R Documentation of KINGUII

June 28, 2011

`chi2err` *chi2err*

Description

This function calculates the smallest relative error still resulting in passing the chi-squared test as defined in the FOCUS kinetics report from 2006.

Usage

```
chi2err(errdata, n.parms, alpha = 0.05)
```

Arguments

`errdata`	A data frame with mean observed values in column named 'value_mean' and predicted values in column 'value_pred'.
`n.parms`	The number of optimized parameters to be taken into account for the data series.
`alpha`	The confidence level chosen for the chi-squared test.

Details

This function is used internally by 'mkinfit.gui', 'IRLSkinfit.gui' and 'mcmckinfit.gui'.

Value

`err.min`	The relative error, expressed as a fraction.
`n.optim`	The number of optimised parameters attributed to the data series.
`df`	The number of remaining degrees of freedom for the chi2 error level calculations. Note that mean values are used for the chi2 statistic and therefore every time point with observed values in the series only counts one time.

References

FOCUS (2006) "Guidance Document on Estimating Persistence and Degradation Kinetics from Environmental Fate Studies on Pesticides in EU Registration" Report of the FOCUS Work Group on Degradation Kinetics, EC Document Reference Sanco/10058/2005 version 2.0, 434 pp, http://focus.jrc.ec.europa.eu/dk

See Also

mkinerrmin

IRLSkinfit.gui *Fit a kinetic model using the IRLS algorithm.*

Description

This function does kinetic evaluations using the IRLS algorithm.

Usage

```
IRLSkinfit.gui(mkinmodini, eigen = FALSE, plot = FALSE, plottitle = "",
quiet = FALSE, err = NULL, weight = "none", scaleVar = FALSE,
 ctr = kingui.control(), irls.control = list(), update = NULL, ...)

## S3 method for class 'kingui':
summary(object, data = TRUE, distimes = TRUE, ff =
TRUE, cov = FALSE, ...)
```

Arguments

mkinmodini	A list of class mkinmod.gui, containing the kinetic model to be fitted to the data, and the initial parameter values, the observed data.
eigen	If TRUE, the solution of the system of differential equations should be based on the spectral decomposition of the coefficient matrix in cases that this is possible.
plot	If TRUE,the observed values and the numerical solutions should be plotted at each stage of the optimisation.
plottitle	The title of the plot for visualizing the optimization process.
quiet	If TRUE, suppress printing out the current model cost after each(>1) improvement.
err	See argumetns of mkinfit.gui
weight	See argumetns of mkinfit.gui
scaleVar	See argumetns of mkinfit.gui
ctr	a list of control values for the estimation algorithm to replace the default values including maximum iterations and absolute error tolerance. Defaults to the output of kingui.control.
irls.control	A list of control values for the estimation algorithm to replace the default values including the maximum number of iterations for the outer iteration and the error tolerance level for the error variance estimation updating.
update	If not NULL, should be a list of starting values obtained from other optimization methods.
...	Further arguments that will be passed to modFit.

Value

A list with "kingui", "mkinfit" and "modFit" in the class attribute. A summary can be obtained by summary.kingui.

Author(s)

Zhenglei Gao

See Also

IRLSkinfit, mkinfit.gui

Examples

```
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab", sink = TRUE,
                           k = list(ini = 0.1,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
               M0 = list(ini = 195,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
                           FF = list(ini = c(.1),
                      fixed = c(0),
                      lower = c(0),
                      upper = c(1)),
                      time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                    residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                           Metab = list(type = "SFO",
                           k = list(ini = 0.1   ,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
               M0 = list(ini = 0,
                      fixed = 1,
                      lower = 0,
                      upper = Inf),
                    residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))                              )
fit <- IRLSkinfit.gui(SFO_SFO_gui,plot=T,ctr=kingui.control(maxIter=100,
tolerance=1e-6,odesolver='lsoda'))
```

---

kingraph *Function to generate graph data for the GUI to make fit plot.*

Description

A data table is generated and stored in file=filename for later usage.

Usage

```
kingraph(fit, filename = "graphdata.txt",xlim = c(1, 1.05) *
range(fit$data$time), ylim = c(1, 1.05)* range(fit$data$observed, na.rm
= TRUE))
```

Arguments

fit	An object of class 'kingui'.
filename	The file in which the graph data will be stored.
xlim	The plot range for x axis.
ylim	The plot range for y axis.

