

Package ‘epivizrData’

October 16, 2018

Type Package

Title Data Management API for epiviz interactive visualization app

Version 1.8.0

URL <http://epiviz.github.io>

Description Serve data from Bioconductor Objects through a WebSocket connection.

BugReports <https://github.com/epiviz/epivizrData/issues>

biocViews Infrastructure, Visualization

Depends R (>= 3.4), methods, epivizrServer (>= 1.1.1), Biobase

Imports S4Vectors, GenomicRanges, SummarizedExperiment (>= 0.2.0), OrganismDbi, GenomicFeatures, GenomeInfoDb, IRanges, ensemblDb

Suggests testthat, roxygen2, bumpHunter, hgu133plus2.db, Mus.musculus, TxDb.Mmusculus.UCSC.mm10.knownGene, rjson, knitr, rmarkdown, BiocStyle, EnsDb.Mmusculus.v79, AnnotationHub, rtracklayer, utils, RMySQL, DBI

License MIT + file LICENSE

LazyData true

RoxygenNote 6.0.1

Collate 'epivizrData-package.R' 'EpivizMeasurement-class.R'
'EpivizDataMgr-class.R' 'createMgr.R' 'EpivizData-class.R'
'EpivizTrackData-class.R' 'EpivizBlockData-class.R'
'EpivizBpData-class.R' 'EpivizGeneInfoData-class.R'
'EpivizFeatureData-class.R' 'make_gene_info_gr.R'
'register-methods.R' 'utils.R'

VignetteBuilder knitr

NeedsCompilation no

Author Hector Corrada Bravo [aut, cre], Florin Chelaru [aut]

Maintainer Hector Corrada Bravo <hcorrada@gmail.com>

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

git_branch RELEASE_3_7

git_last_commit fea1420

git_last_commit_date 2018-04-30

Date/Publication 2018-10-15

R topics documented:

ahToMySQL	2
as.data.frame,EpivizData-method	3
as.list,EpivizMeasurement-method	4
createMgr	4
EpivizBlockData-class	5
EpivizBpData-class	5
EpivizData-class	5
EpivizDataMgr-class	6
EpivizFeatureData-class	7
EpivizGeneInfoData-class	7
EpivizMeasurement-class	7
epivizrData	8
register	8
show,EpivizMeasurement-method	9

Index 11

ahToMySQL	<i>Utility function to import data to a MySQL database from Annotation Hub</i>
-----------	--

Description

Utility function to import data to a MySQL database from Annotation Hub

Usage

```
ahToMySQL(ah, annotations = list(), ...)
```

Arguments

ah	AnnotationHub object with records to add to database.
annotations	A named list of lists (key/value pairs). Keys must be the AH ID for the corresponding record and the value is a named list representing an annotation. An annotation is automatically inferred by the record's metadata. Any annotation that is passed for a particular record is concatenated to its inferred annotation. If the annotation has a subtype column, it is used to name the data object being added to the db, otherwise the record's tags is used.
...	arguments for toMySQL (connection, db_name, batch, index)

Examples

```
library(epivizrData)
library(AnnotationHub)
library(DBI)
library(RMySQL)

ah <- AnnotationHub()
db_annotations <- list()

# Query Patterns
```

```
roadmap <- "EpigenomeRoadMap"
bisulphite <- "bisulphite"

esophagus <- query(ah, c("esophagus", "roadmap", "bisulphite"))
eso_anno <- list(tissue="Digestive", subtype="Esophagus")
eso_id <- names(esophagus)
db_annotations[[eso_id]] <- eso_anno

gastric <- query(ah, c("gastric", "roadmap", "bisulphite"))
gas_anno <- list(tissue="Digestive", subtype="Gastric")
gas_id <- names(gastric)
db_annotations[[gas_id]] <- gas_anno

sigmoid_colon <- query(ah, c("sigmoid colon", "roadmap", "bisulphite"))
colon_anno <- list(tissue="Digestive", subtype="Sigmoid Colon")
colon_id <- names(sigmoid_colon)
db_annotations[[colon_id]] <- colon_anno

small_intestine <- query(ah, c("small intestine", "roadmap", "bisulphite"))
intestine_anno <- list(tissue="Digestive", subtype="Small Intestine")
intestine_id <- names(small_intestine)
db_annotations[[intestine_id]] <- intestine_anno

# This collapses our 4 records into one AnnotationHub object.
records <- c(esophagus, gastric, sigmoid_colon, small_intestine)

# connection <- dbConnect(MySQL(), host=host, user=user, password=pass)
# db_name="my_database"

# ahToMySQL(ah=records, annotations=db_annotations,
# connection=connection, db_name=db_name)
```

as.data.frame,EpivizData-method

Generic as.data.frame method for EpivizData objects

Description

Generic as.data.frame method for EpivizData objects

Usage

```
## S4 method for signature 'EpivizData'
as.data.frame(x, query = NULL, ...)
```

Arguments

x	EpivizData object to coerce.
query	GRanges object
...	other param to send to data.frame

as.list,EpivizMeasurement-method
Convert [EpivizMeasurement](#) object to list

Description

Convert [EpivizMeasurement](#) object to list

Usage

```
## S4 method for signature 'EpivizMeasurement'  
as.list(x)
```

Arguments

x [EpivizMeasurement](#) object to coerce.

