Package 'cBioPortalData'

January 27, 2021

Title Exposes and makes available data from the cBioPortal web resources

Version 2.2.6

Description The cBioPortalData package takes compressed resources from repositories such as cBioPortal and assembles a MultiAssayExperiment object with Bioconductor classes.

Depends R (>= 4.0.0), AnVIL, MultiAssayExperiment

Imports BiocFileCache (>= 1.5.3), digest, dplyr, GenomeInfoDb, GenomicRanges, httr, IRanges, methods, readr, RaggedExperiment, RTCGAToolbox (>= 2.19.7), S4Vectors, SummarizedExperiment, stats, tibble, tidyr, TCGAutils (>= 1.9.4), utils

Suggests BiocStyle, knitr, testthat

License AGPL-3

Encoding UTF-8

LazyData true

VignetteBuilder knitr

BugReports https://github.com/waldronlab/cBioPortalData/issues

biocViews Software, Infrastructure, ThirdPartyClient

RoxygenNote 7.1.1

Collate 'utils.R' 'cBioDataPack.R' 'cBioPortal.R' 'cBioPortalData-pkg.R' 'cBioPortalData.R' 'cache.R' 'data.R'

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

git_branch RELEASE_3_12

git_last_commit ee55be4

git_last_commit_date 2021-01-20

Date/Publication 2021-01-26

Author Levi Waldron [aut], Marcel Ramos [aut, cre]

Maintainer Marcel Ramos <marcel.ramos@roswellpark.org>

R topics documented:

`cBioCache` *Manage cache / download directories for study data*

Description

Managing data downloads is important to save disk space and re-downloading data files. This can be done effortlessly via the integrated BiocFileCache system.

Usage

```
cBioCache(...)

setCache(
  directory = tools::R_user_dir("cBioPortalData", "cache"),
  verbose = TRUE,
  ask = interactive()
)

removePackCache(cancer_study_id, dry.run = TRUE)

removeDataCache(
  api,
  studyId = NA_character_,
  genePanelId = NA_character_,
  molecularProfileIds = NULL,
  sampleListId = NULL,
  sampleIds = NULL,
  dry.run = TRUE,
  ...
)
```

Arguments

`...`	For `cBioCache`, arguments passed to `setCache`
`directory`	The file location where the cache is located. Once set future downloads will go to this folder.
`verbose`	Whether to print descriptive messages
`ask`	logical (default TRUE when interactive session) Confirm the file location of the cache directory

cancer_study_id
: A single string from studiesTable associated with a study tarball

dry.run
: logical Whether or not to remove cache files (default TRUE).

api
: An API object of class ‘cBioPortal‘ from the ‘cBioPortal‘ function

studyId
: character(1) Indicates the "studyId" as taken from ‘getStudies‘

genePanelId
: character(1) Identifies the gene panel, as obtained from the ‘genePanels‘ function

molecularProfileIds
: character() A vector of molecular profile IDs

sampleListId
: character(1) A sample list identifier as obtained from ‘sampleLists()“

sampleIds
: character() Sample identifiers

Value

cBioCache: The path to the cache location

cBioCache

Get the directory location of the cache. It will prompt the user to create a cache if not already created. A specific directory can be used via setCache.

setCache

Specify the directory location of the data cache. By default, it will go to the user directory as given by:

```
tools::R_user_dir("cBioPortalData", "cache")
```

removePackCache

Some files may become corrupt when downloading, this function allows the user to delete the tarball associated with a cancer_study_id in the cache. This only works for the cBioDataPack function. To remove the entire cBioPortalData cache, run unlink("~/.cache/cBioPortalData").

