

Package ‘restfulSE’

January 25, 2022

Title Access matrix-like HDF5 server content or BigQuery content through a SummarizedExperiment interface

Description This package provides functions and classes to interface with remote data stores by operating on SummarizedExperiment-like objects.

Version 1.16.0

Suggests knitr, testthat, Rtsne, org.Mm.eg.db, org.Hs.eg.db, BiocStyle, restfulSEData, rmarkdown

Imports utils, stats, methods, S4Vectors, Biobase, reshape2, AnnotationDbi, DBI, GO.db, rhdf5client, dplyr (>= 0.7.1), magrittr, bigrquery, ExperimentHub, AnnotationHub, rlang

Depends R (>= 3.6), SummarizedExperiment, DelayedArray

License Artistic-2.0

LazyLoad yes

biocViews Infrastructure, SingleCell, Transcriptomics, Sequencing, Coverage

RoxygenNote 7.1.1

Collate localReplaceSlots.R RESTfulSE.R demos.R seByTumor.R isbCgc.R BQ3_Array.R BQM_Array.R pancan_SE.R

VignetteBuilder knitr

Encoding UTF-8

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

git_branch RELEASE_3_14

git_last_commit 61a70ed

git_last_commit_date 2021-10-26

Date/Publication 2022-01-25

Author Vincent Carey [aut],
Shweta Gopaulakrishnan [cre, aut]

Maintainer Shweta Gopaulakrishnan <shwetagopau192@gmail.com>

R topics documented:

assay,BQSummarizedExperiment,missing-method	3
assayNames,BQSummarizedExperiment-method	3
basic_clinvars	4
BQ3_Array	5
BQ3_Array-class	5
BQ3_ArraySeed-class	6
BQ3_Matrix-class	6
BQ3_Source	6
BQ3_Source-class	7
bqConn	7
BQM_Array	8
BQM_Array-class	9
BQM_ArraySeed-class	9
BQM_Matrix-class	9
BQM_Source	10
BQM_Source-class	10
BQSummarizedExperiment-class	11
cgcConn	11
DelayedArray,BQ3_ArraySeed-method	12
dim,BQ3_ArraySeed-method	13
dim,BQM_ArraySeed-method	13
dim,RESTfulSummarizedExperiment-method	14
dimnames,BQ3_ArraySeed-method	14
dimnames,BQM_ArraySeed-method	15
goPatt	15
gtexTiss	16
isbCgcDatasets	17
isbCgcTables	17
newDA	18
pancan_SE	18
RESTfulSummarizedExperiment	20
RESTfulSummarizedExperiment-class	21
se1.3M	22
seByTumor	23
seByTumor_2016	24
tasicCortex	25
Index	26

```
assay, BQSummarizedExperiment, missing-method
  extract assay data
```

Description

extract assay data

Usage

```
## S4 method for signature 'BQSummarizedExperiment,missing'
assay(x, i, withDimnames = TRUE, ...)
```

Arguments

x	BQSummarizedExperiment instance
i	index for retrieval, ignored at present
withDimnames	logical(1)
...	not used

Value

matrix

Note

Very experimental approach to retrieving numerical data given a SummarizedExperiment 'shell'. We need more checking of consistency between assay and clinical data before creating the shell. We use dcast to transform query result to a matrix, and some 'individuals' may have multiple contributions ... we use fun.aggregate = max and will see warnings until this is cleared up.

```
assayNames, BQSummarizedExperiment-method
  Placeholder for assay name extractor for a BQSummarizedExperiment instance.
```

Description

Placeholder for assay name extractor for a BQSummarizedExperiment instance.

