

explorase

February 9, 2012

`exp_addEntityType` *Add an entity type*

Description

Adds an entity type of the given name with a plural human-readable label.

Usage

```
exp_addEntityType(ent_type, label = paste(ent_type, "s", sep=""))
```

Arguments

<code>ent_type</code>	The unique identifier of the entity type
<code>label</code>	The plural label used for the tab in the metadata notebook.

Details

When an entity type is added to `exploRase`, a new tab is added to the metadata notebook with the provided plural label. Data and metadata corresponding to the type may then be loaded into `exploRase`.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_addFilterRule *Add a filter rule*

Description

Adds a filter rule to the exploRase filter model for the given entity type

Usage

```
exp_addFilterRule(id, column, op, expr, active = T, ent_type = exp_entityType())
```

Arguments

id	A unique identifier for the filter rule
column	The metadata column checked by the rule
op	The operator (<, >, ==, etc) used for checking the values
expr	The right hand expression against which the values are checked
active	Whether the rule should be immediately active
ent_type	The entity type of the metadata being filtered

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_calcAngleDist *Caclulate angle distance*

Description

Calculates angle distance between sample vectors x and y

Usage

```
exp_calcAngleDist(x, y)
```

Arguments

x	normally a column from the experimental data matrix
y	normally a column from the experimental data matrix

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

```
exp_calcCanberraDist
```

Calculate Canberra distance

Description

Calculates Canberra distance between `ent` and the other entities across the conditions the samples (columns) in the data frame `ent_data`

Usage

```
exp_calcCanberraDist(ent_data, ent)
```

Arguments

<code>ent_data</code>	a data frame of experimental data, according to <code>exploRase</code> conventions
<code>ent</code>	the id of an entity that is compared to the entities in <code>ent_data</code>

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

```
exp_calcCorrelationDist
```

Calculate correlation distance

Description

Calculates correlation distance between `ent` and the other entities across the conditions the samples (columns) in the data frame `ent_data`

Usage

```
exp_calcCorrelationDist(ent_data, ent)
```

Arguments

<code>ent_data</code>	a data frame of experimental data, according to <code>exploRase</code> conventions
<code>ent</code>	the id of an entity that is compared to the entities in <code>ent_data</code>

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

`exp_calcDiff`*Calculate difference*

Description

Just calculates $y - x$. You're probably better off just doing that.

Usage

```
exp_calcDiff(x, y)
```

Arguments

<code>x</code>	normally a column from the experimental data matrix
<code>y</code>	normally a column from the experimental data matrix

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

`exp_calcEuclideanDist`*Calculate Euclidean distance*

Description

Calculates Euclidean distance between `ent` and the other entities across the samples (columns) in the data frame `ent_data`

Usage

```
exp_calcEuclideanDist(ent_data, ent)
```

Arguments

<code>ent_data</code>	a data frame of experimental data, according to <code>exploRase</code> conventions
<code>ent</code>	the id of an entity that is compared to the entities in <code>ent_data</code>

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

`exp_calcMahalanobisDist` *Calculate Mahalanobis distance*

Description

Calculates mahalanobis distance between the samples (columns) in the data frame `ent_data`

Usage

```
exp_calcMahalanobisDist (ent_data)
```

Arguments

`ent_data` a data frame of experimental data, according to `exploRase` conventions

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

`exp_calcResiduals` *Calculate residuals*

Description

Calculates the residuals from a linear regression of sample `y` against sample `x`

Usage

```
exp_calcResiduals (x, y)
```

Arguments

`x` normally a column from the experimental data matrix

`y` normally a column from the experimental data matrix

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_calcZeroCorDist

Calculate zero-correlation distance

Description

Calculates zero (uncentered) correlation distance between `ent` and the other entities across the samples (columns) in the data frame `ent_data`

Usage

```
exp_calcZeroCorDist(ent_data, ent)
```

Arguments

`ent_data` a data frame of experimental data, according to `exploRase` conventions
`ent` the id of an entity that is compared to the entities in `ent_data`

