R documentation

of all in '`./Pmetrics/man`'

July 27, 2015

R topics documented:

`Pmetrics-package` *Parametric and non-parametric modeling and simulation of pharmacokinetic-pharmacodynamic systems.*

Description

This package contains functions to run and analyze the output from all three components of the Pmetrics software suite for population pharmacometric data analysis: 1) IT2B (Iterative Two-Stage Bayesian) for parametric models; 2) NPAG (Non-parametric Adaptive Grid) for non-parametric models; 3) Simulator for semi-parametric Monte-Carlo simulations.

Author(s)

Michael Neely, MD http://www.lapk.org

ERRreport *Summarize ERR Run*

Description

Generates a summary of an ERR run

Usage

```
ERRreport(wd, icen, type)
```

Arguments

wd	The working directory containing the ASS0001 file
icen	Not used, but included for compatibility with other report functions
type	Not used, but included for compatibility with other report functions

Details

Creates an HTML page summarizing an ERR run. This report is generated automatically at the end of a successful run.

Value

Two files are placed in the wd

ASS0001	A text file of the results
errlog	A text file with a log of the session

,

Author(s)

Michael Neely

ERRrun *Execute an Assay Error Estimation run.*

Description

Runs Assay Error Module

Usage

```
ERRrun(model = "model.txt", data = "data.csv", run, include, exclude,
  ode = -4, tol = 0.001, salt, cycles = 100, search = "cursory",
  xdev = 5, auto = T, intern = F, silent = F, overwrite = F,
  nocheck = F)
```

Arguments

model	Name of a suitable model file template in the working directory or an existing (previous) run number corresponding to a folder in the current working directory that used the same model file as will be used in the current run. If this is supplied, then the model file will be copied into the current working directory for convenience. If not supplied, the default is "model.txt". This file will be converted to a fortran model file. If it is detected to already be a fortran file, then the analysis will proceed without any further file conversion.
data	Name of a suitable data file (see PMwriteMatrix) or an existing (previous) run number corresponding to a folder in the current working directory that used the same data file as will be used in the current run. If this is supplied, then previously made '.ZMQ' files will be copied into the current working directory, bypassing the need to re-convert the .csv file and speeding up the run..
run	Specify the run number of the output folder. Default if missing is the next available number.
include	Vector of subject id values in the data file to include in the analysis. The default (missing) is all.
exclude	A vector of subject IDs to exclude in the plot, e.g. c(4,6:14,16:20)
ode	Ordinary Differential Equation solver log tolerance or stiffness. Default is -4, i.e. 0.0001. Higher values will result in faster runs, but parameter estimates may not be as accurate.
tol	Tolerance for convergence, with default of 0.001.
salt	Vector of salt fractions for each ndrug, default is 1 for each drug. This is not the same as bioavailability.
cycles	Number of cycles to run. Default is 100.
search	Default is "cursory", but can be "medium" or "extensive", which take progressively longer times to converge, but are more accurate.
xdev	Multiple of standard deviations for parameters to be sent to NPAG as a range. Default is 5.
auto	If auto is False you can answer all questions about the run environment manually. This might be helpful for beginners. Default is True.

intern	MacOSX only: Run ERR in the R console without a batch script. Default is false. This will be ignored on Windows systems. On the latter, the behavior of cmd.exe (aka the "DOS" window) with R is poor - it does not update until the end of execution, so you cannot see any output that indicates that ERR is running. If intern=T the HTML summary page will not be automatically loaded at the end of the run, but all post-run processing will occur normally, and you can find the HTML summary page in the /outputs folder: ERRreport.html.
silent	Boolean operator controlling whether a model summary report is given. Default is True.
overwrite	Overwrite existing run result folders. Default is FALSE.
nocheck	Suppress the automatic checking of the data file with PMcheck. Default is FALSE.

Details

ERRrun will execute an Assay Error run to estimate error polynomial coefficients.

