Package 'wnl'

June 1, 2024

Version 0.8.1

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 (>= 3.5.0), 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

Date/Publication 2024-06-01 03:10:02 UTC

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

This is a set of minimization tools to solve all the examples in the book 'Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications' 5th ed. 2016.

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)
}
```

`cmpChi` *Compare model with Chi-square test*

Description

It performs chi-square test for two models comparison.

Usage

```
cmpChi(r1, r2)
```

Arguments

`r1`	A result from `nlr`
`r2`	Another result from `nlr`

Details

One model should include the other model.

Value

Returns a p-value from `pchisq`

Author(s)

Kyun-Seop Bae <k@acr.kr>

Comp1 *One compartment model - analytical*

Description

It calculates using one compartment model.

Usage

```
Comp1(Ke, Ka=0, DH)
```

Arguments

Ke	Elimination rate constant
Ka	Absorption rate constant
DH	Expanded dosing history table

Details

First compartment is the gut compartment for oral dosing. IV bolus and infusion dosing should be done at the second compartment.

Value

This returns a table with the gut and the central compartment columns

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
DAT
DAT2 = ExpandDH(DAT)
X1 = Comp1(Ke=0.1, Ka=1, DAT2)
X1
matplot(DAT2[, "TIME"], X1, type="l")
```

DAT *An Example of Dosing History Table*

Description

This is a conventional NONMEM input data format.

Usage

```
DAT
```

Format

This data frame has 5 columns with 18 time-points for the simulation.

`TIME` Time

`AMT` Amount given for the compartment of `CMT` column

`RATE` Infusion rate

`CMT` Compartment number, 1=gut, 2=central, 3=peripheral, etc.

`DV` Currently blank and not used.

Details

To be used at `Comp1` or `nComp`, expand dosing history with `ExpandDH` function.

dx *Simplest diagnostic plot for minimization result*

Description

It performs a simple diagnostic plot from the result of `nlr`.

Usage

```
dx(r)
```

Arguments

`r`	a result from `nlr` or `wnl5`

Details

This plots 'Observation vs. Prediction' and 'Normalized Redisual vs. Prediction' only. Normalized residual are meant to be distributed as standard normal distribution, N(0, 1).

Value

This just draws a plot.

Author(s)

Kyun-Seop Bae <k@acr.kr>

`EnvObj` *Environment's Objects*

Description

Get an environment's visible objects as a list.

Usage

```
EnvObj(envir = e)
```

Arguments

`envir`	environment to get its content

Details

All the visible objects in the environment including functions and data will be returned.

Value

All visible objects as a list

Author(s)

Kyun-Seop Bae <k@acr.kr>

ExpandDH *Expand Dosing History Table*

Description

It expands dosing history table.

Usage

```
ExpandDH(DH, Fo = 1)
```

Arguments

DH	Dosing history table of NONMEM type
Fo	Bioavailability of the first (gut) compartment

Details

It expands dosing history table of conventional NONMEM data format. It calculate bioavailable amount, then add time points of non-differentiable, e.g. stopping points of infusion.

Value

Returns expanded dosing history table.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
DAT
ExpandDH(DAT) # One observation point is increased at the time of 27.
```

hSkew *Hougaard Measure of Skewness*

Description

Hougaard measure of skewness with nonlinear regression

Usage

```
hSkew(rx)
```

Arguments

rx	a result of nls function

Details

Hougaard measure of skewness can be used to check if the parameters of nonlinear regression behavior in linear fashion, i.e. symmetric confidence interval. Be cautious on the variable name conflict. All the variables in the nonlinear function should be able to be accessed by the function.

Value

Hougaard estimate of skewness for each parameter

(0, 0.1]	The estimate is very close-to-linear in behavior.
(0.1, 0.25]	The estimate is reasonably close-to-linear in behavior.
(0.25, 1]	The skweness is apparent.
>1	The estimate is considerably nonlinear in behavior.

Author(s)

Kyun-Seop Bae k@acr.kr

References

EL-Shehawy SA. On Calculating the Hougaard Measure of Skewness in a Nonlinear Regression Model with Two Parameters. J Math & Stat. 2009;5(4):360-364.

Examples

