

Package ‘mitoODE’

April 14, 2017

Version 1.12.0

Title Implementation of the differential equation model described in
“Dynamical modelling of phenotypes in a genome-wide RNAi
live-cell imaging assay”

Author Gregoire Pau

Maintainer Gregoire Pau <pau.gregoire@gene.com>

SystemRequirements <not required>

Depends R (>= 2.14.0), minpack.lm, MASS, parallel, mitoODEdata,
KernSmooth

Description

The package contains the methods to fit a cell-cycle model on cell count data and the code to reproduce the results shown in our paper “Dynamical modelling of phenotypes in a genome-wide RNAi live-cell imaging assay” by Pau, G., Walter, T., Neumann, B., Heriche, J.-K., Ellenberg, J., & Huber, W., BMC Bioinformatics (2013), 14(1), 308. doi:10.1186/1471-2105-14-308

License LGPL

LazyLoad yes

biocViews ExperimentData, TimeCourse, CellBasedAssays, Preprocessing

NeedsCompilation yes

R topics documented:

figures	1
mitoODE	2
plotfit	4

Index	5
--------------	----------

figures *Reproduce the results of the mitoODE paper*

Description

The functions `loadFittedData` loads all the fitted parameters of the Mitocheck screen. The figure functions contain the code to produce the figures of the paper.

Usage

```
loadFittedData()  
figure1()  
figure2()  
figure3a()  
figure3b()  
figure4()
```

Details

The function `loadFittedData` loads the data.frame `pheno` in the global environment. The figure functions produce PDF plots in the working directory.

Value

No values.

Author(s)

Gregoire Pau, <pau.gregoire@gene.com>, 2012

See Also

[fitmodel](#), [plotfit](#)

Examples

```
## Not run:  
loadFittedData()  
figure1()  
figure2()  
figure3a()  
figure3b()  
figure4()  
  
## End(Not run)
```

mitoODE

Dynamical modelling of phenotypes in a genome-wide RNAi live-cell imaging assay

Description

The `mitoODE` package implements the differential model fitting used in the “Dynamical modelling of phenotypes in a genome-wide RNAi live-cell imaging assay” paper and includes the code to reproduce the presented figures and results. See the vignette for details about the model.

Usage

```
getp0(y, p0=NULL, sd=0)  
fitmodel(y, p0, pconst, nfits=1, sd=0, mc.cores=1, best=TRUE)
```

Arguments

<code>y</code>	A matrix with 4 columns containing the cell count time series, such as the one returned by <code>readspot</code> .
<code>p0</code>	A vector of 10 numeric parameters, such as returned by <code>getp0</code> , containing the initial conditions. See the vignette for details.
<code>pconst</code>	A vector of 10 numeric parameters, containing <code>g.kim</code> , <code>g.kmi</code> , <code>g.mit0</code> and <code>p.lambda</code> . See the vignette for details.
<code>nfits</code>	The number of fits to be done. Default is 1.
<code>sd</code>	A numeric indicating the standard deviation of some Gaussian noise added to the initial conditions before fitting. Default is 0.
<code>mc.cores</code>	A numeric indicating how many parallel cores should be used to fit the cell count time series. Default is 1.
<code>best</code>	A logical. See return value.

Details

The function `fitmodel` fits the cell count data to the differential equation model described in the vignette, using the penalized Levenberg-Marquardt `nls.lm` algorithm.

Value

`getp0` returns a vector of 10 parameters that can be used as initial conditions. If `TRUE`, `fitmodel` returns the best fitted parameters, otherwise returns the all `n` fitted parameters.

Author(s)

Gregoire Pau, <pau.gregoire@gene.com>, 2012

See Also

[loadFittedData](#), [figure1](#), [plotfit](#)

Examples

```
## read spot
spotid <- 1000
y <- readspot(spotid)

## fit data
pconst <- c(g.kim=0.025, g.kmi=0.57, g.mit0=0.05, p.lambda=4)
p0 <- getp0()
pp <- fitmodel(y, p0, pconst)
round(pp, 2)

## plot fitted data
plotfit(spotid, pp)
```

<code>plotfit</code>	<i>Plot fitted values</i>
----------------------	---------------------------

Description

The functions `plotfit` and `plotk` plot the fitted values and the transition functions across time.

Usage

```
plotfit(spot, p=pheno[spot,], showfit=TRUE, legend="topleft", kk=NULL, cex=1, kcol='#ffaa77', lwd=1)
plotk(spot, p=pheno[spot,], kk=c("kim"), height, kcol, lwd=1, xlab="Time after seeding (h)", ylab="Transition rate")
```

Arguments

<code>spot</code>	A numeric indicating a spot ID, synchronised with the global Mitocheck tab object.
<code>p</code>	A vector of 10 numeric parameters, such as returned by <code>getp0</code> or <code>fitmodel</code> , containing the fitted parameters. Default are the previously fitted parameters in the pheno matrix.
<code>showfit</code>	A logical. If TRUE, the fitted data is plotted over the cell count time series. Default is TRUE.
<code>kk</code>	A character vector containing the transition rates to plot. Values can be <code>kim</code> , <code>kmi</code> , <code>kmp</code> or <code>kd</code> .
<code>kcol</code>	A named vector of colors for inflection points. Names take the same values as <code>kk</code> .
<code>legend</code> , <code>cex</code> , <code>lwd</code> , <code>xlab</code> , <code>ylab</code> , <code>height</code> , <code>type</code> , ...	Graphical parameters, see <code>par</code> .

Details

The function `plotfit` plots the cell count times series and the fitted data. The function `plotk` plots the transition rate functions.

Value

No values.

Author(s)

Gregoire Pau, <pau.gregoire@gene.com>, 2012

See Also

[fitmodel](#), [loadFittedData](#)

Examples

```
## load previously fitted data
loadFittedData()

## plot fitted data
plotfit(156205)
```

Index

figure1, [3](#)
figure1 (figures), [1](#)
figure2 (figures), [1](#)
figure3a (figures), [1](#)
figure3b (figures), [1](#)
figure4 (figures), [1](#)
figures, [1](#)
fitmodel, [2](#), [4](#)
fitmodel (mitoODE), [2](#)

getp0 (mitoODE), [2](#)

loadFittedData, [3](#), [4](#)
loadFittedData (figures), [1](#)

mitoODE, [2](#)

pheno (figures), [1](#)
plotfit, [2](#), [3](#), [4](#)
plotk (plotfit), [4](#)