Package 'epivizr'

November 11, 2026

Maintainer Hector Corrada Bravo <hcorrada@gmail.com>

Author Hector Corrada Bravo, Florin Chelaru, Llewellyn Smith, Naomi Goldstein, Jayaram Kancherla, Morgan Walter, Brian Gottfried

Version 2.42.0

License Artistic-2.0

Title R Interface to epiviz web app

Description This package provides connections to the epiviz web app (http://epiviz.cbcb.umd.edu) for interactive visualization of genomic data. Objects in R/bioc interactive sessions can be displayed in genome browser tracks or plots to be explored by navigation through genomic regions. Fundamental Bioconductor data structures are supported (e.g., GenomicRanges and RangedSummarizedExperiment objects), while providing an easy mechanism to support other data structures (through package epivizrData). Visualizations (using d3.js) can be easily added to the web app as well.

VignetteBuilder knitr

Depends R (>= 3.0), methods,

Imports epivizrServer (>= 1.1.1), epivizrData (>= 1.3.4), GenomicRanges, S4Vectors, IRanges, bumphunter, GenomeInfoDb

Suggests testthat, roxygen2, knitr, Biobase, SummarizedExperiment, antiProfilesData, hgu133plus2.db, Mus.musculus, BiocStyle, minfi, rmarkdown

Collate 'EpivizChartMgr-class.R' 'EpivizApp-class.R' 'EpivizChart-class.R' 'restartEpiviz.R' 'settings_utils.R' 'startEpiviz.R' 'vignette_data.R'

biocViews Visualization, Infrastructure, GUI

Video https://www.youtube.com/watch?v=099c4wUxozA

RoxygenNote 7.1.0

git_url https://git.bioconductor.org/packages/epivizr

git_branch RELEASE_3_23

git_last_commit 9cd98d7

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

EpivizApp-class 2
EpivizChart-class 3
EpivizChartMgr-class 4
restartEpiviz 6
show,EpivizChart-method 7
startEpiviz 7
tcga_colon_blocks 9
tcga_colon_curves 10
tcga_colon_expression 10

Index **12**

`EpivizApp-class` *Class managing connection to epiviz application.*

Description

Class managing connection to epiviz application.

Fields

`server` An object of class `EpivizServer` used to communicate with epiviz app.

`data_mgr` An object of class `EpivizDataMgr` used to serve data to epiviz app.

`chart_mgr` An object of class `EpivizChartMgr` used to manage charts added to epiviz app session.

Methods

`get_current_location(callback)` Obtain current genome location on epiviz app and evaluate callback function on result.

callback A callback function to evaluate on response data. Response data will be a list with slots `seqName`, `start` and `end`

`get_ms_object(chart_id_or_object, index = 1)` Get object of class `EpivizData` used as a data source in a given chart.

chart_id_or_object An object of class `EpivizChart` or an id for a chart loaded to the epiviz app.

index Index into `.measurements`list of chart object to obtain data object for.

`is_server_closed()` Check if underlying server connection is closed.

`load_remote_measurements()` Load remote measurements into R session.

`load_workspace(workspace_id = NULL)` load an epiviz workspace.

workspace_id Workspace id to load from the database.

`navigate(chr, start, end)` Navigate to given position on the epiviz app.

`plot( data_object, datasource_name = NULL, send_request = TRUE, settings = NULL, colors = NULL, ... )` Visualize data on epiviz app using its default chart type. Measurements from the `data-object` are first added to the epiviz app using the `add_measurements` method for class `EpivizData`. See documentation for `register` for information on supported data types and the `EpivizData` class encapsulating this type of data. Once measurements are loaded, the `plot` method of class `EpivizChartMgr` is used to plot the data, using the default chart type for this type of data.

data_object An object to plot in epiviz app.

datasource_name Name to use for datasource, parses `data_object` if missing or NULL

... Additional arguments passed to `add_measurements` method for class `EpivizData`

`print_workspace(file_name = NULL, file_type = "pdf")` Save epiviz workspace as a pdf or png.

`save(file, include_data = TRUE)` Save EpivizApp object representation of a workspace into .RData file.

file (character) The name of the file to save the EpivizApp object into, ending in .rda.

stop_app (logical) Stop the EpivizApp session.

include_data (logical) Include EpivizApp's data when saving the EpivizApp object.