Usage

```
## S4 method for signature 'BQSummarizedExperiment'
assayNames(x, ...)
```

Arguments

x instance of BQSummarizedExperiment
... not used

Value

string indicating that assay is served by BigQuery, nameless

Note

This function supplies a placeholder for this early version of a SummarizedExperiment instance to BigQuery. At present there is only one assay supported; future work will help to reduce special coding for BigQuery back end.

basic_clinvars	<i>provide a collection of basic clinical variables to limit size of clinical data return</i>
----------------	---

Description

provide a collection of basic clinical variables to limit size of clinical data return

Usage

```
basic_clinvars()
```

Value

character(20) vector of variable names

Note

Use `pancan_app()` to determine other variables available

BQ3_Array *create BQ3_Array instance given url (filepath) and entity (host) name*

Description

create BQ3_Array instance given url (filepath) and entity (host) name

Usage

```
BQ3_Array(filepath)
```

Arguments

filepath a BQ3_Source instance

Value

an instance of [DelayedArray-class](#)

Examples

```
#
# authentication issues may arise.  if you are authorized
# to use bigquery with GPC project isb-cgc, a token may
# be generated through the following
# options(httr_oob_default=TRUE)
# example(BQ3_Source)
# a browser authentication event may occur, or if you are in
# a browserless session, a URL will be emitted, possibly in
# the context of warnings ... browse to this URL and an
# authentication event will occur, and a token will be provided
# this can be provided back to the R session to allow the
# query to proceed
#
if (interactive()) {
  con = cgcConn("TCGA_hg38_data_v0")
  ss = BQ3_Source(con, filterval="TCGA-LUAD")
  BQ3_Array(ss)
}
```

BQ3_Array-class *extension of DelayedArray for BigQuery content*

Description

extension of DelayedArray for BigQuery content

BQ3_ArraySeed-class *BQ3_Array for BigQuery matrix content*

Description

BQ3_Array for BigQuery matrix content

BQ3_Matrix-class *extension of DelayedMatrix for HDF Server content*

Description

extension of DelayedMatrix for HDF Server content

BQ3_Source *construct a BigQuery resource interface*

Description

construct a BigQuery resource interface

Usage

```
BQ3_Source(
  bqconn,
  tblnm = "RNAseq_Gene_Expression",
  rowkeyfield = "Ensembl_gene_id",
  colkeyfield = "case_barcode",
  filtervbl = "project_short_name",
  filterval = "TCGA-GBM",
  assayvbl = "HTSeq__Counts",
  assaysamplotype = "TP",
  maxnrec = Inf
)
```

Arguments

bqconn	instance of BigQueryConnection from bigquery
tblnm	character(1) table name known to bqconn
rowkeyfield	character(1) field in the table that will generate rownames for matrices derived from table
colkeyfield	character(1) field in the table that will generate colnames for matrices derived from table

filtervbl	character(1) field in the table that will be used to filter out a group of records, for example, all records pertaining to a given tumor in TCGA
filterval	character(1) value in the range of filtervbl to identify records to retain –
assayvbl	character(1) field with assay quantifications
assaysamplotype	character(1) value for filtering pancancer-atlas assays, which include normals and other sample types, defaulting to "TP"; ignored if project element of bqconn is not 'pancancer-atlas'
maxnrec	numeric(1) defaults to Inf: number of records to convert with as.data.frame

Value

instance of BQ3_Source

Note

Tailored to ISB-CGC projects 'isb-cgc' and 'pancancer-atlas'.

Examples

```
if (interactive()) {
  con = cgcConn("TCGA_hg38_data_v0")
  lungConn = BQ3_Source(con, filterval="TCGA-LUAD")
  lungConn
}
```

BQ3_Source-class *a class for BigQuery tables with keys and filters baked in*

Description

a class for BigQuery tables with keys and filters baked in

bqConn *generate a connection to BigQuery for specific dataset*

Description

generate a connection to BigQuery for specific dataset

Usage

```
bqConn(dataset, project, billing)
```

Arguments

dataset	character(1) name of dataset in project
project	character(1) name of project
billing	character(1) billing code for project

Value

an instance of BigQueryConnection

Note

You will need to authenticate with Google.

