Package 'Rivivc'

January 20, 2025

Type Package

Title In Vitro in Vivo Correlation Linear Level ``A"

Version 0.9.1

Date 2022-04-24

Author Aleksander Mendyk <mfmendyk@cyf-kr.edu.pl>, with contributions from Sebastian Polak <mfpolak@@cyf-kr.edu.pl>.

Maintainer Aleksander Mendyk <mfmendyk@cyf-kr.edu.pl>

Depends signal, compiler

Suggests graphics

Description It is devoted to the IVIVC linear level A with numerical deconvolution method. The latter is working for inequal and incompatible timepoints between impulse and response curves. A numerical convolution method is also available. Application domains include pharamaceutical industry QA/QC and R&D together with academic research.

License GPL (>= 3)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-23 23:50:10 UTC

Contents

Rivivc-package *IVIVC LEVEL A*

Description

This package performs linear iv vitro in vivo correlation of linear level A. It provides numerical convolution/deconvolution procedures with unequal time steps and no assumptions about the function shapes.

Details

Package:	Rivivc
Type:	Package
Version:	0.9
Date:	2012-10-03
License:	GPLv3

Author(s)

Aleksander Mendyk and Sebastian Polak

Maintainer: Aleksander Mendyk <mfmendyk@cyf-kr.edu.pl>

References

Langenbucher (2003) F. Handling of computational in vitro/in vivo correlation problems by Microsoft Excel: III. Convolution and deconvolution. *Eur J Pharm Biopharm.* **56**, 429-37.

impulse *PK profile after drug intravenous administration*

Description

This data set gives the time and concentration of the hypothetical drug after its intravenous administration. This is the simulated data set.

Usage

```
data(impulse)
```

Format

matrix

input *In vivo absorption of the drug*

Description

This data set gives the time and cumulative amount of the hypothetical drug absorbed. It is also used as in vitro dissolution for Rivivc example of IVIVC level A. This is the simulated data set.

Usage

```
data(input)
```

Format

matrix

NumConv *Numerical convolution*

Description

Performs numerical convolution independent of the sampling points but requiring the same timescale of the input and impulse profiles.

Usage

```
NumConv(impulse.matrix,input.matrix,conv.timescale = NULL,
    explicit.interpolation = 1000)
```

Arguments

`impulse.matrix`	matrix of the PK profile after the drug intravenous (i.v.) administration
`input.matrix`	cumulative in vivo absorption profile
`conv.timescale`	a timescale of convolution defined either as a whole vector with specific timepoints `c(t1,t2,...tN)` or two-element vector containing only lower and upper boundery of the required prediction timescale `c(lower,upper)`; in the latter case system creates the time vector based on the parameter `explicit.interpolation`; if omitted it computes convolution timescale based on the input matrix
`explicit.interpolation`	sampling accuracy used by the interpolation method to find the same timepoints for input and impulse profiles

Value

Output values are:

`$par`	convolved time profile based on the original timescale
`$par_explicit`	provides convolution with the explicit interpolation

Author(s)

Aleksander Mendyk and Sebastian Polak

See Also

NumDeconv,

Examples

```
require(Rivivc)
require(graphics)

#i.v. data
data("impulse")
#p.o. PK profile
data("resp")
#in vitro dissolution for correlation purposes
data("input")

#preparing data matrices
input_mtx<-as.matrix(input)
impulse_mtx<-as.matrix(impulse)
resp_mtx<-as.matrix(resp)

#setting interpolation accuracy
accur_explic<-1000

#run convolution
result<-NumConv(impulse_mtx,input_mtx,explicit.interp=accur_explic)

print("Raw results")
print(result$par)

print("Raw results explicit")
print(result$par_explicit)

dev.new()
plot(resp_mtx)
lines(result$par, type="l", col="blue")

dev.new()
plot(resp_mtx)
lines(result$par_explicit, type="l", col="blue")
```

NumDeconv *Numerical deconvolution method*

Description

Numerical deconvolution method based on the convolution and the `optim()` BFGS method to find in vivo absorption profile through the convolution approach. The function works iteratively with the cumulative in vivo absorption profile optimization performed by the BFGS method in regard to the convolved PK profile and its proximity to the real known p.o. profile.

Usage

```
NumDeconv(impulse.matrix,resp.matrix,dose_iv=NULL,dose_po=NULL,
    deconv.timescale = NULL, explicit.interpolation = 20,
    implicit.interpolation = 10, optim.maxit = 200)
```

Arguments

`impulse.matrix`	matrix of the PK profile after the drug intravenous (i.v.) administration
`resp.matrix`	PK profile after oral (p.o.) administration of the drug
`dose_iv`	drug dose after i.v. administration; not obligatory but if provided must be in the same units like the dose p.o.
`dose_po`	drug dose after p.o. administration; not obligatory but if provided must be in the same units like the dose i.v.
`deconv.timescale`	a timescale of deconvolution defined either as a whole vector with specific time-points `c(t1,t2,...tN)` or two-element vector containing only lower and upper boundery of the required prediction timescale `c(lower,upper)`; in the latter case system creates the time vector based on the parameter `explicit.interpolation`; if omitted it computes deconvolution timescale based on the impulse matrix
`explicit.interpolation`	deconvolution explicit interpolation parameter, namely number of the curve interpolation points used directly by the `optim()` method
`implicit.interpolation`	implicit interpolation - a factor multiplying `explicit.interpolation` for better accuracy
`optim.maxit`	maximum number of iterations used by `optim()` method

Details

This method is an empirical approach to the deconvolution method with minimum mechanistic assumptions. Yet the latter involve kinetics linearity when the doses of i.v. and p.o. are different, thus the i.v. profile is scaled by multiplication with the factor of `dose_po/dose_iv`. It is also important to know that large values of explicit and/or implicit accuracy lead to the long execution times. The recommended values are `explicit = 20` and `implicit = 10`, however this is only a rule of thumb used here. When looking for higher accuracy it is advisable to increase implicit interpolation prior to the explicit.