`service(verbose = TRUE)` Block interactive R session to service websocket requests.

`slideshow(granges, n = length(granges), .callback = NULL)` Navigate on epiviz app successively to given positions.

granges An object of class `GenomicRanges` indicating set of genomic regions to navigate in epiviz app.

n (integer) The number of regions in `granges` to navigate to.

.callback (function) function to call after navigating to each region. Used for testing purposes.

`stop_app()` stop and clean connection to epiviz app.

`EpivizChart-class` *Class encapsulating a chart in epiviz app*

Description

Class encapsulating a chart in epiviz app

Methods

`get_app_id(id)` Get chart's id assigned by epiviz app.

`get_colors()` Get color palette applied to the chart.

`get_id()` Get chart's id assigned by chart manager

`get_measurements()` Get measurements for the chart.

`get_settings()` Get custom settings for the chart.

`get_source_name()` Get chart's id assigned by chart manager

`is_connected()` Returns `TRUE` if chart is connected to a chart on epiviz app.

`print_info()` Print settings and color currently used in chart object.

`set(settings = NULL, colors = NULL)` Set settings and colors used in chart.

settings List of settings to apply.

colors Character vector of HEX colors to use in chart.

`set_app_id(id)` Set chart's id assigned by epiviz app.

`set_colors(colors)` Set custom colors for the chart.

`set_id(id)` Set chart's id assigned by chart manager.

`set_settings(settings)` Set custom settings for the chart. Returns subset of valid settings.

settings List of settings to apply.

`EpivizChartMgr-class` *Class providing chart manager for epiviz app*

Description

Class providing chart manager for epiviz app

Methods

`add_chart(chart_object, send_request = TRUE)` Add a chart to the chart manager.

chart_object an object of class `EpivizChart`

send_request send request to app through websocket

`get_chart_colors(chart_object_or_id)` List colors used in a specific chart object.

chart_object_or_id An object of class `EpivizChart` or a string indicating the chart's id assigned by chart manager

`get_chart_settings(chart_object_or_id)` List chart settings for a specific chart object.

chart_object_or_id An object of class `EpivizChart` or a string indicating the chart's id assigned by chart manager

`is_server_closed()` Returns `TRUE` if underlying server is closed. See `is_closed` method in class `EpivizServer`.

`list_chart_type_colors(chart_type)` List colors currently used in given chart type

`list_chart_type_settings(chart_type)` List available settings for a specific chart type.

chart_type the name for a chart type in R (e.g., 'BlocksTrack', 'StackedLineTrack')

`list_chart_types(col_width = 80)` List charts types registered in epivizr with their default settings and colors. Returns a `data.frame` listing available chart types and a summary of the settings that can be modified.

col_width Maximum length of settings list displayed.

`list_charts()` Return `data.frame` describing charts loaded by chart manager

`num_charts()` Return the number of charts currently loaded through manager.

`plot(measurement_object, settings = NULL, colors = NULL, send_request = TRUE)` Visualize data in an `EpivizData` object using its default chart type. The method `get_default_chart_type` in class `EpivizData` is used to determine which chart type is used.

measurement_object an object of class `EpivizData`

settings list of settings to use in chart (uses default chart settings if NULL)

colors character vector of HEX colors to use in chart (uses default chart colors if NULL)

`print_chart(chart_object_or_id, file_name = NULL, file_type = c("pdf", "png"))` Print specific chart to file.

chart_object_or_id An object of class `EpivizChart` or a string indicating the chart's id assigned by chart manager.

file_name Name of file to print to.

file_type Type of file to print. One of `pdf` or `png`.

`print_chart_info(chart_object_or_id)` Print settings and colors used in a specific chart object.

chart_object_or_id An object of class `EpivizChart` or a string indicating the chart's id assigned by chart manager

`print_chart_type_info(chart_type)` Print settings and color information for a given chart type

chart_type the name for a chart type in R (e.g., 'BlocksTrack', 'StackedLineTrack')

`register_chart_type( chart_type, js_chart_type = paste0("epiviz.plugins.charts.", chart_type), js_char` Register a chart type name to a JavaScript chart type in the epiviz app.

chart_type the name to use for chart type in R (e.g., 'BlocksTrack')

js_chart_type the full JavaScript class name of the corresponding chart type (e.g. 'epiviz.plugins.charts.BlocksTrack'). If missing it is taken from the `chart_type` argument

js_chart_settings custom settings that can be applied to this chart type in JS.

js_chart_colors default color palette applied to this chart type in JS.