If all function arguments are default, the simplest execution of this command is ERRrun(). This will result in generation of a batch file. On Unix (Mac) systems will be launched automatically in a terminal window. On Windows systems, the user must execute the batch file from the current working directory, which will launch the estimation program in a command prompt (DOS-like) window. In either case, it will run independently of R so that R can be used for other purposes if desired.

Value

A successful run will result in creation of a new folder in the working directory. This folder will be named with a date-time stamp in the format "out-YYYYMMMDD-hhmm", e.g. out-2011Apr10-1015. Under this folder will be four subfolders: etc, inputs, outputs, and wrkcopy, described below.

- **etc** Control files generally not needed by the user after a completed run.
- **inputs** This folder will contain the .csv data file and the model file.
- **outputs** This folder will contain the output from the run: a file that will be prefixed by ASS with appended numbers, usually 0001. This file contains all the output of the run, with the estimated assay error polynomical coefficients at the end.
- **wrkcopy** The working copy format which is used by the program. Invisibly to the user, the .csv input file is converted to these text files, one file per subject.

Author(s)

Michael Neely

See Also

ITrun, NPrun

getPMoptions *Get Pmetrics User Options*

Description

Get user options for Pmetrics

Usage

```
getPMoptions(opt)
```

Arguments

opt	The option to retrieve. If omitted, all option values will be returned.

Details

This function will get user options for Pmetrics. Current user options are

- sep Field separator in data files
- dec Decimal separator in numbers

Value

The user options file will be updated. This will persist from session to session.

Author(s)

Michael Neely

growth *CDC Pediatric and Adolescent Growth Data Table*

Description

Centers for Disease Control Pediatric and Adolescent Growth Data Table

Usage

```
growth
```

Format

A data frame with the following 9 columns: KNOT (integer age in months); A, B1, B2, B3 (coefficients for calculating percentiles), SEX, AGE, PERCENTILE, and CHART (length x age, wt x age, wt x length, hc x age, or ht x age).

Details

Coefficients to calculate sex-specific percentiles of length, weight and head cicumference data in children from 0 to 18 years. Downloaded and combined from http://www.cdc.gov/growthcharts/data_tables.htm. Used with the `qgrowth` function to generate height and weight percentiles for the purposes of simulation.

Author(s)

Michael Neely

`ITparse` *Parse Pmetrics IT2B Output*

Description

`ITparse` processes the output from an IT2B run into a list.

Usage

```
ITparse(outfile = "IT_RF0001.TXT")
```

Arguments

`outfile`	This is the filename of the output from IT2B. Typically, the file will be called IT_RF0001.txt, and this is the default.

Details

This function can take some time to process the RFILE, depending on the number of subjects, doses, observations, etc. Typical wait times are a few seconds up to 5 minutes. When processing is complete a summary of the extracted data will be reported on the console.

Value

The output of `ITparse` is a list with the following objects and of the class *IT2B*.

`nsub`	Number of subjects
`nvar`	Number of random variables or parameters in the model
`nofix`	Number of fixed variables or parameters in the model
`par`	Names of random parameters
`parfix`	Names of fixed parameters
`covnames`	Names of covariates
`ab`	Suggested boundaries for each random parameter to be passed to NPAG
`fixedpos`	Index of variables fixed to be positive
`valfix`	Values for fixed parameters
`icycmax`	Maximum number of cycles specified by the user
`icyctot`	Number of cycles run. If less than `icycmax`, convergence occurred.
`stoptol`	Stopping tolerance for convergence, default 0.001

