Package ‘wnl’

December 1, 2017

Version 0.3.2

Title Minimization Tool for Pharmacokinetic-Pharmacodynamic Data Analysis

Description This is a set of minimization tools (maximum likelihood estimation and least square fitting) to solve examples in the Johan Gabrielsson and Dan Weiner's book ``Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications'' 5th ed. (ISBN:9198299107). Examples include linear and nonlinear compartmental model, turn-over model, single or multiple dosing bolus/infusion/oral models, allometry, toxicokinetics, reversible metabolism, in-vitro/in-vivo extrapolation, enterohepatic circulation, metabolite modeling, Emax model, inhibitory model, tolerance model, oscillating response model, enantiomer interaction model, effect compartment model, drug-drug interaction model, receptor occupancy model, and rebound phenomena model.

Depends R (>= 3.0.0), numDeriv

Author Kyun-Seop Bae [aut]

Maintainer Kyun-Seop Bae <k@acr.kr>

Copyright 2017, Kyun-Seop Bae

License GPL-3

NeedsCompilation no

LazyLoad yes

Repository CRAN

URL https://cran.r-project.org/package=wnl

R topics documented:

wnl-package *Minimization Tool for Pharmacokinetic-Pharmacodynamic Data Analysis*

Description

This is a minimization tool to solve the examples in the book Gabrielsson J, Weiner D. 'Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications' 5th ed. 2016. (ISBN:9198299107).

Details

Description: This is a minimization tool to solve all the examples(PK1-PK53, PD1-PD52) in the book 'Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications'.

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016.

Examples

```
tData = Theoph
colnames(tData) = c("ID", "BWT", "DOSE", "TIME", "DV")

fPK = function(THETA) # Prediction function
{
  DOSE = 320000 # in microgram
  TIME = e$DATA[,"TIME"]  # use data in e$DATA

  K    = THETA[1]
  Ka   = THETA[2]
  V    = THETA[3]

  Cp  = DOSE/V*Ka/(Ka - K)*(exp(-K*TIME) - exp(-Ka*TIME))
  return(Cp)
}

IDs = unique(tData[,"ID"])
nID = length(IDs)
for (i in 1:nID) {
  Data = tData[tData$ID == IDs[i],]
  Res = nlr(fPK, Data, pNames=c("k", "ka", "V"), IE=c(0.1, 3, 500),
            SecNames=c("CL", "Thalf", "MRT"), SecForms=c(~V*k, ~log(2)/k, ~1/k))
  print(paste("## ID =", i, "##"))
  print(Res)
}
```

nlr *Nonlinear Regression in R*

Description

It performs nonlinear regression usually for pharmacokinetic and pharmacodynamic models.

Usage

```
nlr(Fx, Data, pNames, IE, LB, UB, Error="A", ObjFx=ObjDef, SecNames, SecForms,
    Method="L-BFGS-B")
```

Arguments

`Fx`	Function for structural model. It should return a vector of same length to observations.
`Data`	Data table which will be used in Fx. Fx should access this with `e$DATA`.
`pNames`	Parameter names in the order of Fx arguments
`IE`	Initial estimates of parameters
`LB`	Lower bound for `optim` function. Default value is 0.
`UB`	Upper bound for `optim` function. Default value is 1e+06.
`Error`	Error model. One of `"A"` for additive error, `"POIS"` for poisson error, `"P"` for proportional error, and `"C"` for combined error model.
`ObjFx`	Objective function to be minimized. Default is maximum likelihood estimation function.
`SecNames`	Names of secondary parameter estimates
`SecForms`	Formula to calculate the secondary parameter estimates
`Method`	`"L-BFGS-B"` is default. See `optim` for more detail.

Details

This uses scaled transformed parameters and environment `e` internally.

