

Package ‘BiocOncoTK’

October 15, 2018

Title Bioconductor components for general cancer genomics

Description Provide a central interface to various tools for genome-scale analysis of cancer studies.

Version 1.0.3

Author Vince Carey

Suggests knitr, dbplyr, DBI, org.Hs.eg.db, MultiAssayExperiment, BiocStyle, ontoProc, ontologyPlot, pogos, ggplot2, GenomeInfoDb, restfulSE, BiocFileCache, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg18.knownGene

Imports ComplexHeatmap, S4Vectors, bigquery, shiny, httr, rjson, dplyr, magrittr, grid, utils, DT

Depends R (>= 3.5.0), methods

Maintainer VJ Carey <stvjc@channing.harvard.edu>

License Artistic-2.0

LazyLoad yes

LazyData yes

biocViews CopyNumberVariation, CpGIIsland, DNAMethylation, GeneExpression, GeneticVariability, SNP, Transcription

VignetteBuilder knitr

RoxygenNote 6.0.1

git_url <https://git.bioconductor.org/packages/BiocOncoTK>

git_branch RELEASE_3_7

git_last_commit 5f05421

git_last_commit_date 2018-08-24

Date/Publication 2018-10-15

R topics documented:

annotTabs	2
brcaMAE	3
CCLC_DRUG_BROAD	4
cell_70138	4
clueDemos	5

clueServiceNames	5
darmGBMcls	6
icd10_c	6
loadPatel	7
load_ccleNRAS	7
oncoPrintISB	8
pancan.clin.varnames	9
pancan_app	9
pancan_BQ	10
pancan_clinicalTabVarnames	10
pancan_longname	11
pancan_sampTypeMap	11
pancan_tabulate	12
pertClasses	12
pert_70138	13
query_clue	13
TcgaMutCounts	14
TcgaNIndWithAnyMut	15
Index	16

annotTabs	<i>table names in Annotated pancancer data release</i>
-----------	--------------------------------------------------------

Description

table names in Annotated pancancer data release

Usage

annotTabs

Format

character vector

Source

pancancer-atlas in BigQuery

Examples

```
BiocOncoTK::annotTabs
```

brcaMAE

*a virtual MultiAssayExperiment for pancancer-atlas BRCA data***Description**

a virtual MultiAssayExperiment for pancancer-atlas BRCA data

Usage

```
brcaMAE
```

Format

MultiAssayExperiment instance with DelayedArray (BQ3_Array) assay data

Note

Constructed as

```
library(BiocOncoTK)
pcbq = pancan_BQ()
library(restfulSE)
BRCA_mir = pancan_SE(pcbq)
BRCA_mrna = pancan_SE(pcbq,
  assayDataTableName = pancan_longname("rnaseq"),
  assayFeatureName = "Entrez",
  assayValueFieldName = "normalized_count")
BRCA_rppa = pancan_SE(pcbq,
  assayDataTableName = pancan_longname("RPPA"),
  assayFeatureName = "Protein",
  assayValueFieldName = "Value")
BRCA_meth = pancan_SE(pcbq,
  assayDataTableName = pancan_longname("27k")[2],
  assayFeatureName = "ID",
  assayValueFieldName = "Beta")
library(MultiAssayExperiment)
library(dplyr)
library(magrittr)
clinBRCA = pcbq
  filter(acronym=="BRCA")
rownames(clinBRCA) = clinBRCA[,2]
clinDF = DataFrame(clinBRCA)
library(MultiAssayExperiment)
brcaMAE = MultiAssayExperiment(
  ExperimentList(rnaseq=BRCA_mrna, meth=BRCA_meth, rppa=BRCA_rppa,
    mirna=BRCA_mir),colData=clinDF)
upsetSamples(brcaMAE) # to view display
```

Source

ISB BigQuery pancan-atlas project

Examples

```
if (requireNamespace("MultiAssayExperiment"))
  BiocOncoTK::brcaMAE
```

CCLE_DRUG_BROAD	<i>CCLE_DRUG_BROAD: serialization of legacy CCLE 'Drug data' from Broad Institute</i>
-----------------	---------------------------------------------------------------------------------------