Examples

```
cBioCache()

removePackCache("acc_tcga", dry.run = TRUE)


cbio <- cBioPortal()

cBioPortalData(
    cbio, by = "hugoGeneSymbol",
    studyId = "acc_tcga",
    genePanelId = "AmpliSeq",
    molecularProfileIds =
        c("acc_tcga_rppa", "acc_tcga_linear_CNA", "acc_tcga_mutations")
)

removeDataCache(
```

```
    cbio,
    studyId = "acc_tcga",
    genePanelId = "AmpliSeq",
    molecularProfileIds =
        c("acc_tcga_rppa", "acc_tcga_linear_CNA", "acc_tcga_mutations"),
    dry.run = TRUE
)
```

`cBioDataPack` *Obtain pre-packaged data from cBioPortal and represent as a MultiAssayExperiment object*

Description

The `cBioDataPack` function allows the user to download and process cancer study datasets found in MSKCC's cBioPortal. Output datasets use the MultiAssayExperiment data representation to faciliate analysis and data management operations.

Usage

```
cBioDataPack(
  cancer_study_id,
  use_cache = TRUE,
  names.field = c("Hugo_Symbol", "Entrez_Gene_Id", "Gene"),
  ask = TRUE
)
```

Arguments

`cancer_study_id`	character(1) The study identifier from cBioPortal as in `https://cbioportal.org/webAPI`
`use_cache`	logical(1) (default TRUE) create the default cache location and use it to track downloaded data. If data found in the cache, data will not be re-downloaded. A path can also be provided to data cache location.
`names.field`	A character vector of possible column names for the column that is used to label ranges from a mutations or copy number file.
`ask`	A logical vector of length one indicating whether to prompt the the user before downloading and loading study `MultiAssayExperiment`. If TRUE, the user will be prompted to continue for studies that are not currently building as `MultiAssayExperiment` based on previous testing (in a non-interactive session, no data will be downloaded and built unless `ask = FALSE`).

Details

The list of datasets can be found in the `studiesTable` dataset by doing `data("studiesTable")`. Some datasets may not be available for download and are not guaranteed to be represented as MultiAssayExperiment data objects. After taking a random sample of 100 (using `set.seed(1234)`), we were able to succesfully represent about 76 percent of the study identifiers as MultiAssayExperiment objects. Please refer to the #' website for the full list of available datasets. Users who would

like to prioritize particular datasets should open GitHub issues at the URL in the DESCRIPTION file. For a more fine-grained approach to downloading data from the cBioPortal API, refer to the cBioPortalData function.

Value

A MultiAssayExperiment object

cBio_URL

The cBioDataPack function accesses data from the cBio_URL option. By default, it points to an Amazon S3 bucket location. Previously, it pointed to 'http://download.cbioportal.org'. This recent change (> 2.1.17) should provide faster and more reliable downloads for all users. See the URL using cBioPortalData:::.url_location. This can be changed if there are mirrors that host this data by setting the cBio_URL option with getOption("cBio_URL","https://some.url.com/") before running the function.

Author(s)

Levi Waldron, Marcel R., Ino dB.

See Also

https://www.cbioportal.org/datasets, cBioPortalData

Examples

```
data(studiesTable)

head(studiesTable[["cancer_study_id"]])

# ask=FALSE for non-interactive use
mae <- cBioDataPack("acc_tcga", ask = FALSE)
```

---

cBioPortal *The R interface to the cBioPortal API Data Service*

Description

This section of the documentation lists the functions that allow users to access the cBioPortal API. The main representation of the API can be obtained from the 'cBioPortal' function. The supporting functions listed here give access to specific parts of the API and allow the user to explore the API with individual calls. Many of the functions here are listed for documentation purposes and are recommended for advanced usage only. Users should only need to use the 'cBioPortalData' main function to obtain data.

