Package ‘biobtreeR’

July 11, 2025

Type Package

Title Using biobtree tool from R

Version 1.21.0

Author Tamer Gur

Maintainer Tamer Gur <`tgur@ebi.ac.uk`>

Description The biobtreeR package provides an interface to [biobtree](https://github.com/tamerh/biobtree) tool which covers large set of bioinformatics datasets and allows search and chain mappings functionalities.

License MIT + file LICENSE

Encoding UTF-8

LazyLoad yes

NeedsCompilation no

URL `https://github.com/tamerh/biobtreeR`

BugReports `https://github.com/tamerh/biobtreeR/issues`

VignetteBuilder knitr

Suggests BiocStyle, knitr,testthat,rmarkdown,markdown

Imports httr, httpuv, stringi,jsonlite,methods,utils

biocViews Annotation

RoxygenNote 6.1.1

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

git_branch devel

git_last_commit dc151e6

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-11

Contents

`bbBuildCustomDB` *Build custom DB*

Description

biobtree covers all the genomes in ensembl and ensembl genomes. If the the studied organism genome is not included in the default pre built in databases then this function is used and build the biobtree database locally for given genomes.

Usage

```
bbBuildCustomDB(taxonomyIDs = NULL, rawArgs = NULL)
```

Arguments

`taxonomyIDs`	Comma seperated list of taxonomy identifiers for building the genomes
`rawArgs`	For using all available biobtree command line arguments directly

Value

returns empty

Author(s)

Tamer Gur

Examples

```
## Not run:

bbUseOutDir("your directory path")
bbBuildCustomDB(taxonomyIDs="1408103,206403")


## End(Not run)
```

`bbBuiltInDB` *Get pre build biobtree database*

Description

Pre build biobtree database for commonly studied datasets and model organism genomes. Once this function called it retrieves the pre build database saves to users output directory.

Usage

```
bbBuiltInDB(type = "1")
```

Arguments

`type`	built in database type accepted values are 1,2,3 and 4. Currently there are 4 different builtin database; Type 1 Requires ~ 5 GB free storage Included datasets hgnc,hmdb,taxonomy,go,efo,eco,chebi,interpro Included uniprot proteins and ensembl genomes belongs to following organisms homo_sapiens 9606 –> ensembl danio_rerio 7955 zebrafish –> ensembl gallus_gallus 9031 chicken –> ensembl mus_musculus 10090 –> ensembl Rattus norvegicus 10116 —> ensembl saccharomyces_cerevisiae 4932–> ensembl,ensembl_fungi arabidopsis_thaliana 3702–> ensembl_plants drosophila_melanogaster 7227 –> ensembl,ensembl_metazoa caenorhabditis_elegans 6239 –> ensembl,ensembl_metazoa Escherichia coli 562 –> ensembl_bacteria Escherichia coli str. K-12 substr. MG1655 511145 –> ensembl_bacteria Escherichia coli K-12 83333 –> ensembl_bacteria Type 2 Requires ~ 5 GB free storage Instead of genomes in in the type 1 it contains human and all the mouse strains genomes with their uniprot proteins. In addition hgnc,hmdb,taxonomy,go,efo,eco,chebi,interpro datasets are included Type 3 Requires ~ 4 GB storage Contains no genome but it contains all the uniprot data with hgnc,hmdb,taxonomy,go,efo,eco,chebi,interpro Type 4 Requires ~ 13 GB storage Contains no genome but full uniprot and chembl data with hgnc,hmdb,taxonomy,go,efo,eco,chebi,interpro

Value

returns empty

Author(s)

Tamer Gur

Examples

```
bbUseOutDir(tempdir()) # temp dir for demo purpose
bbBuiltInDB("demo") # small demo database for real database use 1, 2, 3 or 4
```

`bbConfig-class` *Class for biobtreeR config*

Description

This class holds the datasets meta data and web service endpoints and used while executing the search/mapping queries. Instance of this class with name bbConfig is globally set by bbStart function. About dataset meta data, this class instance holds all the datasets unqiue identifers, entry url templates etc. In biobtree each dataset has unique character and numeric identifier. For instance Uniprot's charachter identifier is "uniprot" and numeric identifier is 1. When performing queries the dataset charachter identifier is used for convinience but in actual database it is saved numerically.

