Package 'pksensi'

January 29, 2019

Type Package

Title Global Sensitivity Analysis in Pharmacokinetic Modeling

Version 1.0.1

Depends R (>= 3.1)

Imports ggplot2, data.table, deSolve, dplyr, getPass, magrittr, reshape

Description Applying the global sensitivity analysis workflow to investigate the parameter uncertainty and sensitivity in pharmacokinetic (PK) models, especially the physiologically-based pharmacokinetic (PBPK) model with multivariate outputs. The package also provide some functions to check the sensitivity measures and its convergence of model parameters.

License GPL-3 | file LICENSE

URL https://github.com/nanhung/pksensi, https://nanhung.rbind.io/pksensi

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat, viridis

LazyData true

BugReports https://github.com/nanhung/pksensi/issues

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author Nan-Hung Hsieh [aut, cre] (<https://orcid.org/0000-0003-0163-2766>), Brad Reisfeld [aut], Weihsueh A. Chiu [aut] (<https://orcid.org/0000-0002-7575-2368>)

Maintainer Nan-Hung Hsieh <nhsieh@cvm.tamu.edu>

Repository CRAN

Date/Publication 2019-01-29 18:00:03 UTC

R topics documented:

about-pksensi 2
APAP 3
check 3
compile_model 5
mcsim_install 6
models 7
pksim 8
rfast99 9
solve_fun 10
solve_mcsim 11
tell2 13

Index **15**

`about-pksensi` *About* **pksensi** *package*

Description

Applying a global sensitivity analysis approach to reduce parameter dimensionality in pharmacokinetic modeling and evaluate the robustness of the algorithm under the given sampling number.

Details

The "extended Fourier amplitude sensitivity testing (eFAST)" method, a variance-based sensitivity analysis method is used to estimate the parameter impact on model output (Saltelli et al. 1999). The eFAST is the effective algorithm to determine the influential parameter in physiologically-based pharmacokinetic model calibration (Hsieh et al. 2018). The eFAST algorithm is sourced from **sensitivity** package but implemented the random-phase shift to evaluating the robustness of sensitivity measurement under the given sample size.

References

A. Saltelli, S. Tarantola and K. Chan, 1999, A quantitative, model independent method for global sensitivity analysis of model output, *Technometrics*, 41, 39-56

N-H Hsieh, B Reisfeld, FY Bois, WA, Chiu, 2018, Applying a global sensitivity analysis workflow to improve the computational efficiencies in physiologically-based pharmacokinetic modeling, *Froniers in Pharmacology*, 9, 588.

APAP *Pharmacokinetics of Acetaminophen*

Description

The APAP dataset contains a human experiment of pharmacokinetic data for acetaminophen, and it's major metabolites.

Usage

```
APAP
```

Format

A data frame containing the following columns:

- Wt: averaged weight of the study population (kg).
- Dose: given oral dose of acetaminophen administered (mg/kg).
- Time: time after drug administration (hr).
- APAP: concentration of acetaminophen in the sample (mg/L).
- AG: concentration of acetaminophen-glucuronide in the sample (mg/L).
- AS: concentration of acetaminophen-sulfate in the sample (mg/L).
- Study: Sourced reference.

References

T. J. Zurlinden and B. Reisfeld, 2016, Physiologically based modeling of the pharmacokinetics of acetaminophen and its major metabolites in humans using a Bayesian population approach, *European Journal of Drug Metabolism and Pharmacokinetics*, 79(4), 1-26.

check *Check the Parameter Sensitivity*

Description

Visualize and check the sensitivity (or convergence) measurement with a given result.