Examples

```
bqConn
```

```
BQM_Array          create BQM_Array instance given url (filepath) and entity (host) name
```

Description

create BQM_Array instance given url (filepath) and entity (host) name

Usage

```
BQM_Array(filepath)
```

Arguments

filepath	a BQM_Source instance
----------	-----------------------

Value

an instance of [DelayedArray-class](#)

Examples

```
# authentication issues may arise.  if you are authorized
# to use bigquery with GPC project isb-cgc, a token may
# be generated through the following
# options(httr_oob_default=TRUE)
# example(BQM_Source)
# a browser authentication event may occur, or if you are in
# a browserless session, a URL will be emitted, possibly in
# the context of warnings ... browse to this URL and an
# authentication event will occur, and a token will be provided
# this can be provided back to the R session to allow the
```



```
# query to proceed
#
if (interactive()) {
  con = bqConn(dataset="yriMulti", project=Sys.getenv("CGC_BILLING"),
               billing=Sys.getenv("CGC_BILLING"))
  ss = BQM_Source(con, "banovichSE_methylationData", "cg_Methyl450")
  #BQM_Array(ss)
  BQM_Array(ss)["cg000000029",c("NA18498", "NA18499", "NA18501"),drop=FALSE]
}
```

BQM_Array-class *extension of DelayedArray for BigQuery content*

Description

extension of DelayedArray for BigQuery content

BQM_ArraySeed-class *BQM_Array for BigQuery matrix content*

Description

BQM_Array for BigQuery matrix content

BQM_Matrix-class *extension of DelayedMatrix for HDF Server content*

Description

extension of DelayedMatrix for HDF Server content

BQM_Source	<i>construct a BigQuery resource interface, for a matrix-like table with one column devoted to row identification (rowkeyfield) and all other columns assumed numeric</i>
------------	---

Description

construct a BigQuery resource interface, for a matrix-like table with one column devoted to row identification (rowkeyfield) and all other columns assumed numeric

Usage

```
BQM_Source(bqconn, tblnm, rowkeyfield, maxdfsize = 5e+05)
```

Arguments

bqconn	instance of BigQueryConnection from bigquery
tblnm	character(1) table name known to bqconn
rowkeyfield	character(1) field in the table that will
maxdfsize	numeric(1) field in the table that will constrain as.data.frame for determining rownames generate rownames for matrices derived from table

Value

instance of BQM_Source

Examples

```
if (interactive()) {
  con = bqConn(dataset="yriMulti", project=Sys.getenv("CGC_BILLING"),
    billing=Sys.getenv("CGC_BILLING"))
  banoMeth = BQM_Source(con, tblnm="banovichSE_MethylationData",
    rowkeyfield="cg_methyl450")
  banoMeth
}
```

BQM_Source-class	<i>Represent information about a BigQuery resource with a 'triple' database schema. This is targeting the isb-cgc TCGA layout. Big-Query Records are regarded as triples, within major groups defined by filtervbl. Triples have content subject - gene - value, to be pivoted to genes(rows) x subjects(columns) with values as entries.</i>
------------------	---

Description

Represent information about a BigQuery resource with a 'triple' database schema. This is targeting the isb-cgc TCGA layout. BigQuery Records are regarded as triples, within major groups defined by filtervbl. Triples have content subject - gene - value, to be pivoted to genes(rows) x subjects(columns) with values as entries.

