Get uniprot info from organism

### Usage
```r
getUniProtInfo(organism, path = cachedir(), update = TRUE)
```

### Arguments
- **organism**: character, name of organism
- **path**: path at which the info file shall be stored locally.
- **update**: shall the internal `cachedir` file be deleted and the info file freshly downloaded from the UniProt API?

---

**getUniProtSTATS**  
*Retrieve UniProt Database Information File (STATS)*

### Description
The UniProt stores a STATS file to summarise all available information for their reference proteomes. Users can now download this file and process it with biomart.

### Usage
```r
getUniProtSTATS(update = FALSE)
```

### Arguments
- **update**: shall the internal `cachedir` file be deleted and the STATS file freshly downloaded from the UniProt FTP servers?

### Author(s)
Hajk-Georg Drost
is.genome.available

Examples

```r
## Not run:
# retrieve STATS file from UniProt
uniprot_info <- getUniProtSTATS(update = TRUE)
# look at results
uniprot_info

## End(Not run)
```

is.genome.available  Check Genome Availability

Description

This function checks the availability of a given genome on the NBCI servers specified as scientific name.

Usage

```r
is.genome.available(
  db = "refseq",
  organism,
  skip_bacteria = TRUE,
  details = FALSE
)
```

Arguments

db a character string specifying the database from which the genome shall be retrieved:

- db = "refseq"
- db = "genbank"
- db = "ensembl"
- db = "uniprot"

organism there are three options to characterize an organism:

- by scientific name: e.g. organism = "Homo sapiens"
- by database specific accession identifier: e.g. organism = "GCF_000001405.37"
  (= NCBI RefSeq identifier for Homo sapiens)
- by taxonomic identifier from NCBI Taxonomy: e.g. organism = "9606"
  (= taxid of Homo sapiens)

skip_bacteria Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they need to actively specify skip_bacteria = FALSE. When skip_bacteria = FALSE is set then the bacterial summary file will be downloaded.

details a logical value specifying whether or not details on genome size, kingdom, etc. shall be printed to the console instead of a boolean value.
Details

Internally this function calls the `listGenomes` function to detect all available genomes and checks whether or not the specified organism is available for download.

Value

a logical value specifying whether or not the genome of the input organism is available. In case `details = TRUE` only a character string specifying the genome details is being returned.

Author(s)

Hajk-Georg Drost

Examples

```r
## Not run:
# checking whether the Homo sapiens genome is stored on NCBI
is.genome.available(organism = "Homo sapiens", db = "refseq")

# and printing details
is.genome.available(organism = "Homo sapiens", db = "refseq", details = TRUE)

# checking whether the Homo sapiens genome is stored on ENSEMBL
is.genome.available(organism = "Homo sapiens", db = "ensembl")

# and printing details
is.genome.available(organism = "Homo sapiens",
                   details = TRUE,
                   db = "ensembl")

## End(Not run)
```
listGenomes

List All Available Genomes either by kingdom, group, or subgroup

Description

This function retrieves the names of all genomes available on the NCBI ftp:// server and stores the results in a file named 'overview.txt' inside the directory _ncbi_downloads' that is built inside the workspace.

Usage

listDatabases(db = "nr", update = FALSE)
listNCBIDatabases(db = "nr", update = FALSE)

Arguments

db: a character string specifying the name of the database that shall be searched for.
update: a logical value specifying whether or not the local listDatabases.txt file shall be updated by remote access to NCBI.

Author(s)

Hajk-Georg Drost

See Also

download.database, download.database.all

Examples

## Not run:
# retrieve all versions of the NCBI 'nr' database that can be downloaded
listNCBIDatabases(db = "nr")

# analogous:
# listNCBIDatabases(db = "cdd")
# listNCBIDatabases(db = "nt")
# listNCBIDatabases(db = "gss")
# listNCBIDatabases(db = "refseq_protein")

## End(Not run)
Usage

```r
listGenomes(
  db = "refseq",
  type = "all",
  subset = NULL,
  details = FALSE,
  update = FALSE,
  skip_bacteria = FALSE
)
```

Arguments

- **db**: a character string specifying the database for which genome availability shall be checked. Available options are:
  - `db = "refseq"`
  - `db = "genbank"`
  - `db = "ensembl"`

- **type**: a character string specifying a potential filter of available genomes. Available options are:
  - `type = "all"`, no subset
  - `type = "kingdom"`, subset on kingdom
  - `type = "group"`, subset on group
  - `type = "subgroup"`, subset on subgroup

- **subset**: a character string or character vector specifying a subset of `type`. E.g. if users are interested in retrieving all Eukaryota species, they can specify: `type = "kingdom"` and `subset = "Eukaryota"`.