`bbEntry` *Retrieve entry*

Description

Returns entry for an identifier and dataset. Entry contains all the data raw data for and entry such as mappings, attiributes and paging info if exists.

Usage

```
bbEntry(identifer, source)
```

Arguments

`identifer`	Identifer for the entry. Note that keywords are not accepted. For instance insted of "vav_human" keyword "p15498" identifier must be passed
`source`	Dataset identifier

Value

returns biobtree json object

Author(s)

Tamer Gur

Examples

```
bbStart() # if not already started
bbEntry("HGNC:12009","hgnc")
```

`bbEntryFilter` *Retrieve entry with filtered dataset*

Description

Similar with entry retrieval but filtered mapping entries with given datasets.

Usage

```
bbEntryFilter(identifer, source, filters, page = NULL)
```

Arguments

`identifer`	Identifer for the entry.
`source`	Dataset identifier
`filters`	Comma seperated dataset identifer to retrieve
`page`	Page index if results is more than default biobtree paging size.

Value

returns biobtree json object

Author(s)

Tamer Gur

Examples

```
bbStart() # if not already started
bbEntryFilter("HGNC:12009","hgnc","uniprot,ensembl")
```

`bbEntryPage` *Retrieve entry result page*

Description

If an entry contains large set of mapping entries it is paginated by biobtree with confiGured paging size. This function retrieve these paging for an entry. Biobtree paging size for each entry is 200.

Usage

```
bbEntryPage(identifer, source, page, totalPage)
```

Arguments

`identifer`	Identifer for the entry.
`source`	Dataset identifier
`page`	Page index it starts from 0
`totalPage`	Total number of page for the entry. This value needs to calculate by user via using total number of entries which is available at the root result for the entry and divide it to the paging sizeb of 200

Value

returns biobtree json object

Author(s)

Tamer Gur

Examples

```
bbStart() # if not already started
bbEntryPage("ENSG00000141956","ensembl",0,0)
```

`bbListAttrs` *Retrieve attributes of dataset*

Description

Provides list of available attributes for a dataset to use in search and mapping queries.

Usage

```
bbListAttrs(dataset)
```

Arguments

`dataset`	Dataset identifier

Value

attributes names

Examples

```
bbListAttrs("hgnc")
bbListAttrs("ensembl")
```

`bbListDatasets` *List available datasets*

Description

Lists the available source and target datasets with their numeric identifiers.

Usage

```
bbListDatasets()
```

Value

returns datasets

Examples

```
bbListDatasets()
```

`bbMapping` *Chain mapping and filtering*

Description

Chain mapping identifiers or keywords with filtering and retrieving attributes if available.

Usage

```
bbMapping(terms, mapfilter, page = NULL, source = NULL, lite = TRUE,
  limit = 1000, inattrs = NULL, attrs = NULL,
  showInputColumn = FALSE)
```

Arguments

`terms`	Input terms for the mapping. Same with search functionality they can be comma seperated identifers or keywords
`mapfilter`	Mapping query which consist of map and optional filter functions in the form of map(dataset).filter(Boolean query expression) The boolean expressions are based on datasets attributes and dataset attributes can be list with bbAttr function. Dataset attributes which used in the filters starts with their dataset name. In biobtree boolean expressions feature has been implemented via Google common expression language so its full capability can be checked in its documentation.
`page`	Optional parameter works similar with bbSearch page paramter.
`source`	Optional dataset identifiers for searching input terms within the given dataset.
`lite`	By default it is TRUE and allow function return quickly with data.frame with mapping identifiers and attributes. If set to TRUE function return raw results converted from json.