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_close

Close exploRase

Description

Close (shutdown) `exploRase`

Usage

```
exp_close()
```

Author(s)

Michael Lawrence

exp_colorEntities *Coloring entities*

Description

Set the color for the specified entities of the current entity type in GGobi

Usage

```
exp_colorEntities(entities = getEntityIds(), color)
```

Arguments

entities the entities to color
color the color index (as interpreted by GGobi) for the entities

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_dataset *Get experimental data*

Description

Get experimental data of the specified type out of exploRase

Usage

```
exp_dataset(ent_type = exp_entityType())  
exp_dataFrame(ent_type = exp_entityType())
```

Arguments

ent_type the entity type ("gene", "met", ...) of the data

Details

The result of `exp_dataset` allows one to directly manipulate the GGobi dataset. If only a `data.frame` is required, `exp_dataFrame` is a more convenient path.

Value

For `exp_dataset`, a GGobi dataset, see `rggobi` documentation. For `exp_dataFrame`, a `data.frame` coerced from the result of `exp_dataset`.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_designFactors *Get the exp. design factors*

Description

Gets a vector of the names of the factors in the experimental design

Usage

```
exp_designFactors(ent_type = exp_entityType(), treatments_only = FALSE)
```

Arguments

ent_type the entity type ("gene", "met", etc)
treatments_only if TRUE, only include treatments (eg genotype), leaving out eg ID and replicate.

Value

A vector of factor names

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_designFrame *Get the exp. design*

Description

Gets a data frame containing the experimental design information for the given entity type.

Usage

```
exp_designFrame(ent_type = exp_entityType(), treatments_only = FALSE)
```

Arguments

ent_type the entity type ("gene", "met", etc)
treatments_only if TRUE, only include treatment columns (eg genotype), leaving out eg ID and replicate.

Value

A data frame with conditions as rows and factors as columns.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

`exp_designSelection`

Get selected conditions

Description

Gets a vector of the names of the selected conditions in the condition list.

Usage

```
exp_designSelection()
```

Value

A vector of condition names

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

`exp_entitiesInList` *Get the entities in a list*

Description

Gets the ID's of the entities in a specified list and of the specified types.

Usage

```
exp_entitiesInList(list, types = exp_entityTypes())
```

Arguments

<code>list</code>	The name of the entity list
<code>types</code>	The types of entities to return

Value

The entity ID's of the specified types belonging to the list

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_entityFrame *Get entity metadata*

Description

Retrieves the entity metadata table for the given type as a data frame

Usage

```
exp_entityFrame(ent_type = exp_entityType())
```

Arguments

ent_type the entity type for which the metadata is retrieved

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_entitySelection
 Get selected entities

Description

Gets the ID's of the entities selected in the metadata table of the given type.

Usage

```
exp_entitySelection(ent_type = exp_entityType())
```

Arguments

ent_type The entity type for which the selected is retrieved

Value

The entity ID's

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_entityType *Get the current entity type*

Description

Retrieves the entity type with its tab selected in the metadata notebook.

Usage

```
exp_entityType()
```

Value

Name of the current entity type

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_entityTypes *Get all entity types*

Description

Retrieves the identifiers for all entity types in exploRase.

Usage

```
exp_entityTypes()
```

Value

Vector of entity type names

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_filterRules *Get filter rules*

Description

Gets the filter rules for the metadata of the given type

Usage

```
exp_filterRules(ent_type = exp_entityType())
```

Arguments

ent_type the entity type of the metadata being filtered by the rules

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_findPatterns *Find Patterns*

Description

Finds patterns in data. Transitions within range of 'fraction' (centered on median) are considered flat (unchanging). Those below are falling and those above are rising.

Usage

```
exp_findPatterns(data, flat_fraction)
```

Arguments

data A data frame of experimental data according to `exploRase` conventions.

flat_fraction

The fraction of transitions considered unchanged, centered on median.

