

# Package ‘metavizr’

October 18, 2017

**Type** Package

**Version** 1.0.2

**Maintainer** Hector Corrada Bravo <hcorrada@gmail.com>

**License** MIT + file LICENSE

**Title** R Interface to the metavizr web app for interactive metagenomics data analysis and visualization

**Description** This package provides Websocket communication to the metavizr web app (<http://metavizr.cbcb.umd.edu>) for interactive visualization of metagenomics data. Objects in R/bioc interactive sessions can be displayed in plots and data can be explored using a facetzoom visualization. Fundamental Bioconductor data structures are supported (e.g., MRExperiment objects), while providing an easy mechanism to support other data structures. Visualizations (using d3.js) can be easily added to the web app as well.

**VignetteBuilder** knitr

**Depends** R (>= 3.3), metagenomeSeq (>= 1.17.1), methods, data.table, Biobase, digest

**Imports** epivizr, epivizrData, epivizrServer, epivizrStandalone, vegan, GenomeInfoDb, phyloseq, httr

**Suggests** knitr, BiocStyle, matrixStats, msd16s (>= 0.109.1), etec16s, testthat, gss

**Collate** 'metavizrControl.R' 'startMetavizr.R' 'utils.R'  
'EpivizrMetagenomicsData-class.R' 'register-methods.R'  
'validateMRExperiment.R' 'MetavizrApp-class.R'  
'MetavizrGraph-class.R'

**biocViews** Visualization, Infrastructure, GUI, Metagenomics

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Hector Corrada Bravo [cre, aut],  
Florin Chelaru [aut],  
Justin Wagner [aut],  
Jayaram Kancherla [aut],  
Joseph Paulson [aut]

**R topics documented:**

buildMetavizGraph . . . . .	2
EpivizMetagenomicsData-class . . . . .	3
generateSelection . . . . .	4
MetavizApp-class . . . . .	5
metavizControl . . . . .	5
MetavizGraph-class . . . . .	6
register,MRExperiment-method . . . . .	7
register,phyloseq-method . . . . .	7
startMetaviz . . . . .	8
startMetavizStandalone . . . . .	8
validateObject . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

buildMetavizGraph	<i>Build a MetavizTree object from another object</i>
-------------------	---

---

**Description**

Build a MetavizTree object from another object

**Usage**

```
buildMetavizGraph(object, ...)

## S4 method for signature 'MRExperiment'
buildMetavizGraph(object, feature_order, ...)
```

**Arguments**

object	The object from which taxonomy data is extracted
...	Additional arguments
feature_order	Ordering of leaves (features) in taxonomy tree

**Value**

a [MetavizGraph](#) object

**Methods (by class)**

- MRExperiment: Build graph from a [MRExperiment-class](#) object

---

 EpivizMetagenomicsData-class

*Data container for MExperiment objects*


---

## Description

Used to serve metagenomic data (used in e.g., icicle plots and heatmaps). Wraps [MExperiment-class](#) objects.

## Methods

`df_to_tree(root, df)` Helper function to recursively build nested response for `getHierarchy`

**root** Root of subtree

**df** data.frame containing children to process

`getAlphaDiversity(measurements = NULL)` Compute alpha diversity using vegan for the given samples

**measurements** Samples to compute alpha diversity

**start** Start of feature range to query

**end** End of feature range to query

`getCombined(measurements = NULL, seqName, start = 1, end = 1000, order = NULL, nodeSelection = NULL)` Return the counts aggregated to selected nodes for the given samples

**measurements** Samples to get counts for

**seqName** name of datasource

**start** Start of feature range to query

**end** End of feature range to query

**order** Ordering of nodes

**nodeSelection** Node-id and selectionType pairs

**selectedLevels** Current aggregation level

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

`getHierarchy(nodeId = NULL)` Retrieve feature hierarchy information for subtree with specified root

