

# Package ‘epivizrData’

October 16, 2022

**Type** Package

**Title** Data Management API for epiviz interactive visualization app

**Version** 1.24.0

**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, matrixStats

**License** MIT + file LICENSE

**LazyData** true

**RoxygenNote** 7.1.1

**Collate** 'epivizrData-package.R' 'SparseEpivizMeasurement-class.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\_15

**git\_last\_commit** 8a48f19

**git\_last\_commit\_date** 2022-04-26

**Date/Publication** 2022-10-16

## R topics documented:

.emptyEpivizMeasurement . . . . .	2
.emptyEpivizSparseMeasurement . . . . .	3
ahToMySQL . . . . .	3
as.data.frame,EpivizData-method . . . . .	4
as.list,EpivizMeasurement-method . . . . .	4
as.list,SparseEpivizMeasurement-method . . . . .	5
createMgr . . . . .	5
EpivizBlockData-class . . . . .	6
EpivizBpData-class . . . . .	6
EpivizData-class . . . . .	7
EpivizDataMgr-class . . . . .	8
EpivizFeatureData-class . . . . .	8
EpivizGeneInfoData-class . . . . .	9
EpivizMeasurement-class . . . . .	9
epivizrData . . . . .	9
register . . . . .	10
show,EpivizMeasurement-method . . . . .	11
show,SparseEpivizMeasurement-method . . . . .	12
SparseEpivizMeasurement-class . . . . .	12
<b>Index</b>	<b>13</b>

---

.emptyEpivizMeasurement

*Create empty Epiviz Measurement*

---

### Description

Create empty Epiviz Measurement

### Usage

.emptyEpivizMeasurement()

---

.emptyEpivizSparseMeasurement  
*Create empty Epiviz Measurement*

---

### Description

Create empty Epiviz Measurement

### Usage

```
.emptyEpivizSparseMeasurement()
```

---

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

---

### 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

```
## Not run:  
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

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

ahToMySQL(ah=record, annotations=db_annotations,
          connection=connection, db_name=db_name)

## End(Not run)

```

---

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	<a href="#">EpivizData</a> 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

---

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

---

**Description**

Convert [SparseEpivizMeasurement](#) object to list

**Usage**

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

**Arguments**

x [SparseEpivizMeasurement](#) 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)), send\_request = FALSE) 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

Infrastructure package for the epivizr interactive visualization system in Bioconductor. It provides connections between Bioconductor infrastructure objects and the epivizr visualization framework.

---

register	<i>Generic method to register data to the data server</i>
----------	---

---

### 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, ...)

## S4 method for signature 'data.frame'
register(object, columns = 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 <a href="#">GenomicRanges</a> 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
- data.frame: Register an [data.frame](#)

**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 <- epivizData::register(gr, type="bp", columns="score")

server <- epivizServer::createServer(port=7123L)
data_mgr <- epivizData::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

---

show, SparseEpivizMeasurement-method

*Display measurement datasourceId and id*

---

**Description**

Display measurement datasourceId and id

**Usage**

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

**Arguments**

object            a [SparseEpivizMeasurement](#) to display

**Value**

A string describing measurement

---

SparseEpivizMeasurement-class

*Class encapsulating a measurement description for epiviz app.*

---

**Description**

Class encapsulating a measurement description for epiviz app.

# Index

.emptyEpivizMeasurement, [2](#)  
.emptyEpivizSparseMeasurement, [3](#)

ahToMySQL, [3](#)  
as.data.frame, EpivizData-method, [4](#)  
as.list, EpivizMeasurement-method, [4](#)  
as.list, SparseEpivizMeasurement-method, [5](#)

createMgr, [5](#)

data.frame, [11](#)

EnsDb, [11](#)  
EpivizBlockData  
    (EpivizBlockData-class), [6](#)  
EpivizBlockData-class, [6](#)  
EpivizBpData (EpivizBpData-class), [6](#)  
EpivizBpData-class, [6](#)  
EpivizData, [4](#), [11](#)  
EpivizData (EpivizData-class), [7](#)  
EpivizData-class, [7](#)  
EpivizDataMgr, [6](#)  
EpivizDataMgr (EpivizDataMgr-class), [8](#)  
EpivizDataMgr-class, [8](#)  
EpivizFeatureData  
    (EpivizFeatureData-class), [8](#)  
EpivizFeatureData-class, [8](#)  
EpivizGeneInfoData  
    (EpivizGeneInfoData-class), [9](#)  
EpivizGeneInfoData-class, [9](#)  
EpivizMeasurement, [4](#), [5](#), [11](#)  
EpivizMeasurement  
    (EpivizMeasurement-class), [9](#)  
EpivizMeasurement-class, [9](#)  
epivizrData, [9](#)  
EpivizServer, [6](#)  
ExpressionSet, [11](#)

GenomicRanges, [6](#), [7](#), [9–11](#)

OrganismDb, [11](#)

RangedSummarizedExperiment, [8](#), [11](#)  
register, [10](#)  
register, data.frame-method (register), [10](#)  
register, EnsDb-method (register), [10](#)  
register, ExpressionSet-method (register), [10](#)  
register, GenomicRanges-method (register), [10](#)  
register, OrganismDb-method (register), [10](#)  
register, RangedSummarizedExperiment-method (register), [10](#)  
register, TxDb-method (register), [10](#)

show, EpivizMeasurement-method, [11](#)  
show, SparseEpivizMeasurement-method, [12](#)

SparseEpivizMeasurement, [5](#), [12](#)  
SparseEpivizMeasurement  
    (SparseEpivizMeasurement-class), [12](#)  
SparseEpivizMeasurement-class, [12](#)

TxDb, [11](#)