Examples

```
## Not run:
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab",
               sink = TRUE,
               k = list(ini = 0.1,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
               M0 = list(ini = 195,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                             FF = list(ini = c(.1),
                        fixed = c(0),
                        lower = c(0),
                        upper = c(1)),
                        time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                      residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                             Metab = list(type = "SFO",
                             k = list(ini = 0.1   ,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
               M0 = list(ini = 0,
                        fixed = 1,
                        lower = 0,
                        upper = Inf),
                      residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))                              )
fit <- mkinfit.gui(SFO_SFO_gui,plot=T)
kingraph(fit)

## End(Not run)
```

---

kingui.control *Choose the algorithm to use and related control parameters in kinetic evaluations using NLS and IRLS.*

Description

Allow the user to set some characteristics of the optimization procedure.

Usage

```
kingui.control(method = "solnp", maxIter = 100, tolerance = 1e-08,
odesolver = "lsoda", atol = 1e-09, rtol = 1e-10, rhobeg = 0.05, iprint =
1, trace = 0, goMarq = 0, delta = 1e-06, rho = 1, submethod = "Marq",
...)
```

Arguments

method	The method to be used, one of "Rsolnp", "minqa","spg", "Marq", "Port", "Newton", "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", "Pseudo" - see details.
maxIter	Numerber of maximum iterations
tolerance	A positive numeric value specifying the tolerance level for the relative offset convergence criterion.
odesolver	the integration routines used. see ode
atol	absolute error tolerance, either a scalar or an array as long as 'y'. See ode
rtol	relative error tolerance, either a scalar or an array as long as 'y'.See ode.
rhobeg	Initial trust region radius for method 'bobyqa'. See details of see bobyqa.
iprint	control the amount of printing by setting iprint to 0, 1 2, or 3. See details of see bobyqa.
trace	A logical variable (TRUE/FALSE). If 'TRUE', information on the progress of optimization is printed. See details of see spg.
goMarq	If TRUE, using "Marq" for the iterations after the first step in IRLS.
delta	Control parameters for 'solnp'. See details of see solnp.
rho	Control parameters for 'solnp'. See details of see solnp.
submethod	If the method chosen failed to produce results, run the optimization using a substitute method.
...	Other characteristics of different optimizer for the users to play with.

Value

A list of control parameters for the ODE solver and the optimization routines.

Author(s)

Zhenglei Gao

Examples

```
kingui.control()
```

KINGUII-package *Routines for fitting kinetic models NLS, IRLS and MCMC methods.*

Description

R-package KINGUII is a GUI version of package mkin. It contains functions to fit kinetic models using NLS,IRLS and MCMC methods. It also includes a function to set up the kineticmodel with differential equations.

The main functions are:

- Function mkinmod.gui to set up the kinetic model, the data, and the initial values.
- Functions to allow fitting of the model to data: mkinfit.gui, IRLSkinfit.gui, and mcmckinfit.gui.
- Functions to summarize the results and produce output which can be read by the GUI: summary.kingui, summary.mcmckingui, kingraph, etc.

Details

Package:	KINGUII
Type:	Package
Version:	1.0
Date:	2011-06-21
License:	GPL
LazyLoad:	yes

Author(s)

Johannes Ranke and Zhenglei Gao

Maintainer: Johannes Ranke <jranke@users.berlios.de>

See Also

<package-mkin>

Examples

```
## Not run:
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab", sink = TRUE,
                             k = list(ini = 0.1,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                  M0 = list(ini = 195,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                             FF = list(ini = c(.1),
```

```
                      fixed = c(0),
                      lower = c(0),
                      upper = c(1)),
                      time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                    residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                         Metab = list(type = "SFO",
                         k = list(ini = 0.1   ,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
             M0 = list(ini = 0,
                      fixed = 1,
                      lower = 0,
                      upper = Inf),
                    residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))                          )
fit1 <- mkinfit.gui(SFO_SFO_gui)
fit2 <- IRLSkinfit.gui(SFO_SFO_gui)
fit3 <- mcmckinfit.gui(SFO_SFO_gui)

## End(Not run)
```

`mcmckinfit.gui` *Fit a kinetic model using MCMC methods.*

Description

This function does kinetic evaluations using MCMC algorithm with function `modMCMC` in the `FME` package.