- **details**: a boolean value specifying whether only the scientific names of stored genomes shall be returned (`details = FALSE`) or all information such as
  - `organism_name`
  - `kingdoms`
  - `group`
  - `subgroup`
  - `file_size_MB`, etc.

- **update**: logical, default `FALSE`. If `TRUE`, update cached list, if `FALSE` use existing cache (if it exists). For cache location see `cachedir()`

- **skip_bacteria**: Due to its enormous dataset size (> 700MB as of July 2023), the bacterial summary file will not be loaded by default anymore. If users wish to gain insights for the bacterial kingdom they needs to actively specify `skip_bacteria = FALSE`. When `skip_bacteria = FALSE` is set then the bacterial summary file will be downloaded.

Details

Internally this function loads the overview.txt file from NCBI and creates a directory `'_ncbi_downloads'` in the `temdir()` folder to store the overview.txt file for future processing. In case the overview.txt
file already exists within the '_ncbi_downloads' folder and is accessible within the workspace, no
download process will be performed again.

Note

Please note that the ftp:// connection relies on the NCBI or ENSEMBL server and cannot be accu-
rately accessed via a proxy.

Author(s)

Hajk-Georg Drost

Examples

```r
## Not run:
# print details for refseq
listGenomes(db = "refseq")
# print details for all plants in refseq
listGenomes(db = "refseq", type = "kingdom")
# print details for all plant groups in refseq
listGenomes(db = "refseq", type = "group")
# print details for all plant subgroups in refseq
listGenomes(db = "refseq", type = "subgroup")
# Ensembl
listGenomes(db = "ensembl", type = "kingdom", subset = "EnsemblVertebrates")
## End(Not run)
```

listGroups

List number of available genomes in each taxonomic group

Description

Users can retrieve the available number of sequenced genomes per group. Only available for `db = "refseq"` and `db = "genbank"`.

Usage

`listGroups(db = "refseq", kingdom = "all", details = FALSE)`

Arguments

- `db` a character string specifying the database for which genome availability shall be checked. Available options are:
  - `db = "refseq"`
  - `db = "genbank"`
- `kingdom` a kingdom specification retrieved by `getKingdoms`.
- `details` shall all species corresponding to the specified kingdom be returned? Default is `details = FALSE`. 
listKingdoms

Author(s)

Hajk-Georg Drost

See Also

listGenomes, is.genome.available, listKingdoms

Examples

```r
## Not run:
# example for refseq
listGroups(db = "refseq")
# example for genbank
listGroups(db = "genbank")
### in case groups should be specified by kingdom
# first, retrieve available kingdom names
listKingdoms()
# now we choose kingdom "bacteria"
listGroups(db = "refseq", kingdom = "bacteria")
# or
listGroups(db = "genbank", kingdom = "bacteria")
## End(Not run)
```

listKingdoms  List number of available genomes in each kingdom of life

Description

Users can retrieve the available number of sequenced genomes per kingdom.

Usage

```r
listKingdoms(db = "refseq")
```

Arguments

db  a character string specifying the database for which genome availability shall be checked, e.g. db = "refseq", db = "genbank", db = "ensembl".

Author(s)

Hajk-Georg Drost

See Also

listGenomes, is.genome.available, listGroups
(listMetaGenomes)

Description

List available metagenomes on NCBI genbank. NCBI genbank allows users to download entire metagenomes of several metagenome projects. This function lists all available metagenomes that can then be downloaded via `getMetaGenomes`.

Usage

```r
listMetaGenomes(details = FALSE)
```

Arguments

- `details` a boolean value specifying whether only the scientific names of stored metagenomes shall be returned (details = FALSE) or all information such as "organism_name","bioproject", etc (details = TRUE).

Author(s)

Hajk-Georg Drost

See Also

`getMetaGenomes`,`getMetaGenomeSummary`

Examples

```r
## Not run:
# retrieve available metagenome projects at NCBI Genbank
listMetaGenomes()
# retrieve detailed information on available metagenome projects
# at NCBI Genbank
listMetaGenomes(details = TRUE)

## End(Not run)
```
meta.retrieval

Perform Meta-Genome Retrieval

Description

Download genomes, proteomes, cds, gff, rna, or assembly stats files of all species within a kingdom of life.