Value

Three matrices are returned at the output of the function:

`$par`	represents original timescale provided at the input
`$par_explicit`	provides deconvolution with the explicit interpolation
`$par_implicit`	provides deconvolution with the implicit interpolation

Author(s)

Aleksander Mendyk and Sebastian Polak

See Also

RivivcA

Examples

```
require(Rivivc)
require(graphics)

#i.v. data
data("impulse")
#p.o. PK profile
data("resp")
#in vitro dissolution for correlation purposes
data("input")

#preparing data matrices
input_mtx<-as.matrix(input)
impulse_mtx<-as.matrix(impulse)
resp_mtx<-as.matrix(resp)

#setting accuracy for both interpolation modes
accur_explic<-10
accur_implic<-5


#for deconvolution
result<-NumDeconv(impulse_mtx,resp_mtx,explicit.interp=accur_explic,implicit.interp=accur_implic)


print("Raw results")
print(result$par)

print("Explicit interpolation")
print(result$par_explicit)


print("Implicit interpolation")
print(result$par_implicit)
```

```
#let's compare the deconvolved curve with known input
dev.new()
plot(input_mtx)
lines(result$par, type="l", col="blue")
```

---

resp *PK profile after drug oral administration*

Description

This data set gives the time and concentration of the hypothetical drug after its oral administration. This is the simulated data set.

Usage

```
data(resp)
```

Format

matrix

RivivcA *Level A linear correlation for a single formulation*

Description

This is the major function to be called where numerical convolution ad/or deconvolution might be used for a linear in vitro in vivo correlation level A. It performes either numerical convolution via /codeNumConv() or deconvolution via /codeNumDeconv() and correlates their results with the known.data object via linear regression `lm()`. If you just want raw results of convolution/deconvolution then call explicitely NumConv or `link{NumDeconv}`

Usage

```
RivivcA(known.data, impulse.data, second.profile.data,dose_iv=NULL,dose_po=NULL,
mode = "deconv", explicit.interp = 20, implicit.interp = 10,
optimization.maxit = 200)
```

Arguments

known.data	the data matrix to be correlated with; depending on the state of the mode variable it represents either in vitro dissolution profile (mode = "deconv") or PK profile after oral administration of the drug (mode="conv")
impulse.data	matrix of the PK profile after the drug i.v. administration
second.profile.data	matrix of the second PK profile; depending on the mode variable it represents either PK profile after oral administration of the drug (mode = "deconv") or a drug cumulative absorption profile (mode="conv"), sometimes substituted directly by the in vitro dissolution profile
dose_iv	drug dose after i.v. administration; not obligatory but if provided must be in the same units like the dose p.o.
dose_po	drug dose after p.o. administration; not obligatory but if provided must be in the same units like the dose i.v.
mode	represents the method used here; two states are allowed: mode="conv" for numerical convolution method or mode="deconv" for numerical deconvolution (default)
explicit.interp	convolution and deconvolution explicit interpolation parameter, namely number of the curve interpolation points
implicit.interp	implicit interpolation - a factor multiplying explicit.interp for better accuracy; applies to the deconvolution procedure only
optimization.maxit	maximum number of iterations used by optim() method; applies to the deconvolution procedure only

Details

The function represents either convolution or deconvolution data together with linear regression of the above functions outputs and known data supplied as a parameter. Please bear in mind that NumDeconv() procedure is iterative and therefore depending on the parameters might require substantial amount of time to converge. Please refer to the NumDeconv description.

Value

$regression	returns a whole object of the linear regression - a result from the lm() procedure
$numeric	returns results from NumConv() or NumDeconv() functions

Author(s)

Aleksander Mendyk and Sebastian Polak

See Also

NumConv, NumDeconv

Examples

```
require(Rivivc)
require(graphics)

#i.v. data
data("impulse")
#p.o. PK profile
data("resp")
#in vitro dissolution for correlation purposes
data("input")

#preparing data matrices
input_mtx<-as.matrix(input)
impulse_mtx<-as.matrix(impulse)
resp_mtx<-as.matrix(resp)

#setting accuracy
accur_explic<-20
accur_implic<-5

#run deconvolution
result<-RivivcA(input_mtx,impulse_mtx,resp_mtx,
    explicit.interp=accur_explic,implicit.interp=accur_implic)

summary(result$regression)

print("Raw results of deconvolution")
print(result$numeric$par)

predicted<-predict(result$regression)
deconvolved_data<-unname(predicted)
orig_data<-input_mtx[,2]

dev.new()
plot(orig_data,result$numeric$par[,2])
lines(orig_data,deconvolved_data, type="l", col="blue")
dev.new()
plot(input_mtx)
lines(result$numeric$par, type="l", col="blue")
```

Index