Usage

```
mcmckinfit.gui(mkinmodini, eigen = FALSE, ctr = kingui.control(),
plottitle = "", plot = FALSE, quiet = FALSE, err = NULL, weight =
"none",scaleVar = FALSE, commonsigma = FALSE, jump = NULL, prior = NULL,
wvar0 = 0.1, niter = 1000, outputlength = niter, burninlength = 0,
updatecov = niter, ntrydr = 1, drscale = NULL, verbose = TRUE,
fitstart = TRUE, update = NULL, ...)

## S3 method for class 'mcmckingui':
summary(object, remove = NULL, ...)

## S3 method for class 'mcmckingui':
plot(object, fname1, fname2, pch = 1, device = "wmf", ...)
```

Arguments

`mkinmodini`	A list of class `mkinmod.gui`, containing the kinetic model to be fitted to the data, and the initial parameter values, the observed data.
`eigen`	If TRUE, the solution of the system of differential equations should be based on the spectral decomposition of the coefficient matrix in cases that this is possible.

ctr	Used when fitstart is TRUE. A list of control values for the estimation algorithm to replac the default values including maximum iterations and absolute error tolerance. Defaults to the output of kingui.control.
plottitle	The title of the plot for visualizing the optimization process.
plot	When fitstart==TRUE, if TRUE,the observed values and the numerical solutions should be plotted at each stage of the first optimization step.
quiet	If TRUE, suppress printing out the current model cost after each(>1) improvement.
commonsigma	If TRUE,the error model has constant error variance and the NLS algorithm will be used for the first optimizatio step.
jump	jump length, either a number, a vector with length equal to the total number of parameters, a covariance matrix, or a function that takes as input the current values of the parameters and produces as output the perturbed parameters. See details of modMCMC.
prior	prior probability of the parameters, either a function of the parameters or 'NULL'; in the latter case a flat prior is used (i.e. all parameters are equally likely).
wvar0	"weight" for the initial model variance. See details of modMCMC.
niter	number of iterations for the MCMC.
outputlength	number of iterations kept in the output.
burninlength	number of discarded initial iterations.
updatecov	number of iterations after which the parameter covariance matrix is (re)evaluated based on the parameters kept thus far, and used to update the MCMC jumps.
ntrydr	maximal number of tries for the delayed rejection procedure.
drscale	for each try during delayed rejection, the cholesky decomposition of the proposal matrix is scaled with this amount; if 'NULL', it is assumed to be 'c(0.2,0.25, 0.333, 0.333, ...)'
verbose	if 'TRUE': prints extra output.
fitstart	if 'TRUE': first perform an optimization step and using the fitted parameters as the starting values for MCMC.
update	if not NULL, using the values in the update as the starting values for MCMC.
...	Further arguments that will be passed to modFit.

Value

A list with "mcmckingui" and "modMCMC" in the class attribute. A summary can be obtained by summary.mcmckingui.

Author(s)

Zhenglei Gao

See Also

modMCMC

Examples

```
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab", sink = TRUE,
                           k = list(ini = 0.1,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
               M0 = list(ini = 195,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
                           FF = list(ini = c(.1),
                      fixed = c(0),
                      lower = c(0),
                      upper = c(1)),
                      time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                    residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                           Metab = list(type = "SFO",
                           k = list(ini = 0.1   ,
                      fixed = 0,
                      lower = 0,
                      upper = Inf),
               M0 = list(ini = 0,
                      fixed = 1,
                      lower = 0,
                      upper = Inf),
                    residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))                              )

fit <- mcmckinfit.gui(SFO_SFO_gui)

summary(fit)
plot(fit)
```

`mkinfit.gui` *Fit a kinetic model using the NLS or WNLS algorithm.*

Description

GUI version of `mkinfit`. This function uses the Flexible Modelling Environment package `FME` to create a function calculating the model cost with weigths, which is then minimised , using the specified initial or fixed parameters and starting values.

