Package ‘Xeva’

January 10, 2017

Type Package

Title Integration of molecular and pharmacological profiles of patient-derived xenograft models

Version 0.1.0

Author Arvind Mer

Maintainer Who to complain to <amer@uhnresearch.ca>

Description More about what it does (maybe more than one line)

License Give me Beer, I will fix it license

LazyData TRUE

RoxygenNote 5.0.1

VignetteBuilder knitr

Suggests knitr

Imports methods,
BBmisc,
Biobase,
plyr,
stringr,
ggplot2,
ComplexHeatmap,
reshape2,
grDevices

R topics documented:

addExperimentalDesign 2
batchNames 3
calculateAngle 3
computemRECIST 4
creatXevaSet 5
drugInfo 6
drugInfo<- 6
expDesign 7
expDesignInfo 7

expDesignInfo<- 8
experimentType 8
getBatchName 9
getControls 9
getExpDesignDF 10
getExperiment 11
getmRECIST 11
getTimeVarData 12
getTreatment 13
mapModelSlotIds 13
ModelInfo 14
ModelInfo<- 14
Old_getTimeVarData 15
pasteColTogather 16
pasteWithoutNA 16
plotDrugResponse 17
plotmRECIST 18
selectModelIds 18
setmRECIST 19
show,XevaSet-method 19
XevaSet-class 20

Index **21**

`addExperimentalDesign` *Add a new experimental design*

Description

Add a new experimental design in expDesign slot.

Usage

```
addExperimentalDesign(object, treatment, control = NULL, batch.name = NULL,
  replace = FALSE)
```

Arguments

`object`	The `Xeva` dataset
`treatment`	The `model.id` of treatment
`control`	The `model.id` of control
`batch.name`	The `batch.name` for new batch
`replace`	If TRUE will replace the old batch with new values

Value

returns `Xeva` dataset with new experimental design added

Examples

```
data(pdxe)
pdxe = addExperimentalDesign(object=pdxe, treatment= c("X.010.BG98"), control=c("X.010.uned"),
                             batch.name="new.batch", replace=FALSE)
```

`batchNames` *Get all batch names*

Description

Get all batch.name from a Xeva dataset

Usage

```
batchNames(object)
```

Arguments

`object`	The `XevaSet` to replace drug info in

Value

A `Vector` with all batch.name

Examples

```
data(pdxe)
batchNames(pdxe)
```

`calculateAngle` *Calculate angle between control and treatment groups*

Description

Given a batch (control and treatment model ids) it will return angle (in Degree) between the linear fit of treatment and control group

Usage

```
calculateAngle(object, ExpDesign, var = "volume", plot = TRUE)
```

Arguments

`object`	The `Xeva` dataset
`ExpDesign`	A list with batch.name, treatment and control
`var`	Name of the variable, default `volume`

Value

a `data.fram` with treatment, control and batch.name

Examples

```
data(pdxe)
# creat a experiment desing
ExpDesign = list(batch.name="myBatch", treatment=c("X.015.BY19"), control=c("X.015.uned"))
df = calculateAngle(object=pdxe, ExpDesign, var = "volume", plot=TRUE)
```

`computemRECIST` *Computes the mRECIST*

Description

`computemRECIST` returns the mRECIST for given volume response

Usage

```
computemRECIST(best.response, best.average.response)
```

Arguments

`best.response` Value of best response

`best.average.response`
Value of best average response

Value

Returns the mRECIST for given volume response

Examples

```
computemRECIST(best.response=8.722, best.average.response=8.722)
```

creatXevaSet *Creat Xeva class object*

Description

`creatPharmacoPxSet` returns Xeva class object

Usage

```
creatXevaSet(name, molecularProfiles = list(), experiment = data.frame(),
  expDesign = list(), model = data.frame(), drug = data.frame())
```

Arguments

`name`	A `character` string detailing the name of the dataset
`molecularProfiles`	A `list` of ExpressionSet objects containing molecular profiles
`experiment`	A `data.frame` containg all experiment information
`model`	A `data.frame` containg the annotations for all models used in the experiment
`drug`	A `data.frame` containg the annotations for all the drugs profiled in the data set, across all data types

Value

Returns Xeva object

Examples

```
geoExp = readRDS("DATA-raw/Geo_Exp.Rda")
pdxe = creatPharmacoPxSet(name = "PDXE",
                          molecularProfiles = list(RNASeq = geoExp$RNASeq),
                          experiment = geoExp$experiment,
                          model = geoExp$model,
                          drug  = geoExp$drug )
save(pdxe, file = "data/pdxe.rda")
data("pdxe")
```