Usage

meta.retrieval(
  db = "refseq",
  kingdom,
  group = NULL,
  type = "genome",
  restart_at_last = TRUE,
  reference = FALSE,
  combine = FALSE,
  path = NULL
)

Arguments

db a character string specifying the database from which the genome shall be retrieved:
  - db = "refseq"
  - db = "genbank"
  - db = "ensembl"

kingdom a character string specifying the kingdom of the organisms of interest, e.g.
  - For NCBI RefSeq:
    - kingdom = "archaea"
    - kingdom = "bacteria"
    - kingdom = "fungi"
    - kingdom = "invertebrate"
    - kingdom = "plant"
    - kingdom = "protozoa"
    - kingdom = "viral"
    - kingdom = "vertebrate_mammalian"
    - kingdom = "vertebrate_other"
  - For NCBI Genbank:
    - kingdom = "archaea"
    - kingdom = "bacteria"
    - kingdom = "fungi"
- kingdom = "invertebrate"
- kingdom = "plant"
- kingdom = "protozoa"
- kingdom = "vertebrate_mammalian"
- kingdom = "vertebrate_other"

• For ENSEMBL:
  - kingdom = "EnsemblVertebrates"
  - kingdom = "EnsemblPlants"
  - kingdom = "EnsemblFungi"
  - kingdom = "EnsemblMetazoa"
  - kingdom = "EnsemblBacteria"
  - kingdom = "EnsemblProtists"

Available kingdoms can be retrieved with `getKingdoms`.

**group**

only species belonging to this subgroup will be downloaded. Groups can be retrieved with `getGroups`.

**type**

type of sequences that shall be retrieved. Options are:

• type = "genome": (for genome assembly retrieval; see also `getGenome`),
• type = "proteome": (for proteome retrieval; see also `getProteome`),
• type = "cds": (for coding sequence retrieval; see also `getCDS`),
• type = "gff": (for annotation file retrieval in gff format; see also `getGFF`),
• type = "gtf": (for annotation file retrieval in gtf format (only for ensembl and ensemblgenomes); see also `getGTF`)  
• type = "rna": (for RNA file retrieval in fasta format; see also `getRNA`),
• type = "rm": (for Repeat Masker output file retrieval; see also `getRepeatMasker`),
• type = "assemblystats": (for genome assembly quality stats file retrieval; see also `getAssemblyStats`).

**restart_at_last**

a logical value indicating whether or not `meta.retrieval` should pick up at the last species when re-running the function.

• If `restart_at_last = TRUE` (Default) then `meta.retrieval` will skip all organisms that are already present in the folder and will start downloading all remaining species. However, this way `meta.wretrieval` will not be able to check whether already downloaded organism files are corrupted or not by checking the md5 checksum.
• If `restart_at_last = FALSE` then `meta.retrieval` will start from the beginning and crawl through already downloaded organism files and check whether already downloaded organism files are corrupted or not by checking the md5 checksum. After checking existing files the function will start downloading all remaining organisms.

**reference**

a logical value indicating whether or not a genome shall be downloaded if it isn’t marked in the database as either a reference genome or a representative genome. Options are:

• reference = FALSE (Default): all organisms (reference, representative, and non-representative genomes) are downloaded.
• reference = TRUE: organisms that are downloaded must be either a reference or representative genome. Thus, most genomes which are usually non-reference genomes will not be downloaded.

combine just in case type = "assemblystats" is specified, shall assembly stats of individual species be imported and combined to a data.frame?

path path to the folder in which downloaded genomes shall be stored. By default the kingdom name is used to name the output folder.

Details

This function aims to perform bulk retrieval of the genomes, proteomes, cds, etc. of species that belong to the same kingdom of life or to the same subgroup.

Value

a character vector storing the file paths of the retrieved files.