Value

a data frame, with a row for each gene. The first column is the sum of x^i over all i from 1 to `ncol(data) - 1`, where x is 1, 2, or 3, depending on whether the pattern is up, same, or down, respectively, for transition i . The second column contains the magnitude of the pattern.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_ggobi *Get the GGobi session*

Description

Gets the GGobi session being used by exploRase. Will restart GGobi if it died

Usage

```
exp_ggobi ()
```

Value

The GGobi session

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_isRunning *Check whether exploRase is running*

Description

Currently a hack that checks if the main window exists

Usage

```
exp_isRunning ()
```

Value

whether exploRase is currently running

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_listFrame *Get entity lists*

Description

Gets the entity lists loaded in exploRase

Usage

```
exp_listFrame()
```

Value

A data frame with a single column "name" holding the names of the lists

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_listSelection *Get selected lists*

Description

Gets the names of the lists selected in the exploRase GUI.

Usage

```
exp_listSelection()
```

Value

The names of the selected lists

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_loadData *Load experimental data*

Description

Load experimental data of a specified type into exploRase (and GGobi).

Usage

```
exp_loadData(exp_data, data_name = "expression", ent_type =  
"gene", sync = FALSE, add_to_design = !nrow(exp_designFrame(ent_type)))
```

Arguments

exp_data	a data frame containing experimental data, with observations as rows and conditions as columns.
data_name	name of the dataset in GGobi
ent_type	the entity type ("gene", "met", ...) of the data
sync	whether to synchronize the other data models with the experimental data. If sync is TRUE, the entity information and experimental design tables will be limited to the observations and conditions, respectively, in the experimental data. This is experimental.
add_to_design	whether to add the columns in the dataset as conditions in the experimental design, if they are not there already

Details

Organizes experimental data, loads it into GGobi and synchronizes it with other data (design and entity info)

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_loadDesign *Load experimental design*

Description

Loads a matrix describing the experimental conditions

Usage

```
exp_loadDesign(design_info, ent_type = exp_entityType())
```

Arguments

design_info a data frame with conditions for rows and factors for cols

ent_type the type of entity to which this design applies. This links This links the design to the experimental data and entity information.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_loadFiles *Load files*

Description

Loads a set of files into exploRase

Usage

```
exp_loadFiles(filenamees, data_type = NULL, entity_type = "gene",
              ignoreUnknown = TRUE)
```

Arguments

filenamees filenamees The paths to the files to load

data_type The data type ("data", "design", "info", "list")

entity_type The entity type (by default: "gene", "met", "prot"), only used if data_type is provided.

ignoreUnknown If FALSE, throw an error if the data/entity type cannot be determined for a file; otherwise, the file is ignored.

Details

If the data_type is specified, it is assumed that all the files are of the given data_type and entity_type. Otherwise, the types are autodetected based on file extensions.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_loadInfo	<i>Load entity metadata</i>
--------------	-----------------------------

Description

Loads a data frame of entity metadata into the metadata table of the given type(s).

Usage

```
exp_loadInfo(ent_info, ent_types = "gene", append_col = TRUE, keywords = NULL, u
```

Arguments

ent_info	the data frame of entity information
ent_types	a single type identifier (applies to all rows) or a
append_col	whether the column should be appended or inserted after ID
keywords	a vector of identifiers that describe the added metadata
update_view	whether the metadata view should be updated
sync	whether metadata rows should be filtered out if they don't

Details

The loaded entity metadata is merged with any existing data and the view is optionally updated to reflect the change. The new set of columns may be appended to the end or inserted just after the "ID" column.

Metadata for multiple entity types may be added simultaneously, if `ent_types` is a vector with the same number of elements as the number of rows in `ent_info` and specifies the type of the entity described by each row.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_loadLists	<i>Load entity lists</i>
---------------	--------------------------

Description

Loads a list of entity lists (matrices) into `exploRase`

Usage

```
exp_loadLists(ent_lists)
```

Arguments

ent_lists	a list of entity lists (1 or 2 column matrices)
-----------	---

Details

An entity list matrix may have one or two columns. The last column specifies the entity ID's and its name is the name of the entity list. If there are two columns, the first specifies the type of each entity, allowing entity lists holding entities of different types.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_loadProject *Load a project*

Description

Loads a project (a file system directory) into exploRase

Usage

exp_loadProject (project)

Arguments

project The path to the directory holding the project

Details

Loads all of the files in a specified directory, using their file extensions to determine their purpose.