```
r1 = nls(density ~ b1*conc/(b2 + conc), DNase[DNase$Run == 1, ], start=list(b1=3, b2=5))
hSkew(r1)
```

---

nComp *Get Amounts of Each Compartments using Lambdas and Coefficients of Multi-compartment Model*

Description

It calculates using multi-compartment model.

Usage

```
nComp(Sol, Ka=0, DH)
```

Arguments

Sol	Solution list of lambdas and coefficients
Ka	Absorption rate constant
DH	Expanded dosing history table

Details

First compartment is the gut compartment for oral dosing. IV bolus and infusion dosing should be done at the second compartment. If a bolus dose was given at time T, it is reflected at times of larger than T. This is more close to real observation. ADAPT does like this, but NONMEM does not.

Value

This returns a table with the gut and the other compartment columns

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
DAT
DAT2 = ExpandDH(DAT)
Sol = SolComp2(K10=0.1, K12=3, K21=1)
X2 = nComp(Sol, Ka=1, DAT2)
X2
matplot(DAT2[, "TIME"], X2, type="l")
```

---

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", Sx, conf.level=0.95, k, fix=0)
```

Arguments

Fx	Function for structural model. It should return a vector of the 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. The default value is 0.
UB	Upper bound for optim function. The default value is 1e+06.
Error	Error model. One of "A" for additive error, "POIS" for Poisson error, "P" for proportional error, "C" for combined error model, "S" for general error model. With Error="S", Sx should be provieded.
ObjFx	Objective function to be minimized. The default is maximum likelihood estimation function(-2 log likelihood).
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.
Sx	Scale function. This is usually the inverse of weight. It should return the same length(nrow) of Y. When Error="S", Scale function should be provided as Sx.
conf.level	Confidence level for confidence interval
k	1/k likelihood interval(LI) will be provided. Currently recommended value is exp(qf(1 - alpha, 1, nRec-nPara)/2) + 1.
fix	indices of parameters to fix

Details

This uses scaled transformed parameters and environment e internally.

Value

Est	Point estimate(PE) with standard error(SE) and relative standard error(RSE)
LI	1/k likelihood interval, at which likelihood drops to 1/k of maximum likelihood. This reflects asymmetry better than confidence interval. This is estimated likelihood interval, not profile likelihood interval.
Skewness	Hougaard's skewness measure. This is printed only with additive error model. See also hSkew
Cov	Variance-covariance matrix of the objective function at the value of point estimates
run$m	Count of positive residuals
run$n	Count of negative residuals
run$run	Count of runs of residuals
run$p.value	P value of run test with excluding zero points
Objective Function Value	Minimum value of the objective function
-2LL	-2 times log likelihood
AIC	Akaike Information Criterion
AICc	Corrected Akaike Information Criterion
BIC	Schwarz Bayesian Information Criterion

Convergence	Convergence code from optim
Message	Message from optim.
Prediction	Fitted(predicted) values
Residuals	Residuals
Scale	Scales with Error="S". Variances for each points are scale vector multiplied by ScaleErrVar in Est.
Elapsed Time	Consumed time by minimization

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)
}

# Another example from radioimmunoassay(RIA)
d1 = data.frame(conc = c(200, 100, 50, 25, 12.5, 6.25, 3.125, 0),
                DV = c(1.78, 1.5, 1.17, 0.74, 0.51, 0.31, 0.19, 0.04))

PRED = function(TH) TH[1] + TH[2]*d1$conc^TH[4]/(TH[3]^TH[4] + d1$conc^TH[4])
Scale = function(TH) 1/(PRED(TH) - (TH[1] + TH[2])/2)^2

nlr(PRED, d1, pNames=c("R0", "Rmax", "RC50", "Hill"), IE=c(0.1, 3, 50, 1),
    Error="S", Sx=Scale)
```

pComp *Plot Compartment Model Diagram*

Description

It plots the diagrom of a comparment model.

Usage

```
pComp(dComp, dRate, Shape="rect", Col=NA, Bx=0.3, By=0.2, Cex=1.0, Lwd=3,
      Radius=0.3, thIn=pi/2, thOut=pi/2, ...)
```

Arguments

dComp	data.frame for a compartment model. See the example.
dRate	data.frame for rate information. See the example.
Shape	rectangle or cricle
Col	filling color
Bx	half width of compartment box
By	half height of compartment box
Cex	character expansion
Lwd	line width
Radius	radius of compartment circle
thIn	Input angle in radian
thOut	Output angle in radian
...	arguments to be passed to plot function

Details

Flow direction is from the top to bottom.

