R Documentation of PKtools

March 26, 2007

Version 1.3-0

Date 2007-03-23

Title unified computational interfaces for population PK

Author M. Suzette Blanchard <sblanchard@coh.org>

Depends lattice, xtable, R2HTML

Maintainer M. Suzette Blanchard <sblanchard@coh.org>

Description computations for WinBUGS, NONMEM, NLME

License GPL2; incorporates by permission code of W. Bachman (wrtab 5.for, infnx5u.for), A. Gelman, (bugs.R) and J.Pinheiro (getPsi).

R topics documented:

`AICcomp` *AICcomp*

Description

AICcomp calculates and or prints the AIC, AICc (small sample AIC) and the loglikelihood from NONMEM and NLME for each of any number of models

Usage

```
AICcomp(PKNLMEobjects, NONMEMobjects)
```

Arguments

`PKNLMEobjects`
a list of PKNLME objects

`NONMEMobjects`
a list of NONMEM objects

Details

The lists of PKNLME objects and NONMEM objects must be in the same order and must be of the same length.

Value

data frame of the AIC, AICc (small sample AIC), the loglikelihood and the K, number of population parameters including both means and variance parameters.

Note

Author(s)

M.S. Blanchard <sblanchard@coh.org>

References

Burnham, K.P. and Anderson,D.R., (2002). Model Selection and Multimodel Inference: A Practical Information - Theoretic Approach (2nd edition). Springer: New York.

See Also

AIC

Examples

```
if (.Platform$OS.type =="windows"){
  library(PKtools)
  library(nlme)
curwd=getwd()
if (file.exists("C:/nmv/run"))  {
  setwd("C:/nmv/run")
  #NLME code models 3 and 6
  #data definition for NLME and NONMEM
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  conc<-round(sqrt(Theoph$conc),4)
  Theo<-data.frame(cbind(id,dose,time,conc))
  names(Theo)<-c("id","dose","time","conc")
  wt.v<-Theoph$Wt
  data<-list(pkvar=Theo, cov=wt.v)

  #model 3
  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka","Cl"),
               params=c("Ka","V int", "V slope", "Cl"),
               tparams=c("log(Ka)","log(V) int"," log(V) slope", "log(CL)"))

  model.def<-list(fixed.model=list(lKa~1,lV~wt,lCl~1), random.model=lKa+lCl~1,
  start.lst=c(.3,-.6,0,-3),form=conc~sonecpmt(dose, time, lV, lKa, lCl),
  control=nlmeControl(returnObject=FALSE))
  results.nlme3<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)

  #model 6
  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka","V","Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

  model.def<-list(fixed.model=c(lKa+lV+lCl~1),random.model=pdDiag(form=lKa+lV+lCl~1),
  start.lst=c(.5,-.6,-3), form=conc~sonecpmt(dose, time, lV, lKa, lCl),
  control=nlmeControl(returnObject=FALSE))
  results.nlme6<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)


  #NONMEM code models 3 and 6
  #note control files must be placed in the C:/nmv/run directory

  #model 3
  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "Cl"),
               params=c("Ka", "V", "Cl", "V slope"),
               tparams=c("log(Ka)", "log(V)", "log(Cl)","log(V slope)"),
               varnames=c("D[1,1]","D[1,2]","D[2,2]")
               )

  results3<-RunNM(inputStructure="control.model3", data=data, nameData=nameData)

  #model 6
  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)", "log(V)", "log(CL)"),
               varnames=c("D[1,1]","D[1,2]","D[2,2]","D[1,3]","D[2,3]","D[3,3]")
               )

  results6<-RunNM(inputStructure="control.model6", data=data, nameData=nameData)

  #Multimodel Code
  PKNLMEobjects=list(results.nlme3,results.nlme6)
  NONMEMobjects=list(results3,results6)
  print(AICcomp(PKNLMEobjects=PKNLMEobjects, NONMEMobjects=NONMEMobjects))
  setwd(curwd)
  }
  else{
    "You do not have NONMEM."
  }
}
```

`bugs` *bugs*

Description

Interface from R to WinBUGS by A. Gelman

Usage

```
bugs(data, inits, parameters.to.save, model.file="model.bug",
n.chains=3, n.iter=2000, n.burnin=floor(n.iter/2),
n.thin=max(1, floor(n.chains * (n.iter - n.burnin)/1000)),
debug=FALSE, attach.sims=TRUE, print.summary=TRUE, plot.summary=TRUE,
digits.summary=1, display.parallel=FALSE, DIC=TRUE,
bugs.directory="c:/Program Files/WinBUGS14/",
dos.location="c:/progra~1/winbug~1/winbug~1")
```

Arguments

`data` a list of the data for the Winbugs model, or a vector of the names of the data objects used by the model

`inits` a list with n.chains elements; each element of the list is itself a list of starting values for the Winbugs model, or a function creating (possibly random) initial values `parameters.to.save` vector of the names of the parameters to save

`model.file` location of the model. (Default is "model.txt".)

`n.chains` number of chains. Must be at least 2. (Default is 3.)

`n.iter` number of iterations per chain. (Default is 2000.)

`n.burnin` number of iterations to discard at the beginning. (Default is n.burnin=n.iter/2, that is, discarding the first half of the simulations.)

`n.thin` thinning rate. Must be a positive integer. Set n.thin>1 to save memory and computation time if n.iter is large. (Default is n.thin=max(1,floor(n.chains*(n.iter-n.burnin)/1000)) which will only thin if there are at least 2000 simulations.)

`debug` option to not automatically quit out of WinBugs when the script has finished running, so that you can look at what's going on within WinBugs. (Default is debug=F.)

`attach.sims` option to save all the parameters in parameters.to.save as R objects, overwriting any existing variables with these names. (Default is attach.sims=T.)

`print.summary` option to print summary statistics and convergence information. (Default is print.summary=T.)

`plot.summary` option to display summary statistics and convergence information as a graph. (Default is plot.summary=T.)

`digits.summary` rounding for tabular output on the console. (Default is to round to 1 decimal place.)