converge	Boolean value if convergence occurred.
ODEtol	Ordindary Differential Equation solver tolerance.
numeqt	Number of output equations
ERRmod	Vector of length equal to numeqt whose values are 0 if gamma was estimated for that output equation or 1 if gamma was fixed to 1 for that output equation
ndrug	Number of drug inputs
salt	Vector of values of the salt fraction for each ndrug
ndose	Vector of the number of doses for each subject in the population
ncov	Number of covariates in the model
nobs	Vector of the number of observations for each subject in the population
nobsmax	Maximum number of observation in any individual subject
ypredpop	Array of population model predictions for each subject at each observation time point. *ypredpop[nsub,numeqt,time,type]* where *type* is 1=mean, 2=median of the population prior used to calculate ypredpop
ypredbay	Array of Bayesian posterior model predictions for each subject at each observation time point. *ypredbay[nsub,numeqt,time,type]* where *type* is 1=mean, 2=median of the population prior used to calculate ypredbay
parbay	Array of Bayesian posterior parameter estimates for each subject, *parbay[nsub,nvar,type]* where *type* is 1=mean, 2=median of the population prior used to calculate parbay
ic	Data frame with one row and two columns for final cycle Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)
ilog	Vector of cycle number and associated log-likelihood
imean	Matrix of cycle numbers and associated means for each random parameter
imed	Matrix of cycle numbers and associated medians for each random parameter
isd	Matrix of cycle numbers and associated standard deviations for each random parameter
icv	Matrix of cycle numbers and associated coefficients of variation for each random parameter
igamlam	Matrix of cycle number and associated gamma or lambda with each output equation in a column
lpar	Matrix of subjects in rows and MAP Bayesian parameter estimates in columns for each parameter, based on population means from the next to last cycle.
lsd	Matrix of subjects in rows and SD of Bayesian posterior parameter distributions in columns for each parameter, based on population means from the next to last cycle.
lcv	Matrix of subjects in rows and CV of Bayesian posterior parameter distributions in columns for each parameter, based on population means from the next to last cycle.
sdata	Subject data consisting of 5 columns: [id, nsub, age, sex, ht], *id* is the original identification number in the .csv matrix file; *nsub* is the sequential subject number in the IT2B run; *age*, *sex* and *ht* will be missing for .csv input and present if included in .wrk input files
dosecov	Data frame with all dosing information for each subject, including times, routes, amounts, and associated covariate values

outputs	Data frame with measured outputs for each subject and associated assay error polynomials. The order of the columns is nsub, time, numeqt, observation, c0, c1, c2, c3, where the last four columns are the coefficients of the assay error polynomial for that observation, such that SD[obs] = c0 + c1*[obs] + c2*[obs]**2 + c3*[obs]**3
negflag	A flag indicating that some negative predictions were changed to missing. This means that the model may be misspecified.
mdata	The filename of the data used in the run.

Author(s)

Michael Neely

`ITrun` *Execute an IT2B run.*

Description

Runs IT2B

Usage

```
ITrun(model = "model.txt", data = "data.csv", run, include, exclude,
  ode = -4, tol = 0.001, salt, cycles = 100, xdev = 5,
  icen = "median", auto = T, intern = F, silent = F, overwrite = F,
  nocheck = F)
```

Arguments

model	Name of a suitable model file template in the working directory or an existing (previous) run number corresponding to a folder in the current working directory that used the same model file as will be used in the current run. If this is supplied, then the model file will be copied into the current working directory for convenience. If not supplied, the default is "model.txt". This file will be converted to a fortran model file. If it is detected to already be a fortran file, then the analysis will proceed without any further file conversion.
data	Name of a suitable data file (see PMwriteMatrix) or an existing (previous) run number corresponding to a folder in the current working directory that used the same data file as will be used in the current run. If this is supplied, then previously made '.ZMQ' files will be copied into the current working directory, bypassing the need to re-convert the .csv file and speeding up the run..
run	Specify the run number of the output folder. Default if missing is the next available number.
include	Vector of subject id values in the data file to include in the analysis. The default (missing) is all.
exclude	A vector of subject IDs to exclude in the plot, e.g. c(4,6:14,16:20)
ode	Ordinary Differential Equation solver log tolerance or stiffness. Default is -4, i.e. 0.0001. Higher values will result in faster runs, but parameter estimates may not be as accurate.