Usage

```
check(x, times, vars, SI.cutoff, CI.cutoff)

heat_check(x, order = c("first order", "interaction", "total order"),
  vars = NULL, times = NULL, SI.cutoff = c(0.05, 0.1),
  CI.cutoff = c(0.05, 0.1), index = "SI", level = T, text = F)

## S3 method for class 'rfast99'
plot(x, vars = 1, SI.cutoff = 0.1, ...)

## S3 method for class 'rfast99'
print(x, ...)
```

Arguments

x	a list of storing information in the defined sensitivity function.
times	a logical value or character to specific the display time in simulation.
vars	a logical value or character to specific the display variable in simulation.
SI.cutoff	a value or vector to set the cut-off for sensitivity index. The default is 0.05.
CI.cutoff	a value or vector to set the cut-off for convergence index. The default is 0.05.
order	a vector of interested output index included `first order`, `interaction`, and `total order`.
index	a character to choose sensitivity index `SI` (default) or convergence index `CI`.
level	a logical value to use continuous or discrete (default) output.
text	a logical value to display the calculated indices in the plot.
...	additional arguments to customize the graphical parameters.

Details

The convergence of sensitivity indices for each parameter is using the approach proposed by Sarrazin et al. (2016). This method quantitatively assesses the convergence by computing the range of 95 Using a global approach based on a heatmap visualization combined with an index "cut-off," can systematically distinguish between "influential" and "non-influential" parameters (Hsieh et al. 2018).

Value

The `print` function returns sensitivity and convergence indices with given time-step in console. The `check` method provides the summary of parameter sensitivity and convergence according to the given `SI.cutoff` and `CI.cutoff`. It can distinguish the influential and non-influential parameter by the providing value of `SI.cutoff`. The `plot` function can generate the time-course functional outputs of first order and interaction indices for each parameter. The default output is the first model variable. The `heat_check` provides a convenient way to visualize and distinguish the influential and non-influential parameter by the setting cut-off. The convergence index can examine the stability of sensitivity index. To check convergence, be sure to conduct the replication in `rfast99`.

References

F Sarrazin, F Pianosi, T Wagener, 2016, Global sensitivity analysis of environmental models: convergence and validation, *Environ. Model. Softw*, 79, 135–152.

N-H Hsieh, B Reisfeld, FY Bois, WA, Chiu, 2018, Applying a global sensitivity analysis workflow to improve the computational efficiencies in physiologically-based pharmacokinetic modeling, *Front. Pharmacol*, 9, 588.

See Also

tell2

Examples

```
q <- "qunif"
q.arg <- list(list(min = 0.6, max = 1),
  list(min = 0.5, max = 1.5),
  list(min = 0.02, max = 0.3),
  list(min = 20, max = 60))

params <- c("F","KA","KE","V")

set.seed(1234)
x <- rfast99(params = params, n = 200, q = q, q.arg = q.arg, rep = 20)

time <- seq(from = 0.25, to = 12.25, by = 0.5)
y <- solve_fun(x, model = FFPK, time = time, vars = "output")


tell2(x,y) # Link decoupling simulation result
x

# Check results of sensitivity measures
check(x)
plot(x)
heat_check(x)
heat_check(x, index = "CI")
```

---

compile_model *Model Compiler*

Description

The compile_model is used to compile the C file or MCSim's model file to generate the executable file in numerical analysis.

Usage

```
compile_model(mName, application = "mcsim", use_model_file = TRUE,
  version = NULL)
```

Arguments

mName	a string giving the name of the model or C file (without extension).
application	a character to assign the specific methods (mcsim or R) that will be applied to the numerical analysis (default is mcsim).
use_model_file	a logical value to operate the compiler to use model or C file, the default is set to TRUE to assign the MCSim's model file in compiling.
version	a character to assign the version of MCSim that had been installed. The version must be assigned for Windows user.

Details

Generally, the solving function through MCSim can provide faster speed than exporting C in R. Therefore, this function set `use_model_file = TRUE` and `application = 'mcsim'` as a default setting and suggest to use MCSim to solve the differential equation. To compile MCSim in Windows, be sure to install Rtools or MinGW first. For Windows user, to compile MCSim's model file, the `version` of MCSim should provide to conduct model compiling.