**nodeId** Feature identifier with level info

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

`getPCA(measurements = NULL)` Compute PCA over all features for given samples

**measurements** Samples to compute PCA over

**start** Start of feature range to query

**end** End of feature range to query

`getRows(measurements = NULL, start = 1, end = 1000, selectedLevels = 3, selections = NULL)` Return the sample annotation and features within the specified range and level for a given sample and features

**measurements** Samples to retrieve for

**start** Start of feature range to query

**end** End of feature range to query

**selections** Node-id and selectionType pairs

**selectedLevels** Current aggregation level

getValues(measurements = NULL, start = 1, end = 1000, selectedLevels = 3, selections = NULL)  
Return the counts for a sample within the specified range

**measurements** Samples to get counts for

**start** Start of feature range to query

**end** End of feature range to query

**selections** Node-id and selectionType pairs

**selectedLevels** Current aggregation level

propagateHierarchyChanges(selection = NULL, order = NULL, selectedLevels = NULL, request\_with\_la  
Update internal state for hierarchy

**selection** Node-id and selectionType pairs

**order** Ordering of features

**selectedLevels** Current aggregation level

**request\_with\_labels** For handling requests using fData entries from MRExperiment

row\_to\_dict(row) Helper function to format each node entry for getHierarchy response

**row** Information for current node.

searchTaxonomy(query = NULL, max\_results = 15) Return list of features matching a text-  
based query

**query** String of feature for which to search

**max\_results** Maximum results to return

toNEO4JDbHTTP(batch\_url, neo4juser, neo4jpass, datasource, description = NULL) Write  
an 'EpivizMetagenomicsData' object to a Neo4j graph database

@param batch\_url (character) Neo4j database url and port for processing batch http requests

@param neo4juser (character) Neo4j database user name @param neo4jpass (character) Neo4j  
database password @param datasource (character) Name of Neo4j datasource node for this  
'EpivizMetagenomicsData' object

@examples library(metagenomeSeq) data("mouseData") mobj <- metavizr:::EpivizMetagenomicsData\$new(object  
mobj\$toNEO4JDbHTTP(batch\_url = "http://localhost:7474/db/data/batch", neo4juser = "neo4juser",  
neo4jpass = "neo4jpass", datasource = "mouse\_data")

update(new\_object, send\_request = TRUE) Update underlying data object with new object

## Examples

```
library(metagenomeSeq)
data(mouseData)
obj <- metavizr:::EpivizMetagenomicsData$new(mouseData, feature_order = colnames(fData(mouseData)))
```

---

generateSelection      *Method to select and set aggregation type to nodes in FacetZoom*

---

## Description

Method to select and set aggregation type to nodes in FacetZoom

**Usage**

```
generateSelection(feature_names, aggregation_level, selection_type,
  feature_order = NULL)
```

**Arguments**

```
feature_names  Selected Features
aggregation_level
                Level in the hierarchy
selection_type Expanded, aggregated, or removed
feature_order  Order of features at that level
```

**Value**

A selection object for a metavizControl object to accept

**Examples**

```
generateSelection("Bacteroidales", 1L, 2L)
```

---

MetavizApp-class	<i>Class managing connection to metaviz application.</i>
------------------	--

---

**Description**

Class managing connection to metaviz application.

---

metavizControl	<i>metavizr settings</i>
----------------	--------------------------

---

**Description**

Default settings for the various plotting functions in metavizr.