`drugInfo` *drugInfo Generic Generic for drugInfo method*

Description

drugInfo Generic Generic for drugInfo method

Usage

```
drugInfo(object)
```

Arguments

`object`	The `XevaSet` to retrieve drug info from

Value

a `data.frame` with the drug annotations

Examples

```
data(pdxe)
drugInfo(pdxe)
```

`drugInfo<-` *drugInfo<- Generic Generic for drugInfo replace method*

Description

drugInfo<- Generic Generic for drugInfo replace method

Usage

```
drugInfo(object) <- value
```

Arguments

`object`	The `XevaSet` to replace drug info in
`value`	A `data.frame` with the new drug annotations

Value

Updated `XevaSet`

Examples

```
data(pdxe)
drugInfo(pdxe) <- drugInfo(pdxe)
```

`expDesign` *Given a batch.name get batch*

Description

Given a batch.name get batch from a Xeva dataset

Usage

```
expDesign(object, batch.name)
```

Arguments

`object`	The `XevaSet`
`object`	The `batch.name`

Value

A `Vector` with all batch.name

Examples

```
data(pdxe)
expDesign(pdxe, batch.name = "X-6047.paclitaxel")
```

`expDesignInfo` *expDesignInfo Generic Generic for expDesignInfo method*

Description

expDesignInfo Generic Generic for expDesignInfo method

Usage

```
expDesignInfo(object)
```

Arguments

`object`	The `XevaSet` to retrieve drug info from

Value

a `list` with the all experiment designs

Examples

```
data(pdxe)
expDesignInfo(pdxe)
```

expDesignInfo<- *expDesignInfo<- Generic Generic for expDesignInfo replace method*

Description

expDesignInfo<- Generic Generic for expDesignInfo replace method

Usage

```
expDesignInfo(object) <- value
```

Arguments

object	The XevaSet to replace drug info in
value	A list with the experiment designs

Value

Updated XevaSet

Examples

```
data(pdxe)
expDesignInfo(pdxe) <- expDesignInfo(pdxe)
```

experimentType *Get experiment type (treatment or control) for a given model.id*

Description

Get experiment type (treatment or control) for a given model.id

Usage

```
experimentType(object, model.id)
```

Arguments

object	The Xeva dataset
model.id	The model.id

Value

returns treatment or control

Examples

```
data(pdxe)
# get experiment type for model.id
experimentType(object=pdxe, model.id="X.1655.LE11.biib")
experimentType(object=pdxe, model.id="X.1655.uned")
```

getBatchName *Get batch.name for a given model.id*

Description

Get batch.name for a given model.id. If no batch.name found it will return NULL

Usage

```
getBatchName(object, model.id)
```

Arguments

object	The Xeva dataset
model.id	The model.id for which batch name required

Value

a vector with all batch names

Examples

```
data(pdxe)
# extract batch.name for a given model.id
getBatchName(object=pdxe, model.id="X.1655.uned")
getBatchName(object=pdxe, model.id="X.010.fiab")
```

getControls *Get controls for a given model.id*

Description

Get controls for a given model.id. If no control found it will return NULL

Usage

```
getControls(object, model.id)
```

Arguments

`object`	The `Xeva` dataset
`model.id`	The `model.id`

Value

a `vector` with control model.id

Examples

```
data(pdxe)
# extract controls for a given model.id
getControls(object=pdxe, model.id="X.1655.LE11.biib")
# if no control found it will return NULL
getControls(object=pdxe, model.id="X.1655.uned")
```

getExpDesignDF *Given a model.id it will return a data.fram of experiemt design with columns as "treatment", "control", "batch.name"*

Description

Given a model.id it will return a data.fram of experiemt design with columns as "treatment", "control", "batch.name"

Usage

```
getExpDesignDF(object, model.id)
```

Arguments

`object`	The `Xeva` dataset
`model.id`	The `model.id`

Value

a `data.fram` with treatment, control and batch.name

Examples

```
data(pdxe)
# This will give a data.fram with columns as "treatment", "control", "batch.name"
getExpDesignDF(object=pdxe, model.id="X.1655.LE11.biib")
```

getExperiment — *For a given model.id, it will return a data.fram containing all data stored in experiment slot*

Description

For a given model.id, it will return a data.fram containing all data stored in experiment slot