Author(s)

Hajk-Georg Drost

See Also

Other meta_retrieval: meta.retrieval.all()

Examples

```r
## Not run:
# get all available kingdoms for refseq
getKingdoms(db = "refseq")
# download all vertebrate genomes from refseq
meta.retrieval(kingdom = "vertebrate_mammalian",
               db = "refseq",
               type = "genome")

# get all available kingdoms for genbank
getKingdoms(db = "genbank")
# download all vertebrate genomes from genbank
meta.retrieval(kingdom = "vertebrate_mammalian",
               db = "genbank",
               type = "genome")

# In case users do not wish to retrieve genomes from an entire kingdom,
# but rather from a subgroup (e.g. from species belonging to the
# Gammaproteobacteria class, a subgroup of the bacteria kingdom),
# they can use the following workflow
# First, users can again consult the getKingdoms() function to retrieve
# kingdom information.
getKingdoms(db = "refseq")
```
# In this example, we will choose the bacteria kingdom.
# Now, the getGroups() function allows users to obtain available
# subgroups of the bacteria kingdom.
groups <- getGroups(db = "refseq", kingdom = "bacteria")

# Now we choose the group Gammaproteobacteria and specify
# the group argument in the meta.retrieval() function
meta.retrieval(kingdom = "bacteria",
               group = "Gammaproteobacteria",
               db = "refseq",
               type = "genome")

## End(Not run)

---

**meta.retrieval.all**

Perform Meta-Genome Retrieval of all organisms in all kingdoms of life

**Description**

Download genomes, proteomes, cds, gff, rna, or assembly stats files of individual species of all kingdoms of life.

**Usage**

```r
meta.retrieval.all(db = "refseq", type = "genome", reference = FALSE)
```

**Arguments**

- `db`:
  - a character string specifying the database from which the genome shall be retrieved:
    - `db = "refseq"`
    - `db = "genbank"`
    - `db = "emsembl"`
    - `db = "ensemblgenomes"`

- `type`:
  - type of sequences that shall be retrieved. Options are:
    - `type = "genome"` (for genome assembly retrieval; see also `getGenome`),
    - `type = "proteome"` (for proteome retrieval; see also `getProteome`),
    - `type = "cds"` (for coding sequence retrieval; see also `getCDS`),
    - `type = "gff"` (for annotation file retrieval in gff format; see also `getGFF`),
    - `type = "gtf"` (for annotation file retrieval in gtf format (only for ensembl and ensemblgenomes); see also `getGTF`),
    - `type = "rma"` (for RNA file retrieval in fasta format; see also `getRNA`),
    - `type = "rm"` (for Repeat Masker output file retrieval; see also `getRepeatMasker`),
    - `type = "assemblystats"` (for genome assembly quality stats file retrieval; see also `getAssemblyStats`).
organismAttributes

**Reference**

A logical value indicating whether or not a genome shall be downloaded if it isn't marked in the database as either a reference genome or a representative genome. Options are:

- `reference = FALSE` (Default): all organisms (reference, representative, and non-representative genomes) are downloaded.
- `reference = TRUE`: organisms that are downloaded must be either a reference or representative genome. Thus, most genomes which are usually non-reference genomes will not be downloaded.

Details

This function aims to perform bulk retrieval of all genomes of species for all kingdoms of life.

Value

A character vector storing the file paths of the retrieved files.

Author(s)

Hajk-Georg Drost

See Also

Other meta_retrival: `meta.retrieval()`

Examples

```r
## Not run:
# download all genomes from refseq
meta.retrieval.all(db = "refseq", type = "genome")
# download all vertebrate genomes from genbank
meta.retrieval.all(db = "genbank", type = "genome")
# download all vertebrate genomes from ensemblgenomes
meta.retrieval.all(db = "genbank", type = "ensemblgenomes")
## End(Not run)
```

---

**organismAttributes**

Retrieve Ensembl Biomart attributes for a query organism

Description

In addition to the `organismBM` function, this function returns all available attributes that can be accessed through different marts and datasets for a given query organism.

Usage

`organismAttributes(organism, update = FALSE, topic = NULL)`
**Arguments**

- **organism**: a character string specifying the scientific name of a query organism.
- **update**: a logical value specifying whether or not the local listMarts.txt, listDatasets.txt, and listAttributes_organism.txt files shall be updated by remote access to BioMart.
- **topic**: a character string specifying a topic (category) of attributes, e.g. `topic = "id"`.

**Details**

For a given query organism, this function retrieves all available attributes that can be accessed through different marts and datasets.

Sometimes the same attribute names correspond to different datasets and marts causing problems when using `getMarts`. The approach introduced by this function provides (again) a organism centric way of accessing organism specific attributes.

The `topic` argument allows the user to search for specific attribute topics/categories for faster filtering.

**Value**

a data.frame storing corresponding attribute names, description, datasets, and marts.