Description

CCLE_DRUG_BROAD: serialization of legacy CCLE 'Drug data' from Broad Institute

Usage

```
CCLE_DRUG_BROAD
```

Format

S4Vectors DataFrame instance

Source

["https://data.broadinstitute.org/ccle_legacy_data/pharmacological_profiling/CCLE_NP24.2009_Drug_data_2015.02.24.csv"](https://data.broadinstitute.org/ccle_legacy_data/pharmacological_profiling/CCLE_NP24.2009_Drug_data_2015.02.24.csv)

Examples

```
data(CCLE_DRUG_BROAD)
requireNamespace("S4Vectors")
S4Vectors::metadata(CCLE_DRUG_BROAD) # imported using read.csv, stringsAsFactors=FALSE, coerced to DataFrame
head(CCLE_DRUG_BROAD)
```

cell_70138	<i>cell_70138: a table with cell-line information from LINCS</i>
------------	------------------------------------------------------------------

Description

cell_70138: a table with cell-line information from LINCS

Usage

```
cell_70138
```

Format

data.frame

Source

GEO GSE70138 GSE70138_Broad_LINCS_cell_info_2017-04-28.txt.gz

Examples

```
data(cell_70138)
```

clueDemos	<i>generate lists to generate clue API queries</i>
-----------	----------------------------------------------------

Description

generate lists to generate clue API queries

Usage

```
clueDemos()
```

Value

a list of lists of strings with 'where' and substructure as appropriate

Note

These are converted to JSON (

Examples

```
clueDemos()
```

clueServiceNames	<i>Provide names of some clue.io services for which examples are available in this package.</i>
------------------	-------------------------------------------------------------------------------------------------

Description

Provide names of some clue.io services for which examples are available in this package.

Usage

```
clueServiceNames()
```

Value

a character vector of service names

Note

See <https://clue.io/api>.

Examples

```
clueServiceNames()
```

darmGBMcls	<i>Data in count_1stpm format from Darmanis 2017 (PMC 5810554) single cell RNA-seq in GBM</i>
------------	-----------------------------------------------------------------------------------------------

Description

Data in count_1stpm format from Darmanis 2017 (PMC 5810554) single cell RNA-seq in GBM

Usage

```
darmGBMcls
```

Format

SummarizedExperiment with HDF Object store back end

Note

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5810554/> is the main source article.

Source

http://imlspenticton.uzh.ch/robinson_lab/conquer/data-mae/GSE84465.rds

Examples

```
BiocOncoTK::darmGBMcls
```

icd10_c	<i>helper for interpreting ICD-10 codes</i>
---------	---------------------------------------------

Description

helper for interpreting ICD-10 codes

Usage

```
icd10_c
```

Format

data.frame

Source

ICD-10

Examples

```
BiocOncoTK::icd10_c
```

loadPatel	<i>use BiocFileCache discipline to acquire patelGBMSC SummarizedExperiment</i>
-----------	--------------------------------------------------------------------------------

Description

use BiocFileCache discipline to acquire patelGBMSC SummarizedExperiment

Usage

```
loadPatel(remotePath = "https://s3.us-east-2.amazonaws.com/biocfound-scrna/patelGBMSC.rds",
  cache = BiocFileCache::BiocFileCache())
```

Arguments

remotePath	character(1) identifying remote RDS
cache	instance of BiocFileCache, defaults to BiocFileCache::BiocFileCache()

Value

a SummarizedExperiment instance

Note

The RDS for the SummarizedExperiment is in an AWS S3 bucket. This function will check local cache for the data and will download to cache if not found. That download is a one-time operation for any given value of cache.

Examples

```
loadPatel
```

load_cc1eNRAS	<i>utilities for mock data (not involving internet access for vignette)</i>
---------------	-----------------------------------------------------------------------------

Description

utilities for mock data (not involving internet access for vignette)

Usage

```
load_cc1eNRAS()
```

```
load_NRAS_AHR()
```

```
load_nrasdf()
```

Value

a list of DRProfSet instances

a data.frame with fields 'Cell_line_primary_name', 'RMA_normalized_expression', 'HGNC_gene_symbol'

a data.frame

Note

These functions are provided only for avoiding reliance on internet connectivity for document production.