Value

The default `application` is set to `'mcsim'` to generate the executable file to solve differential equations by MCSim. If `application = 'R'`, the function will compile and create dynamic-link libraries (.dll) on Windows and shared objects (.so) on Unix-likes systems (e.g., Linux and MacOS).

---

mcsim_install *Install MCSim*

Description

Download the latest or specific version of GNU MCSim from the official website (https://www.gnu.org/software/mcsim/) and install it to the system directory.

Usage

```
mcsim_install(version = "6.0.1", directory = NULL, mxstep = 500)

mcsim_version()
```

Arguments

version	a character of MCSim version number.
directory	a character to assign the directory to put the GNU MCSim files.
mxstep	a numeric value to assign the maximum number of (internally defined) steps allowed during one call to the solver.

Details

This function aims to help users install GNU MCSim more easily. However, if you can not install it through this function. You might need to follow the instruction of GNU MCSim and install it, manually: https://www.gnu.org/software/mcsim/mcsim.html#Installation

The default `mxstp` is setting to 500. The user can increase `mxstp` to avoid possible error return. If you meet any error when conduct sensitivity analysis, you can this function to reinstall GNU MCSim and set the higher `mxstp`. The default `directory` to install MCSim is under `/home/username` (Linux), `/Users/username` (MacOS), and `C:/Users/` (windows). To install MCSim in Windows, be sure to install Rtools or MinGW first.

Functions

- `mcsim_version`: Return the version number of GNU MCSim.

References

F.Y. Bois, and D. Maszle, 1997, MCSim: A Monte Carlo Simulation Program, *Journal of Statistical Software*, 2(9): 1–60.

https://www.gnu.org/software/mcsim/

Examples

```
## Not run: mcsim_install(version = 6.0.1, mxstep = 10000)
```

`models` *Example PK Model for Sensitivity Analysis*

Description

Three examples are included: Flip-flop pharmacokinetic model, one-compartment toxicokinetic model from **httk** (Pearce et al. 2017), and acetaminophen pharmacokinetic model (Zurlinden et al. 2016).

Usage

```
FFPK(params, time, dose = 1)

pbtk1cpt_model()

pbpk_apap_model()
```

Arguments

`params`	a parameter matrix containing the input sample.
`time`	the given time-points.
`dose`	a given dose.

Functions

- pbtk1cpt_model: Download pbtk1cpt.model file.
- pbpk_apap_model: Download pbpk_apap.model file.

References

R. Pearce, R. Setzer, C. Strope, N. Sipes and J. Wambaugh, 2017, httk: R Package for High-Throughput Toxicokinetics, *Journal of Statistical Software*, 79(4), 1-26.

T. J. Zurlinden and B. Reisfeld, 2016, Physiologically based modeling of the pharmacokinetics of acetaminophen and its major metabolites in humans using a Bayesian population approach, *European Journal of Drug Metabolism and Pharmacokinetics*, 79(4), 1-26.

Examples

```
params <- c(F = 0.9, KA = 1.2, KE = 0.2, V = 1.5)
t <- seq(0, 12, 0.1)
C <-FFPK(params = params, time = t)
plot(t, C, type = "l", xlab = "time", ylab = "concentration")
```

---

pksim *Pharmacokinetic Simulation from Sampling Parameter*

Description

Pharmacokinetic plot of the output results based on the given parameter (Uncertainty analysis). If the user define the multiple output in model, the generated result will based on first model variable (default). A pharmacokinetic plot with median and the range of min-max, 10

Usage

```
pksim(y, vars = 1, log = F, legend = T, ...)
```

Arguments

y	a numeric array created from solve_fun or solve_mcsim function.
vars	a logical value or character to specific the display variable in simulation (default 1).
log	a logical value to transform the y-axis to log scale.
legend	a logical value to display the legend in the created plot.
...	additional arguments to customize the graphical parameters.

`rfast99` *Extended Fourier Amplitude Sensitivity Test with Random Phase Shift*

Description

`rfast99` is used to create the sequances for each parameter. It is based on the `fast99` function in **sensitivity** package.