`limit`	Limits the number of mapping results. By default without any limit all the results returned.
`inattrs`	Optional comma seperated attribute names for input identifiers and if available their values includes in result data.frame
`attrs`	Optional comma seperated attribute names for mapping identifiers and if available their values includes in result data.frame
`showInputColumn`	Optional logical parameter to show the input identifers in the result data.frame

Value

returns mapping results in data.frame by default if lite set it true returns json object

Author(s)

Tamer Gur

Examples

```
bbStart()

bbMapping("tpi1",'map(uniprot)')

bbMapping("shh",'map(ensembl)')

## Not run:
# run these examples with building the default dataset with bbBuildData()
#Map protein to its go terms and retrieve go term types
bbMapping("AT5G3_HUMAN",'map(go)',attrs = "type")

#Map protein to its go terms with filter by its type and retrieve their types
bbMapping("AT5G3_HUMAN",'map(go).filter(go.type=="biological_process")',attrs = "type")

#Map gene names to exon identifiers and retrieve the region
bbMapping("ATP5MC3,TP53",'map(transcript).map(exon)',attrs = "seq_region_name")

#Map Affymetrix identifiers to Ensembl identifiers and gene names
bbMapping("202763_at,213596_at,209310_s_at",source ="affy_hg_u133_plus_2"
,'map(transcript).map(ensembl)',attrs = "name")


## End(Not run)
```

`bbSearch` *Search identifiers or special keywords*

Description

Search identifiers or special keywords terms uniformly and resolve their actual unique identifiers and datasets. Keywords can be several things for instance for uniprot an accession like "vav_human" can be a keyword which points to its original identifier "P15498". Or gene name can be also a keyword like "tpi1" which could points multiple dataset like ensembl and hgnc.

Usage

```
bbSearch(terms, source = NULL, filter = NULL, page = NULL,
  limit = 1000, showURL = FALSE, lite = TRUE)
```

Arguments

`terms`	Comma seperated identifers or keywords
`source`	Optional dataset identifiers to search only within this dataset.
`filter`	Filter expression useful to filter out results when a keyword point several results. For instance if the biobtree index with multiple organism a same gene search could hit several results for different species to filter only a specific species a filter can apply to search function.
`page`	By default no need to pass this parameter since it returns all the results. It can be used with limit parameter for very large results to process them in paginated manner. About paging every long search or mapping result paginated in biobtree and for paginated results every response contains a key to get the next page results. So if this parameter is set with this key specified next page results returned for the given search term.
`limit`	Limits the number of search results. By default without any limit all the results returned.
`showURL`	allows returning the dataset source url
`lite`	By default it is TRUE and allow function return quickly with data.frame containing most important fields. If set to TRUE function return raw results converted from json.

Value

returns search results in data.frame by default if lite set it true returns json object

Author(s)

Tamer Gur

Examples

```
bbSearch("hunk,vav_human")

bbSearch("hunk","ensembl",filter='ensembl.genome=="homo_sapiens"')
```

`bbStart` *Start biobtreeR*

Description

Once target datasets is built with `bbBuildData` this function used to start biobtree server in the background for performing search/mapping queries.

Usage

```
bbStart()
```

Value

character

Examples

```
bbStart()
bbStop()
```

`bbStop` *Stop biobtree*

Description

Stops running background biobtree process which started with `bbStart`

Usage

```
bbStop()
```

Value

returns empty

Examples

```
bbStop()
```

`bbUseOutDir` *Output directory for biobtreeR*

Description

Allows to set the directory for the package for its files. It is required to set a valid directory.

Usage

```
bbUseOutDir(outDir)
```

Arguments

`outDir`	path for the output directory.

Value

returns empty

Examples

```
bbUseOutDir(tempdir())
```

Index