Package ‘restfulSE’

March 28, 2024

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 1.24.0

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, bigquery, ExperimentHub, AnnotationHub, rlang

Depends R (>= 3.6), SummarizedExperiment, DelayedArray

License Artistic-2.0

LazyLoad yes

biocViews Infrastructure, SingleCell, Transcriptomics, Sequencing, Coverage

RoxygenNote 7.2.1

Collate localReplaceSlots.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_18

git_last_commit 8154538

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-27

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

Maintainer Shweta Gopaulakrishnan <shwetagopaul92@gmail.com>
R topics documented:

- assay,BQSummarizedExperiment,missing-method
- assayNames,BQSummarizedExperiment-method
- basic_clinvars
- BQ3_Array
- BQ3_Array-class
- BQ3_ArraySeed-class
- BQ3_Matrix-class
- BQ3_Source
- BQ3_Source-class
- bqConn
- BQM_Array
- BQM_Array-class
- BQM_ArraySeed-class
- BQM_Matrix-class
- BQM_Source
- BQM_Source-class
- BQSummarizedExperiment-class
- cgcConn
- DelayedArray,BQ3_ArraySeed-method
- dim,BQ3_ArraySeed-method
- dim,BQM_ArraySeed-method
- dimnames,BQ3_ArraySeed-method
- dimnames,BQM_ArraySeed-method
- goPatt
- gtexTiss
- isbCgcDatasets
- isbCgcTables
- newDA
- pancan_SE
- sel.3M
- seByTumor
- seByTumor_2016
- tasicCortex

Index

- assay,BQSummarizedExperiment,missing-method

extract assay data

Description

eXtract assay data
assayNames, BQSummarizedExperiment-method

Usage

```r
## 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

```r
## 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**

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

---

Description

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

Usage

```r
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

```r
BQ3_Array(filepath)
```

Arguments

- `filepath` a BQ3_Source instance

Value

an instance of `DelayedArray-class`
Examples

```r
# 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
construct a BigQuery resource interface

Description

construct a BigQuery resource interface

Usage

```r
BQ3_Source(
  bqconn,
  tblnm = "RNAseq_Gene_Expression",
  rowkeyfield = "Ensembl_gene_id",
  colkeyfield = "case_barcode",
  filtervbl = "project_short_name",
  filterval = "TCGA-GBM",
  assayvbl = "HTSeq__Counts",
  assaysampletype = "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
- **assaysampletype**: 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

```r
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-class

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**

```r
# 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)["cg00000029",c("NA18498", "NA18499", "NA18501"),drop=FALSE]
}
```

BQM_Array-class

extension of DelayedArray-class for BigQuery content

**Description**

extension of DelayedArray for BigQuery content
BQM_ArraySeed-class

**Description**

BQM_Array for BigQuery matrix content

BQM_Matrix-class

**Description**

extension of DelayedMatrix for HDF Server content

BQM_Source

**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 bigrquery
- **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

```r
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

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.

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)
}
}
Description

constructor

Usage

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

Arguments

seed instance of 'array'

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
Description

dim derived from saved dimnames

Usage

```r
## 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

Description

dimnames are saved in the BQM_ArraySeed

Usage

```r
## 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
Description

dimnames are saved in the BQM_ArraySeed

Usage

```r
## 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 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**

```r
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
**gtexTiss**

**Value**

data.frame

**Examples**

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

---

**Description**

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

**Usage**

```r
gtexTiss(endpoint = URL_hsds(), 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**

```r
gtexTiss()
```
### isbCgcDatasets

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

**Description**

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

**Usage**

```r
isbCgcDatasets()
```

**Value**

character vector

**Examples**

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

### isbCgcTables

**List the tables in a selected dataset**

**Description**

List the tables in a selected dataset

**Usage**

```r
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

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

---

### newDA

**DelayedArray bridge**

**Description**

DelayedArray bridge

**Usage**

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

**Arguments**

- **seed**: instance of 'array'
- **Class**: character(1)

---

### pancan_SE

**illustrate DelayedArray assay from BigQuery backend in Summarized-Experiment**

**Description**

illustrate DelayedArray assay from BigQuery backend in SummarizedExperiment

**Usage**

```r
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_annot",
  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_04_16_annot"
assayFeatureName character(1) defaulting to "ID"
assaySampleTypeCode character(1) defaulting to "ID"
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::bigrquery(), ...). 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://g.co/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

```r
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
}
```

---

**se1.3M**

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

**Description**

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

**Usage**

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

**Value**

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

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

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
seByTumor_2016

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

```r
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_barcodes 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

```r
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 bigquery

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/doifinder/10.1038/nn.4216, Fig 1C

Examples
head(tasicCortex())
Index

assay,BQSummarizedExperiment,missing-method, se1.3M, 19
2
assayNames
seByTumor, seByTumor_2016, 21
(assayNames,BQSummarizedExperiment-method),
tasicCortex, 22
assayNames,BQSummarizedExperiment-method,
3

basic_clinvars, 4
BQ3_Array, 4
BQ3_Array-class, 5
BQ3_ArraySeed-class, 5
BQ3_Matrix-class, 5
BQ3_Source, 6
BQ3_Source-class, 7
bqConn, 7
BQM_Array, 8
BQM_Array-class, 8
BQM_ArraySeed-class, 9
BQM_Matrix-class, 9
BQM_Source, 9
BQM_Source-class, 10
BQSummarizedExperiment-class, 10
cgcConn, 11

DelayedArray,BQ3_ArraySeed-method, 12
dim,BQ3_ArraySeed-method, 12
dim,BQM_ArraySeed-method, 13
dimnames,BQ3_ArraySeed-method, 13
dimnames,BQM_ArraySeed-method, 14

goPatt, 14
gtexTiss, 15

isbCgcDatasets, 16
isbCgcTables, 16

newDA, 17

pancan_SE, 17