`display.parallel` option to display parallel intervals in both halves of the summary plots. This is a convergence-monitoring tool and is not necessary once you have approximate convergence. (Default is display.parallel=F.)

`DIC` option to compute deviance, pD, and DIC. (Default is DIC=T.)

`bugs.directory` bugs.directory

`dos.location` dos.location

Value

Output is a list (sims.array, sims.list, sims.matrix, summary):

n.keep: number of iterations kept per chain (equal to (n.iter-n.burnin)/n.thin)

n.sims: number of posterior simulations (equal to n.chains*n.keep)

sims.array: 3-way array of simulation output, with dimensions n.keep, n.chains, and length of combined parameter vector

sims.list: list of simulated parameters: for each scalar parameter, a vector of length n.sims for each vector parameter, a 2-way array of simulations, for each matrix parameter, a 3-way array of simulations, etc.

sims.matrix: matrix of simulation output, with n.chains*n.keep rows and one column for each element of each saved parameter. (For convenience, the n.keep*n.chains simulations in sims.array and sims.list have been randomly permuted.)

summary: summary statistics and convergence information for each element of each saved parameter. Approximate convergence has been reached when R-hat < 1.2 for all parameters.

mean: a list of the estimated parameter means

sd: a list of the estimated parameter sd's

median: a list of the estimated parameter medians (The information in "mean", "sd", and "median" is already included in "summary"; it is included in list form for convenience in later analyses.)

pD: var(deviance)/2, an estimate of the effective number of parameters (The variance is computed as the average of the within-chain variances, which gives a more reasonable estimate when convergence has not been reached.)

DIC: mean(deviance) + pD

last.values: list of simulations from the most recent iteration. They can be used as starting points if you wish to run Bugs for further iterations

In addition, the simulated parameter values are automatically saved as R objects (in the same form as the elements of sims.list). And the summary elements are also saved as R objects. (For example, if "beta" is a 10 x 3 array in the model, then it will be saved as an array of dimensions n.sims x 10 x 3.)

Author(s)

A. Gelman

References

Gelman, A. and Carlin, J.B. and Stern, H.S. and Rubin, D.B. (2003). "Bayesian Data Analysis (2nd edition)". Chapman & Hall/CRC:New York.

See Also

RunWB

Examples

```
if (.Platform$OS.type =="windows"){
library(PKtools)
library(nlme)
curwd=getwd()
if (file.exists("C:/bugsR")) {
  setwd("C:/bugsR")
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  conc<-round(sqrt(Theoph$conc),4)
  sid<-split(id,id)
  hist<-sapply(sid,length)
  n.ind<-12
  off.data<-matrix(NA,n.ind+1,1)
  off.data[1,1]<-1
  for (i in 2:(n.ind+1)) off.data[i,1]<-off.data[i-1,1]+ hist[i-1]
  off.data<-c(off.data)
  mean <- c(.5, -.6, -3)
  R<-structure(.Data=diag(rep(.1,3)))
  prec<-structure(.Data=diag(rep(.000001,3)))
  data<-list(n.ind=n.ind,off.data=off.data,dose=dose,conc=conc,
  time=time,mean=mean,R=R,prec=prec)

  inits<- function(){
    list(beta = structure(
        .Data = c(rep(.5,12),rep(-.6,12),rep(-3,12)),
        .Dim = c(12, 3)),
        mu = c(.5, -.6,  -3),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

    list(beta = structure(
        .Data = c(rep(-.5,12),rep(-.8,12),rep(-3.5,12)),
        .Dim = c(12, 3)),
        mu = c(-.5, -.8,  -3.5),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

    list(beta = structure(
        .Data = c(rep(1.5,12),rep(-.4,12),rep(-2.8,12)),
        .Dim = c(12, 3)),
        mu = c(1.5, -.4,  -2.8),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

  }

  parameters <- c("sigma2","ka","cl","v","beta","mu","re","itau","ipredwb","ppredwb")

  theo.sim <- bugs(data = data, inits = inits,
        parameters.to.save = parameters, model.file = "theosw.txt",
        n.chains = 3, n.iter = 12000, debug = T,
        n.burnin = 4000 , n.thin = 8 , print.summary = F,
        plot.summary = T)
  print(names(theo.sim))
  setwd(curwd)
  }
else{
    "You do not have a C:/BugsR directory."
    }
}
```

`coVar.id` *coVar.id*

Description

coVar.id creates a data set of the covariates one line per id with id as the first column.

Usage

```
coVar.id(id, coVar, nameData)
```

Arguments

`id`	cluster id
`coVar`	data set of the covariates with length equal to the full data set
`nameData`	list of names, including, covnames

Value

coVar.id outputs a data set of the covariates one line per id with id as the first column.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

RunNM, RunNLME, RunWB

Examples

```
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt
data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"))

descStructure<-list(pcts=c(.025,.05,.95,.975),nsig=4)

cov.id <- coVar.id(data$pkvar$id, data$cov, nameData)
cov.id
```

`desc` *desc*

Description

desc calculates select descriptive statistics for the variable X.

Usage

```
desc(y, pcts, nsig)
```

Arguments

`y`	variable of interest
`pcts`	percentiles of interest, the default is c(0.025, 0.05, 0.95, 0.975)
`nsig`	number of significant figures, the default is 4

Value

desc prints descriptive statistics including mean, median, standard deviation, range, min, max, and select percentiles.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

`tex`

Examples

```
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
concblq<-round(sqrt(Theoph$conc),4)
conc<-concblq
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
desc(Theo$conc)
```

`diagplot` *diagplot*

Description

diagplot creates plots of observed versus predicted values and residuals (ordinary and standardized) versus predicted values for both the population (marginal) and individual (conditional) levels.