BQSummarizedExperiment-class

Define a class to use BigQuery data through SummarizedExperiment interface

Description

Define a class to use BigQuery data through SummarizedExperiment interface

Slots

rowQref a BigQueryConnection wrapped in tbl_dbi that holds rowData for the SummarizedExperiment instance

colQref a BigQueryConnection wrapped in tbl_dbi that holds colData for the SummarizedExperiment instance

rowkey character(1) name of a field in the table referenced by rowQref to be used as key for features

colkey character(1) name of a field in the table referenced by colQref to use as key for samples

assayvbl character(1) name to be used to select table providing assay content

Note

This is an experimental structure to probe the concept that one can use a SummarizedExperiment object to interact with BigQuery data, particularly TCGA data. The slots rowQref and colQref are expected to be BigQuery connections which supply information on features and samples respectively, in a way that is consistent with the assay representation. See [seByTumor](#) for illustration.

cgcConn

Simplify connection to a BigQuery dataset for the project "isb-cgc"

Description

Simplify connection to a BigQuery dataset for the project "isb-cgc"

Usage

```

cgcConn(
  dataset = "TCGA_bioclin_v0",
  project = "isb-cgc",
  billing = Sys.getenv("CGC_BILLING")
)

```

Arguments

dataset	character string with dataset name
project	character string with project name
billing	character(1) with billing code

Value

instance of [BigQueryConnection-class](#)

Note

This function operates on a BigQuery project to select a dataset and return a connection. If the google billing code is assigned to environment variable CGC_BILLING, that will be used to authenticate the user and collect charges. Alternately the billing code can be given as a parameter.

Examples

```

if (interactive()) {
  cgcConn
  require(bigrquery)
  # defaults concern new GDC-compliant format
  if (nchar(Sys.getenv("CGC_BILLING"))>0) {
    clin = cgcConn()
    dbListTables(clin)
  }
}

```

DelayedArray,BQ3_ArraySeed-method
constructor

Description

constructor

Usage

```

## S4 method for signature 'BQ3_ArraySeed'
DelayedArray(seed)

```

Arguments

seed instance of 'array'

dim,BQ3_ArraySeed-method

dim derived from saved dimnames

Description

dim derived from saved dimnames

Usage

```
## S4 method for signature 'BQ3_ArraySeed'  
dim(x)
```

Arguments

x instance of BQ3_ArraySeed

Value

integer(2) vector of dimensions corresponding to R's layout, assuming 2-d data

dim,BQM_ArraySeed-method

dim derived from saved dimnames

Description

dim derived from saved dimnames

Usage

```
## S4 method for signature 'BQM_ArraySeed'  
dim(x)
```

Arguments

x instance of BQM_ArraySeed

Value

integer(2) vector of dimensions corresponding to R's layout, assuming 2-d data

dim,RESTfulSummarizedExperiment-method

Dimension access for RESTfulSummarizedExperiment

Description

Dimension access for RESTfulSummarizedExperiment

Usage

```
## S4 method for signature 'RESTfulSummarizedExperiment'  
dim(x)
```

Arguments

x instance of RESTfulSummarizedExperiment

Value

vector of nrows, ncols

dimnames,BQ3_ArraySeed-method

dimnames are saved in the BQ3_ArraySeed

Description

dimnames are saved in the BQ3_ArraySeed

Usage

```
## S4 method for signature 'BQ3_ArraySeed'  
dimnames(x)
```

Arguments

x instance of BQ3_ArraySeed

Value

currently returns list(NULL, NULL) as we do not store dimnames in HDF5

```
dimnames,BQM_ArraySeed-method
      dimnames are saved in the BQM_ArraySeed
```

Description

dimnames are saved in the BQM_ArraySeed

Usage

```
## S4 method for signature 'BQM_ArraySeed'
dimnames(x)
```

Arguments

x instance of BQM_ArraySeed

Value

currently returns list(NULL, NULL) as we do not store dimnames in HDF5

```
goPatt                    Create a data.frame with ENSEMBL and SYMBOL identifiers as-
                           associated with a GO TERM specified by a regular expression in
                           termPattern
```

Description

Create a data.frame with ENSEMBL and SYMBOL identifiers associated with a GO TERM specified by a regular expression in termPattern

Usage

```
goPatt(
  termPattern = "neurotro",
  targets = c("ENSEMBL", "SYMBOL"),
  organism = "Hs",
  inst = "eg"
)
```