tol	Tolerance for convergence, with default of 0.001.
salt	Vector of salt fractions for each ndrug, default is 1 for each drug. This is not the same as bioavailability.
cycles	Number of cycles to run. Default is 100.
xdev	Multiple of standard deviations for parameters to be sent to NPAG as a range. Default is 5.
icen	Summary of parameter distributions to be used to calculate predictions in HTML report. Default is "median", but could be "mean". #Predictions based on both summaries will be available in objects loaded by PMload.
auto	If auto is False you can answer all questions about the run environment manually. This might be helpful for beginners. Default is True.
intern	MacOSX only: Run IT2B in the R console without a batch script. Default is false. This will be ignored on Windows systems. On the latter, the behavior of cmd.exe (aka the "DOS" window) with R is poor - it does not update until the end of execution, so you cannot see any output that indicates that IT2B is running. If intern=T the HTML summary page will not be automatically loaded at the end of the run, but all post-run processing will occur normally, and you can find the HTML summary page in the /outputs folder: IT2Breport.html.
silent	Boolean operator controlling whether a model summary report is given. Default is True.
overwrite	Overwrite existing run result folders. Default is FALSE.
nocheck	Suppress the automatic checking of the data file with PMcheck. Default is FALSE.

Details

ITrun will execute an IT2B run.

If all function arguments are default, the simplest execution of this command is ITrun(). This will result in generation of a batch file. On Unix (Mac) systems will be launched automatically in a terminal window. On Windows systems, the user must execute the batch file from the current working directory, which will launch IT2B in a command prompt (DOS-like) window. In either case, IT2B will run independently of R so that R can be used for other purposes if desired.

Value

A successful IT2B run will result in creation of a new folder in the working directory. This folder will be named with a date-time stamp in the format "out-YYYYMMMDD-hhmm", e.g. out-2011Apr10-1015. Under this folder will be four subfolders: etc, inputs, outputs, and wrkcopy, described below.

- **etc** Control files for IT2B generally not needed by the user after a completed run.
- **inputs** This folder will contain the .csv data file and the model file.
- **outputs** This folder will contain the output from the IT2B run. These files will be prefixed by DENF, ILOG, OUTF, OUFF, LAST, FROM and RFILE, with appended numbers, usually 0001. DEN is the density file which contains the joint posterior density which can be passed to IT2B. OUTF and OUFF are full and truncated textfiles containing all output of IT2B. OUFF is missing density file. LAST contains last cycle Bayesian posterior parameters and predictions for each subject. FROM contains estimated parameter ranges which can be passed to IT2B. RFILE contains IT2B output formatted for easy import into R, and is the file read by the ITparse command. Finally, there will also be an itlog.txt file containing additional run information.

- **wrkcopy** The working copy format which is used by IT2B. Invisibly to the user, the .csv input file is converted to these text files, one file per subject.

Author(s)

Michael Neely

See Also

ITparse, NPrun

---

makeAUC *Calculation of AUCs*

Description

Calculates AUC from a variety of inputs

Usage

```
makeAUC(data, formula, include, exclude, start = 0, end = Inf,
  icen = "median", outeq = 1, block = 1, method = "linear")
```

Arguments

data	A suitable data object of the *PMpop* class (see makePop), *PMpost* class (see makePost), *PMop* class (see makeOP), the *PMsim* class (see SIMparse), or some other suitable dataframe with at least time/observation columns referred to by formula, with an "id" column (so named) if necessary.
formula	A formula of the form obs ~ time. This is only required with data that is not of class PMpop, PMpost, PMop or PMsim.
include	A vector of subject IDs to include in the AUC calculations, e.g. c(1:3,5,15)
exclude	A vector of subject IDs to exclude in the AUC calculations, e.g. c(4,6:14,16:20)
start	Specify the time to begin AUC calculations. Default is 0.
end	Specify the time to end AUC calculations so that AUC is calculated from start to end. Default for end is the maximum observation time for each subject. Subjects with insufficient data for a specified interval will have AUC calculated for the available data, not to exceed the specified interval.
icen	Only relevant for PMpost or PMpop objects which have predictions based on median or mean of each subject's Bayesian posterior parameter distribution. Default is "median", but could be "mean".
outeq	Specify which output equation is to be used. Default is 1.
block	Specify which observation block (separated by EVID=4) is to be used for each subject. Default is 1.
method	Default is "linear" for AUC trapezoidal calculation. Any other value will result in linear up, log down.