Usage

```
cBioPortal(
  hostname = "www.cbioportal.org",
  protocol = "https",
  api. = "/api/api-docs"
)

getStudies(api)

clinicalData(api, studyId = NA_character_)

molecularProfiles(
  api,
  studyId = NA_character_,
  projection = c("SUMMARY", "ID", "DETAILED", "META")
)

mutationData(
  api,
  molecularProfileIds = NA_character_,
  entrezGeneIds = NULL,
  sampleIds = NULL
)

molecularData(
  api,
  molecularProfileIds = NA_character_,
  entrezGeneIds = NULL,
  sampleIds = NULL
)

searchOps(api, keyword)

geneTable(api, pageSize = 1000, pageNumber = 0, ...)

samplesInSampleLists(api, sampleListIds = NA_character_)

sampleLists(api, studyId = NA_character_)

allSamples(api, studyId = NA_character_)

genePanels(api)

getGenePanel(api, genePanelId = NA_character_)

genePanelMolecular(
  api,
  molecularProfileId = NA_character_,
  sampleListId = NULL,
  sampleIds = NULL
)
```

```
getGenePanelMolecular(api, molecularProfileIds = NA_character_, sampleIds)

getSampleInfo(
  api,
  studyId = NA_character_,
  sampleListIds = NULL,
  projection = c("SUMMARY", "ID", "DETAILED", "META")
)

getDataByGenePanel(
  api,
  studyId = NA_character_,
  genePanelId = NA_character_,
  molecularProfileIds = NULL,
  sampleListId = NULL,
  sampleIds = NULL
)
```

Arguments

hostname	character(1) The internet location of the service (default: 'www.cbioportal.org')
protocol	character(1) The internet protocol used to access the hostname (default: 'https')
api.	character(1) The directory location of the API protocol within the hostname (default: '/api/api-docs')
api	An API object of class 'cBioPortal' from the 'cBioPortal' function
studyId	character(1) Indicates the "studyId" as taken from 'getStudies'
projection	character(default: "SUMMARY") Specify the projection type for data retrieval for details see API documentation
molecularProfileIds	character() A vector of molecular profile IDs
entrezGeneIds	numeric() A vector indicating entrez gene IDs
sampleIds	character() Sample identifiers
keyword	character(1) Keyword or pattern for searching through available operations
pageSize	numeric(1) The number of rows in the table to return
pageNumber	numeric(1) The pagination page number
...	Additional arguments to lower level API functions
sampleListIds	character() A vector of 'sampleListId' as obtained from 'sampleLists'
genePanelId	character(1) Identifies the gene panel, as obtained from the 'genePanels' function
molecularProfileId	character(1) Indicates a molecular profile ID
sampleListId	character(1) A sample list identifier as obtained from 'sampleLists()"

Value

cBioPortal: An API object of class 'cBioPortal'

cBioPortalData: A data object of class 'MultiAssayExperiment'

API Metadata

* getStudies - Obtain a table of studies and associated metadata

* searchOps - Search through API operations with a keyword

* geneTable - Get a table of all genes by 'entrezGeneId' or 'hugoGeneSymbol'

* sampleLists - obtain all 'sampleListIds' for a particular 'studyId'

* allSamples - obtain all samples within a particular 'studyId'

* genePanels - Show all available gene panels

Patient Data

* clinicalData - Obtain clinical data for a particular study identifier ('studyId')

Molecular Profiles

* molecularProfiles - Produce a molecular profiles dataset for a given study identifier ('studyId')

* molecularData - Produce a dataset of molecular profile data based on 'molecularProfileId', 'entrezGeneIds', and 'sampleIds'

Mutation Data

* mutationData - Produce a dataset of mutation data using 'molecularProfileId', 'entrezGeneIds', and 'sampleIds'

Sample Data

* samplesInSampleLists - get all samples associated with a 'sampleListId'

* getSampleInfo - Obtain sample metadata for a particular 'studyId' or 'sampleListId'

Gene Panels

* getGenePanels - Obtain the gene panel for a particular 'genePanelId'

* genePanelMolecular - get gene panel data for a paricular 'molecularProfileId' and 'sampleListId' combination

* getGenePanelMolecular - get gene panel data for a combination of 'molecularProfileId' and 'sampleListId' vectors

* getDataByGenePanel - Download data for a gene panel and 'molecularProfileId' combination, optionally a 'sampleListId' can be provided.

