

SQUADD

April 20, 2011

`getFittedTable`-methods

Build the table of fitted values

Description

This method creates a table containing the interpolated/fitted value of the simulation curve at the user-defined time point. Each column represents the condition and each line, the component node. It is used as basis for the Correlation Circle and the Prediction Map.

Methods

`signature(object = "SquadSimResServiceImpl")` See [SquadSimResServiceImpl-class](#) for more information about object construction.

`getFittedTable`

Build the table of fitted values.

Description

This method creates a table containing the interpolated/fitted value of the simulation curve at the user-defined time point. Each column represents the condition and each line, the component node. It is used as basis for the Correlation Circle and the Prediction Map.

Usage

`getFittedTable(object)`

Arguments

`object` **Object of the Class SquadSimResServiceImpl.**

Details

This method firstly interpolate the SQUAD Simulation curve with a global (linear) or local (lowess) interpolation. A local interpolation is more sensitive to change in the curve when applying perturbation or pulse.

Value

returns a matrix of size length of component x length of conditions.

Author(s)

Martial Sankar

References

(1) Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatterplots. J. Amer. Statist. Assoc. 74, 829:836. (2) Chambers, J. M. (1992) Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth And Brooks/Cole.

See Also

More specific information in [getFittedTable-methods](#). For more informations about the input object see [SquadSimResServiceImpl-class](#).

Examples

```
fpath <- system.file("extdata", package="SQUADD")
fileModel <- file.path(fpath,"data_IAA")
nm <- c("ARF(a)", "ARF(i)", "AR_Genes", "Aux/IAA", "BES1/BZR1",
        "BIN2", "BR", "BRI1-BAK1", "BRR_Genes", "BRX", "BR_Biosynthesis", "BZR1",
        "DO", "IAA", "IAA_Biosynthesis", "NGA1", "PIN", "SCFTir1",
        "StimAux", "StimBR")
t <- 50

## call constructor
obj <- simResService(
  folder=fileModel,
  time=t,
  ncolor=5,
  legend=nm,
  indexDeno=1,
  method="lowess")

## Apply method
tab <- getFittedTable(obj)
```

plotCC-methods

plot the correlation circle

Description

This method generates the correlation circle. Correlation circle is used to assess the coherence of a model. Indeed, each vector represents our conditions and the angle between each vector give the correlation between the variables. if the angle is higher than 90 degree there is no correlation.

Methods

`signature(object = "SquadSimResServiceImpl")` See [SquadSimResServiceImpl-class](#) for more information about object construction.

plotCC

*Generate the Correlation Circle.***Description**

This method generates the correlation circle. Correlation circle is used to assess the coherence of a model. Indeed, Each vector represents our conditions and the angle between each vector give the correlation between the variables. if the angle is higher than 90 degrees there is no correlation.

Usage

```
plotCC(object)
```

Arguments

object Object of the Class SquadSimResServiceImpl.

Details

The percentage of variability explained by the two first principal components are displayed in parenthesis in the axis label.

Author(s)

Martial Sankar

See Also

More specific information in [plotCC-methods](#). For more informations about the input object see [SquadSimResServiceImpl-class](#).

Examples

```
fpath <- system.file("extdata", package="SQUADD")
fileModel <- file.path(fpath,"data_IAA")
nm <- c("ARF(a)", "ARF(i)", "AR_Genes", "Aux/IAA", "BES1/BZR1",
        "BIN2", "BR", "BRI1-BAK1","BRR_Genes","BRX","BR_Biosynthesis","BZR1",
        "DO", "IAA", "IAA_Biosynthesis", "NGA1", "PIN", "SCFTir1",
        "StimAux", "StimBR")
t <- 50

## call constructor
obj <- simResService(
  folder=fileModel,
  time=t,
  ncolor=5,
  legend=nm,
  indexDeno=1,
  method="lowess",
  conditionList=c("Normal", "brxlof", "BRrescue","brxarfilof",
                  "brxarfiBRrescue", "brxgof")
)
## call method
plotCC(obj)
```

plotPredMap-methods

Display a prediction map

Description

This method is used to visualize the prediction heat-map. The prediction map is generated by mapping a color code on the ratio of the fitted values.

Methods

`signature(object = "SquadSimResServiceImpl")` See [SquadSimResServiceImpl-class](#) for more information about object construction.

plotPredMap

Prediction map

Description

This method is used to visualize the prediction heat-map. The prediction map is generated by mapping a color code on the ratio of the fitted values.

Usage

```
plotPredMap(object)
```

Arguments

`object` object of the Class `SquadSimResServiceImpl`.

Details

This method firstly interpolate the SQUAD Simulation Result. Then, it performs a ratio at a user-defined time point. Finally, a color code is assigned to the ratio value. The darker the color, the higher the increase of the node activation.