Details

makeAUC will calculate the area under the time concentration curve using the trapezoidal approximation from a variety of inputs. If a PMpost, PMop, or PMsim object is specified, formula is not required. AUCs from PMop objects are based on observations. For AUCs based on predictions, use a PMpost object.

Value

The output of makeAUC is a dataframe of class *PMauc*, which has 2 columns:

id	subject identification
tau	AUC from start to end

Author(s)

Michael Neely

See Also

makeOP, SIMparse

Examples

```
data(PMex1)
op <- makeOP(NPdata.1)
makeAUC(op)
```

---

makeCov *Extract covariate data*

Description

Generates an data.frame with subject-specific covariate data from an *NPAG* or *IT2B* object

Usage

```
makeCov(data)
```

Arguments

data	A suitable data object of the *NPAG* or *IT2B* class (see NPparse or ITparse).

Details

For each subject, makeCov extracts covariate information and Bayesian posterior parameter estimates. This output of this function is suitable for exploration of covariate-parameter, covariate-time, or parameter-time relationships.

Value

The output of makeCov is a dataframe of class *PMcov*, which has the following columns:

id	Subject identification
time	Times of covariate observations
covnames...	Columns with each covariate observations in the dataset for each subject and time
parnames...	Columns with each parameter in the model and the icen summary for each subject, replicated as necessary for covariate observation times and duplicated for Bayesian parameter means and medians
icen	The type of summarized Bayesian posterior individual parameter values: mean or median.

Author(s)

Michael Neely

See Also

NPparse, ITparse, plot.PMcov, summary.PMcov

Examples

```
data(PMex1)
cov <- makeCov(NPdata.1)
cov
names(cov)
```

---

makeCycle *Summarize Pmetrics Run Cycle Information*

Description

Parses the cycle information from an NPAG or an IT2B object.

Usage

```
makeCycle(data)
```

Arguments

data	A suitable data object of the *NPAG* or *IT2B* class (see NPparse or ITparse).

Details

This function will parse the output of NPparse or ITparse to generate a list suitable for analysis and plotting of NPAG or IT2B cycle information.

Value

The output of `makeCycle` is a list of class *PMcycle*, which has 8 objects from NPAG or 6 objects from IT2B :

`names`	Vector of names of the random parameters
#'	
`names`	Vector of names of the random parameters
`cycnum`	Vector cycle numbers, which may start at numbers greater than 1 if a non-uniform prior was specified for the run (NPAG only)
`ll`	Matrix of cycle number and -2*Log-likelihood at each cycle
`gamlam`	A matrix of cycle number and gamma or lambda at each cycle
`mean`	A matrix of cycle number and the mean of each random parameter at each cycle, normalized to initial mean
`sd`	A matrix of cycle number and the standard deviation of each random parameter at each cycle, normalized to initial standard deviation
`median`	A matrix of cycle number and the median of each random parameter at each cycle, normalized to initial median
`aic`	A matrix of cycle number and Akaike Information Criterion at each cycle
`bic`	A matrix of cycle number and Bayesian (Schwartz) Information Criterion at each cycle

A plot method exists in `plot` for *PMcycle* objects.

Author(s)

Michael Neely

See Also

`NPparse`, `ITparse`, `plot.PMcycle`

Examples

```
data(PMex1)
cycle <- makeCycle(NPdata.1)
cycle
names(cycle)
```

`makeErrorPoly` *Assay error polynomial coefficients*

Description

This function plots first, second, and third order polynomial functions fitted to pairs of observations and associated standard deviations for a given output assay. In this way, the standard deviation associated with any observation may be calculated and used to appropriately weight that observation in the model building process. Observations are weighted by the reciprocal of the variance, or squared standard deviation.