Usage

```
diagplot(x,...)
```

Arguments

`x`	object of class, NONMEM, PKNLME, WinBUGS
`...`	additional arguments to be passed to lower level functions

Value

Plots of observed versus predicted values and residuals (ordinary and standardized) versus predicted values for both the population (marginal) and individual (conditional) levels.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

`residplot`, `obvsprplot`, `identify`

Examples

```
if (.Platform$OS.type =="windows"){
curwd=getwd()
if (file.exists("C:/nmv/run")) {
  setwd("C:/nmv/run")
  library(nlme)
  library(PKtools)
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  conc<-round(sqrt(Theoph$conc),4)
  Theo<-data.frame(cbind(id,dose,time,conc))
  names(Theo)<-c("id","dose","time","conc")
  wt.v<-Theoph$Wt

  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)", "log(V)", "log(Cl)"),
               varnames=c("D[1,1]","D[1,2]","D[2,2]","D[1,3]","D[2,3]","D[3,3]")
               )

  data<-list(pkvar=Theo, cov=wt.v)
  NM<-RunNM(inputStructure="control.model5", data=data, nameData=nameData)
  diagplot(NM)
  setwd(curwd)
  }
  else{
    "You do not have NONMEM."
    }
}
```

---

diagtrplot *diagtrplot*

Description

diagtrplot creates a trellis plot of the observed concentrations and predicted values vs time by subject.

Usage

```
diagtrplot(x, level, xvarlab, yvarlab, pages,...)
```

Arguments

`x`	variable identifying the clustering variable
`level`	level of mixed model ("p"-population, "i"-individual)
`xvarlab`	label for x variable
`yvarlab`	label for y variable
`pages`	number of pages to print, 1 prints first page
`...`	additional arguments to be passed to lower level functions

Value

diagtrplot produces a trellis plot of observed concentrations and predicted values vs time by subject.

Author(s)

M.S. Blanchard<sblanchard@coh.org>

See Also

`trplot`, `diagplot`, `residplot`, `obvsprplot`, `tex`, `HTMLtools`

Examples

```
library(nlme)
library(PKtools)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt
data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

model.def<-list(fixed.model=lKa+lV+lCl~1,random.model=lCl~1,
start.lst=c(lKa=.3,lV=-.6,lCl=-3), form=conc~sonecpmt(dose, time,
lV, lKa, lCl), control=nlmeControl(returnObject=FALSE))
```

```
MM<-RunNLME(inputStructure=model.def,data=data, nameData=nameData)

diagtrplot(x=MM,level="p", xvarlab=nameData$xvarlab,
yvarlab=nameData$xvarlab, pages=1)
```

HTMLtools *HTMLtools*

Description

HTMLtools is a method that outputs a HTML file of the parameter estimates and diagnostic plots from an object of class NONMEM, PKNLME, or WinBUGS for a single dose population PK model with hierarchical data.

Usage

```
HTMLtools(x, nameData, nameDir, nameFile, descStructure, drive, ...)
```

Arguments

`x`	an object from one of the following classes NONMEM, PKNLME, or WinBUGS
`nameData`	list of names, including, covnames, yvarlab, xvarlab, paramester names
`nameDir`	the path and name of the directory where the HTML file will reside
`nameFile`	list of the names of the plots being output to the .html file
`descStructure`	list of arguments (pcts,nsig) for the function desc
`drive`	graphics drive; the default is X11()
`...`	additional arguments to be passed to lower level functions

Details

RunNM, RunNLME, and RunWB create the objects of the respective classes NONMEM, PKNLME and WinBUGS that can be read by this method.

nameData is a list of the labels including the names of the covariates in the order there are given in the covariate dataset, y and x variable, the parameters names as defined by RunNM, RunNLME, and RunWB.

nameFile lists the name of the output html file and the names of the plots being output to the html file. Note the html file name should not have an html extension that will be added by the program and the plots should not have a png extension again that will be added by the program. Finally, note that there is a file0 in HTMLtools for the WinBUGS class to allow for inclusion of the density plots of the model coefficients.

- nameFile<-list(file="wb", file0=hist, file1="trplt.wb",file2="diagplt.wb", file3="qqploti.wb", file4="qqnormre.wb", file5="covre.wb", file6="diagtrplti.wb", file7="diagtrpltp.wb")

Finally for the HTML file to be in color the correct path must be given in nameDir.

Value

An HTML file of the results from the selected object.

The trellis plots including those from trplot, diagtrplot output only the first page of plots to the HTML file and a png file of all pages is also created. The covariate plot allows for up to 16 covariates again printing the first page in the HTML file and any additional plots are sent to an accompanying png file.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

RunNM, RunNLME, RunWB

Examples

```
    #NLME example
      setwd(tempdir())
      library(PKtools)
      library(R2HTML)
      library(nlme)
      data(Theoph)
      Theoph<-Theoph[Theoph$Time!=0,]
      id<-as.numeric(as.character(Theoph$Subject))
      dose<-Theoph$Dose
      time<-Theoph$Time
      conc<-round(sqrt(Theoph$conc),4)
      Theo<-data.frame(cbind(id,dose,time,conc))
      names(Theo)<-c("id","dose","time","conc")
      wt.v<-Theoph$Wt
      data<-list(pkvar=Theo, cov=wt.v)

      nameData<-list(covnames=c("wt"),
                     yvarlab="Sqrt(Theop. Conc.) (mg/L)",
                     xvarlab="Time since dose (hrs)",
                     reparams=c("Cl"),
                     params=c("Ka","V", "Cl"),
                     tparams=c("log(Ka)","log(V)","log(CL)"))

      nameFile<-list(file="nlme.output", file1="trplt.nl",
                     file2="diagplt.nl", file3="qqploti.nl",
                     file4="qqnormre.nl", file5="covre.nl",
                     file6="diagtrplti.nl", file7="diagtrpltp.nl")

      descStructure<-list(pcts=c(.025,.05,.95,.975),nsig=4)

      model.def<-list(fixed.model=lKa+lV+lCl~1,random.model=lCl~1,
      start.lst=c(lKa=.3,lV=-.6,lCl=-3), form=conc~sonecpmt(dose, time,
      lV, lKa, lCl), control=nlmeControl(returnObject=FALSE))

MM<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)

HTMLtools(x=MM, nameData = nameData, nameDir=tempdir(), nameFile = nameFile,
       descStructure = descStructure, drive=X11)
```

`indEst` *indEst*

Description

indEst outputs the individual level parameter estimates from NONMEM, PKNLME and WinBUGS.