**Note**

When you run this function for the first time, the data retrieval procedure will take some time, due to the remote access to BioMart. The corresponding result is then saved in a *.txt file within the `tempdir` directory named "_biomart/listMarts.txt","_biomart/listDatasets.txt","_biomart/listAttributes_organism.txt", allowing subsequent queries to perform much faster.

**Author(s)**

Hajk-Georg Drost

**References**

http://biomart.org/


**See Also**

`organismFilters`, `organismBM`, `biomart`, `listAttributes`
Examples

```r
## Not run:
# search for attribute topic id
head(organismAttributes("Homo sapiens", topic = "id"), 20)

## End(Not run)
```

organismBM

Retrieve Ensembl Biomart marts and datasets for a query organism

Description

This function returns either all available biomart connections for all available organisms for which biomart access is possible, or (when specified) returns all organism specific biomart connections.

Usage

```r
organismBM(organism = NULL, update = FALSE, mute_citation = TRUE)
```

Arguments

- `organism` a character string specifying the scientific name of a query organism. Default is `organism = NULL`. In this case all available biomart connections are returned.
- `update` a logical value specifying whether or not the local listMart.txt and listDatasets.txt files shall be updated by remote access to BioMart.
- `mute_citation` logical value indicating whether citation message should be muted.

Details

This function collects all available biomart connections and returns a table storing the organism for which biomart connections are available as well as the corresponding mart and database.

Note

When you run this function for the first time, the data retrieval procedure will take some time, due to the remote access to BioMart. The corresponding result is then saved in a *.txt file named ".biomart/listDatasets.txt" in the `tempdir` directory, allowing subsequent queries to perform much faster.

Author(s)

Hajk-Georg Drost
organismFilters

Retrieve Ensembl Biomart filters for a query organism

Description

In addition to the organismBM and organismAttributes functions, this function returns all available filters that can be accessed through different marts and datasets for a given query organism.

Usage

organismFilters(organism, update = FALSE, topic = NULL)

Arguments

organism a character string specifying the scientific name of a query organism.
update a logical value specifying whether or not the local listMart.txt, listDatasets.txt, and listFilters_organism.txt files shall be updated by remote access to BioMart.
topic a character string specifying a topic (category) of filters, e.g. topic = "id".

References

http://biomart.org/


See Also

Other biomaRt: biomart(), getAttributes(), getDatasets(), getMarts(), organismFilters()
organismFilters

Details

For a given query organism, this function retrieves all available filters that can be accessed through different marts and datasets.

Sometimes the same filter names correspond to different datasets and marts causing problems when using `getMarts`. The approach introduced by this function provides (again) a organism centric way of accessing organism specific filters.

The `topic` argument allows the user to search for specific filters topics/categories for faster selection.

Value

a data.frame storing corresponding filter names, description, datasets, and marts.

Note

When you run this function for the first time, the data retrieval procedure will take some time, due to the remote access to BioMart. The corresponding result is then saved in a *.txt file within the `tempdir` directory named "_biomart/listMarts.txt","_biomart/listDatasets.txt", and "_biomart/listFilters_organism.txt", allowing subsequent queries to perform much faster.

Author(s)

Hajk-Georg Drost

References

http://biomart.org/


See Also

Other biomaRt: `biomart()`, `getAttributes()`, `getDatasets()`, `getMarts()`, `organismBM()`

Examples

```r
## Not run:
# search for filter topic "id"
head(organismFilters("Homo sapiens", topic = "id"), 20)

## End(Not run)
```
read_assemblystats Import Genome Assembly Stats File

Description
This function reads an organism specific Genome Assembly Stats file that was retrieved with getAssemblyStats.

Usage
read_assemblystats(file, type = "raw")

Arguments
file a character string specifying the path to the file storing the Genome Assembly Stats file.
type a tibble object, either type = "raw" to import the entire genome assembly stats file or type = "stats" to import overall statistics including all chromosomes, mitochondria and plastids.

Details
This function takes a string specifying the path to the Genome Assembly Stats file of interest (e.g. the path returned by getAssemblyStats) and imports it.