Usage

```
mkinfit.gui(mkinmodini, eigen = FALSE, plot = FALSE, plottitle = "",
quiet = FALSE, err = NULL, weight = "none", scaleVar = FALSE,
ctr = kingui.control(), ...)

## S3 method for class 'kingui':
summary(object, data = TRUE, distimes = TRUE, ff =
TRUE, cov = FALSE, ...)
```

Arguments

mkinmodini	A list of class mkinmod.gui, containing the kinetic model to be fitted to the data, and the initial parameter values, the observed data.
eigen	If TRUE, the solution of the system of differential equations should be based on the spectral decomposition of the coefficient matrix in cases that this is possible.
plot	If TRUE,the observed values and the numerical solutions should be plotted at each stage of the optimization.
plottitle	The title of the plot for visualizing the optimization process.
quiet	If TRUE, suppress printing out the current model cost after each(>1) improvement.
err	either NULL, or the name of the column with the *error* estimates, used to weigh the residuals (see details of modCost); if NULL, then the residuals are not weighed. In the GUI version, there is no need to consider this argument since a default weight one matrix is setup in mkinmod.gui. The err argument turned into 'err' automatically in the codes.
weight	only if err=NULL: how to weigh the residuals, one of "none", "std", "mean", see details of modCost.
scaleVar	Will be passed to modCost. Default is not to scale Variables according to the number of observations.
ctr	A list of control values for the estimation algorithm to replace the default values including maximum iterations and absolute error tolerance. Defaults to the output of kingui.control.
...	Further arguments that will be passed to modFit.

Value

A list with "kingui", "mkinfit" and "modFit" in the class attribute. A summary can be obtained by summary.kingui.

Author(s)

Zhenglei Gao

See Also

mkinfit

Examples

```
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab", sink = TRUE,
                                k = list(ini = 0.1,
                           fixed = 0,
                           lower = 0,
                           upper = Inf),
                      M0 = list(ini = 195,
                           fixed = 0,
                           lower = 0,
                           upper = Inf),
```

```
                          FF = list(ini = c(.1),
                        fixed = c(0),
                        lower = c(0),
                        upper = c(1)),
                        time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                      residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                          Metab = list(type = "SFO",
                          k = list(ini = 0.1   ,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                M0 = list(ini = 0,
                        fixed = 1,
                        lower = 0,
                        upper = Inf),
                      residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))                              )
fit <- mkinfit.gui(SFO_SFO_gui,plot=T,ctr=kingui.control(maxIter=100,
tolerance=1e-6,odesolver='lsoda'))
```

`mkinmod.gui` *Function to set up a kinetic model with one or more compartments.*

Description

GUI version of `mkinmod`. The function takes a specification, consisting of a list of the compartments in the data. Each compartment is again represented by a list, specifying the kinetic model type, reaction or transfer to other observed compartments, the initial parameter values, lower and upper bounds, fixed or not, and observed data.

Usage

```
mkinmod.gui(...)
```

Arguments

`...`	Each list cell represents a comparment which contains a list of comonents including 'type'(kinetic reaction type, single first order kinetics "SFO" are implemented for all compartments, while "FOMC", "DFOP" and "HS" can additionally be chosen for the first variable which is assumed to be the source compartment), each parameter name(a list of 'ini','fixed','lower','upper'),'residue'(measured concentrations),'time'(sampling time),'weight'(weights to be used, defaul 1),'sink'(Default TRUE, tranformation to unspecified compartments.),'to'(a vector of compartment names that the source compartment will be transferred to).

Value

A list of class 'mkinmod.gui' for use with `mkinfit.gui`,`IRLSkinfit.gui` and `mcmckinfit.gui` containing:

`diffs`	A vector of string representations of differential equations, one for each modelling compartment.

parms	A vector of parameter names occurring in the differential equations.
map	A list containing named character vectors for each compartments in the model.
parms.ini	Initial values for all kinetic parameters in the model.
state.ini	Initial state values for all compartments in the model.
lower	Lower bounds for the parameters(including state variables) to be optimized.
upper	upper bounds for the parameters(including state variables) to be optimized.
fixed_parms	The names of the kinetic parameters that are fixed during optimization.
fixed_initials	The names of the initial states that are fixed during optimization.
residue	The observed data matrix with a time column.
weightmat	The weights matrix.
ff	A vector of string representations of the transformation between the formation fractions in the model and the transfomed formation fractions in the optimization process.