Usage

```
makeErrorPoly(obs, sd, data, outeq = 1, col = "red", cex = 3, pch = "+",
  lcol = "blue", lwd = 2, ref = T, legend = T, ...)
```

Arguments

obs	A vector of observations
sd	A vector of standard deviations obtained from repeated measurements at each observation in obs
data	A Pmetrics data file. From this, the maximum and mininimum observations will be retrieved. This is useful to ensure that calculated standard deviations are not negative at any observation in the dataset. If not specified, the default is the maximum *obs*.
outeq	The output equation in *data*. Default is 1.
col	Color of the data points. Default is red.
cex	Relative size of the data points. Default is 3. See par.
pch	Ploting symbol. Default is "+". See par.
lcol	Color of the fitted polynomial lines. Default is blue.
lwd	Width of the lines. Default is 2.
ref	Add a reference line at SD 0 to help evaluate that all fitted SDs are >0. Default is true.
legend	Boolean argument to plot legend. Default is TRUE.
...	Other plotting parameters as in plot.default and par

Value

A plot of the measured observations and fitted polynomial curves and a list with the first, second, and third order coefficients

Author(s)

Michael Neely

Examples

```
makeErrorPoly(obs=c(0,5,50,100,250,500,1000),sd=c(1,0.4,4.5,12,34,60,190))
```

---

makeFinal *Summarize NPAG or IT2B Final Cycle Population Values*

Description

Extracts final cycle information from NPAG or IT2B run.

Usage

```
makeFinal(data)
```

Arguments

data	A suitable data object of the *NPAG* or *IT2B* class (see NPparse or ITparse).

Details

This function will parse the output of NPparse or ITparse to generate a list suitable for analysis and plotting of NPAG or IT2B final cycle population values.

Value

The output of makeFinal is a list of class *PMfinal*, which has 11 objects from NPAG, or 9 objects from IT2B:

popPoints	(NPAG only) Dataframe of the final cycle joint population density of grid points with column names equal to the name of each random parameter plus *prob* for the associated probability of that point
popMean	The final cycle mean for each random parameter distribution
popSD	The final cycle standard deviation for each random parameter distribution
popCV	The final cycle coefficient of variation (SD/Mean) for each random parameter distribution
popVar	The final cycle variance for each random parameter distribution
popCov	The final cycle random parameter covariance matrix
popCor	The final cycle random parameter correlation matrix
popMedian	The final cycle median values for each random parameter
gridpts	(NPAG only) Initial number of support points
nsub	Number of subjects
ab	Matrix of boundaries for random parameter values

A plot method exists in plot for *PMfinal* objects.

Author(s)

Michael Neely

See Also

NPparse, ITparse, plot.PMfinal

Examples

```
data(PMex1)
final <- makeFinal(NPdata.1)
final
names(final)
```

makeNCA *Non-compartmental analysis*

Description

Performs a non-compartmental analysis from observed concentrations in the raw data file or from an individual Bayesian posterior predicted time-observation profiles (PMpost object) generated automatically after an NPAG run by the makePost command and loaded with PMload.

Usage

```
makeNCA(x, postPred = F, include, exclude, input = 1, icen = "median",
  outeq = 1, block = 1, start = 0, end = Inf, first = NA, last = NA)
```