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_newList *Create an entity list*

Description

Forms the actual entity list matrix from the name, ents, and types

Usage

exp_newList (name, ents, types)

Arguments

name The name of the entity list
 ents The ID's of the entities in the list
 types The types of the entities in the list

Value

A matrix conforming to the structure for entity lists

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_removeFilterRules
Remove filter rules

Description

Removes the filter rules with the given identifiers for the given type

Usage

```
exp_removeFilterRules(rules, ent_type = exp_entityType())
```

Arguments

rules	the identifiers of the rules to remove
ent_type	the entity type of the metadata being filtered

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_showClustering *Show a hierarchical clustering*

Description

Shows a hierarchical clustering using the "GGobi" dendrogram viewer.

Usage

```
exp_showClustering(ids, data, clustering)
```

Arguments

ids	The ids of the entities that were clustered
data	The experimental data that was clustered
clustering	the clustering, as returned by hclust()

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_showPatterns *Show patterns*

Description

Show the calculated patterns in the GUI

Usage

```
exp_showPatterns(patterns, desc, samples = exp_designSelection())
```

Arguments

patterns	a data frame, with a row for each gene and the first column being the pattern codes and the second the magnitude of the pattern (as returned by <code>exp_findPatterns</code>).
desc	a description of the patterns, for labeling them in the GUI
samples	the samples involved in the calculation, for labeling

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_showResults *Show analysis results*

Description

Add a column of analysis results to the exploRase table and the GGobi dataset.

Usage

```
exp_showResults(results, label, sublabels = "", types = exp_entityType(), keyword)
```

Arguments

results	the data frame of results. To include results for multiple entity types at once, the result for each type should be concatenated in the same order as the <code>types</code> parameter.
label	a root label for the result (like the type of analysis)
sublabels	other labels concatenated to the root (such as the conditions involved)
types	the entity type(s) of the data from which this result was derived.
keyword	a keyword identifying this result, for use in context-sensitive help
explorase	whether to show the result in the explorase table
ggobi	whether to add the result to the GGobi dataset

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

exp_toggleFilterRules
Toggle filter rules

Description

Toggles (activates or deactivates) the specified rules for the specified type.

Usage

```
exp_toggleFilterRules(rules, active = T, ent_type = exp_entityType())
```

Arguments

rules	The identifiers of the rules to toggle
active	Whether to activate or deactivate the rules
ent_type	The entity type of the metadata being filtered by the rules

Author(s)

Michael Lawrence <mflawren@fhcrc.org>

bioc_integrations *ExpressionSet Integration*

Description

Functions loading and retrieving Bioconductor ExpressionSet objects to and from exploRase.

Usage

```
exp_loadExpressionSet(exprset, type = exp_entityType())  
exp_phenoData(type = exp_entityType())  
exp_featureData(type = exp_entityType())  
exp_expressionSet(type = exp_entityType())
```

Arguments

exprset	The ExpressionSet to load.
type	The type of the biological entity (e.g. gene).

Details

To load a ExpressionSet instance into exploRase, call exp_loadExpressionSet.

The function exp_expressionSet retrieves the ExpressionSet representing the data stored in exploRase for the given entity type. exp_phenoData and exp_featureData retrieve only the phenoData and featureData components, respectively.

Value

For `exp_expressionSet`, an `ExpressionSet`. For `exp_phenoData`, an `AnnotatedDataFrame` representing the phenotype data (i.e. experimental design, see [exp_designFrame](#)). For `exp_featureData`, an `AnnotatedDataFrame` representing the feature annotations (i.e. the entity info, see [exp_entityFrame](#)).

Author(s)

Michael Lawrence

See Also

[explorase](#) for loading data and starting `exploRase` in one call. `exp_designFrame`, `exp_entityFrame`, `exp_dataset`, etc for retrieving data from `exploRase`.

explorase

Start exploRase

Description

The main function of `exploRase`, normally invoked without arguments, unless one already has data/metadata in the R session to load.

