Package 'KOMA'

July 22, 2011

Type Package

Title Kinetic Operating Microarray Analyzer (KOMA)

Version 1.0

Date 2011-07-15

Author Florian Heigwer <florian_heigwer@web.de>

Maintainer Stefan Woelfl <wolfl@uni-hd.de>

Description Kinetic Operating Microarray Analyzer (KOMA) enables
calibration and high-throughput analysis of quantitative
microarray data collected by using kinetic detection protocol.
This tool can be also helpful for analyzing data from any other
analytical assays employing enzymatic signal amplification, in
which a broader range of quantification is reached by the time-resolved recording of readouts.

License GPL-2

LazyLoad yes

Depends epicalc,fields,gplots,magic,rgl,tcltk2,tkrplot,R (>= 2.12)

OS_type windows

References James Wettenhall, Philippe Grosjean 2005 Felix Bonowski 2010 Pavlo Holenya 2011

R topics documented:

`analyte.choice` *Choosing an analyte*

Description

This function opens a window showing all accessible analyte names of the specified array. Here the one to analyse or calibrate should be chosen.

Usage

```
analyte.choice(analyte.names)
```

Arguments

`analyte.names`
names of all the defined analytes

Value

`analyte` the name of the chosen analyte as character string

Author(s)

Florian Heigwer

`calculate.plot.mean.and.error`
Calculate and plot the mean and error surfaces

Description

This function gives the possibility to evaluate the goodness of calibration from the several independent replicates.

Usage

```
calculate.plot.mean.and.error(workpath, datasets.to.leeve.out)
```

Arguments

`workpath` the specified working directory from the maindirectory
`datasets.to.leeve.out`
vector of dataset IDs to exclude, passed by from the function plotting.overlapping.cals

Note

please close the windows after analysis

Author(s)

Florian Heigwer

See Also

`plotting.overlapping.cals`

`calibrationsettings` *Calibration settings*

Description

This function opens a panel to specify settings for the calibration analysis.

Usage

```
calibrationsettings()
```

Details

This function will then auotmatically run the subfunctions timecourse and dilutionseries.

Value

`times`	a vector of time points in the given interval
`concentrations`	a vector of concentrations

Author(s)

Florian Heigwer

See Also

`timecourse` `dilutionseries`

collect.and.save.cal *Save the analysed calibration run*

Description

This functions generates a calibration file for the specified calibration run.

Usage

```
collect.and.save.cal(data, chosen.IDs, substance, run)
```

Arguments

data	a list of tables with confidence and signal values for each spotreplicate, timepoint and concentration
chosen.IDs	a vector of chosen IDs
substance	the analytes name
run	the name of the calibration run to name the files subsequently

Note

The file will be saved under the analytes name, calibration, and the run ID.

Author(s)

Florian Heigwer

See Also

calculate.plot.mean.and.error

dilutionseries *Calculate the dilutionseries*

Description

This function calculates a series of concentrations depending on the highest concentration, the number of dilutionsteps and the diltionfactor.

Usage

```
dilutionseries(highestconc, dilutions, dilutionfactor)
```

Arguments

highestconc	the highest concentration as numeric object
dilutions	the number of diltionsteps performed in the experiment
dilutionfactor	the used factor of dilution

Value

concentrations
: a series of ideal calculated concentrations used to calibrate

Author(s)

Florian Heigwer

Examples

```
dilutionseries(10,8,2) #give the highest concentration, the number of dilutions and the d
```

---

extractconfidence_strips
Extract confidence values

Description

This function parses all the folders of an experiment for the result.txt tables and sorts or reformats them.

Usage

```
extractconfidence_strips(workpath, substance, run, times, concentrations)
```

Arguments

workpath	the directory inheritating data to analyse
substance	the analytes name chosen by analyte.choice
run	the name of the certain run
times	timeseries as given from timecourse
concentrations	series of concentration as given from dilutionseries

Value

confidencedata
: a list consisting of the sortet and formatted confindencedata, plus the timecourse and the dilutionseries

Author(s)

Florian Heigwer

See Also

extractdata_strips

`extractdata_strips` *Extract experimental data from folder with rawdata*

Description

This function parses all the folders of an experiment for the result.txt tables and sorts or reformats them.

Usage

```
extractdata_strips(workpath, substance, run, times, concentrations)
```

Arguments

`workpath`	the directory inheritating data to analyse
`substance`	the analytes name chosen by `analyte.choice`
`run`	the name of the certain run
`times`	timeseries as given from `timecourse`
`concentrations`	series of concentration as given from `dilutionseries`

Value

`signaldata`	a list consisting of the sortet and formatted signaldata, plus the timecourse and the dilutionseries

Author(s)

Florian Heigwer

See Also

`extractconfidence_strips`

`extractdata_unknown_predict_conc` *Measure an concentration*

Description

This function provides a workflow for calculation of an unknown concentration by using the analytes forprediction.RData file.