Arguments

termPattern a character string encoding a regular expression to be matched to keys of type TERM in GO.db

targets columns to be returned from org.[organism].[inst].db

organism two-letter code for organism in the OrgDb family of packages

inst two- or three-letter code (e.g., eg for ENTREZ GENE or sgd for yeastgenome.org) identifying institute responsible for annotation

Value

data.frame

Examples

```
gp = goPatt()
dim(gp)
head(gp)
```

gtexTiss	<i>Convenience function for access to gene-level GTEx tissues, as quantified in recount</i>
----------	---

Description

Convenience function for access to gene-level GTEx tissues, as quantified in recount

Usage

```
gtexTiss(endpoint = URL_hsd(), svrtype = "hsds", dsetname = "/assay001")
```

Arguments

endpoint	endpoint URL of remote server
svrtype	type of server, must be either 'hsds' or 'h5serv'
dsetname	complete internal path to dataset in H5 file

Value

SummarizedExperiment instance

Examples

```
gtexTiss()
```

isbCgcDatasets	<i>Access the dataset names in the "isb-cgc" project</i>
----------------	--

Description

Access the dataset names in the "isb-cgc" project

Usage

```
isbCgcDatasets()
```

Value

character vector

Examples

```
isbCgcDatasets() # might be out of date ... can use list_datasets in bigrquery
```

isbCgcTables	<i>List the tables in a selected dataset</i>
--------------	--

Description

List the tables in a selected dataset

Usage

```
isbCgcTables(  
  dataset = "TCGA_hg19_data_v0",  
  billing = Sys.getenv("CGC_BILLING")  
)
```

Arguments

dataset	character string identifying a table in "isb-cgc"
billing	Google BigQuery billing code, which can be set in an environment variable CGC_BILLING

Value

character vector

Examples

```

if (interactive()) {
  # be sure that .cgcBilling is set
  code = Sys.getenv("CGC_BILLING")
  if (!(nchar(code)==0)) {
    isbCgcTables()
  }
}

```

newDA	<i>DelayedArray bridge</i>
-------	----------------------------

Description

DelayedArray bridge

Usage

```
newDA(seed = new("array"), Class = "DelayedArray")
```

Arguments

seed	instance of 'array'
Class	character(1)

pancan_SE	<i>illustrate DelayedArray assay from BigQuery backend in Summarized-Experiment</i>
-----------	---

Description

illustrate DelayedArray assay from BigQuery backend in SummarizedExperiment

Usage

```

pancan_SE(
  bqcon,
  colDataTableName = "clinical_PANCAN_patient_with_followup",
  clinVars = basic_clinvars(),
  colDSubjectIdName = "bcr_patient_barcode",
  colDFilterField = "acronym",
  colDFilterValue = "BRCA",

  assayDataTableName = "pancanMiRs_EBadjOnProtocolPlatformWithoutRepsWithUnCorrectMiRs_08_04_16_ann",
  assayFeatureName = "ID",

```

```

    assaySampleTypeCode = "TP",
    subjectIDName = "ParticipantBarcode",
    tumorFieldName = "Study",
    tumorFieldValue = "BRCA",
    assayValueFieldName = "miRNAexpr"
)

```

Arguments

bqcon	a BigQueryConnection instance
colDataTableName	character(1) defaulting to "clinical_PANCAN_patient_with_followup"
clinVars	character() vector of variables to be retained from the table named by 'colDataTableName', defaults to vector returned by clinVars()
colDSubjectIdName	character(1) defaulting to "bcr_patient_barcode"
colDFilterField	character(1) defaulting to "acronym"
colDFilterValue	character(1) defaulting to "BRCA"; a vector may be used, in which case multiple tumor types will be represented – must agree with tumorFieldValue, as clinical and assay data are collected separately
assayDataTableName	character(1) defaulting to "pancanMiRs_EBadjOnProtocolPlatformWithoutRepsWithUnCorrectMiRs_08"
assayFeatureName	character(1) defaulting to "ID"
assaySampleTypeCode	character(1) defaulting to "TP"
subjectIDName	character(1) defaulting to "ParticipantBarcode"
tumorFieldName	character(1) defaulting to "Study"
tumorFieldValue	character() defaulting to "BRCA"; a vector may be used in which case multiple tumor types will be represented – must agree with colDFilterValue
assayValueFieldName	character(1) defaulting to "miRNAexpr"