Author(s)

Martial Sankar

See Also

More information about heatmap, [heatmap](#) More specific information in [plotPredMap-methods](#). For more informations about the input object see [SquadSimResServiceImpl-class](#).

Examples

```
fpath <- system.file("extdata", package="SQUADD")
fileModel <- file.path(fpath,"data_IAA")
nm <- c("ARF(a)", "ARF(i)", "AR_Genes", "Aux/IAA", "BES1/BZR1",
        "BIN2", "BR", "BRI1-BAK1", "BRR_Genes", "BRX", "BR_Biosynthesis", "BZR1", "DO",
        "IAA", "IAA_Biosynthesis", "NGA1", "PIN", "SCFTir1", "StimAux", "StimBR")
t <- 50

## call constructor
obj <- simResService(
  folder=fileModel,
  time=t,
  ncolor=5,
  legend=nm,
  indexDeno=1,
  method="lowess")

## Apply method
plotPredMap(obj)
```

plotSimMatrix-methods

Generate a SQUAD simulation matrix

Description

This method is used to create a simulation matrix. A simulation matrix permits to follow the activation signal of an user-defined number of Node. Each columns correspond to the user-selected component node and the each rows, to the conditions. (more details in [plotSimMatrix](#))

Methods

signature(object = "SquadSimResServiceImpl") See [SquadSimResServiceImpl-class](#) for more information about object construction.

plotSimMatrix

Generate the SQUAD Simulation matrix display.

Description

This method is used to create a simulation matrix. Each columns correspond to the user-selected component node and the each rows, to the conditions.

Usage

```
plotSimMatrix(object)
```

Arguments

object object of the Class SquadSimResServiceImpl.

Author(s)

Martial Sankar

See Also

More specific information in [plotSimMatrix-methods](#). For more informations about the input object see [SquadSimResServiceImpl-class](#).

Examples

```
fpath <- system.file("extdata", package="SQUADD")
fileModel <- file.path(fpath,"data_IAA")
nm <- c("ARF(a)", "ARF(i)", "AR_Genes", "Aux/IAA", "BES1/BZR1",
        "BIN2", "BR", "BRI1-BAK1", "BRR_Genes", "BRX", "BR_Biosynthesis", "BZR1",
        "DO", "IAA", "IAA_Biosynthesis", "NGA1", "PIN", "SCFTir1", "StimAux", "StimBR")
t <- 50

## call constructor
obj <- simResService(
  folder=fileModel,
  time=t,
  ncolor=5,
  legend=nm,
  indexDeno=1,
  method="lowess",
  selectNode=c("DO", "IAA_Biosynthesis", "BR_Biosynthesis", "IAA", "BR"))

## call method
plotSimMatrix(obj)
```

SQUADD-package

Collection of visualization tools to perform a SQUAD-based quantitative analysis ~SQUADD~

Description

The SQUAD (Standardized Qualitative Dynamical Systems) software is dedicated to the analysis of boolean logical model (Di Cara et al 2007). It was the first tool to combine the traditional boolean approach and a continuous system. This novel feature permits the dynamical analysis of regulatory network in the absence of kinetic data. In order to get the best of this approach, we provide here a R package, SQUADD, a SQUAD add-on for : (i) simplify the visualization of the initial SQUAD simulation results. (ii) generate qualitative predictions of the component change between two conditions. (iii) Assess the coherence of a model by PCA analysis.

Details

Package:	SQUADD
Type:	Package
Version:	1.0.0

Date: 2010-06-14
 License: GPL(>=2)
 LazyLoad: yes
 Depends: R (>= 2.11.0)
 Imports: graphics, grDevices, methods, RColorBrewer, stats, utils

Author(s)

Martial Sankar <martial.sankar@unil.ch>

References

Di Cara A, Garg A, De Micheli G, Xenarios I, Mendoza L (2007) Dynamic simulation of regulatory networks using SQUAD. BMC Bioinformatics 8: 462. Dominik M Wittmann, Jan Krumsiek, Julio Saez-Rodriguez, Douglas A Lauffenburger, Steffen Klamt and Fabian J Theis (2009) Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. PLOS Computational Biology 3: 98

SquadSimResServiceImpl-class

Class "SquadSimResServiceImpl"

Description

The Class SquadSimResServiceImpl contains the information needed to plot the simulation matrix, the prediction map and the correlation circle.

Slots

folder: Object of class "character", path to the folder to find the SQUAD (1) data.
time: Object of class "numeric", time point used to get the interpolated activate state value.
ncolor: Object of class "numeric", number of color code to used in the prediction map. default = 5
legend: Object of class "vector", contains the name of each node of the model (Must appear in the same order than the simulation).
indexDeno: Object of class "numeric", the index of the file to used as denominator (the names of each file in the folder should contain an index, i.e 1_XXX.txt), default = 1
method: Object of class "character", the fitting method to interpolate the curve ("linear" to get a straight line (2) or "lowess" to get a smoothed interpolation (3)).
selectNode: Object of class "character", contains the name of the node to display.
conditionList: Object of class "character", list of the perturbation condition names.

