

# Package ‘rPBK’

February 26, 2024

**Title** Inference and Prediction of Generic Physiologically-Based Kinetic Models

**Version** 0.2.4

**BugReports** <https://gitlab.in2p3.fr/mosaic-software/rPBK/-/issues>

**URL** <https://gitlab.in2p3.fr/mosaic-software/rPBK/>

**Description** Fit and simulate any kind of physiologically-based kinetic ('PBK') models whatever the number of compartments. Moreover, it allows to account for any link between pairs of compartments, as well as any link of each of the compartments with the external medium. Such generic PBK models have today applications in pharmacology (PBPK models) to describe drug effects, in toxicology and ecotoxicology (PBTK models) to describe chemical substance effects. In case of exposure to a parent compound (drug or chemical) the 'rPBK' package allows to consider metabolites, whatever their number and their phase (I, II, ...). Last but not least, package 'rPBK' can also be used for dynamic flux balance analysis (dFBA) to deal with metabolic networks. See also Charles et al. (2022) <[doi:10.1101/2022.04.29.490045](https://doi.org/10.1101/2022.04.29.490045)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Biarch** true

**Depends** R (>= 3.4.0)

**Imports** ggplot2, methods, Rcpp (>= 0.12.0), rstan (>= 2.26.0), rstantools

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

**SystemRequirements** GNU make

**Suggests** knitr, loo, rmarkdown, testthat

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**LazyData** true

**NeedsCompilation** yes

**Author** Virgile Baudrot [aut, cre],  
 Sandrine Charles [aut],  
 Christelle Lopes [aut],  
 Ophélie Gestin [ctb],  
 Dominique Lamonica [ctb],  
 Aurélie Siberchicot [ctb]

**Maintainer** Virgile Baudrot <virgile.baudrot@qonfluens.com>

**Repository** CRAN

**Date/Publication** 2024-02-26 17:00:02 UTC

## R topics documented:

dataCompartment4 . . . . .	2
dataMaleGammarusSingle . . . . .	3
dataPBK . . . . .	3
df_quant95_ . . . . .	5
export_interpolate . . . . .	6
fitPBK . . . . .	6
fitPBK_C4 . . . . .	7
plot.fitPBK . . . . .	8
ppc . . . . .	8

**Index** **10**

---

dataCompartment4	<i>An example data set with 4 compartment</i>
------------------	---

---

### Description

A dataset containing an example with 4 compartments.

### Usage

```
data(dataCompartment4)
```

### Format

A data frame with 21 rows and 7 variables:

**temps** vector of time

**condition** exposure concentration

**replicat** replicate of experiment

**intestin** compartiment 'intestin'

**caecum** compartment 'caecum'  
**cephalon** compartment 'cephalon'  
**reste** compartment 'reste#'

---

dataMaleGammarusSingle

*An example data set with 1 compartment*

---

### Description

A dataset containing an example with a single compartment.

### Usage

```
data(dataMaleGammarusSingle)
```

### Format

A data frame with 22 rows and 4 variables:

**time** vector of time  
**expw** exposure concentration  
**replicate** replicat of experiment  
**conc** internal measured concentration#'

---

dataPBK

*Create a list giving data and parameters to use in the model inference.*

---

### Description

Create a list giving data and parameters to use in the model inference.

### Usage

```
dataPBK(object, ...)
```

```
## S3 method for class 'data.frame'  
dataPBK(  
  object,  
  col_time = NA,  
  col_replicate = NA,  
  col_exposure = NA,  
  col_compartment = NA,  
  time_accumulation = NA,
```

```

    ku_nest = NA,
    ke_nest = NA,
    k_nest = NA,
    ...
)

nested_model(object)

## S3 method for class 'stanPBKdata'
nested_model(object)

```

### Arguments

<code>object</code>	An object of class <code>stanPBKdata</code> (from <code>dataPBK()</code> function).
<code>...</code>	Further arguments to be passed to generic methods
<code>col_time</code>	Column name of the time column
<code>col_replicate</code>	Column name of the replicate column
<code>col_exposure</code>	Column name of the exposure column.
<code>col_compartment</code>	Column names of the compartment column. If several columns, give a vector with the column names.
<code>time_accumulation</code>	A scalar giving accumulation time.
<code>ku_nest</code>	Vector of binary (0 or 1) to select the uptake route. Use the <code>nested_model()</code> on the <code>stanPBKdata</code> object to check it.
<code>ke_nest</code>	Vector of binary (0 or 1) to select the excretion route. Use the <code>nested_model()</code> on the <code>stanPBKdata</code> object to check it.
<code>k_nest</code>	Matrix of binary (0 or 1) to select interaction routes. Use the <code>nested_model()</code> on the <code>stanPBKdata</code> object to check it.

### Value

A list with data and parameters require for model inference.