See Also

mkinmod

Examples

```
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab", sink = TRUE,
                           k = list(ini = 0.1,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                 M0 = list(ini = 195,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                           FF = list(ini = c(.1),
                        fixed = c(0),
                        lower = c(0),
                        upper = c(1)),
                        time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                      residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                           Metab = list(type = "SFO",
                           k = list(ini = 0.1   ,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                 M0 = list(ini = 0,
                        fixed = 1,
                        lower = 0,
                        upper = Inf),
                      residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))
```

`plot.mcmckingui` *%% ~~function to do ... ~~ S3 method to plot for calss 'mcmckingui'*

Usage

```
plot.mcmckingui(object, fname1, fname2, pch = 1, device = "wmf", ...)
```

Arguments

`object`	An object of class 'mcmckingui'
`fname1`	The file name of the density plot.
`fname2`	The file name of the correlation plot.
`pch`	What kind of points to use in the plots.
`device`	The plot device to be used.
`...`	Other arguments to be passed to 'plot'.

Value

Density and Correlation plots of the sampled parameters in 'wmf' or other format.

Author(s)

Zhenglei Gao

`schaefer07_complex_case`
Metabolism data set used for checking the software quality of KinGUI

Description

This dataset was used for a comparison of KinGUI and ModelMaker to check the software quality of KinGUI in the original publication (Schäfer et al., 2007). The results from the fitting are also included.

Usage

```
data(schaefer07_complex_case)
```

Format

The data set is a data frame with 8 observations on the following 6 variables.

`time` a numeric vector

`parent` a numeric vector

`A1` a numeric vector

`B1` a numeric vector

`C1` a numeric vector

`A2` a numeric vector

The results are a data frame with 14 results for different parameter values

Source

Schäfer D, Mikolasch M, Rainbird P and Harvey B (2007). KinGUI: a new kinetic software tool for evaluations according to FOCUS degradation kinetics. In: Del Re AAM, Capri E, Fragoulis G and Trevisan M (Eds.). Proceedings of the XIII Symposium Pesticide Chemistry, Piacenza, 2007, p. 916-923.

Examples

```
data <- mkin_wide_to_long(schaefer07_complex_case, time = "time")
model <- mkinmod(
  parent = list(type = "SFO", to = c("A1", "B1", "C1"), sink = FALSE),
  A1 = list(type = "SFO", to = "A2"),
  B1 = list(type = "SFO"),
  C1 = list(type = "SFO"),
  A2 = list(type = "SFO"))
## Not run: mkinfit(model, data, plot=TRUE)
```

---

summary.kingui *S3 method for class 'kingui'*

Usage

```
summary.kingui(object, data = TRUE, distimes = TRUE, ff = TRUE, cov =
FALSE, ...)
```

Arguments

object	An object of class 'kingui' from the result of NLS or IRLS fit.
data	If TRUE, include in the returned values a data frame containing the observed and predicted values with residuals and estimated standard deviations or weights.
distimes	If TRUE, DT50 and DT90 values should be included.
ff	If TRUE, the formation fraction should be calculated from the estimated transformed parameters.
cov	If TRUE, parameter covariances should be calculated.
...	Optional arguments passed to methods like 'print'

Value

The summary function returns a list with the same components as 'summary.mkinfit', and the additional components from the results of the optimization routine used.

See Also

summary.mkinfit, summary.modFit

Examples

```
## Not run:
SFO_SFO_gui <- mkinmod.gui(Parent = list(type = "SFO", to = "Metab", sink = TRUE,
                           k = list(ini = 0.1,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                M0 = list(ini = 195,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                           FF = list(ini = c(.1),
                        fixed = c(0),
                        lower = c(0),
                        upper = c(1)),
                        time=c(0.0,2.8,   6.2,  12.0,  29.2,  66.8,  99.8,
127.5, 154.4, 229.9, 272.3, 288.1, 322.9),
                      residue = c( 157.3, 206.3, 181.4, 223.0, 163.2,
144.7,  85.0,  76.5,  76.4,  51.5,  45.5,  47.3, 42.7)),
                           Metab = list(type = "SFO",
                           k = list(ini = 0.1   ,
                        fixed = 0,
                        lower = 0,
                        upper = Inf),
                M0 = list(ini = 0,
                        fixed = 1,
                        lower = 0,
                        upper = Inf),
                      residue =c( 0.0,  0.0,  0.0,  1.6,  4.0, 12.3, 13.5,
12.7, 11.4, 11.6, 10.9,  9.5,  7.6))                              )
fit1 <- mkinfit.gui(SFO_SFO_gui)
summary(fit1)
fit2 <- IRLSkinfit.gui(SFO_SFO_gui)
summary(fit2)

## End(Not run)
```

Index