Examples

```
cbio <- cBioPortal()

getStudies(api = cbio)

searchOps(api = cbio, keyword = "molecular")

## obtain clinical data
acc_clin <- clinicalData(api = cbio, studyId = "acc_tcga")
acc_clin

molecularProfiles(api = cbio, studyId = "acc_tcga")
```

```
genePanels(cbio)

(gp <- getGenePanel(cbio, "AmpliSeq"))

muts <- mutationData(
    api = cbio,
    molecularProfileIds = "acc_tcga_mutations",
    entrezGeneIds = 1:1000,
    sampleIds = c("TCGA-OR-A5J1-01", "TCGA-OR-A5J2-01")
)
exps <- molecularData(
    api = cbio,
    molecularProfileIds = c("acc_tcga_rna_seq_v2_mrna", "acc_tcga_rppa"),
    entrezGeneIds = 1:1000,
    sampleIds = c("TCGA-OR-A5J1-01", "TCGA-OR-A5J2-01")
)

sampleLists(api = cbio, studyId = "acc_tcga")

samplesInSampleLists(
    api = cbio,
    sampleListIds = c("acc_tcga_rppa", "acc_tcga_cnaseq")
)

genePanels(api = cbio)

getGenePanel(api = cbio, genePanelId = "IMPACT341")


getDataByGenePanel(cbio, studyId = "acc_tcga", genePanelId = "IMPACT341",
   molecularProfileId = "acc_tcga_rppa", sampleListId = "acc_tcga_rppa")
```

`cBioPortal-class` *A class for representing the cBioPortal API protocol*

Description

The ‘cBioPortal‘ class is a representation of the cBioPortal API protocol that directly inherits from the ‘Service‘ class in the ‘AnVIL‘ package. For more information, see the ’AnVIL’ package.

Details

This class takes the static API as provided at https://www.cbioportal.org/api/api-docs and creates an R object with the help from underlying infrastructure (i.e., ’rapiclient’ and ’AnVIL’) to give the user a unified representation of the API specification provided by the cBioPortal group. Users are not expected to interact with this class other than to use it as input to the functionality provided by the rest of the package.

See Also

cBioPortal, Service

Examples

```
cBioPortal()
```

`cBioPortalData` *Download data from the cBioPortal API*

Description

Obtain a `MultiAssayExperiment` object for a particular gene panel, `studyId`, `molecularProfileIds`, and `sampleListIds` combination. Default `molecularProfileIds` and `sampleListIds` are set to NULL for including all data. This option is best for users who wish to obtain a section of the study data that pertains to a specific molecular profile and gene panel combination. For users looking to download the entire study data as provided by the https://cbioportal.org/datasets, refer to `cBioDataPack`.

Usage

```
cBioPortalData(
  api,
  studyId = NA_character_,
  genePanelId = NA_character_,
  molecularProfileIds = NULL,
  sampleListId = NULL,
  by = c("entrezGeneId", "hugoGeneSymbol")
)
```

Arguments

`api`	An API object of class ‘cBioPortal‘ from the ‘cBioPortal‘ function
`studyId`	character(1) Indicates the "studyId" as taken from ‘getStudies‘
`genePanelId`	character(1) Identifies the gene panel, as obtained from the ‘genePanels‘ function
`molecularProfileIds`	character() A vector of molecular profile IDs
`sampleListId`	character(1) A sample list identifier as obtained from ‘sampleLists()“
`by`	character(1) Either ’entrezGeneId’ or ’hugoGeneSymbol’ for row metadata

Details

As of May 2020, there were about 96.6 percent of the 268 datasets successfully imported. The datasets that currently fail to import are:

```
c("all_stjude_2015", "sclc_ucologne_2015", "skcm_ucla_201
"sclc_jhu", "gbm_tcga_pub2013", "hnsc_tcga_pub", "kirc_tc
"brca_tcga_pub", "brca_tcga_pub2015")
```

Note that changes to the cBioPortal API may affect this rate at any time. If you encounter any issues, please open a GitHub issue at the https://github.com/waldronlab/cBioPortalData/issues/ page with a fully reproducible example.