Usage

```
indEst(PKNLMEobject, NMobject, WBobject, outputType)
```

Arguments

`PKNLMEobject`	PKNLME object from RunNLME
`NMobject`	NONMEM object from RunNM
`WBobject`	WinBUGS object from RunWB
`outputType`	"tex" or "R" outputs are available

Details

The PKNLME, NM and WB objects should all be from the same model

Value

The output is a dataframe of the indiviudal parameter estimates.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

References

Boeckmann, A.J. and Sheiner, L.B. and Beal, S.L. (1994). "NONMEM Users Guide- Part V, Introductory Guide". NONMEM Project Group:UCSF.

Pinheiro, J.C. and Bates, D.M. (2000). "Mixed-Effects Models in S and SPLUS." Springer: New York.

Spiegelhalter, D. and Thomas, A. and Best, N. and Lunn, D. (2001). "Winbugs Version 1.4 User Manual.", Imperial College School of Medicine:London.

See Also

RunNM, RunNLME, RunWB

Examples

```
if (.Platform$OS.type =="windows"){
library(PKtools)
library(nlme)
out<-0
curwd=getwd()
if (file.exists("C:/bugsR")) {
  setwd("C:/bugsR")
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  conc<-round(sqrt(Theoph$conc),4)
  sid<-split(id,id)
  hist<-sapply(sid,length)
  n.ind<-12
  off.data<-matrix(NA,n.ind+1,1)
  off.data[1,1]<-1
  for (i in 2:(n.ind+1)) off.data[i,1]<-off.data[i-1,1]+ hist[i-1]
  off.data<-c(off.data)
  mean <- c(.5, -.6, -3)
  R<-structure(.Data=diag(rep(.01,3)))
  prec<-structure(.Data=diag(rep(.000001,3)))
  data<-list(n.ind=n.ind,off.data=off.data,dose=dose,conc=conc,
  time=time,mean=mean,R=R,prec=prec)

  inits<- function(){
    list(beta = structure(
        .Data = c(rep(.5,12),rep(-.6,12),rep(-3,12)),
        .Dim = c(12, 3)),
        mu = c(.5, -.6,  -3),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

    list(beta = structure(
        .Data = c(rep(-.5,12),rep(-.8,12),rep(-3.5,12)),
        .Dim = c(12, 3)),
        mu = c(-.5, -.8,  -3.5),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

    list(beta = structure(
        .Data = c(rep(1.5,12),rep(-.4,12),rep(-2.8,12)),
        .Dim = c(12, 3)),
        mu = c(1.5, -.4,  -2.8),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)
  }

  #covariates
  wt.v<-Theoph$Wt

  parameters <-
  c("sigma2","ka","cl","v","beta","mu","itau","ipredwb","ppredwb")

  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"),
               varnames=c("D[1,1]","D[1,2]","D[1,3]",
               "D[2,1]","D[2,2]","D[2,3]",
               "D[3,1]","D[3,2]","D[3,3]")
               )

  data<-list(data=data, cov=wt.v, id=id)

  WBargs<-list(parameters=parameters, inits=inits, n.chains=3,
             n.iter=12000, n.burnin=4000, n.thin=3, debug=T)

  WB2<-RunWB(inputStructure="theosw.txt", data=data, nameData=nameData, WBargs=WBargs)
  setwd(curwd)
  }
else {
   print("You do not have C:/BugsR directory.")
   out<-1
   }
#NLME code model 5
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt
data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka","V","Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

#mat<-matrix(c(.5, 0, 0, 0,.03, 0, 0,0,.08),nrow=3)
model.def<-list(fixed.model=c(lKa+lV+lCl~1),random.model=lKa+lV+lCl~1,
start.lst=c(.5,-.6,-3), form=conc~sonecpmt(dose, time, lV, lKa, lCl),
control=nlmeControl(returnObject=TRUE, opt=c("nlm")))
results.nlme5<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)
```

```
#NONMEM code model 5
curwd=getwd()
if (file.exists("C:/nmv/run")) {
  setwd("C:/nmv/run")
  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)", "log(V)", "log(Cl)"),
               varnames=c("D[1,1]","D[1,2]","D[1,3]","D[2,2]","D[2,3]","D[3,3]")
               )

  results5<-RunNM(inputStructure="control.model5", data=data, nameData=nameData)
  setwd(curwd)
  }
  else {
  print("You do not have NONMEM.")
  out<-1
    }
if (out==0) print(try(indEst(PKNLMEobject=results.nlme5, NMobject=results5, WBobject=WB2,
}
```

obvsprplot *obvsprplot*

Description

obvsprplot creates individual observed vs predicted plots at the population (marginal) and individual (conditional) levels of the mixed model the can be used with the method identify to identify outliers.

Usage

```
obvsprplot(x,...)
```

Arguments

`x`	object of class, NONMEM, PKNLME, WinBUGS
`...`	additional arguments to be passed to lower level functions

Details

The method identify can be used with objects of class NONMEM, PKNLME, and WinBUGS by including the following code.