Usage

```
getExperiment(object, model.id)
```

Arguments

`object`	The `XevaSet`
`model.id`	The `model.id` for which data is required

Value

a `data.fram` will all the the values stored in experiment slot

Examples

```
data(pdxe)
getExperiment(pdxe, model.id="X.1004.pael")
```

getmRECIST — *getmRECIST Generic Generic for getmRECIST method*

Description

getmRECIST Generic Generic for getmRECIST method

Usage

```
getmRECIST(object, group.by = "biobase.id")
```

Arguments

`object`	The `XevaSet` to retrieve mRECIST from
`group.by`	The name of column which will be mapped to model.id

Value

a `data.frame` with the mRECIST values, rows are drugs and columns are model.id

Examples

```
data(pdxe)
# calculate mRECIST for each experiment
setmRECIST(pdxe)<- setmRECIST(pdxe)
getmRECIST(pdxe, group.by="biobase.id")
```

---

getTimeVarData *Get time vs volume data with standard error*

Description

Given a batch (treatment and control model ids) it will return a data.fram with time vs volume (or any other variable) with standard error calculated. Note that this function do not check if model.id in given batch belongs to same patient or biobase id Note: Write a function to check integrity of a batch

Usage

```
getTimeVarData(object, ExpDesign, var = "volume")
```

Arguments

object	The Xeva dataset
ExpDesign	A list with batch.name, treatment and control
var	Name of the variable, default volume

Value

a data.fram with treatment, control and batch.name

Examples

```
data(pdxe)
# creat a experiment desing
ExpDesign = list(batch.name="myBatch", treatment=c("X.010.BG98", "X.010.BG98"), control=c("X.010.uned"))
df = getTimeVarData(object=pdxe, ExpDesign, var = "volume")
```

getTreatment *Get treatment for a given model.id*

Description

Get treatment for a given model.id. If no treatment found it will return NULL

Usage

```
getTreatment(object, model.id)
```

Arguments

object	The Xeva dataset
model.id	The model.id

Value

a vector with treatment model.id

Examples

```
data(pdxe)
# extract treatment model.id for a given model.id
getTreatment(object=pdxe, model.id="X.1655.uned")
```

mapModelSlotIds *Map ids of model slot*

Description

Map one id type to another in model slot. For example map a model.id to biobase.id

Usage

```
mapModelSlotIds(object, id, id.name, map.to = "all", unique = TRUE)
```

Arguments

object	The Xeva dataset
id	The id
id.name	The id name
map.to	The name of the mapped id. Default all

Value

a data.fram with id and mapped id

Examples

```
data(pdxe)
mapModelSlotIds(object=pdxe, id="X-007", id.name="biobase.id", map.to="model.id")
```

ModelInfo *modelInfo Generic Generic for ModelInfo method*

Description

modelInfo Generic Generic for ModelInfo method

Usage

```
ModelInfo(object)
```

Arguments

object	The XevaSet to retrieve drug info from

Value

a data.frame with the model annotations

Examples

```
data(pdxe)
ModelInfo(pdxe)
```

ModelInfo<- *ModelInfo<- Generic Generic for ModelInfo replace method*

Description

ModelInfo<- Generic Generic for ModelInfo replace method

Usage

```
ModelInfo(object) <- value
```

Arguments

object	The XevaSet to replace drug info in
value	A data.frame with the new model annotations

Value

Updated XevaSet

Examples

```
data(pdxe)
ModelInfo(pdxe) <- ModelInfo(pdxe)
```

Old_getTimeVarData *Given a treatment and control model ids it will return a data.fram with time vs volume (or any other variable)*

Description

Given a treatment and control model ids it will return a data.fram with time vs volume (or any other variable)

Usage

```
Old_getTimeVarData(object, ExpDesign, var, collapse)
```

Arguments

object	The Xeva dataset
ExpDesign	A list with batch.name, treatment and control
var	Name of the variable, default volume
collapse	Default TRUE. It will summerize all models which belongs to same treatment or control group

Value

a data.fram with treatment, control and batch.name

Examples

```
data(pdxe)
# creat a experiment desing
ExpDesign = list(batch.name="myBatch", treatment=c("X.010.BG98"), control=c("X.010.uned"))
df = getTimeVarData(object=pdxe, ExpDesign, var = "volume", collapse=TRUE)
## if collapse=FALSE it will not calculate standard error
df2= getTimeVarData(object=pdxe, ExpDesign, var = "volume", collapse=FALSE)
```

pasteColTogather *paste a data.frame columns togather while removing NA*

Description

pasteColTogather paste a data.frame columns togather while removing NA

Usage

```
pasteColTogather(df, collapse = " + ")
```

Arguments

df	A data.frame
collapse	Collapse string default " + "