Usage

```
rfast99(params, n, M = 4, omega = NULL, q = NULL, q.arg = NULL,
  replicate = 1, conf = 0.95)
```

Arguments

`params`	an integer for the giving number of parameters, or a vector of character strings giving their names.
`n`	an integer for the sampling number.
`M`	an integer specifying the interference parameter. The default is 4.
`omega`	a vector giving the set of frequencies.
`q`	a vector of quantile functions names corresponding to wanted parameters distributions.
`q.arg`	a list of quantile functions parameters.
`replicate`	an integer to define the number of replication. The default is set to 1 turn off the replication.
`conf`	the confidence level for replication confidence intervals. The default is 0.95.

Value

The returned parameter value will be stored in an array with c(model evaluation, replication, parameters).

References

A. Saltelli, S. Tarantola and K. Chan, 1999, A quantitative, model independent method for global sensitivity analysis of model output, *Technometrics*, 41, 39-56.

R. I. Cukier, H. B. Levine and K. E. Schuler, 1978, Nonlinear sensitivity analysis of multiparameter model systems. *Journal of Computational Physics*, 26, 1–42.

Examples

```
# Generate the parameter matrix with 20 replications
q <- "qunif"
q.arg <- list(min = 0, max = 1)

set.seed(1234)
x <- rfast99(params = 3, n = 100, replicate = 20, q = q, q.arg = q.arg)
dim(x$a) # the array of c(model evaluation, replication, parameters).

## Not run:
save(x, file = "input_parameters.rda")

## End(Not run)
```

`solve_fun` *Solve PK Model Through deSolve Package or Analytical Function*

Description

The `solve_fun` can solve time-dependent quantities/concentrations of different variables in PK model through the imported deSolve function. It can be used to solve the function with analytical solution.

Usage

```
solve_fun(x, time = NULL, initParmsfun = "initParms", initState,
  dllname, func = "derivs", initfunc = "initmod", outnames,
  method = "lsode", rtol = 1e-08, atol = 1e-12, model = NULL,
  lnparam = F, vars = NULL, ...)
```

Arguments

`x`	a list of storing information in the defined sensitivity function.
`time`	a vector to define the given time sequence.
`initParmsfun`	a character for the given specific initial parameter function.
`initState`	a vector that define the initial values of state variables for the ODE system.
`dllname`	a string giving the name of the shared library (without extension) that contains the compiled function.
`func`	the name of the function in the dynamically loaded shared library.
`initfunc`	the name of the initialization function (which initialises values of parameters), as provided in dllname.
`outnames`	the names of output variables calculated in the compiled function `func`.
`method`	method used by integrator (**desolve**).
`rtol`	argument passed to integrator (**desolve**).

atol	argument passed to integrator (**desolve**).
model	the defined analytical equation with functional output.
lnparam	a logical value that make the statement of the log-transformed parameter (default FALSE).
vars	a character for the selected output.
...	additional arguments for deSolve::ode method.

References

S. Karline, T. Petzoldt and R. Setzer. 2010. Solving Differential Equations in R: Package deSolve. *Journal of Statistical Software*, 33(9), 1–25.

K. Soetaert, T. Petzoldt, R.W. Setzer, 2010, Solving differential equations in R: package deSolve, *J. Stat. Soft.*, 33:9

See Also

pksim

Examples

```
q <- "qunif"
q.arg <- list(list(min = 0.6, max = 1.0),
 list(min = 0.5, max = 1.5),
 list(min = 0.02, max = 0.3),
 list(min = 20, max = 60))

params <- c("F","KA","KE","V")

set.seed(1234)
x <- rfast99(params = params, n = 200, q = q, q.arg = q.arg, rep = 20)

time <- seq(from = 0.25, to = 12.25, by = 0.5)
y <- solve_fun(x, model = FFPK, time = time, vars = "output")

pksim(y) # Visualize uncertainty of model output
```

---

solve_mcsim *Solve PK Model Through MCSim*

Description

The solve_mcsim can solve the differential equations of time-dependent quantity/concentration in different tissues/compartments through MCSim.