Value

a list describing measurement object

createMgr *Create a data manager for epiviz app*

Description

Create a data manager for epiviz app

Usage

```
createMgr(server = server)
```

Arguments

server An object of class [EpivizServer](#)

Value

An object of class [EpivizDataMgr](#)

Examples

```
server <- epivizrServer::createServer(port=7123L)  
data_mgr <- epivizrData::createMgr(server)
```

EpivizBlockData-class *Data container for interval data.*

Description

Used to serve data for visualizations of genomic regions only. Wraps [GenomicRanges](#) objects.

Methods

`get_default_chart_type()` Get name of default chart type for this data type

`get_measurements()` Get description of measurements served by this object

See Also

[EpivizData](#)

EpivizBpData-class *Container for basepair level numeric data*

Description

Used to serve data to genomic line tracks. Wraps [GenomicRanges](#) objects. Numeric values obtained from `mcols` slot.

Methods

`get_default_chart_type()` Get name of default chart type for this data type

`get_measurements()` Get description of measurements served by this object

See Also

[EpivizData](#)

EpivizData-class *Data container for epiviz data server*

Description

Data container for epiviz data server

Methods

get_default_chart_type() Get name of default chart type for this data type
 get_id() Get id provided by manager [EpivizDataMgr-class](#)
 get_measurements() Get description of measurements served by this object
 get_name() Get datasource name, usually set by manager [EpivizDataMgr-class](#)
 get_rows(query, metadata, useOffset = FALSE) Get genomic interval information overlapping query [<GenomicRanges>](#) region
 get_source_name() Get original datasource name provided by manager [EpivizDataMgr-class](#)
 get_values(query, measurement, round = TRUE) Get measurement values for features overlapping query region [<GenomicRanges](#)
 parse_measurement(ms_id = NULL) Parse a measurement description for data served by this object
 set_id(id) Set id, used by manager [EpivizDataMgr-class](#)
 set_limits(ylim) Set plotting limits for continuous data
 set_mgr(mgr) Set data manager, [EpivizDataMgr-class](#)
 set_name(name) Set datasource name, usually set by manager [EpivizDataMgr-class](#)
 set_source_name(source_name) Set original datasource name, used by manager [EpivizDataMgr-class](#)
 toMySQL(connection, db_name, annotation = NULL, batch = 50, index = TRUE) Send Epiviz-Data to a MySQL Database
 connection DBIConnection to a database
 db_name Name of MySQL database
 annotation Annotation for index table
 batch Batch size for data sent to the MySQL database
 index Insert into respective index table
 update(new_object, send_request = TRUE) Update underlying data object with new object

EpivizDataMgr-class *Class providing data manager for epiviz app*

Description

Class providing data manager for epiviz app

Methods

add_measurements(obj, datasource_name = NULL, datasource_origin_name = deparse(substitute(obj)), register measurements in data manager
 is_ms_connected(ms_obj_or_id) check if measurement object was properly added to JS app
 is_server_closed() Check if underlying server is closed, [<logical>](#)
 list_measurements() make a printable list of registered measurements
 rm_all_measurements() remove all registered measurements
 rm_measurements(ms_obj_or_id) remove registered measurements from a given data object
 update_measurements(ms_obj_or_id, new_object, send_request = TRUE) update the underlying data object for a registered measurement (given by object or id)

EpivizFeatureData-class

Data container for RangedSummarizedExperiment objects

Description

Used to serve general data (used in e.g., scatter plots and heatmaps). Wraps [RangedSummarizedExperiment](#) objects. Numeric values obtained from assays slot

Methods

`get_default_chart_type()` Get name of default chart type for this data type

`get_measurements()` Get description of measurements served by this object

See Also

[EpivizData](#)

EpivizGeneInfoData-class

Container for gene annotation data

Description

Used to serve data to gene annotation tracks. Wraps [GenomicRanges](#) objects. Annotation obtained from columns Gene (gene symbols) and Exons (exon start and end locations).

Methods

`get_default_chart_type()` Get name of default chart type for this data type

`get_measurements()` Get description of measurements served by this object

`get_rows(query, metadata, useOffset = FALSE)` Get genomic interval information overlapping query [GenomicRanges](#) region

See Also

[EpivizData](#)

`register.OrganismDb`

EpivizMeasurement-class

Class encapsulating a measurement description for epiviz app.