Author(s)
Hajk-Georg Drost

See Also
getAssemblyStats, read_genome, read_proteome, read_cds, read_gff

read_cds Import CDS as Biostrings or data.table object

Description
This function reads an organism specific CDS stored in a defined file format.
read_cds

Usage

read_cds(
  file,
  format = "fasta",
  obj.type = "Biostrings",
  delete_corrupt = FALSE,
  ...
)

Arguments

file  a character string specifying the path to the file storing the CDS.
format a character string specifying the file format used to store the genome, e.g. format = "fasta" (default) or format = "gbk".
obj.type a character string specifying the object type in which the genomic sequence shall be represented. Either as obj.type = "Biostrings" (default) or as obj.type = "data.table".
delete_corrupt a logical value specifying whether potential CDS sequences that cannot be divided by 3 shall be excluded from the dataset. Default is delete_corrupt = FALSE.
... additional arguments that are used by read.fasta.

Details

The read.cds function takes a string specifying the path to the cds file of interest as first argument. It is possible to read in different proteome file standards such as fasta or genebank.

CDS stored in fasta files can be downloaded from http://www.ensembl.org/info/data/ftp/index.html.

Value

A data.table storing the gene id in the first column and the corresponding sequence as string in the second column.

Author(s)

Hajk-Georg Drost

See Also

Other cds: getCDSSet(), getCDS()
Other readers: read_genome(), read_gff(), read_proteome(), read_rna()
read_genome

Import Genome Assembly as Biostrings or data.table object

Description

This function reads an organism specific genome stored in a defined file format.

Usage

read_genome(file, format = "fasta", obj.type = "Biostrings", ...)

Arguments

file
  a character string specifying the path to the file storing the genome.
format
  a character string specifying the file format used to store the genome, e.g. format = "fasta" (default) or format = "gbk".
obj.type
  a character string specifying the object type in which the genomic sequence shall be represented. Either as obj.type = "Biostrings" (default) or as obj.type = "data.table".
...
  additional arguments that are used by the read.fasta function.

Details

This function takes a string specifying the path to the genome file of interest as first argument (e.g. the path returned by getGenome).

Value

Either a Biostrings or data.table object.

Author(s)

Hajk-Georg Drost

See Also

Other genome: getGenomeSet(), getGenome()
Other readers: read_cds(), read_gff(), read_proteome(), read_rna()
read_gff

Import GFF File

Description

This function reads an organism specific CDS stored in a defined file format.

Usage

read_gff(file)

Arguments

file  a character string specifying the path to the file storing the CDS.

Details

This function takes a string specifying the path to the GFF file of interest (e.g. the path returned by getGFF).

Value

Either a Biostrings or data.table object.

Author(s)

Hajk-Georg Drost

See Also

Other gff: getGFFSet(), getGFF()
Other readers: read_cds(), read_genome(), read_proteome(), read_rna()

read_proteome

Import Proteome as Biostrings or data.table object

Description

This function reads an organism specific proteome stored in a defined file format.

Usage

read_proteome(file, format = "fasta", obj.type = "Biostrings", ...)
Arguments

file  a character string specifying the path to the file storing the proteome.

format a character string specifying the file format used to store the genome, e.g. format = "fasta" (default) or format = "gbk".

obj.type a character string specifying the object stype in which the genomic sequence shall be represented. Either as obj.type = "Biostrings" (default) or as obj.type = "data.table".

... additional arguments that are used by read.fasta.

Details

This function takes a string specifying the path to the proteome file of interest as first argument.

It is possible to read in different proteome file standards such as fasta or genebank.

Value

Either a Biostrings or data.table object.

Author(s)

Hajk-Georg Drost

See Also

Other readers: read_cds(), read_genome(), read_gff(), read_rna()
Other proteome: getProteomeSet(), getProteome()

Description

This function reads an organism specific Repeat Masker output file.

Usage

read_rm(file)

Arguments

file  a character string specifying the path to the file storing the Repeat Masker output (e.g. retrieved with getRepeatMasker).

Details

This function takes a string specifying the path to the Repeat Masker output file of interest as first argument.
Description

This function reads an organism specific RNA stored in a defined file format.

Usage

read_rna(file, format = "fasta", obj.type = "Biostrings", ...)

Arguments

- `file`: a character string specifying the path to the file storing the RNA.
- `format`: a character string specifying the file format used to store the genome, e.g. format = "fasta" (default) or format = "gbk".
- `obj.type`: a character string specifying the object stype in which the genomic sequence shall be represented. Either as obj.type = "Biostrings" (default) or as obj.type = "data.table".
- `...`: additional arguments that are used by read.fasta.

Details

This function takes a string specifying the path to the RNA file of interest as first argument. It is possible to read in different proteome file standards such as fasta or genebank.