Examples

```
load_ccleNRAS()
dim(load_nrasdf())
```

oncoPrintISB

interactive interface to ComplexHeatmap oncoPrint with inputs from ISB Cancer Genomics Cloud BigQuery back end

Description

interactive interface to ComplexHeatmap oncoPrint with inputs from ISB Cancer Genomics Cloud BigQuery back end

Usage

```
oncoPrintISB(bq)
```

Arguments

bq an instance of [BigQueryConnection-class](#) authenticated for ISB Cancer Genomics Cloud access

Value

only used for side effect of running shiny app

Note

This function will start a shiny app and will generate queries to Google BigQuery tables representing TCGA.

Examples

```
if (interactive()) {
  bcode = Sys.getenv("CGC_BILLING")
  if (nchar(bcode)>0) {
    con <- DBI::dbConnect(bigrquery::dbi_driver(), project = "isb-cgc",
      dataset = "tcga_201607_beta", billing = bcode)
    oncoPrintISB(con)
  }
}
```

pancan.clin.varnames *pancan.clin.varnames: a data.frame with a list of variable names for clinical patient data*

Description

pancan.clin.varnames: a data.frame with a list of variable names for clinical patient data

Usage

```
pancan.clin.varnames
```

Format

data.frame

Source

pancancer-atlas in BigQuery

Examples

```
BiocOncoTK::pancan.clin.varnames[1:5,]
```

pancan_app *provide a shiny app to 'glimpse' structure and content of pancan atlas*

Description

provide a shiny app to 'glimpse' structure and content of pancan atlas

Usage

```
pancan_app(dataset = "Annotated", nrecs = 5)
```

Arguments

dataset character(1) name of a BigQuery dataset in the pancan-atlas project
nrecs numeric(1) number of records to request (limited through the n= parameter to as.data.table)

Value

currently only as a side effect of starting app

Examples

```
if (interactive()) pancan_app()
```

pancan_BQ *provide bigquery connection to pancancer Annotated datasets*

Description

provide bigquery connection to pancancer Annotated datasets

Usage

```
pancan_BQ(dataset = "Annotated", billing = Sys.getenv("CGC_BILLING"))
```

Arguments

dataset	character(1) dataset name
billing	character(1) Google cloud platform billing code; authentication will be attempted when using the resulting connection

Value

BigQueryConnection instance

Examples

```
pancan_BQ
```

pancan_clinicalTabVarnames
give an interface to tablenamees

Description

give an interface to tablenamees

Usage

```
pancan_clinicalTabVarnames()
```

Value

interactive datatable from DT

Examples

```
if (interactive()) pancan_clinicalTabVarnames()
```

pancan_longname	<i>utility to help find long table names</i>
-----------------	----------------------------------------------

Description

utility to help find long table names

Usage

```
pancan_longname(guess, ...)
```

Arguments

guess	a regexp to match the table of interest
...	passed to agrep

Value

character vector of matches

Note

Note that `ignore.case=TRUE` is set in the function.

Examples

```
pancan_longname("rnaseq")
```

pancan_sampTypeMap	<i>helper for interpreting pancan-atlas sample type codes</i>
--------------------	---------------------------------------------------------------

Description

helper for interpreting pancan-atlas sample type codes

Usage

```
pancan_sampTypeMap
```

Format

data.frame

Note

The sample type codes are not straightforward to interpret. Primary solid tumor is denoted "TP", and metastatic samples are denoted "TM". This data frame pairs code and natural language terms.

Source

ISB BigQuery pancan-atlas project

Examples

```
BiocOncoTK::pancan_sampTypeMap
```

```
pancan_tabulate      tabulate a variable in a table
```

Description

tabulate a variable in a table

Usage

```
pancan_tabulate(dataset = "Annotated", tblname, vblname)
```

Arguments

dataset	character(1) dataset name
tblname	character(1) table name in dataset
vblname	character(1) field name in table

Value

instance of `tbl_dbi`, constituting summarise result

Examples

```
if (interactive()) pancan_tabulate(tblname=
  "clinical_PANCAN_patient_with_followup", vblname="icd_10")
```

```
pertClasses      enumerate perturbation classes
```

Description

enumerate perturbation classes

Usage

```
pertClasses(key = Sys.getenv("CLUE_KEY"))
```

Arguments

key	character(1) API key provided by clue.io
-----	------------------------------------------