Value

an instance of SummarizedExperiment

Note

The parameters need different assignments for different tables. Field names are not standardized across tables as of August 2018. AUTHENTICATION CONCERNS: You must have a valid BigQuery project identifier in the environment variable CGC_BILLING, or pass such as 'billing' when using DBI::dbConnect(bigquery::bigquery(), ...). To get such a project identifier, you need to have

a Google identity and you must have created a BigQuery project with that identity. Notes at <https://isb-cancer-genomics-cloud.readthedocs.io/en/latest/sections/progapi/bigqueryGUI/WalkthroughOfGoogleBigQuery.html> provide details. Another approach that can be used involves the Google Cloud SDK. Assuming you have a Google identity and have installed a recent SDK instance, you can use (in the shell) `gcloud auth login` to pick the identity that has a project with id `PROJECT_ID`. Use that project id as the billing code for `bigquery dbConnect`, or set `CGC_BILLING` in the environment to evaluate to that project id.

Examples

```
if (interactive() & requireNamespace("DBI") &
    requireNamespace("bigquery")) {
  billco = Sys.getenv("CGC_BILLING")
  if (nchar(billco)==0)
    stop("need CGC_BILLING set to your BigQuery project ID, see note in ?pancan_SE")
  bqcon = DBI::dbConnect(bigquery::bigquery(), project = "pancancer-atlas",
    dataset = "Annotated", billing = billco)
  brca_mirSE = pancan_SE(bqcon)
  brca_mirSE
}
```

RESTfulSummarizedExperiment

Construct RESTfulSummarizedExperiment

Description

Construct RESTfulSummarizedExperiment

hidden constructor

Usage

```
RESTfulSummarizedExperiment(se, source)
```

```
.RESTfulSummarizedExperiment(se, source)
```

```
## S4 method for signature 'RESTfulSummarizedExperiment,missing'
assay(x, i, withDimnames = TRUE, ...)
```

Arguments

<code>se</code>	SummarizedExperiment instance, assay component can be empty SimpleList
<code>source</code>	instance of H5S_dataset
<code>x</code>	instance of RESTfulSummarizedExperiment
<code>i</code>	not used
<code>withDimnames</code>	logical(1)
<code>...</code>	not used

Value

instance of RESTfulSummarizedExperiment
matrix

Note

RESTfulSummarizedExperiment contains a global dimnames list generated at creation. It is possible that standard operations on a SummarizedExperiment will engender dimnames components that differ from the initial global dimnames, principally through unification (adding suffixes when dimname elements are repeated). When this is detected, assay() will fail with a complaint about length(setdiff(*names(x), x@globalDimnames[...])).

Examples

```
require("rhdf5client")
hsds = H5S_source(serverURL=URL_hsds())
hsdsCon = setPath(hsds, "/shared/bioconductor/bano_meQTLex.h5")
fetchDatasets(hsdsCon)
banoh5 = H5S_dataset2(hsdsCon, "d-3ac1ab59-672d1634-7880-3f2f37-c09c9d")
ehub = ExperimentHub::ExperimentHub()
myfiles <- AnnotationHub::query(ehub, "restfulSEData")
myfiles[["EH551"]] -> banoSEMeta
rr = RESTfulSummarizedExperiment(banoSEMeta, banoh5)
rr
rr2 = rr[1:4, 1:5] # just modify metadata
rr2
assay(rr2) # extract data
```