Constructor

simResService(folder, time, ncolor, legend, indexDeno, method, selectNode, conditionList): construct a instance of SquadSimResServiceImpl.

Methods

getFittedTable signature(object = "SquadSimResServiceImpl"): Build and display the table of interpolated/fitted values (see [getFittedTable](#) for more details).

plotSimMatrix signature(object = "SquadSimResServiceImpl"): plot the simulation matrix (see [plotSimMatrix](#) for more details).

plotPredMap signature(object = "SquadSimResServiceImpl"): plot the prediction map (see [plotPredMap](#) for more details).

plotCC signature(object = "SquadSimResServiceImpl"): plot the correlation circle from PCA analysis (see [plotCC](#) for more details).

Accessor methods

See examples Section.

Author(s)

Martial Sankar

References

(1) Di Cara A, Garg A, De Micheli G, Xenarios I, Mendoza L (2007) Dynamic simulation of regulatory networks using SQUAD. BMC Bioinformatics 8: 462. (2) Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatterplots. J. Amer. Statist. Assoc. 74, 829:836. (3) Chambers, J. M. (1992) Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth And Brooks/Cole.

Examples

```
fpath <- system.file("extdata", package="SQUADD")
fileModel <- file.path(fpath,"data_IAA")
nm <- c("ARF(a)", "ARF(i)", "AR_Genes", "Aux/IAA", "BES1/BZR1", "BIN2", "BR", "BRI1-BAK1")
t <- 50

## call constructor
obj <- simResService(folder=fileModel, time=t, ncolor=5, legend=nm, indexDeno=1, method="local")

## Apply method
tab <- getFittedTable(obj)
obj["selectNode"] <- c("DO", "IAA_Biosynthesis", "BR_Biosynthesis", "IAA", "BR")

plotSimMatrix(obj)
plotPredMap(obj)

obj["conditionList"] <- c("Normal", "brxlof", "BRrescue", "brxarfilof", "brxarfiBRrescue", "brxarfiBRrescue")
plotCC(obj)

## getter/setter test
obj["time"] <- 45
```


Index

*Topic **PCA**

plotCC, 3
plotCC-methods, 2

*Topic **accessor**

SquadSimResServiceImpl-class,
7

*Topic **circle**

plotCC, 3

*Topic **classes**

SquadSimResServiceImpl-class,
7

*Topic **constructor**

SquadSimResServiceImpl-class,
7

*Topic **correlation**

plotCC, 3
plotCC-methods, 2

*Topic **fitted**

getFittedTable-methods, 1

*Topic **fit**

getFittedTable, 1

*Topic **getter**

SquadSimResServiceImpl-class,
7

*Topic **heatmap**

plotPredMap, 4

*Topic **interpolated**

getFittedTable-methods, 1

*Topic **interpolate**

getFittedTable, 1

*Topic **methods**

getFittedTable-methods, 1
plotCC-methods, 2
plotPredMap-methods, 4
plotSimMatrix-methods, 5

*Topic **package**

SQUADD-package, 6

*Topic **prediction**

plotPredMap, 4
plotPredMap-methods, 4

*Topic **setter**

SquadSimResServiceImpl-class,
7

*Topic **simulation**

plotSimMatrix, 5
plotSimMatrix-methods, 5

[, SquadSimResServiceImpl-method
(SquadSimResServiceImpl-class),
7

[<-, SquadSimResServiceImpl-method
(SquadSimResServiceImpl-class),
7

getFittedTable, 1, 8

getFittedTable, SquadSimResServiceImpl-method
(getFittedTable-methods), 1

getFittedTable-methods, 2

getFittedTable-methods, 1

heatmap, 4

plotCC, 3, 8

plotCC, SquadSimResServiceImpl-method
(plotCC-methods), 2

plotCC-methods, 3

plotCC-methods, 2

plotPredMap, 4, 8

plotPredMap, SquadSimResServiceImpl-method
(plotPredMap-methods), 4

plotPredMap-methods, 4

plotPredMap-methods, 4

plotSimMatrix, 5, 5, 8

plotSimMatrix, SquadSimResServiceImpl-method
(plotSimMatrix-methods), 5

plotSimMatrix-methods, 6

plotSimMatrix-methods, 5

simResService

(SquadSimResServiceImpl-class),
7

SQUADD (SQUADD-package), 6

SQUADD-package, 6

SquadSimResServiceImpl

(SquadSimResServiceImpl-class),
7

SquadSimResServiceImpl-class, 1-6

SquadSimResServiceImpl-class, 7