Description

Class encapsulating a measurement description for epiviz app.

 epivizrData

epivizrData

Description

 epivizrData

 register

Generic method to register data to the data server

Description

Generic method to register data to the data server

Usage

```

register(object, columns = NULL, ...)

## S4 method for signature 'GenomicRanges'
register(object, columns, type = c("block", "bp",
  "gene_info"), ...)

## S4 method for signature 'RangedSummarizedExperiment'
register(object, columns = NULL,
  assay = 1, metadata = NULL)

## S4 method for signature 'ExpressionSet'
register(object, columns, annotation = NULL,
  assay = "exprs")

## S4 method for signature 'OrganismDb'
register(object, kind = c("gene", "tx"),
  keepSeqlevels = NULL, ...)

## S4 method for signature 'TxDb'
register(object, kind = c("gene", "tx"),
  keepSeqlevels = NULL, ...)

## S4 method for signature 'EnsDb'
register(object, kind = c("gene", "tx"),
  keepSeqlevels = NULL, ...)

```

Arguments

object	The object to register to data server
columns	Name of columns containing data to register
...	Additional arguments passed to object constructors

type	Which type of data object to register for a GenomicRanges object. block: only region data, bp base-pair resolution quantitative data (see columns argument), geneInfo information about gene location.
assay	Which assay in object to register
metadata	Additional metadata about features
annotation	Character string indicating platform annotation (only hgu133plus2 supported for now)
kind	Make gene or transcript annotation (only gene supported for now)
keepSeqlevels	character vector indicating seqlevels in object to keep

Value

Object inheriting from [EpivizData](#) class

Methods (by class)

- GenomicRanges: Register a [GenomicRanges](#) object
- RangedSummarizedExperiment: Register a [RangedSummarizedExperiment](#) object
- ExpressionSet: Register an [ExpressionSet](#) object
- OrganismDb: Register an [OrganismDb](#) object
- TxDb: Register a [TxDb](#) object
- EnsDb: Register an [EnsDb](#) object

Examples

```
library(GenomicRanges)
# create an example GRanges object
gr <- GRanges("chr10", IRanges(start=1:1000, width=100), score=rnorm(1000))
# this returns an EpivizData object without adding to data manager
# this is not the preferred way of creating these object, but is shown
# here for completeness.
ms_obj <- epivizrData::register(gr, type="bp", columns="score")

server <- epivizrServer::createServer(port=7123L)
data_mgr <- epivizrData::createMgr(server)

# This adds a data object to the data manager
data_mgr$add_measurements(gr, "example_gr", type="bp", columns="score")
```

show,EpivizMeasurement-method

Display measurement datasourceId and id

Description

Display measurement datasourceId and id

Usage

```
## S4 method for signature 'EpivizMeasurement'  
show(object)
```

Arguments

object a [EpivizMeasurement](#) to display

Value

A string describing measurement

Index

ahToMySQL, [2](#)
AnnotationHub, [2](#)
as.data.frame, EpivizData-method, [3](#)
as.list, EpivizMeasurement-method, [4](#)

createMgr, [4](#)

EnsDb, [9](#)
EpivizBlockData
 (EpivizBlockData-class), [5](#)
EpivizBlockData-class, [5](#)
EpivizBpData (EpivizBpData-class), [5](#)
EpivizBpData-class, [5](#)
EpivizData, [3](#), [9](#)
EpivizData (EpivizData-class), [5](#)
EpivizData-class, [5](#)
EpivizDataMgr, [4](#)
EpivizDataMgr (EpivizDataMgr-class), [6](#)
EpivizDataMgr-class, [6](#)
EpivizFeatureData
 (EpivizFeatureData-class), [7](#)
EpivizFeatureData-class, [7](#)
EpivizGeneInfoData
 (EpivizGeneInfoData-class), [7](#)
EpivizGeneInfoData-class, [7](#)
EpivizMeasurement, [4](#), [10](#)
EpivizMeasurement
 (EpivizMeasurement-class), [7](#)
EpivizMeasurement-class, [7](#)
epivizrData, [8](#)
epivizrData-package (epivizrData), [8](#)
EpivizServer, [4](#)
ExpressionSet, [9](#)

GenomicRanges, [5–7](#), [9](#)

OrganismDb, [9](#)

RangedSummarizedExperiment, [7](#), [9](#)
register, [8](#)
register, EnsDb-method (register), [8](#)
register, ExpressionSet-method
 (register), [8](#)
register, GenomicRanges-method
 (register), [8](#)
register, OrganismDb-method (register), [8](#)
register, RangedSummarizedExperiment-method
 (register), [8](#)
register, TxDb-method (register), [8](#)

show, EpivizMeasurement-method, [9](#)

TxDb, [9](#)