RESTfulSummarizedExperiment-class

HDF5Server-based assay for SummarizedExperiment

Description

HDF5Server-based assay for SummarizedExperiment

Usage

```
## S4 method for signature 'RESTfulSummarizedExperiment'
assayNames(x, ...)

## S4 method for signature 'RESTfulSummarizedExperiment,numeric,numeric,ANY'
x[i, j, ..., drop = FALSE]
```

Arguments

x	instance of RESTfulSummarizedExperiment
...	not used
i	numeric selection vector
j	numeric selection vector
drop	not used

Value

instance of RESTfulSummarizedExperiment

se1.3M	<i>Convenience functions using HSDS server to extract tenx neurons full or subset data</i>
--------	--

Description

Convenience functions using HSDS server to extract tenx neurons full or subset data

Usage

```
se1.3M(endpoint = URL_hsdS(), svrtype = "hsds", dsetname = "/newassay001")
```

Arguments

endpoint	endpoint URL of remote server
svrtype	type of server, must be either 'hsds' or 'h5serv'
dsetname	complete internal path to dataset in H5 file

Value

RESTfulSummarizedExperiment single-cell RNA-seq data from TENx, 1.3mm neurons

SummarizedExperiment instance

Note

se1.3M provides access to the full 1.3 million neurons with features in their order as given in the original HDF5

seByTumor	<i>Given a BigQueryConnection to the 2017 GDC-oriented ISB TCGA bigtables, obtain a SummarizedExperiment 'shell' rowData and colData</i>
-----------	--

Description

Given a BigQueryConnection to the 2017 GDC-oriented ISB TCGA bigtables, obtain a SummarizedExperiment 'shell' rowData and colData

Usage

```
seByTumor(
  tumorCode = "LUAD",
  assayTblName = "RNAseq_Gene_Expression",
  rdColsToKeep = c("gene_name", "Ensembl_gene_id", "gene_type"),
  bqConnClinical,
  bqConnAssay,
  rowkey = "Ensembl_gene_id",
  colkey = "case_barcode",
  assayvbl = "HTSeq__Counts"
)
```

Arguments

tumorCode	one of the concise TCGA codes in a character string – not checked, defaults to "LUAD", lung adenocarcinoma
assayTblName	the name of the assay whose annotation will be used as rowData
rdColsToKeep	columns of assay table to use in rowData component
bqConnClinical	instance of BigQueryConnection from bigrquery, for access to clinical metadata – current expectation is that the BigQuery dataset is named "TCGA_bioclin_v0" and has a table called "Clinical"
bqConnAssay	instance of BigQueryConnection from bigrquery – current expectation is that the BigQuery dataset is named "TCGA_hg19_data_v0"
rowkey	name of a field to be used as key for rows
colkey	name of a field to use as key for samples
assayvbl	name of field to use for numerical values

Value

SummarizedExperiment

Note

This function demonstrates the use of external resources for rowData, colData and assay components of a SummarizedExperiment instance. The intention is that the full complement of activities supported by [SummarizedExperiment-class](#) are likewise supported through this class, with assay data and sample and feature metadata all external and in BigQuery projects. The seByTumor function is provided to generate an example of this approach with minimal user configuration.

Examples

```
if (interactive()) {
  require(bigrquery)
  # be sure that .cgcBilling is set
  code = Sys.getenv("CGC_BILLING")
  if (!(nchar(code)==0)) {
    clinQ = cgcConn(billing=code)
    assayQ = cgcConn( dataset = "TCGA_hg38_data_v0", billing=code )
    myexpShell = seByTumor( bqConnClinical=clinQ,
                          bqConnAssay=assayQ)
    print(myexpShell)
    print(nrow(myexpShell) == 60483)
    print(ncol(myexpShell) == 515) # 7/18/2018
    assay(myexpShell[[11:15,1:4]) # some case_barcode repeat
  }
}
```

seByTumor_2016

Given a BigQueryConnection to the 2016 ISB TCGA bigtables, obtain a SummarizedExperiment 'shell' rowData and colData