Value

a character vector

Examples

```
if (nchar(Sys.getenv("CLUE_KEY"))>0) {
  pc = pertClasses()
  head(vapply(pc, "[", character(1), 1))
}
```

pert_70138

pert_70138: a table with perturbagen information from LINCS

Description

pert_70138: a table with perturbagen information from LINCS

Usage

```
pert_70138
```

Format

```
data.frame
```

Source

```
GEO GSE70138 GSE70138_Broad_LINCS_pert_info.txt.gz
```

Examples

```
data(pert_70138)
```

query_clue

run the api.clue.io API to acquire information on LINCS experiments

Description

run the api.clue.io API to acquire information on LINCS experiments

Usage

```
query_clue(service = "profiles", filter = list(where = (list(pert_iname =
  "sirolimus", cell_id = "MCF7", assay = "L1000"))),
  key = Sys.getenv("CLUE_KEY"))
```

Arguments

```
service      a character(1) service name
filter       a list to be converted to JSON for submission as a GET request
key          character(1) API key provided by clue.io
```

Value

API return value processed by fromJSON

Examples

```
if (nchar(Sys.getenv("CLUE_KEY"))>0) {
  demos = clueDemos()
  nd = length(demos)
  chk = lapply(seq_len(nd), function(x) query_clue( service=names(demos)[x],
    filter=demos[[x]]) )
  names(chk) = names(demos)
  sapply(chk,length)
}
```

TcgaMutCounts	<i>obtain data frame with counts of mutation per gene symbol for selected tumor type</i>
---------------	------------------------------------------------------------------------------------------

Description

obtain data frame with counts of mutation per gene symbol for selected tumor type

Usage

```
TcgaMutCounts(tumor, limit = NULL, db = "isb-cgc:tcga_201607_beta", project)
```

Arguments

tumor	character(1) defaults to 'BRCA'
limit	numeric(1) defaults to NULL, appended as limit to number of records returned if non-null
db	character(1) BigQuery database name
project	character(1) project code

Value

table as returned by bigquery::query_exec

Note

This function returns overall mutation count, and many individuals have multiple mutations recorded per gene.

Examples

```
if (interactive()) {
  requireNamespace("bigquery")
  tt = TcgaMutCounts("BRCA", project="cgc-05-0009") # substitute your project name
  head(tt)
} # need authentication
```

TcgaNIndWithAnyMut *Give count of individuals with a mutation recorded for selected tumor*

Description

Give count of individuals with a mutation recorded for selected tumor

Usage

```
TcgaNIndWithAnyMut(tumor = "BRCA", limit = NULL,  
  db = "isb-cgc:tcga_201607_beta", project)
```

Arguments

tumor	character(1) defaults to 'BRCA'
limit	numeric(1) defaults to NULL, appended as limit to number of records returned if non-null
db	character(1) BigQuery database name
project	character(1) project code

Value

numeric(1)

Examples

```
if (interactive()) TcgaNIndWithAnyMut(project="cgc-05-0009")
```

Index

*Topic **datasets**

- [annotTabs](#), 2
- [brcaMAE](#), 3
- [CCLE_DRUG_BROAD](#), 4
- [cell_70138](#), 4
- [darmGBMcls](#), 6
- [icd10_c](#), 6
- [pancan.clin.varnames](#), 9
- [pancan_sampTypeMap](#), 11
- [pert_70138](#), 13

- [agrep](#), 11
- [annotTabs](#), 2

- [brcaMAE](#), 3

- [CCLE_DRUG_BROAD](#), 4
- [cell_70138](#), 4
- [clueDemos](#), 5
- [clueServiceNames](#), 5

- [darmGBMcls](#), 6

- [icd10_c](#), 6

- [load_ccleNRAS](#), 7
- [load_NRAS_AHR \(load_ccleNRAS\)](#), 7
- [load_nrasdf \(load_ccleNRAS\)](#), 7
- [loadPatel](#), 7

- [oncoPrintISB](#), 8

- [pancan.clin.varnames](#), 9
- [pancan_app](#), 9
- [pancan_BQ](#), 10
- [pancan_clinicalTabVarnames](#), 10
- [pancan_longname](#), 11
- [pancan_sampTypeMap](#), 11
- [pancan_tabulate](#), 12
- [pert_70138](#), 13
- [pertClasses](#), 12

- [query_clue](#), 13

- [TcgaMutCounts](#), 14
- [TcgaNIndWithAnyMut](#), 15