Value

This returns point estimate, standard error, relative standard error, covariance matrix of estimates, run test results on residuals, AIC, and AICc. Details are stored in the environment `e`.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
tData = Theoph
colnames(tData) = c("ID", "BWT", "DOSE", "TIME", "DV")

fPK = function(THETA) # Prediction function
{
  DOSE = 320000 # in microgram
  TIME = e$DATA[,"TIME"] # use data in e$DATA

  K    = THETA[1]
  Ka   = THETA[2]
  V    = THETA[3]

  P  = DOSE/V*Ka/(Ka - K) * (exp(-K*TIME) - exp(-Ka*TIME))
  return(P)
}

IDs = unique(tData[,"ID"])
nID = length(IDs)
for (i in 1:nID) {
  Data = tData[tData$ID == IDs[i],]
  Res = nlr(fPK, Data, pNames=c("k", "ka", "V"), IE=c(0.1, 3, 500),
            SecNames=c("CL", "Thalf", "MRT"), SecForms=c(~V*k, ~log(2)/k, ~1/k))
  print(paste("## ID =", i, "##"))
  print(Res)
}
```

`Secondary` *Get Secondary Parameter Estimates*

Description

Get standard error and relative standard error (cv) of secondary paramter estimate

Usage

```
Secondary(Formula, PE, COV)
```

Arguments

`Formula`	Formula to calculate the secondary parameter estimate
`PE`	Point estimates of primary estimates with names
`COV`	Variance-covariance matrix of primary estimates

Details

Variables within `Formula` should exist in the names of `PE` vector.

Value

This returns point estimate, standard error, relative standard error of the secondary parameter estimate.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
tData = Theoph
colnames(tData) = c("ID", "BWT", "DOSE", "TIME", "DV") # Table requires DV column

fPK = function(THETA) # Prediction function
{
  AMT  = 320000 # in microgram
  TIME = e$DATA[,"TIME"]
  V  = THETA[1]
  K  = THETA[2]
  Ka = THETA[3]
  Cp = AMT/V*Ka/(Ka - K)*(exp(-K*TIME) - exp(-Ka*TIME))
  return(Cp)
}
Data = tData[tData$ID == 1,]
Res = nlr(fPK, Data, pNames=c("V", "K", "Ka"), IE=c(30000, 0.1, 2))
Secondary(~V*K, Res$Est["PE",1:e$nPara], Res$Cov)
```

wnl5 *Old type WinNonlin - Least Square not MLE*

Description

It performs old type Winnonlin regression.

Usage

```
wnl5(Fx, Data, pNames, IE, LB, UB, Error="A", ObjFx=ObjLS)
```

Arguments

Fx	Function for structural model. It should return a vector of same length to observations.
Data	Data table which will be used in Fx. Fx should access this with e$DATA.
pNames	Parameter names in the order of Fx arguments
IE	Initial estimates of parameters
LB	Lower bound for optim function. Default value is 0.
UB	Upper bound for optim function. Default value is 1e+06.
Error	Error model. One of "POIS" for poisson error, "PROP" for proportional error, and others for additive error model.
ObjFx	Objective function to be minimized. Default is least square function.

Details

This uses scaled transformed parameters and environment e internally. Here we do not provide standard error. If you want standard error, use nlr.

Value

This returns point estimates, run test results on residuals, AIC, and SBC. Details are stored in the environment e.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
tData = Theoph
colnames(tData) = c("ID", "BWT", "DOSE", "TIME", "DV")

fPK = function(THETA) # Prediction function
{
  DOSE = 320000 # in microgram
  TIME = e$DATA[,"TIME"] # use data in e$DATA

  K  = THETA[1]
  Ka = THETA[2]
  V  = THETA[3]
  Cp  = DOSE/V*Ka/(Ka - K)*(exp(-K*TIME) - exp(-Ka*TIME))
  return(Cp)
}

IDs = unique(tData[,"ID"])
nID = length(IDs)
for (i in 1:nID) {
  Data = tData[tData$ID == IDs[i],]
  Res = wnl5(fPK, Data, pNames=c("k", "ka", "V"), IE=c(0.1, 3, 500))
  print(paste("## ID =", i, "##"))
  print(Res)
}
```

Index