Usage

```
solve_mcsim(x, mName, infile.name = NULL, outfile.name = NULL,
  n = NULL, setpoint.name = NULL, params = NULL, vars = NULL,
  time = NULL, condition = NULL, generate.infile = T)

generate_infile(infile.name = NULL, outfile.name = NULL, params, vars,
  time, condition, rtol = 1e-06, atol = 1e-09, n = NULL,
  dist = NULL, q.arg = NULL)
```

Arguments

x	a list of storing information in the defined sensitivity function.
mName	a string giving the name of the model or C file (without extension).
infile.name	a character to assign the name of input file.
outfile.name	a character to assign the name of output file.
n	a numeric to define the sample number.
setpoint.name	a character to assign the name of file for parameter matrix.
params	a character to assign the testing parameters.
vars	a character or a vector to assign the selected output(s).
time	a numeric to define the given time point(s).
condition	a character to set the specific parameter value in the input file.
generate.infile	a logical value to automatically generate the input file, .
rtol	an argument passed to the integrator (default 1e-6).
atol	an argument passed to the integrator (default 1e-9).
dist	a vector of distribution names corresponding to <distribution-name> in MCSim.
q.arg	a list of shape parameters in the sampling distribution (dist).

Details

This function allows users to use external data file that assigned in setpoint.name as parameter matrix. If you want to use it, be sure to define n and setpoint.name.

Value

The output result is the 4-dimension array with c(model evaluations, replications, time-points, output variables).

Functions

- solve_mcsim: Numerical analysis for the PK model by MCSim.
- generate_infile: Generate the MCSim input file.

Examples

```
## Not run:
pbtk1cpt_model()
mName <- "pbtk1cpt"
compile_model(mName)

q <- "qunif"
q.arg <- list(list(min = 0.4, max = 1.1),
              list(min = 0.1, max = 0.4),
              list(min = 1.0, max = 3.0))

params <- c("vdist", "ke", "kgutabs")

set.seed(1234)
x <- rfast99(params = params, n = 200, q = q, q.arg = q.arg, rep = 20)

infile.name <- "example.in"
outfile.name <- "example.csv"
vars <- "Ccompartment"

t <- seq(from = 0.25, to = 12.25, by = 0.5)

y <- solve_mcsim(x, mName = mName, infile.name = infile.name,
                 setpoint.name = "setpoint.dat",
                 outfile.name = outfile.name, params = params, vars = vars, time = t,
                 condition = "Agutlument = 10")
pksim(y)

## End(Not run)
```

`tell2` *The Decoupling Simulations*

Description

Integrate the decoupling simulations (parameter sequences) and estimation results to compute the sensitivity measures.

Usage

```
tell2(x, y)
```

Arguments

`x`	a list of storing information in the defined sensitivity function.
`y`	a numeric array generated from the solutions of `solve_fun` or `solve_mcsim` function.

Source

This function is based on `tell` function in **sensitivity** package, which is an S3 generic method to estimate sensitivity measures by combining sensitivity object (rfast99) and external simulation results.

Index

∗Topic **datasets**
APAP, 3

about-pksensi, 2
APAP, 3

check, 3
compile_model, 5

FFPK *(*models*)*, 7

generate_infile *(*solve_mcsim*)*, 11

heat_check *(*check*)*, 3

mcsim_install, 6
mcsim_version *(*mcsim_install*)*, 6
models, 7

pbpk_apap_model *(*models*)*, 7
pbtk1cpt_model *(*models*)*, 7
pksensi-package *(*about-pksensi*)*, 2
pksim, 8, *11*
plot.rfast99 *(*check*)*, 3
print.rfast99 *(*check*)*, 3

rfast99, 9

solve_fun, 10
solve_mcsim, 11

tell2, *5*, 13