- NONMEM:
  - population level: identify(NM$pred$PRED, NM$pred$CONC)
  - individual level: identify(NM$pred$IPRE, NM$pred$CONC)
- PKNLME:
  - population level: identify(MM$mm$fitted[,1], MM$pkdata$conc)
  - individual level: identify(MM$mm$fitted[,2], MM$pkdata$conc)
- WinBUGS:
  - population level: identify(WB$pred$ppred, WB$pred$conc)
  - individual level: identify(WB$pred$ipred, WB$pred$conc)

Value

plots of observed versus predicted values for both the population (marginal) and individual (conditional) levels.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

identify, obvsprplot, diagplot

Examples

```
if (.Platform$OS.type =="windows"){
library(PKtools)
library(nlme)

curwd=getwd()
if (file.exists("C:/nmv/run")) {
  setwd("C:/nmv/run")
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  conc<-round(sqrt(Theoph$conc),4)
  Theo<-data.frame(cbind(id,dose,time,conc))
  names(Theo)<-c("id","dose","time","conc")
  wt.v<-Theoph$Wt

  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)", "log(V)", "log(Cl)"),
               varnames=c("D[1,1]","D[1,2]","D[2,2]","D[1,3]","D[2,3]","D[3,3]")
               )

  data<-list(pkvar=Theo, cov=wt.v)
  NM<-RunNM(inputStructure="control.model5", data=data, nameData=nameData)
  obvsprplot(NM, "p")
  setwd(curwd)
  }
  else{
      "You do not have NONMEM."
      }
}
```

paramEst *paramEst*

Description

paramEst outputs the parameter estimates from NONMEM, PKNLME and WinBUGS.

Usage

```
paramEst(PKNLMEobject, NMobject, WBobject)
```

Arguments

`PKNLMEobject`	PKNLME object from RunNLME
`NMobject`	NONMEM object from RunNM
`WBobject`	WinBUGS object from RunWB

Details

The PKNLME, NM and WB objects should all be from the same model

Value

The output is a dataframe of the population parameter estimates

Author(s)

M.S. Blanchard <sblanchard@coh.org>

References

Boeckmann, A.J. and Sheiner, L.B. and Beal, S.L. (1994). "NONMEM Users Guide- Part V, Introductory Guide". NONMEM Project Group:UCSF.

Pinheiro, J.C. and Bates, D.M. (2000). "Mixed-Effects Models in S and SPLUS." Springer: New York.

Spiegelhalter, D. and Thomas, A. and Best, N. and Lunn, D. (2001). "Winbugs Version 1.4 User Manual.", Imperial College School of Medicine:London.

See Also

RunNM, RunNLME, RunWB

Examples

```
if (.Platform$OS.type =="windows"){
library(PKtools)
library(nlme)
out<-0
curwd=getwd()
if (file.exists("C:/bugsR")) {
setwd("C:/bugsR")
data(Theoph)
```

```
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
sid<-split(id,id)
hist<-sapply(sid,length)
n.ind<-12
off.data<-matrix(NA,n.ind+1,1)
off.data[1,1]<-1
for (i in 2:(n.ind+1)) off.data[i,1]<-off.data[i-1,1]+ hist[i-1]
off.data<-c(off.data)
mean <- c(.5, -.6, -3)
R<-structure(.Data=diag(rep(.01,3)))
prec<-structure(.Data=diag(rep(.000001,3)))
data<-list(n.ind=n.ind,off.data=off.data,dose=dose,conc=conc,
time=time,mean=mean,R=R,prec=prec)

inits<- function(){
    list(beta = structure(
        .Data = c(rep(.5,12),rep(-.6,12),rep(-3,12)),
        .Dim = c(12, 3)),
        mu = c(.5, -.6,  -3),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

list(beta = structure(
        .Data = c(rep(-.5,12),rep(-.8,12),rep(-3.5,12)),
        .Dim = c(12, 3)),
        mu = c(-.5, -.8,  -3.5),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

list(beta = structure(
        .Data = c(rep(1.5,12),rep(-.4,12),rep(-2.8,12)),
        .Dim = c(12, 3)),
        mu = c(1.5, -.4,  -2.8),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

}

#covariates
wt.v<-Theoph$Wt

parameters <-
c("sigma2","ka","cl","v","beta","mu","itau","ipredwb","ppredwb")

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
                  params=c("Ka", "V", "Cl"),
                  tparams=c("log(Ka)","log(V)","log(CL)"),
                  varnames=c("D[1,1]","D[1,2]","D[1,3]",
                  "D[2,1]","D[2,2]","D[2,3]",
                  "D[3,1]","D[3,2]","D[3,3]")
                  )

data<-list(data=data, cov=wt.v, id=id)

WBargs<-list(parameters=parameters, inits=inits, n.chains=3,
             n.iter=12000, n.burnin=4000, n.thin=3, debug=T)

WB2<-RunWB(inputStructure="theosw.txt", data=data, nameData=nameData, WBargs=WBargs)

  setwd(curwd)
  }
else {
  print("You do not have C:/BugsR directory.")
  out<-1
  }

#NLME code model 5
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt
data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka","V","Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

#mat<-matrix(c(.5, 0, 0, 0,.03, 0, 0,0,.08),nrow=3)
model.def<-list(fixed.model=c(lKa+lV+lCl~1),random.model=lKa+lV+lCl~1,
start.lst=c(.5,-.6,-3), form=conc~sonecpmt(dose, time, lV, lKa, lCl),
control=nlmeControl(returnObject=FALSE, opt=c("nlm")))
results.nlme5<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)


#NONMEM code model 5
curwd=getwd()
if (file.exists("C:/nmv/run")) {
setwd("C:/nmv/run")

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)", "log(V)", "log(Cl)"),
               varnames=c("D[1,1]","D[1,2]","D[2,2]","D[1,3]","D[2,3]","D[3,3]")
               )

results5<-RunNM(inputStructure="control.model5", data=data, nameData=nameData)
setwd(curwd)
  }
  else {
    print("You do not have NONMEM.")
    out<-1
    }
if (out==0) print(try(paramEst(PKNLMEobject=results.nlme5, NMobject=results5, WBobject=WB
}
```

pk *pk*

Description

pk creates two data sets, a rectangular data set for R, and a NONMEM ready data set.

Usage

```
pk(pkvar, covdata, covnames)
```

Arguments

`pkvar`	PK data set including; id, dose, conc, and time
`covdata`	matrix/vector of covariate data
`covnames`	vector of names of covarites in the cov matrix/vector

Value

pk creates a pk data file pkdat including: id, dose, time, conc, plus the covaraites, and also creates NMdata, a NONMEM ready data file.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

`RunNM`

Examples

```
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
```

```
time<-Theoph$Time
concblq<-round(sqrt(Theoph$conc),4)
conc<-concblq
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt

data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"))

pk(pkvar=data$pkvar, covdata=data$cov, covnames=nameData$covnames)
```

`PKtools.AIC` *PKtools.AIC*

Description

PKtools.AIC calculates the AIC and AICc.

