Package: paramGUI (via r-universe)

September 13, 2024

Title A Shiny GUI for some Parameter Estimation Examples

Version 2.2.0

Description Allows specification and fitting of some parameter estimation examples inspired by time-resolved spectroscopy via a Shiny GUI.

URL https://github.com/glotaran/paramGUI/, https://glotaran.github.io/paramGUI/

License GPL (>= 2)

Depends R (>= 3.0.0)

Imports shiny, shinydashboard, TIMP, fields

BugReports https://github.com/glotaran/paramGUI/issues

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.3

Repository https://glotaran.r-universe.dev

RemoteUrl https://github.com/glotaran/paramgui

RemoteRef HEAD

RemoteSha b7908ab79709f2c217b8c59a5923bf210b2669fc

Contents

`calcE` *Calculates a matrix in which each column is a skewed Gaussian*

Description

Calculates a matrix in which each column is a skewed Gaussian. Like `calcEhiergaus` from TIMP package but uses a vector not a list of parameter estimates.

Usage

```
calcE(theta, lambda)
```

Arguments

`theta`	vector of parameter estimates
`lambda`	wavelengths at which to calculate model

Value

`matrix`

`example_dataset` *This is an example dataset included in this package*

Description

Dispersion corrected time-resolved transient-absorption data of the peridinin chlorophyll protein (PCP) excited with 490 nm laser light from the publication of Stokkum et.al. (2009)

Author(s)

Ivo van Stokkum `<i.h.m.van.stokkum@vu.nl>`

References

doi:10.1016/j.chemphys.2008.10.005

is_compressed *is_compressed*

Description

Helper function for is_rdata, checks if the file is a compressed (gzip) file. Does not (yet) check for bzip2 or xz compression.

Usage

```
is_compressed(filename, magic.number = as.raw(c("0x1f", "0x8b")))
```

Arguments

filename	The filename of the file to test for magic compression codes
magic.number	The magic numbers in as a vector of strings with the hexadecimal numbers (e.g. "0x1f")

Value

boolean, TRUE if the file is compressed

is_rdata *is_rdata*

Description

Checks a file is a rdata file by inspecting the file for so called magic bytes

Usage

```
is_rdata(filename)
```

Arguments

filename	The filename of the file to test if it is an rdata file

Value

boolean, TRUE if the file is an rdata file

kroneckercol *kroneckercol: column-wise kronecker product*

Description

The column-wise kronecker product is also called the Khatri–Rao product

Usage

```
kroneckercol(A, B)
```

Arguments

A	numerical matrix
B	numerical matrix

Value

column-wise kronecker product of A and B

linlogtics *Generate linlog tics for a linear-logarithmic axis*

Description

Generate linlog tics for a linear-logarithmic axis

Usage

```
linlogtics(x, mu, alpha)
```

Arguments

x	values for which to calculate a linlog axis
mu	center of axis in the original x axis
alpha	linear part

Value

Returns matrix with new x values in first column and the corresponding labels in the second column.

paramGUI *paramGUI*

Description

Allows specification and fitting of some parameter estimation examples inspired by time-resolved spectroscopy via a Shiny GUI.

plotterforGUI *Master plot function for paramGUI*

Description

Master plot function for paramGUI

Usage

```
plotterforGUI(
  modtype = "kin",
  X = matrix(),
  data,
  model,
  theta = vector(),
  result,
  lin = NA,
  mu = 0,
  guessIRF = FALSE
)
```

Arguments

modtype	either 'kin', 'spec' or 'spectemp'
X	matrix of conditionally linear parameters, if any
data	object of class dat containing data
model	object of class dat containing data
theta	object of class theta containing parameters
result	object returned by fitModel or in the case modtype=='spectemp', by nls
lin	The linear range for the concentration plot
mu	The center of the lin-log axis is lin is specified
guessIRF	Boolean to indicate whether to try and guess the location of the IRF

Value

graphics

runGUI *Run paramGUI*

Description

Runs the shiny paramGUI app.

Usage

```
runGUI()
```

Examples

```
## Not run:
runGUI()

## End(Not run)
```

simndecay_gen_paramGUI *Simulate data*

Description

Calculates an object of class 'kin'. <TODO>

Usage

```
simndecay_gen_paramGUI(
  kinpar,
  tmax,
  deltat,
  specpar = vector(),
  lmin,
  lmax,
  deltal,
  sigma,
  irf = FALSE,
  irfpar = vector(),
  seqmod = FALSE,
  dispmu = FALSE,
  nocolsums = FALSE,
  disptau = FALSE,
  parmu = list(),
  partau = vector(),
```

```
  lambdac = 0,
  fullk = FALSE,
  kmat = matrix(),
  jvec = vector(),
  specfun = "gaus",
  nupow = 1,
  irffun = "gaus",
  kinscal = vector(),
  lightregimespec = list(),
  specdisp = FALSE,
  specdisppar = list(),
  parmufunc = "exp",
  specdispindex = list(),
  amplitudes = vector(),
  specref = 0,
  nosiminfo = TRUE
)
```

Arguments

kinpar	vector of rate constants
tmax	last time point
deltat	time step
specpar	vector of spectral parameters for location, width, skewness
lmin	minimum wavelength (nm)
lmax	maximum wavelength (nm)
deltal	wavelength step
sigma	noise level
irf	logical for IRF usage
irfpar	vector of IRF parameters for location, width
seqmod	logical for sequential model
dispmu	logical for dispersion of IRF location mu
nocolsums	logical for <TODO>
disptau	logical for dispersion of IRF width tau
parmu	vector of dispersion parameters for IRF location mu
partau	vector of dispersion parameters for IRF width tau
lambdac	center wavelength for dispersion
fullk	logical for full K matrix
kmat	K matrix
jvec	input vector
specfun	function for spectral shape
nupow	power of nu in spectral model

irffun	function for IRF
kinscal	vector of kinetic scaling parameters
lightregimespec	<TODO>
specdisp	logical for dispersion parameters of spectral parameters
specdisppar	vector of dispersion parameters of spectral parameters
parmufunc	<TODO>
specdispindex	<TODO>
amplitudes	amplitudes of components
specref	<TODO>
nosiminfo	logical for hiding simulation information

Value

an object of class 'kin'

Author(s)

Katharine M. Mullen

Ivo H. M. van Stokkum

---

spectemp *Spectrotemporal model*

Description

Spectrotemporal model

Usage

```
spectemp(sim, model, iter, kroncol = FALSE, lin = NA, l_posk = FALSE)
```

Arguments

sim	object of class dat representing data
model	object of class dat representing a model
iter	integer number of iterations
kroncol	object of class logical that is TRUE if the kroneckcol function should be used to formulate the model and FALSE if the standard kronecker is to be used instead
lin	defines the range to plot linearly (from -lin to +lin)
l_posk	object of class logical indicating whether positivity-constraints are enforced on the rate parameters

`startGUI` *Start paramGUI*

Description

The same as runGUI(), starts the shiny paramGUI app.

Usage

```
startGUI()
```

Index