Value

Returns an vector of strings where column values paste togather

Examples

```
df = data.frame(x= 1:6, y = c("A", NA, "B", NA, NA, "C"))
pasteColTogather(df, collapse = " + ")
```

pasteWithoutNA *paste a vector elements togather while removing NA*

Description

pasteWithoutNA paste a vector elements togather while removing NA

Usage

```
pasteWithoutNA(L, collapse = " + ")
```

Arguments

L	A vector with values and NA
collapse	Collapse string default " + "

Value

Returns an string with vector values paste togather

Examples

```
L = c("A", NA, "B", NA, NA, "C")
pasteWithoutNA(L, collapse = " + ")
```

`plotDrugResponse` *To plot drug response curve*

Description

To plot drug response curve

Usage

```
plotDrugResponse(object, drug, drug.match.exact, tumor.type, control)

plotDrugResponse(object, drug, drug.match.exact, tumor.type, control)
```

Arguments

`object`	The `XevaSet` to replace drug info in
`drug`	Name of the drug
`object`	The `XevaSet` to replace drug info in
`drug`	Name of the drug

Value

Updated `XevaSet`

Updated `XevaSet`

Examples

```
data(pdxe)
plotDrugResponse(pdxe, drug="LEE011 + binimetinib")
plotDrugResponse(pdxe, drug="paclitaxel", drug.match.exact=TRUE, tumor.type="BRCA")
```

plotmRECIST *Plot mRECIST for models and drugs* `plot.mRECIST` *plots the mRECIST*

Description

Plot mRECIST for models and drugs `plot.mRECIST` plots the mRECIST

Usage

```
plotmRECIST(df, groupBy = "biobase.id", control.name = "untreated")
```

Arguments

`object`	The `Xeva` dataset
`model.id`	The `model.id`

Value

plot

Examples

```
data(pdxe)
df = getmRECIST(pdxe)
df = df[1:500,]
plotmRECIST(df,groupBy = "biobase.id", control.name = "untreated")
```

selectModelIds *To select model ids based on drug name and/or tumor type*

Description

To select model ids based on drug name and/or tumor type

Usage

```
selectModelIds(object, drug = NULL, drug.match.exact = TRUE,
  tumor.type = NULL)
```

Arguments

`object`	The `XevaSet`
`drug`	Name of the drug
`drug.match.exact`	Default `TRUE`
`tumor.type`	Tumor type. Default `NULL`

Value

a `vector` with the matched model.ids

Examples

```
data(pdxe)
selectModelIds(pdxe, drug="paclitaxel", drug.match.exact=TRUE, tumor.type="BRCA")
```

`setmRECIST` *setmRECIST<- Generic for setmRECIST replace method*

Description

setmRECIST<- Generic for setmRECIST replace method

Usage

```
setmRECIST(object)
```

Arguments

`object`	The `XevaSet` object

Value

Updated `XevaSet`

Examples

```
data(pdxe)
#calculate mRECIST for each experiment
setmRECIST(pdxe)<- setmRECIST(pdxe)
getmRECIST(pdxe)
```

`show,XevaSet-method` *A method to display object for "show" setGeneric is already defined*

Description

A method to display object for "show" setGeneric is already defined

Usage

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

XevaSet-class *An S4 class for XevaSet*

Description

An S4 class for XevaSet

Index

addExperimentalDesign, 2

batchNames, 3

calculateAngle, 3
computemRECIST, 4
creatXevaSet, 5

drugInfo, 6
drugInfo<-, 6

expDesign, 7
expDesignInfo, 7
expDesignInfo<-, 8
experimentType, 8

getBatchName, 9
getControls, 9
getExpDesignDF, 10
getExperiment, 11
getmRECIST, 11
getTimeVarData, 12
getTreatment, 13

mapModelSlotIds, 13
ModelInfo, 14
ModelInfo<-, 14

Old_getTimeVarData, 15

pasteColTogather, 16
pasteWithoutNA, 16
plotDrugResponse, 17
plotmRECIST, 18

selectModelIds, 18
setmRECIST, 19
show,XevaSet-method, 19

XevaSet *(XevaSet-class)*, 20
XevaSet-class, 20