Value

A MultiAssayExperiment object

See Also

cBioDataPack

Examples

```
cbio <- cBioPortal()

samps <- samplesInSampleLists(cbio, "acc_tcga_rppa")[[1]]

getGenePanelMolecular(
    cbio, molecularProfileIds = c("acc_tcga_rppa", "acc_tcga_linear_CNA"),
    samps
)

acc_tcga <- cBioPortalData(
    cbio, by = "hugoGeneSymbol",
    studyId = "acc_tcga",
    genePanelId = "AmpliSeq",
    molecularProfileIds =
        c("acc_tcga_rppa", "acc_tcga_linear_CNA", "acc_tcga_mutations")
)
```

---

downloadStudy *Manually download, untar, and load study tarballs*

Description

Note that these functions should be used when a particular study is *not* currently available as a `MultiAssayExperiment` representation. Otherwise, use `cBioDataPack`. Provide a `cancer_study_id` from the `studiesTable` and retrieve the study tarball from cBioPortal. These functions are used by `cBioDataPack` under the hood to download, untar, and load the tarball datasets with caching. As stated in `?cBioDataPack`, not all studies are currently working as `MultiAssayExperiment` objects. As of July 2020, about ~80% of datasets can be successfully imported into the `MultiAssayExperiment` data class. Please open an issue if you would like the team to prioritize a study. You may also check `studiesTable$pack_build` for a more current status.

Usage

```
downloadStudy(
  cancer_study_id,
  use_cache = TRUE,
  force = FALSE,
  url_location = getOption("cBio_URL", .url_location)
)

untarStudy(cancer_study_file, exdir = tempdir())

loadStudy(filepath, names.field = c("Hugo_Symbol", "Entrez_Gene_Id", "Gene"))
```

Arguments

cancer_study_id	character(1) The study identifier from cBioPortal as in https://cbioportal.org/webAPI
use_cache	logical(1) (default TRUE) create the default cache location and use it to track downloaded data. If data found in the cache, data will not be re-downloaded. A path can also be provided to data cache location.
force	logical(1) (default FALSE) whether to force re-download data from remote location
url_location	character(1) (default "https://cbioportal-datahub.s3.amazonaws.com") the URL location for downloading packaged data. Can be set using the 'cBio_URL' option (see ?cBioDataPack for more details)
cancer_study_file	character(1) indicates the on-disk location of the downloaded tarball
exdir	character(1) indicates the folder location to *put* the contents of the tarball (default tempdir(); see also ?untar)
filepath	character(1) indicates the folder location where the contents of the tarball are *located* (usually the same as exdir)
names.field	A character vector of possible column names for the column that is used to label ranges from a mutations or copy number file.

Value

- downloadStudy - The file location of the data tarball
- untarStudy - The directory location of the contents
- loadStudy - A MultiAssayExperiment-class object

See Also

cBioDataPack, MultiAssayExperiment

Examples

```
(acc_file <- downloadStudy("acc_tcga"))

(file_dir <- untarStudy(acc_file, tempdir()))

loadStudy(file_dir)
```

---

studiesTable *A list of available studies from the cBioPortal data repository*

Description

A list of available studies from the cBioPortal data repository

Usage

```
studiesTable
```

Format

A data frame with 220 rows and 4 variables:

cancer_study_id The study code used for input to ‘cBioDataPack‘

study_name A descriptive study title containing data center and year

description A longer description of the study

URL Associated study URLs

Author(s)

Marcel Ramos <marcel.ramos@roswellpark.org>

References

http://www.cbioportal.org/datasets, https://github.com/cBioPortal/cgdsr

Index