Usage

```
PKtools.AIC(loglike,n,K,...)
```

Arguments

`loglike`	loglikelihood
`n`	total number of samples
`K`	number of fixed parameters including both mean and variance parameters
`...`	additional arguments to be passed to lower level functions

Value

This function outputs the AIC and and the small sample AIC, AICc, as well as the objective function (-2 × loglikelihood) and K.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

References

Burnham, K.P. and Anderson,D.R., (2002). Model Selection and Multimodel Inference: A Practical Information - Theoretic Approach (2nd edition). Springer: New York.

See Also

`tex`, `HTMLtools`

Examples

```
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt
data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("V","Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

model.def<-list(fixed.model=lKa+lV+lCl~1,random.model=lV+lCl~1,
start.lst=c(lKa=.3,lV=-.6,lCl=-3), form=conc~sonecpmt(dose, time,
lV, lKa, lCl), control=nlmeControl(returnObject=FALSE))

MM<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)

K = attr(logLik(MM$mm), "df")
n<-nrow(MM$pkdata)
AIC.table<-data.frame(PKtools.AIC(loglike=logLik(MM$mm),n=n,K=K), row.names="")
AIC.table
```

PKtoolsInternal.Rd *PKtoolsInternal*

Description

Files internal to PKtools

residplot *residplot*

Description

resid creates individual residual vs predicted plots at the population (marginal) and individual (conditinal) levels of the mixed model the can be used with the method identify to identify outliers.

Usage

```
residplot(x,...)
```

Arguments

`x`	object of class, NONMEM, PKNLME, or WinBUGS
`...`	additional arguments to be passed to lower level functions

Details

The method identify can be used with objects of class NONMEM, PKNLME, and WinBUGS by including the following code.

- NONMEM:
  - population level: identify(NM$pred$PRED, NM$pred$WRES)
  - individual level: identify(NM$pred$IPRE, NM$pred$IWRE)
- PKNLME:
  - population level: identify(MM$mm$fitted[,1], MM$mm$RES)
  - individual level: identify(MM$mm$fitted[,2], MM$mm$IRES)
- WinBUGS:
  - population level: identify(WB$pred$ppred, WB$pred$presid)
  - individual level: identify(WB$pred$ipred, WB$pred$iresid)

Value

plots of residual versus predicted values for both the population (marginal) and individual (conditional) levels.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

`identify`, `obvsprplot`, `diagplot`

Examples

```
if (.Platform$OS.type =="windows"){
library(PKtools)
library(nlme)
curwd=getwd()
if (file.exists("C:/nmv/run")){
  setwd("C:/nmv/run")
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  conc<-round(sqrt(Theoph$conc),4)
  Theo<-data.frame(cbind(id,dose,time,conc))
  names(Theo)<-c("id","dose","time","conc")
  wt.v<-Theoph$Wt

  nameData<-list(covnames=c("wt"),
                 yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
                 xvarlab="Time since dose (hrs)",
                reparams=c("Ka", "V", "Cl"),
                params=c("Ka", "V", "Cl"),
                tparams=c("log(Ka)", "log(V)", "log(Cl)"),
                varnames=c("D[1,1]","D[1,2]","D[2,2]","D[1,3]","D[2,3]","D[3,3]"))

  data<-list(pkvar=Theo, cov=wt.v)
  NM<-RunNM(inputStructure="control.model5", data=data, nameData=nameData)
  residplot(NM, "p")
  setwd(curwd)
  }
  else{
      "You do not have NONMEM."
      }
}
```

RunNLME *RunNLME*

Description

RunNLME uses the NLME software to estimate parameters for a single dose population PK model with hierarchical data.

Usage

```
RunNLME(inputStructure,data, nameData)
```

Arguments

`inputStructure`
NLME-model.def

`data` list of data files including pk data and covariate data the length of the full dataset

`nameData` list of names, including, covnames, yvarlab, xvarlab, paramester names

Details

model.def is a list of the definitions of the model form, fixed and random effects, the starting values and control argument from the nlme function. The following is an example.

- model.def<-list(fixed.model=lKa+lVol+lCl ~1, random.model=lVol+lCl ~1, start.lst=c(lKa=.3,lVol=-.6,lCl=-3), form=conc ~ sonecpmt(dose, time, lV, lKa, lCl), control=nlmeControl(returnObject=FALSE)).

nameData is a list of the labels including the names of the covariates in the order they are given in the covariate dataset, y and x variable, the random parameters (reparams -should match the list for random.model in the model.def), fixed parameters (params -should match the list for fixed.model in the model.def), label for transformed parameters (in the Theo example the model parameters are on a log scale tparam=c("log(Ka)","log(V)","log(Cl)") and the names of the variance parameters are not required for NLMEoutput.

Value

Output datasets include the input data, the parameter estimates, covariates, model residuals at the population and individual levels,and model predicted values for the population and individual levels.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

References

Pinheiro, J.C. and Bates, D.M. (2000). "Mixed-Effects Models in S and SPLUS." Springer: New York.

See Also

`pk`, `coVar.id`, `RunNM`, `RunWB`

Examples

```
#NLME example
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt
data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

model.def<-list(fixed.model=lKa+lV+lCl~1,random.model=lCl~1,
start.lst=c(lKa=.3,lV=-.6,lCl=-3), form=conc~sonecpmt(dose, time,
lV, lKa, lCl), control=nlmeControl(returnObject=FALSE))

MM<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)
MM
```

`RunNM` *RunNM*

Description

RunNM runs the function pk to create pharmacokinetics datasets for R and NONMEM, runs the system commend to run NONMEM, and reads the NONMEM datasets.