Value

A data.table storing the gene id in the first column and the corresponding sequence as string in the second column.

Author(s)

Hajk-Georg Drost

See Also

Other rna: getRNASet(), getRNA()
Other readers: read_cds(), read_genome(), read_gff(), read_proteome()
refseqOrganisms

Retrieve All Organism Names Stored on refseq

Description

This function extracts all organism names (scientific names) for which genomes, proteomes, and CDS files are stored on the NCBI refseq server.

Usage

refseqOrganisms()

Author(s)

Hajk-Georg Drost

summary_cds

Retrieve summary statistics for a coding sequence (CDS) file

Description

A summary statistics of specific CDS features is returned.

Usage

summary_cds(file, organism)

Arguments

file file path to a CDS file in fasta format.
organism character string specifying the organism at hand.

Details

The summary statistics include:

• total_seqs:
• nnn_abs: The total number of NNN’s (over all chromosomes/scaffolds/contigs) in all coding sequences combined
• nnn_perc: The percentage (relative frequency) of NNN’s (over all chromosomes/scaffolds/contigs) compared to the total number of nucleotides of all coding sequences

Author(s)

Hajk-Georg Drost
**summary_genome**

*Retrieve summary statistics for a genome assembly file*

**Description**

A summary statistics of specific genome features is generated. These statistics are useful to assess the genome quality of retrieved genome assemblies when performing comparative genomics tasks. This way, users can assess whether or not patterns found based on genome comparisons aren’t just a technical artifact of differences in genome assembly quality.

**Usage**

`summary_genome(file, organism)`

**Arguments**

- `file`: file path to a genome assembly file in fasta format.
- `organism`: character string specifying the organism at hand.

**Details**

The summary statistics include:

- `genome_size_mbp`: Genome size in mega base pairs
- `n50_mbp`: The N50 contig size of the genome assembly in mega base pairs
- `n_seqs`: The number of chromosomes/scaffolds/contigs of the genome assembly file
- `n_nnn`: The absolute number of NNNs (over all chromosomes or scaffolds or contigs) in the genome assembly file
- `rel_nnn`: The percentage (relative frequency) of NNNs (over all chromosomes or scaffolds or contigs) compared to the total number of nucleotides in the genome assembly file
- `genome_entropy`: The Shannon Entropy of the genome assembly file (median entropy over all individual chromosome entropies)
- `n_gc`: The total number of GCs (over all chromosomes or scaffolds or contigs) in the genome assembly file
- `rel_gc`: The (relative frequency) of GCs (over all chromosomes or scaffolds or contigs) compared to the total number of nucleotides in the genome assembly file

**Author(s)**

Hajk-Georg Drost

**See Also**

`getCollection, getCDS, read_cds, summary_genome`
See Also

summary_cds, getCollection, getGenome, read_genome

Examples

## Not run:
# retrieve genome from NCBI RefSeq
Sc <- biomartr::getGenome(db = "refseq", organism = "Saccharomyces cerevisiae")
# compute genome assembly summary statistics
Sc_genome_summary <- summary_genome(file = Sc, organism = "Saccharomyces cerevisiae")
# look at results
Sc_genome_summary

## End(Not run)
Index

* **biomaRt**
  biomart, 4
  getAttributes, 13
  getDatasets, 27
  getMarts, 48
  organismBM, 77
  organismFilters, 78

* **cachedir**
  cachedir, 6
  cachedir_set, 6

* **cds**
  getCDS, 19
  getCDSSet, 21
  read_cds, 80

* **collection**
  getCollection, 22
  getCollectionSet, 25

* **genome**
  getGenome, 33
  getGenomeSet, 36
  read_genome, 82

* **getBioSet**
  getBioSet, 16
  getCDSSet, 21
  getCollectionSet, 25
  getGenomeSet, 36
  getGFFSet, 40
  getProteomeSet, 53
  getRNASet, 60

* **getBio**
  getBio, 14
  getCDS, 19
  getCollection, 22
  getGenome, 33
  getGFF, 38
  getProteome, 51
  getRNA, 58

* **gff**
  getGFF, 38

getGFFSet, 40
read_gff, 83

* **meta_retrieval**
  meta.retrieval, 71
  meta.retrieval.all, 74

* **package**
  biomartr-package, 3

* **proteome**
  getProteome, 51
  getProteomeSet, 53
  read_proteome, 83

* **readers**
  read_cds, 80
  read_genome, 82
  read_gff, 83
  read_proteome, 83
  read_rna, 85