Usage

```
explorase(exp_data = NULL, entity_info = NULL, design_info = NULL, type = "gene",
network = NULL, entity_lists = NULL, gobi = ggobi_get(), quit_on_exit = F)
```

Arguments

<code>exp_data</code>	Experimental data to load
<code>entity_info</code>	Entity metadata to load
<code>design_info</code>	Experimental design information to load
<code>type</code>	The entity type of the data being loaded
<code>network</code>	A Bioconductor "graph" to load (not yet implemented)
<code>entity_lists</code>	A list of entity lists to load
<code>gobi</code>	The GGobi session to use
<code>quit_on_exit</code>	Whether R should quit when <code>exploRase</code> is exited

Author(s)

Michael Lawrence

Index

*Topic **IO**

- exp_loadFiles, 16
- exp_loadProject, 18

*Topic **arith**

- exp_calcAngleDist, 2
- exp_calcCanberraDist, 3
- exp_calcCorrelationDist, 3
- exp_calcDiff, 4
- exp_calcEuclideanDist, 4
- exp_calcMahalanobisDist, 5
- exp_calcZeroCorDist, 6
- exp_findPatterns, 12

*Topic **dynamic**

- exp_close, 6
- exp_colorEntities, 7
- exp_ggobi, 13
- exp_showClustering, 19
- exp_showPatterns, 20
- exp_showResults, 20
- explorase, 22

*Topic **manip**

- bioc_integration, 21
- exp_addEntityType, 1
- exp_addFilterRule, 2
- exp_dataset, 7
- exp_designFactors, 8
- exp_designFrame, 8
- exp_designSelection, 9
- exp_entitiesInList, 9
- exp_entityFrame, 10
- exp_entitySelection, 10
- exp_entityType, 11
- exp_entityTypes, 11
- exp_filterRules, 12
- exp_listFrame, 14
- exp_listSelection, 14
- exp_loadData, 15
- exp_loadDesign, 15
- exp_loadInfo, 17
- exp_loadLists, 17
- exp_newList, 18
- exp_removeFilterRules, 19
- exp_showResults, 20

- exp_toggleFilterRules, 21

*Topic **misc**

- exp_isRunning, 13

*Topic **regression**

- exp_calcResiduals, 5

bioc_integration, 21

- exp_addEntityType, 1
- exp_addFilterRule, 2
- exp_calcAngleDist, 2
- exp_calcCanberraDist, 3
- exp_calcCorrelationDist, 3
- exp_calcDiff, 4
- exp_calcEuclideanDist, 4
- exp_calcMahalanobisDist, 5
- exp_calcResiduals, 5
- exp_calcZeroCorDist, 6
- exp_close, 6
- exp_colorEntities, 7
- exp_dataFrame (*exp_dataset*), 7
- exp_dataset, 7
- exp_designFactors, 8
- exp_designFrame, 8, 22
- exp_designSelection, 9
- exp_entitiesInList, 9
- exp_entityFrame, 10, 22
- exp_entitySelection, 10
- exp_entityType, 11
- exp_entityTypes, 11
- exp_expressionSet
(*bioc_integration*), 21
- exp_featureData
(*bioc_integration*), 21
- exp_filterRules, 12
- exp_findPatterns, 12, 20
- exp_ggobi, 13
- exp_isRunning, 13
- exp_listFrame, 14
- exp_listSelection, 14
- exp_loadData, 15
- exp_loadDesign, 15
- exp_loadExpressionSet
(*bioc_integration*), 21

exp_loadFiles, [16](#)
exp_loadInfo, [17](#)
exp_loadLists, [17](#)
exp_loadProject, [18](#)
exp_newList, [18](#)
exp_phenoData (*bioc_integration*),
[21](#)
exp_removeFilterRules, [19](#)
exp_showClustering, [19](#)
exp_showPatterns, [20](#)
exp_showResults, [20](#)
exp_toggleFilterRules, [21](#)
explorase, [22](#), [22](#)