Usage

```
RunNM(inputStructure, data, nameData)
```

Arguments

`inputStructure`
the standard NONMEM control file

`data` list of data files including pk data and covariate data with length of the full dataset

`nameData` list of names, including, covnames, yvarlab, xvarlab, params

Details

nameData is a list of the labels including the names of the covariates in the order there are given in the covariate dataset, y and x variable, the random parameters (reparams -should match the list random effects defined in the control file), fixed parameters (params -should match the list for fixed effects in the control file), label for transformed parameters (in the Theo example the model parameters are on a log scale tparam=c("log(Ka)","log(V)","log(Cl)") and the names of the variance parameters should list the parameters for the upper triangle of variance covariance table.

Value

The output from NMoutput are data tables of the results, including the objective function (ob), population parameters (params), random effects (re), individual parameters (ip), covariates (cov), predicted values (pred). If the objects of class NONMEM is called NM, then the objective function can be accessed by typing NM$ob, similarly the population parameters can be accessed by typing NM$param.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

References

Boeckmann, A.J. and Sheiner, L.B. and Beal, S.L. (1994). "NONMEM Users Guide- Part V, Introductory Guide". NONMEM Project Group:UCSF.

See Also

`pk`, `coVar.id`, `RunNLME`,`RunWB`

Examples

```
#NONMEM example
if (.Platform$OS.type =="windows"){
curwd=getwd()
if (file.exists("C:/nmv/run")) {
  setwd("C:/nmv/run")
  library(PKtools)
  data(Theoph)
  Theoph<-Theoph[Theoph$Time!=0,]
  id<-as.numeric(as.character(Theoph$Subject))
  dose<-Theoph$Dose
  time<-Theoph$Time
  concblq<-round(sqrt(Theoph$conc),4)
  conc<-concblq
  Theo<-data.frame(cbind(id,dose,time,conc))
  names(Theo)<-c("id","dose","time","conc")
  wt.v<-Theoph$Wt

  data<-list(pkvar=Theo, cov=wt.v)

  nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)", "log(V)", "log(Cl)"),
               varnames=c("D[1,1]","D[1,2]","D[2,2]","D[1,3]","D[2,3]","D[3,3]")
               )

  NM<-RunNM(inputStructure="control.model5", data=data, nameData=nameData)
  print(NM)
  setwd(curwd)
  }
else{
    "You do not have NONMEM."
  }
  }
```

RunWB *RunWB*

Description

RunWB uses WinBUGS to estimate parameters for a single dose population PK model with hierarchical data.

Usage

```
RunWB(inputStructure, data, nameData, WBargs)
```

Arguments

`inputStructure`
the inputStructure for WBoutput is .txt file that defines the model using standard WinBUGS code

`data` list of data files following Gelman (2003)

`nameData` list of names, including, covnames, yvarlab, xvarlab, coef, params, tparams,reparams, varparams

`WBargs` Additional arguements required to run WinBUGS, including parameters- the list of parameters to be sampled, initial values (inits), n.chain- number of chains, n.iter-number of iterations, n.burnin- length of the burnin, n.thin, and debug-T/F if T the run will stop at the end of the WinBUGS run to allow use of WinBUGS to study mixing and convergence.

Details

nameData is a list of the labels including the names of the covariates in the order they are given in the covariate dataset, y and x variable, coef are the model coefficient names, params are PK parameter names including fixed PK parameters, reparams are the parameters in the params list that are "not" fixed. tparams are the labels for transformed parameters (in the Theo example the model parameters are on a log scale tparam=c("log(Ka)","log(V)","log(Cl)") and finally varparams are the names of the parameters in the full covariance matrix.

Value

The output from this function is an WinBUGS object that includes the mean and sims.list values as described by Gelman, and the input data set, nameData, model predictions, and a covariate data set by id.

Author(s)

M.S. Blanchard<sblanchard@coh.org>

References

Lunn, D.J. and Best, N. and Thomas, A. and Wakefield, J. and Spiegelhalter, D. (2002). "Bayesian analysis of population PK/PD Models: General concepts and software. Journal of Pharmacokinetics and Pharmacodynamics", 29 (3), 271-307.

Lunn, D.J. and Wakefield, J. and Thomas, A. and Best, N. and Spiegelhalter,D. (1999). PKBugs User Guide (version 1.1). Imperial College: London.

Spiegelhalter, D. and Thomas, A. and Best, N. and Lunn D. (2001). "Winbugs Version 1.4 User Manual.", Imperial College School of Medicine:London.

See Also

`bugs`, `RunNLME`, `RunNM`

Examples

```
if (.Platform$OS.type =="windows"){

curwd=getwd()
if (file.exists("C:/bugsR")) {
setwd("C:/bugsR")
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
sid<-split(id,id)
hist<-sapply(sid,length)
n.ind<-12
off.data<-matrix(NA,n.ind+1,1)
off.data[1,1]<-1
for (i in 2:(n.ind+1)) off.data[i,1]<-off.data[i-1,1]+ hist[i-1]
```

```
off.data<-c(off.data)
mean <- c(.5, -.6, -3)
R<-structure(.Data=diag(rep(.1,3)))
prec<-structure(.Data=diag(rep(.000001,3)))
data<-list(n.ind=n.ind,off.data=off.data,dose=dose,conc=conc,
time=time,mean=mean,R=R,prec=prec)

inits<- function(){
    list(beta = structure(
        .Data = c(rep(.5,12),rep(-.6,12),rep(-3,12)),
        .Dim = c(12, 3)),
        mu = c(.5, -.6,  -3),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

list(beta = structure(
        .Data = c(rep(-.5,12),rep(-.8,12),rep(-3.5,12)),
        .Dim = c(12, 3)),
        mu = c(-.5, -.8,  -3.5),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

list(beta = structure(
        .Data = c(rep(1.5,12),rep(-.4,12),rep(-2.8,12)),
        .Dim = c(12, 3)),
        mu = c(1.5, -.4,  -2.8),
        tau = structure(.Data = c(0.1, 0, 0,
                                  0, 0.1, 0,
                                  0, 0, 0.1), .Dim = c(3, 3)),
        tauC = 20)

}

#covariates
wt.v<-Theoph$Wt

parameters <- c("sigma2","ka","cl","v","beta","mu","re","itau","ipredwb","ppredwb")

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) Sqrt(mg/L)",
               xvarlab="Time since dose (hrs)",
               coef=c("Ka", "V", "Cl"),
               params=c("Ka", "V", "Cl"),
               reparams=c("Ka", "V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"),
               varnames=c("D[1,1]","D[1,2]","D[1,3]",
               "D[2,1]","D[2,2]","D[2,3]",
               "D[3,1]","D[3,2]","D[3,3]")
               )

data<-list(data=data, cov=wt.v, id=id)

WBargs<-list(parameters=parameters, inits=inits, n.chains=3,
                n.iter=12000, n.burnin=4000, n.thin=3, debug=T)

WB<-RunWB(inputStructure="theosw.txt", data=data, nameData=nameData, WBargs=WBargs)
print(WB)
setwd(curwd)
}
else{
    "You do not have the C:/BugsR directory."
    }
}
```