`revisualize(chart_type, chart, ...)` Revisualize uses data the given chart and replots it in the given chart type. The argument `chart` must be non-`NULL`.

chart_type a chart type registered using the `register_chart_type` method

chart a chart that includes measurements to be used in a new chart

`rm_all_charts()` Remove all charts loaded by chart manager.

`rm_chart(chart_object_or_id)` Remove chart from chart manager.

chart_object_or_id An object of class `EpivizChart` or a string indicating the chart's id assigned by chart manager

`set_chart_settings(chart_object_or_id, settings = NULL, colors = NULL)` Apply custom chart settings or colors to a chart object.

chart_object_or_id An object of class `EpivizChart` or a string indicating the chart's id assigned by chart manager

settings a list of settings to apply to the chart

colors a list of (HEX code) colors to use in the chart

`visualize( chart_type, measurements = NULL, datasource = NULL, settings = NULL, colors = NULL, send_reques`
Visualize data use the given chart type. One of arguments `measurements` or `datasource` must be non-`NULL`. If `measurements` is `NULL`, the `get_measurements` method in class `EpivizData` is used to decide which measurements are used in the chart

chart_type a chart type registered using the `register_chart_type` method

measurements a list of `EpivizMeasurement` objects describing measurements to include in the chart

datasource an object of class `EpivizData`, all available measurements from datasource are used as appropriate

---

`restartEpiviz` *Restart epiviz app connection and workspace.*

Description

Restart epiviz app connection and workspace.

Usage

```
restartEpiviz(file, open_browser = TRUE, host = NULL, envir = parent.frame())
```

Arguments

`file`	(character) The name of the file that holds the EpivizApp object to be restarted, ending in .rda.
`open_browser`	(logical) browse to the epiviz URL before exiting function.
`host`	(character) name of epiviz app host to open at restart
`envir`	(environment) environment in which to evaluate expressions needed to reconstruct data sources

Value

An object of class `EpivizApp`

Examples

```
# see package vignette for example usage
app <- startEpiviz(non_interactive=TRUE, open_browser=TRUE)
file_name <- tempfile(fileext=".rda")
app$save(file=file_name)
app$stop_app()

app <- restartEpiviz(file=file_name, open_browser=FALSE)
```

show,EpivizChart-method

Print information about EpivizChart

Description

Print information about EpivizChart

Usage

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

Arguments

object	an EpivizChart object

Value

Nothing, this function is called for its side-effects

startEpiviz

Start epiviz app and create EpivizApp object to manage connection.

Description

Start epiviz app and create EpivizApp object to manage connection.

Usage

```
startEpiviz(
  host = NULL,
  http_port = NULL,
  path = NULL,
  use_devel = FALSE,
  chr = "chr11",
  start = 99800000,
  end = 103383180,
  debug = FALSE,
  workspace = NULL,
  scripts = NULL,
  gists = NULL,
  use_cookie = TRUE,
  register_function = .register_all_the_epiviz_things,
  open_browser = TRUE,
```

```
  server = NULL,
  browser_fun = utils::browseURL,
  ws_host = "localhost",
  ...
)
```

Arguments

host	(character) use a host for the epiviz app other than the cbcb.umd.edu hosts.
http_port	(integer) port at host serving the epiviz app.
path	(character) path at host where epiviz app is located.
use_devel	(logical) use the devel epiviz application server (http://epiviz-dev.cbcb.umd.edu).
chr	(character) chromosome to browse to on app startup.
start	(integer) start location to browse to on app startup.
end	(integer) end location to browse to on app startup.
debug	(logical) start epiviz app in debug mode.
workspace	(character) a workspace id to load in the epiviz app on startup.
scripts	(character) URLs for JavaScript plugin scripts to be imported when epiviz is loaded (see http://epiviz.cbcb.umd.edu/help for details).
gists	(character) Ids for github gists (http://gist.github.com) containing JavaScript plugin scripts to be imported when epiviz is loaded (see http://epiviz.cbcb.umd.edu/help for details).
use_cookie	(logical) use cookies within the epiviz app.
register_function	(function) function used to register actions and charts on the epiviz app.
open_browser	(logical) browse to the epiviz URL before exiting function.
server	(EpivizServer) if not NULL use this object as underlying WebSocket and HTTP server
browser_fun	(function) function used to browse to URL (browseURL by default)
ws_host	(character) host address to use for websocket connection ("localhost" by default)
...	additional parameters passed to createServer.