Value

It plots.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
dA = data.frame(No = c(1, 2, 3, 4), Name=c("Gut Depot", "Skin Depot", "Central", "Peripheral"),
                Level=c(1, 1, 2, 2), xPos=c(-0.5, 0.5, 0, 1))
dB = data.frame(From = c(1, 2, 3, 4, 3, 0, 0), To=c(3, 3, 4, 3, 5, 1, 2),
                Name=c("KA", "KA2", "K12", "K21", "CL", "F1", "F2"))

pComp(dA, dB)
#par(oma=c(0, 0, 0, 0), mar=c(0, 0, 1, 0)) # If need, adjust margin before calling
pComp(dA, dB, "circ", main="Compartmental Model Diagram")

pComp(dA, dB, "circ", main="Compartmental Model Diagram", Col="#DDEEFF", asp=1)
```

---

pProf *Plot Likelihood or Objective Fnnction Value Profile*

Description

It plots estimated likelihood profile. This is not profile likelihood profile.

Usage

```
pProf(Bag = e, Title = "", ...)
```

Arguments

Bag	an environment or an object containing the objects of resultant environment e after nlr()
Title	title for the plot
...	arguments to pass to the plot function

Details

This plots likelihood profile from the result of nlr() function. Bag should contain the results of nlr().

Value

No values but a plot.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Secondary *Get Secondary Parameter Estimates*

Description

Get standard error and relative standard error (cv) of the secondary parameter estimate

Usage

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

Arguments

Formula	Formula to calculate the secondary parameter estimate
PE	Point estimates of primary parameters 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)
```

SolComp2 *Get Lambdas and Coefficients of Two-compartment Model*

Description

It calculates lambdas and coefficients for two-compartment model from K10, K12, and K21.

Usage

```
SolComp2(K10, K12, K21)
```

Arguments

K10	Ke, Elimination rate constant from central compartment
K12	Rate constant from the central to the peripheral compartment
K21	Rate constant from the peripheral to the central compartment

Details

It calculates lambdas and coefficients of two-compartment model from K10, K12, and K21. Lambdas should have no identical values.

Value

This returns a list of lambdas and coefficients.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
DAT
DAT2 = ExpandDH(DAT)
Sol = SolComp2(K10=0.1, K12=3, K21=1)
X2 = nComp(Sol, Ka=1, DAT2)
X2
matplot(DAT2[, "TIME"], X2, type="l")
```

SolComp3 *Get Lambdas and Coefficients of Three-compartment Model*

Description

It calculates lambdas and coefficients for three-compartment model from K10, K12, K21, K13, and K31.

Usage

```
SolComp3(K10, K12, K21, K13, K31)
```

Arguments

K10	Ke, Elimination rate constant from central compartment
K12	Rate constant from the central to the first peripheral compartment
K21	Rate constant from the first peripheral to the central compartment
K13	Rate constant from the central to the second peripheral compartment
K31	Rate constant from the second peripheral to the central compartment

Details

It calculates lambdas and coefficients of two-compartment model from K10, K12, and K21. Lambdas should have no identical values.

Value

This returns a list of lambdas and coefficients.

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
DAT
DAT2 = ExpandDH(DAT)
Sol = SolComp3(K10=0.1, K12=3, K21=1, K13=2, K31=0.5)
X3 = nComp(Sol, Ka=1, DAT2)
X3
matplot(DAT2[, "TIME"], X3, type="l")
```

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 the 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. The default value is 0.
UB	Upper bound for optim function. The default value is 1e+06.
Error	Error model. One of "POIS" for Poisson error, "P" for proportional error, and others for additive error model.
ObjFx	Objective function to be minimized. The 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

PE	Point estimates
WRSS	Weighted Residual Sum of Square
run$m	Count of positive residuals
run$n	Count of negative residuals
run$run	Count of runs of residuals
run$p.value	P value of run test with excluding zero points
Objective Function Value	Minimum value of the objective function
AIC	Akaike Information Criterion
SBC	Schwarz Bayesian Information Criterion
Condition Number	Condition number

Message	Message from optim.
Prediction	Fitted(predicted) values
Residuals	Residuals
Elapsed Time	Consumed time by minimization

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