`tex` *tex*

Description

tex is a method that outputs a tex file of the parameter estimates and diagnostic plots from an object of class NONMEM, PKNLME, or WinBUGS for a single dose population PK model with hierarchical data.

Usage

```
tex(x, nameData, nameDir, nameFile, descStructure,...)
```

Arguments

`x` an object from one of the following classes NONMEM, PKNLME, or WinBUGS

`nameData` list of names, including, covnames, yvarlab, xvarlab, and parameter names

`nameDir` the path and name of the directory where the HTML file will reside

`nameFile` lists the name of the tex file and of the plots being output to the .tex file

`descStructure` list of variables (pcts,nsig) for the function desc

`...` additional arguments to be passed to lower level functions

Details

RunNM, RunNLME, and RunWB create the NONMEM, PKNLME and WinBUGS objects NM, MM, and WB,respectively, that can be read by this method.

nameData is a list of the labels including the names of the covariates in the order they are given in the covariate dataset, y and x variable, and parameter names as listed for the funtions RunNLME, RunNM, or RunWB.

nameFile lists the name of the tex file and the names of the plots being output to .tex file. note the tex file should have a tex extension and the plots should have a ps extension. Finally, note that there is a file0 in tex.WinBUGS which includes the density plots of the model coefficients.

nameFile<-list(file="wb.tex", file0=hist.ps, file1="trplt.wb.ps", file2="diagplt.wb.ps", file3="qqploti.wb.ps", file4="qqnormre.wb.ps", file5="covre.wb.ps", file6="diagtrplti.wb.ps", file7="diagtrpltp.wb.ps")

Value

A tex file of the results from the selected object.

The trellis plots including those from trplt, diagtrplt output the first page of plots to the tex file and all pages to an accompanying postscript file. The covariate plot allows for up to 16 covariates also printing the first page in the tex file and any additional plots to an accompanying postscript file.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

`HTMLtools`

Examples

```
#NLME example
setwd(tempdir())
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt

data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Cl"),
               params=c("Ka","V", "Cl"),
               tparams=c("log(Ka)","log(V)","log(CL)"))

nameFile<-list(file="nlme.tex", file1="trplt.nl.ps",
               file2="diagplt.nl.ps", file3="qqploti.nl.ps",
               file4="qqnormre.nl.ps", file5="covre.nl.ps",
               file6="diagtrplti.nl.ps", file7="diagtrpltp.nl.ps")

descStructure<-list(pcts=c(.025,.05,.95,.975),nsig=4)

model.def<-list(fixed.model=lKa+lV+lCl~1,random.model=lCl~1,
start.lst=c(lKa=.3,lV=-.6,lCl=-3), form=conc~sonecpmt(dose, time,
lV, lKa, lCl), control=nlmeControl(returnObject=FALSE))

MM<-RunNLME(inputStructure=model.def, data=data,
nameData=nameData)

tex(MM, nameData = nameData, nameDir=tempdir(),
    nameFile = nameFile, descStructure = descStructure)
```

trplot *trplot*

Description

trplot creates a trellis plot of concentration vs time by subject.

Usage

```
trplot(x, xvarlab, yvarlab, pages,...)
```

Arguments

`x`	object of class, NONMEM, PKNLME, WinBUGS
`xvarlab`	label for x variable
`yvarlab`	label for y variable
`pages`	number of pages you want to print, pages=1 prints the first page
`...`	additional arguments to be passed to lower level functions

Value

A trellis plot of concentration vs time by subject.

Author(s)

M.S. Blanchard <sblanchard@coh.org>

See Also

`diagtrplot`

Examples

```
#NLME example
library(PKtools)
library(nlme)
data(Theoph)
Theoph<-Theoph[Theoph$Time!=0,]
id<-as.numeric(as.character(Theoph$Subject))
dose<-Theoph$Dose
time<-Theoph$Time
conc<-round(sqrt(Theoph$conc),4)
Theo<-data.frame(cbind(id,dose,time,conc))
names(Theo)<-c("id","dose","time","conc")
wt.v<-Theoph$Wt

data<-list(pkvar=Theo, cov=wt.v)

nameData<-list(covnames=c("wt"),
               yvarlab="Sqrt(Theop. Conc.) (mg/L)",
               xvarlab="Time since dose (hrs)",
               reparams=c("Cl"),
                  params=c("Ka","V", "Cl"),
                  tparams=c("log(Ka)","log(V)","log(CL)"))

model.def<-list(fixed.model=lKa+lV+lCl~1,random.model=lCl~1,
start.lst=c(lKa=.3,lV=-.6,lCl=-3), form=conc~sonecpmt(dose, time,
lV, lKa, lCl), control=nlmeControl(returnObject=FALSE))

MM<-RunNLME(inputStructure=model.def, data=data, nameData=nameData)
trplot(x=MM,yvarlab=nameData$yvarlab,xvarlab=nameData$xvarlab,pages=1)
```

Index