Value

An object of class EpivizApp

Examples

```
# see package vignete for example usage
app <- startEpiviz(non_interactive=TRUE, open_browser=TRUE)
app$stop_app()
```

tcga_colon_blocks *Example methylation data (blocks) for epivizr vignette.*

Description

Example results from methylation analysis of human chromosome 11 using the minfi::minfi-package package of TCGA 450k beadarray samples. This object contains large regions of methylation difference between tumor and normal samples obtained from minfi::blockFinder().

Usage

```
data(tcga_colon_blocks)
```

Format

A GRanges object with 129 and mcols:

value average smooth methylation difference within block

area block area estimate (abs(value) * length)

cluster id of cluster blockgroup within which block occurs

indexStart index of first cluster in block

indexEnd index of last cluster in block

L number of clusters in block

clusterL number of probes in block

p.value permutation p.value based on difference conditioned on length

fwer family-wise error rate estimate based on difference conditioned on length

p.valueArea permutation p.value based on area

fwerArea family-wise error rate estimate based on area

Source

TCGA project: https://tcga-data.nci.nih.gov/tcga/

tcga_colon_curves *Example methylation data (smoothed methylation levels) for epivizr vignette*

Description

Example results from methylation analysis of human chromosome 11 using the `minfi::minfi-package` package of TCGA 450k beadarray samples. This object contains probe cluster level methylation estimates from `minfi::blockFinder()`.

Usage

```
data(tcga_colon_curves)
```

Format

A `GRanges` object with 7135 ranges and `mcols`:

`id` probe cluster id

`type` probe cluster type

`blockgroup` probe cluster block group

`diff` raw methylation percentage difference between normal and tumor

`smooth` smooth methylation percentage difference between normal and tumor

`normalMean` mean methylation estimate for normal samples

`cancerMean` mean methylation estimate for cancer samples

Source

TCGA project: `https://tcga-data.nci.nih.gov/tcga/`

tcga_colon_expression *Example exon-level RNAseq data from TCGA project for epivizr vignette.*

Description

A `SummarizedExperiment::RangedSummarizedExperiment` object containing exon-level counts from RNAseq data for colon tumor and normal tissue from the TCGA project. Only exons in human chromosome 11 are included.

Usage

```
data(tcga_colon_expression)
```

Format

A SummarizedExperiment::RangedSummarizedExperiment object with 12,800 rows (exons) and 40 samples.

`assay(tcga_colon_expression)` exon-level count matrix

`colData(tcga_colon_expression)` a `DataFrame` containing sample information. Normal/Tumor status is given in column `sample_type`

Value

A SummarizedExperiment::RangedSummarizedExperiment object.

Source

TCGA project: https://tcga-data.nci.nih.gov/tcga/

Index

createServer, *8*

EpivizApp, *6–8*
EpivizApp *(*EpivizApp-class*)*, 2
EpivizApp-class, 2
EpivizChart, *4, 5*, *7*
EpivizChart *(*EpivizChart-class*)*, 3
EpivizChart-class, 3
EpivizChartMgr, *2, 3*
EpivizChartMgr *(*EpivizChartMgr-class*)*, 4
EpivizChartMgr-class, 4
EpivizData, *2, 3*, *5, 6*
EpivizDataMgr, *2*
EpivizMeasurement, *6*
EpivizServer, *2*, *4*

GenomicRanges, *3*
GRanges, *9, 10*

minfi::blockFinder(), *9, 10*

plot, *3*

register, *3*
restartEpiviz, 6

show,EpivizChart-method, 7
startEpiviz, 7
SummarizedExperiment::RangedSummarizedExperiment, *10, 11*

tcga_colon_blocks, 9
tcga_colon_curves, 10
tcga_colon_expression, 10