Description

Given a BigQueryConnection to the 2016 ISB TCGA bigtables, obtain a SummarizedExperiment 'shell' rowData and colData

Usage

```
seByTumor_2016(
  tumorCode = "LUAD",
  assayTblName = "mRNA_UNC_HiSeq_RSEM",
  rdColsToKeep = c("original_gene_symbol", "HGNC_gene_symbol", "gene_id", "Study"),
  bqConn
)
```

Arguments

tumorCode one of the concise TCGA codes in a character string – not checked, defaults to "LUAD", lung adenocarcinoma

assayTblName the name of the assay whose annotation will be used as rowData

rdColsToKeep columns of assay table to use in rowData component
bqConn instance of BigQueryConnection from bigrquery

Value

SummarizedExperiment instance, with BigQuery reference as assay

tasicCortex *A set of mouse cortex marker genes.*

Description

A set of mouse cortex marker genes.

Usage

```
tasicCortex()
```

Value

data.frame with columns SYMBOL, GENEID

Note

<http://www.nature.com/doi/10.1038/nn.4216>, Fig 1C

Examples

```
head(tasicCortex())
```

Index

[.RESTfulSummarizedExperiment](#)
[\(RESTfulSummarizedExperiment\)](#), [20](#)
[\[,RESTfulSummarizedExperiment,numeric,numeric,ANY-method](#)
[\(RESTfulSummarizedExperiment-class\)](#), [21](#)
[assay\(RESTfulSummarizedExperiment\)](#), [20](#)
[assay,BQSummarizedExperiment,missing-method](#), [3](#)
[assay,RESTfulSummarizedExperiment,missing-method](#)
[\(RESTfulSummarizedExperiment\)](#), [20](#)
[assayNames](#)
[\(assayNames,BQSummarizedExperiment-method\)](#), [3](#)
[assayNames,BQSummarizedExperiment-method](#), [3](#)
[assayNames,RESTfulSummarizedExperiment-method](#)
[\(RESTfulSummarizedExperiment-class\)](#), [21](#)
[basic_clinvars](#), [4](#)
[BQ3_Array](#), [5](#)
[BQ3_Array-class](#), [5](#)
[BQ3_ArraySeed-class](#), [6](#)
[BQ3_Matrix-class](#), [6](#)
[BQ3_Source](#), [6](#)
[BQ3_Source-class](#), [7](#)
[bqConn](#), [7](#)
[BQM_Array](#), [8](#)
[BQM_Array-class](#), [9](#)
[BQM_ArraySeed-class](#), [9](#)
[BQM_Matrix-class](#), [9](#)
[BQM_Source](#), [10](#)
[BQM_Source-class](#), [10](#)
[BQSummarizedExperiment-class](#), [11](#)
[cgcConn](#), [11](#)
[DelayedArray,BQ3_ArraySeed-method](#), [12](#)
[dim,BQ3_ArraySeed-method](#), [13](#)
[dim,BQM_ArraySeed-method](#), [13](#)
[dim,RESTfulSummarizedExperiment-method](#), [14](#)
[dimnames,BQ3_ArraySeed-method](#), [14](#)
[dimnames,BQM_ArraySeed-method](#), [15](#)
[goPatt](#), [15](#)
[gtexTiss](#), [16](#)
[hbc](#)
[isbCgcDatasets](#), [17](#)
[isbCgcTables](#), [17](#)
[newDA](#), [18](#)
[pancAn_SE](#), [18](#)
[RESTfulSummarizedExperiment](#), [20](#)
[RESTfulSummarizedExperiment,RangedSummarizedExperiment,H5S](#)
[\(RESTfulSummarizedExperiment\)](#), [20](#)
[RESTfulSummarizedExperiment-class](#), [21](#)
[se1.3M](#), [22](#)
[seByTumor](#), [11](#), [23](#)
[seByTumor_2016](#), [24](#)
[tasicCortex](#), [25](#)