Usage

```
extractdata_unknown_predict_conc(workpath, substance)
```

Arguments

workpath	the directory containing the rawdata of the measurement
substance	the analytes name

Details

first it extracts the data second it plots the replicates and lets you choose excludable ones third it will compare the meaned data set with the topographic calibration and give back the estimated analyte concentration

Value

analyteconcentration
: an estimate of the measured concentration

Author(s)

Florian Heigwer

See Also

extractdata_unknown_predict_conc_wholestrip

---

extractdata_unknown_predict_conc_wholestrip

Measure eigth folders at once

Description

This function provides a workflow for calculation of an unknown concentration by using the analytess forprediction file.

Usage

```
extractdata_unknown_predict_conc_wholestrip(foldernames,paths,substance,workpath
```

Arguments

foldernames	the names of the single folders
paths	the directory containing the rawdata of the measurement
substance	the analytes name
workpath	the main working directory

Details

first it extracts the data, second it plots the replicates and lets you choose ecludable ones, third it will compare the meaned data set with the topographic calibration and give back the estimated analyte concentration

Value

`analyteconcentration`
: an estimate of the measured concentration

Author(s)

Florian Heigwer

See Also

`extractdata_unknown_predict_conc`

GUI *Build the GUI*

Description

This function generates a whole functioning graphical user interface.

Usage

```
GUI()
```

Details

The user interface called KOMA consists of a mainmenu containg all neccesary submenues for controlling the data analysis and a tabulary body for showing experimental results.

Author(s)

Florian Heigwer

References

http://www.sciviews.org/_rgui/tcltk/Tktable.html

See Also

`Separation`

Examples

```
GUI()
```

list.dir *List Directories*

Description

This function creates a lists of directories and only directories in a certain path.

Usage

```
list.dir(path)
```

Arguments

path	the path to search the directories in

Value

Q	the names of the found directories

Author(s)

Florian Heigwer

Examples

```
list.dir(path=R.home())
```

numeric.entry.window *Open an interactive window asking for numeric input*

Description

This function opens an interactive window asking for numeric input.

Usage

```
numeric.entry.window(title, question, entryInit, entryWidth = 20, returnValOnCan
```

Arguments

title	the title of the window as character string
question	the question to ask in the window
entryInit	the default text entry when opening the window
entryWidth	the width of the textinput area
returnValOnCancel	what should be printed when Cancel has been pushed

Value

`ReturnVal` the typed character string

Author(s)

Florian Heigwer

References

James Wettenhall, Philippe Grosjean (2005)

See Also

`numeric.entry.window`

`plotting.overlapping.cals` *Plot replicates of several calibration runs.*

Description

This function extracts datasets of different calibration replicates from files in the main working directory. These files are searched by regular expression matching with the analytes name and the word calibration.

Usage

```
plotting.overlapping.cals(workpath)
```

Arguments

`workpath` defines the directory to search in

Value

`IDs.to.leeve.out`
a list of dataset IDs which should be excluded from all further analysis

Note

At least two datasets has to be left over.If more should be left for analysis please stop the function by rightclick–> Stop.

Author(s)

Florian Heigwer

See Also

`collect.and.save.cal`

plotting.timecourses.and.choose.IDs.manually

Plot timecourses of signal and confidence values and choose IDs to exclude manually

Description

This function will open an tcltk interface panel which shows signal an confindence values timeresolved for all spot replicates. Giving as well a chance to choose failed spots for exclusion from further analysis.

Usage

```
plotting.timecourses.and.choose.IDs.manually(data, substance, confidencevalues,
```

Arguments

data	datasets as a list of matrices for each replicate, for each timepoint and eventually each concentration while calibration containing signal values
substance	the analytename as character string
confidencevalues	datasets as a list of matrices for each replicate, for each timepoint and eventually each concentration while calibration containing confidence values
workpath	the directory with the measured rawdata
foldernames	the single foldernames (each representing one well of an ArrayStrip)

Value

IDs.to.left.out
: a vector of spot IDs to be exclude from further analysis

Author(s)

Florian Heigwer

See Also

collect.and.save.cal

Separation

Define everything at once

Description

This function is provided to define all the other needed functions by default and load the needed packages into the R workspace.

Usage

```
Separation()
```

Details

If there is a need to change the source code feel free to edit it in this file.

Author(s)

Florian Heigwer

See Also

GUI

---

textstring.entry.window

Open an interactive window asking for textinput

Description

This function opens an interactive window asking for textinput.

Usage

```
textstring.entry.window(title, question, entryInit, entryWidth = 20, returnValOn
```

Arguments

title	the title of the window as character string
question	the question to ask in the window
entryInit	the default text entry when opening the window
entryWidth	the width of the textinput area
returnValOnCancel	what should be printed when Cancel has been pushed

Value

ReturnVal	the typed character string

Author(s)

Florian Heigwer

References

James Wettenhall, Philippe Grosjean (2005)

See Also

numeric.entry.window

`timecourse` *Define timecourses*

Description

This function calculates a series of timepoints.

Usage

```
timecourse(endpoint, measurementinterval)
```

Arguments

`endpoint`	endpoint of the measurement
`measurementinterval`	time interval between single measurements

Value

`times`	a vector of timespoints

Warning

Time interval and the endpoint of the measurement got to be in the same unit.

Author(s)

Florian Heigwer

See Also

`dilutionseries`

Examples

```
time.gone=60 #minutes
interval=1.5 #minutes

timecourse(time.gone,interval)
```

Index