* **rna**
  getRNA, 58
  getRNASet, 60
  read_rna, 85

biomart, 4, 14, 27, 42, 43, 48, 76, 78, 79
biomartr (biomartr-package), 3
biomartr-package, 3

cachedir, 6, 7, 63
cachedir_set, 6, 6
check_annotation_biomartr, 7
data.frame, 73
download.database, 8, 9, 65, 66
download.database.all, 8, 9, 66

enssembl_divisions, 10, 10, 11, 32
get.ensembl.info, 10, 32, 47
getAssemblyStats, 11, 72, 74, 80
getAttributes, 5, 13, 27, 32, 43, 48, 78, 79
getBio, 14, 18, 20, 24, 35, 39, 45, 52, 60
getBioSet, 16, 22, 26, 37, 42, 55, 62
getBM, 5, 43
getCDS, 13, 16, 19, 22, 24, 35, 39, 44, 45, 47,
52, 58, 60, 72, 74, 81, 87
getCDSSet, 18, 20, 21, 26, 37, 42, 55, 62, 81
getCollection, 13, 16, 20, 22, 26, 35, 39, 45,
52, 58, 60, 87, 88
getCollectionSet, 18, 22, 24, 25, 37, 42, 55,
62
getDatasets, 5, 14, 27, 32, 43, 48, 78, 79
getENSEMBL, 28
getENSEMBL.gtf, 29
getENSEMBL.Seq, 30
getENSEMBLGENOMESInfo, 31
getENSEMBLInfo, 11, 31
getFilters, 32, 43
getGenome, 13, 16, 20, 24, 33, 37, 39, 44, 45,
47, 52, 58, 60, 72, 74, 82, 88
getGENOMEREPORT, 35
getGenomeSet, 18, 22, 26, 35, 36, 42, 55, 62,
82
getGFF, 13, 16, 20, 24, 35, 38, 42, 49, 52, 58,
60, 72, 74, 83
getGFFSet, 18, 22, 26, 37, 39, 40, 45, 55, 62,
83
getGO, 42
getGroups, 43, 47, 72
getGTF, 44, 72, 74
getKingdomAssemblySummary, 11, 32, 46, 50,
62
getKingdoms, 43, 44, 47, 62, 68, 72
getMarts, 4, 5, 14, 27, 32, 43, 48, 76, 78, 79
getMetaGenomeAnnotations, 48, 50
getMetaGenomes, 48, 49, 49, 70
getMetaGenomeSummary, 47, 50, 62, 70
getProteome, 13, 16, 20, 24, 35, 39, 44, 45,
47, 51, 55, 58, 60, 72, 74, 84
getProteomeSet, 18, 22, 26, 37, 42, 52, 53,
62, 84
getReleases, 56
getRepeatMasker, 56, 72, 74, 84, 85
getRNA, 13, 16, 20, 24, 35, 39, 45, 52, 58, 58,
62, 72, 74, 85
getRNASEt, 18, 22, 26, 37, 42, 55, 60, 60, 85
getSummaryFile, 47, 50, 62
getUniprotInfo, 63
getUniprotSTATS, 63

is.genome.available, 64, 69

listAttributes, 76
listDatabases, 8, 9, 65
listGenomes, 65, 66, 69
listGroups, 68, 69
listKingdoms, 69, 69
listMetaGenomes, 49, 50, 70
listNCBIDatabases, 9
listNCBIDatabases(listDatabases), 65

meta.retrieval, 13, 44, 47, 58, 71, 75
meta.retrieval.all, 73, 74

organismAttributes, 32, 75, 78
organismBM, 5, 14, 27, 32, 43, 48, 75, 76, 77,
78, 79
organismFilters, 5, 14, 27, 32, 43, 48, 76,
78, 78

read.fasta, 81, 82, 84, 85
read_assemblystats, 13, 80
read_cds, 20, 22, 80, 80, 82–85, 87
read_genome, 35, 37, 80, 81, 82, 83–85, 88
read_gff, 39, 42, 45, 80–82, 83, 84, 85
read_proteome, 52, 55, 80–83, 83, 85
read_rm, 58, 84
read_rna, 60, 62, 81–85, 85
refseqOrganisms, 86

summary_cds, 86, 88
summary_genome, 33, 87, 87

tempdir, 7, 15, 18, 24, 26, 39, 41, 45, 76, 77, 79