**Usage**

```
metavizControl(aggregateAtDepth = 3, aggregateFun = function(x) colSums(x),
  valuesAnnotationFuns = NULL, maxDepth = 4, maxHistory = 3,
  maxValue = NULL, minValue = NULL, title = "", n = 10000,
  rankFun = stats::sd, norm = TRUE, log = FALSE,
  featureSelection = NULL)
```

**Arguments**

<code>aggregateAtDepth</code>	Level of the tree to aggregate counts at by default.
<code>aggregateFun</code>	Function to aggregate counts by at the <code>aggregateAtDepth</code> level.
<code>valuesAnnotationFuns</code>	Function for error bars.
<code>maxDepth</code>	Level of the tree to display by default in icicle view.
<code>maxHistory</code>	Value for caching.
<code>maxValue</code>	Maximum value to display.
<code>minValue</code>	Minimum value to display.
<code>title</code>	title.
<code>n</code>	Number of OTUs to include in ranking.
<code>rankFun</code>	Ranking function - single vector function.
<code>norm</code>	Normalize MRExperiment object.
<code>log</code>	Log tranformation of MRExperiment object.
<code>featureSelection</code>	List of features to set as <code>nodeSelections</code>

**Value**

List of setting parameters.

**Examples**

```
settings = metavizControl()
```

---

`MetavizGraph-class`      *Graph implementation to query hierarchical feature data*

---

**Description**

Used to manage aggregation and range queries from the Metaviz app UI.

---

register,MRexperiment-method

*Generic method to register data to the epiviz data server*

---

### Description

Generic method to register data to the epiviz data server

### Usage

```
## S4 method for signature 'MRexperiment'  
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

### Value

An [EpivizMetagenomicsData-class](#) object

---

register,phyloseq-method

*Generic method to register data to the epiviz data server*

---

### Description

Generic method to register data to the epiviz data server

### Usage

```
## S4 method for signature 'phyloseq'  
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

### Value

An [phyloseq-class](#) object

---

startMetaviz	<i>Start metaviz app and create <a href="#">MetavizApp</a> object to manage connection.</i>
--------------	---

---

**Description**

Start metaviz app and create [MetavizApp](#) object to manage connection.

**Usage**

```
startMetaviz(host = "http://metaviz.cbcb.umd.edu",
  register_function = .register_all_metaviz_things, ...)
```

**Arguments**

host	(character) host address to launch.
register_function	(function) function used to register actions and charts on the metaviz app.
...	additional parameters passed to <a href="#">startEpiviz</a> .

**Value**

An object of class [MetavizApp](#)

**See Also**

[MetavizApp](#)

**Examples**

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

---

startMetavizStandalone	<i>Start metaviz app in standalone (locally) and create <a href="#">MetavizApp</a> object to manage connection.</i>
------------------------	---

---

**Description**

Start metaviz app in standalone (locally) and create [MetavizApp](#) object to manage connection.

**Usage**

```
startMetavizStandalone(branch = "metaviz-4.1",
  register_function = .register_all_metaviz_things, ...)
```



**Arguments**

branch (character) branch to pull from metaviz github repo to run standalone.  
register\_function (function) function used to register actions and charts on the metaviz app.  
... additional parameters passed to [startStandalone](#).

**Value**

An object of class [MetavizApp](#)

**Examples**

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

---

validateObject      *validate* [MRExperiment-class](#) object

---

**Description**

validate [MRExperiment-class](#) object

**Usage**

```
validateObject(object)
```

**Arguments**

object an object of class [MRExperiment-class](#)

**Value**

TRUE or FALSE

**Examples**

```
library(metagenomeSeq)  
data(mouseData)  
validateObject(mouseData)
```

# Index

buildMetavizGraph, 2  
buildMetavizGraph, MRexperiment-method  
(buildMetavizGraph), 2

EpivizMetagenomicsData  
(EpivizMetagenomicsData-class),  
3  
EpivizMetagenomicsData-class, 3

generateSelection, 4

MetavizApp, 8, 9  
MetavizApp (MetavizApp-class), 5  
MetavizApp-class, 5  
metavizControl, 5  
MetavizGraph, 2  
MetavizGraph (MetavizGraph-class), 6  
MetavizGraph-class, 6  
MRexperiment-class, 9

register, MRexperiment-method, 7  
register, phyloseq-method, 7

startEpiviz, 8  
startMetaviz, 8  
startMetavizStandalone, 8  
startStandalone, 9

validateObject, 9