### Examples

```

# (1) load data file
data("dataCompartment4")
# (2) prepare data set
dataPBK_C4 <- dataPBK(
  object = dataCompartment4,
  col_time = "temps",
  col_replicate = "replicat",
  col_exposure = "condition",
  col_compartment = c("intestin", "reste", "caecum", "cephalon"),
  time_accumulation = 7)

# (1) load data file

```

```

data("dataCompartment4")
# (2) prepare data set
dataPBK_C4 <- dataPBK(
  object = dataCompartment4,
  col_time = "temps",
  col_replicate = "replicat",
  col_exposure = "condition",
  col_compartment = c("intestin", "reste", "caecum", "cephalon"),
  time_accumulation = 7)
# (3) check nesting
nested_model(dataPBK_C4)
# (2bis)
dataPBK_C42 <- dataPBK(
  object = dataCompartment4,
  col_time = "temps",
  col_replicate = "replicat",
  col_exposure = "condition",
  col_compartment = c("intestin", "reste", "caecum", "cephalon"),
  time_accumulation = 7,
  ku_nest = c(1,1,1,1), # No Change here
  ke_nest = c(1,1,1,1), # No Change here
  k_nest = matrix(c(
    c(0,1,1,1),
    c(0,0,1,1),
    c(0,0,0,0),
    c(0,0,0,0)),
    ncol=4,nrow=4,byrow=TRUE) # Remove
  )
# (3bis) re-checking nesting
nested_model(dataPBK_C42)

```

df\_quant95\_

*Compute 95 credible intervals***Description**

Compute quantiles 95 credible intervals

**Usage**

```
df_quant95_(x, ...)
```

**Arguments**

x	An object of class <code>fitPBK</code>
...	Additional arguments

**Value**

An object of class `data.frame` returning median and 95 credible interval

---

export_interpolate	<i>Interpolate function implemented in Stan only export for checking</i>
--------------------	--

---

### Description

This function export the linear interpolation implemented in Stan. It can be use to re-sample the exposure profiles.

### Usage

```
export_interpolate(x, xpt, ypt, chain = 1, iter = 1, ...)
```

### Arguments

x	interpolation point x
xpt	a vector of x axis (has to be same size as ypt vector)
ypt	a vector of y axis (has to be same size as ypt vector)
chain	number of chain
iter	number of iteration
...	Arguments passed to <code>rstan::sampling</code>

### Value

A sample of a stanfit object returning a linear interpolation

---

fitPBK	<i>Bayesian inference of TK model with Stan</i>
--------	---

---

### Description

Bayesian inference of TK model with Stan

### Usage

```
fitPBK(stanPBKdata, ...)

## S3 method for class 'stanPBKdata'
fitPBK(stanPBKdata, ...)
```

### Arguments

stanPBKdata	List of Data require for computing
...	Arguments passed to <code>rstan::sampling</code> (e.g. iter, chains).

**Value**

An object of class `fitPBK` containing two object: `stanPBKdata` the data set used for inference and `stanfit` returned by `rstan::sampling`

**Examples**

```
# (1) load data file
data("dataCompartment4")
# (2) prepare data set
dataPBK_C4 <- dataPBK(
  object = dataCompartment4,
  col_time = "temps",
  col_replicate = "replicat",
  col_exposure = "condition",
  col_compartment = c("intestin", "reste", "caecum", "cephalon"),
  time_accumulation = 7)
# (3) run Bayesian fitting: <5 sec to be executed
# 1 chain and 10 iterations is fast to run but provide
# bad goodness-of-fit
fitPBK_C4_FASTbadGOF <- fitPBK(dataPBK_C4, chains = 1, iter = 10)

# (3) run Bayesian fitting: > 5 sec to be executed
# 4 chains and 2000 iterations provides better estimates
fitPBK_C4 <- fitPBK(dataPBK_C4, chains = 4, iter = 2000)
```

---

`fitPBK_C4`*An example of fitPBK object*

---

**Description**

A `fitPBK` object containing Bayesian inference of the `dataCompartment4` data set.

**Usage**

```
data(fitPBK_C4)
```

**Format**

A `fitPBK` object with:

**stanPBKdata** original data frame wrap in a formatted list for inference

**stanfit** a `stanfit` object resulting from inference with `stan`

---

<code>plot.fitPBK</code>	<i>Plotting method for fitPBK objects</i>
--------------------------	---

---

**Description**

This is the generic plot S3 method for the `fitTK`. It plots the fit obtained for each variable in the original dataset.

**Usage**

```
## S3 method for class 'fitPBK'  
plot(x, ...)
```

**Arguments**

<code>x</code>	And object returned by <code>fitPBK</code>
<code>...</code>	Additional arguments

**Value**

a plot of class `ggplot`

**Examples**

```
# (1) load a fitPBK object  
data("fitPBK_C4")  
# (2) plot result of bayesian fitting  
plot(fitPBK_C4)
```

---

<code>ppc</code>	<i>Posterior predictive check plot</i>
------------------	--

---

**Description**

Plots posterior predictive check for `fitPBK`

**Usage**

```
ppc(x, ...)  
  
## S3 method for class 'fitPBK'  
ppc(x, ...)
```

**Arguments**

*x*                    an object used to select a method *ppc*  
*...*                Further arguments to be passed to generic methods

**Value**

a plot of class *ggplot*

**Examples**

```
# (1) load a fitPBK object  
data("fitPBK_C4")  
# (2) plot ppc of bayesian fitting  
ppc(fitPBK_C4)
```

# Index

## \* datasets

- dataCompartment4, 2
- dataMaleGammarusSingle, 3
- fitPBK\_C4, 7

- dataCompartment4, 2
- dataMaleGammarusSingle, 3
- dataPBK, 3
- df\_quant95\_, 5

- export\_interpolate, 6

- fitPBK, 6
- fitPBK\_C4, 7

- nested\_model (dataPBK), 3

- plot.fitPBK, 8
- ppc, 8