Arguments

x	Data to analyze. This can be specified in a number of ways. • It can be the run number, e.g. 3, that has been previously loaded with PMload. Either the mdata file from the run (NPAG or IT2B) can be used (default) or the post object can be used (NPAG only) by specifying postPred = T below. If x is a run number that corresponds to both an NPAG and IT2B run which have been previously loaded into memory with PMload, the NPAG run will be used. • It can be the run number of a run that has *not* been previously loaded with PMload. In this case, the current working directory should be the Runs folder as makeNCA will call PMload. • It can be the specific name of an mdata.x file already loaded into memory with PMload, e.g. mdata.3. Note that quotation marks are not necessary since mdata.3 is an object, not a label/character vector. • Finally, it can be the name of a Pmetrics data file in the current working directory, which will be loaded with PMreadMatrix and analyzed, e.g. "data.csv". In this case, quotation marks are reqired, because x is now a character vector specifying the filename of the file to load.
postPred	Boolean switch to use the posterior predictions rather than the observed. concentrations. Default is FALSE. Ignored if an IT2B run is used to supply the raw data file.
include	A vector of subject IDs to include in the NCA, e.g. c(1:3,5,15)
exclude	A vector of subject IDs to exclude in the NCA, e.g. c(4,6:14,16:20)
input	The number of the input (e.g. drug) to analyze; default 1.
icen	If post is TRUE, use predictions based on median or mean of each subject's Bayesian posterior parameter distribution. Default is "median", but could be "mean".
outeq	The number of the output equation to analyze; default 1
block	The number of the observation block within subjects, with each block delimited by EVID=4 in the data file; default 1

`start` The beginning of the time interval to look for doses and observations, e.g. 120. It can be a vector to allow for individual start times per subject, e.g. c(120,120,144,168). If the length of `start` is less than the number of subjects, the last value will be recycled as needed. If the `start` time is not 0 (default), then it is assumed that steady state (multiple dose) conditions apply.

`end` Analogous to `start`, set this equal to the end of the dosing interval. It too can be a vector, with the last value recycled as necessary. Default is `Inf`, i.e. all data used.

`first` Alternative way to specify time interval for NCA by choosing dose number, e.g. 1 or 3. May be a numeric vector, like `start` and `end`, e.g. c(1,1,1,3,1,...) to allow for individualization by subject. The last value will be recycled to ensure length equal to the number of subjects. Default is `NA`, which means `start` will be used.

`last` The complement to `first`, specifying the last dose to end the time interval. If `NA`, which is the default, then the maximum time per subject will be the upper bound of the time interval. Like `first`, `last` can be a vector, with the last value recycled as necessary. Use `NA` in the vector to signify maximum time for that subject.

Details

If concentrations from multiple dose intervals are included in the `start`-`end` time interval, `makeNCA` will superpose the concentrations using the time after dose. An error will be generated if different doses are within this interval as superposition would no longer be valid.

A minimum of 5 concentrations must be available to perform NCA for any given subject. Fewer than this will suppress all results for that subject.

Value

A dataframe of class *PMnca* with columns

`id` Subject identification

`auc` Area under the time-observation curve, using the trapezoidal approximation, from time 0 until the second dose, or if only one dose, until the last observation

`aumc` Area under the first moment curve

`k` Slope by least-squares linear regression of the final 3 log-transformed observations vs. time. If the final 3 concentrations are not decreasing such that linear regression results in a positive slope, this value and all others that depend on `k` will be suppressed.

`auclast` Area under the curve from the time of the last observation to infinity, calculated as [Final obs]/k. This value will be suppressed if start != 0.

`aumclast` Area under the first moment curve from the time of the last observation to infinity. This value will be suppressed if start!=0.

`aucinf` Area under the curve from time 0 to infinity, caluculated as auc + auclast

`aumcinf` Area under the first moment curve from time 0 to infinity

`mrt` Mean residence time, calculated as 1/k

`cmax` Maximum predicted concentration after the first dose

`tmax` Time to cmax

`cl` Clearance, calculated as dose/aucinf

vdss	Volume of distribution at steady state, calculated as cl*mrt
thalf	Half life of elimination, calculated as ln(2)/k
dose	Dose for each subject

Author(s)

Michael Neely

---

makeNPDE *Simulation-based model diagnostics*

Description

Use simulations to run model diagnostic tests.

Usage

```
makeNPDE(run, outeq, nsim = 1000, ...)
```

Arguments

run	When the current working directory is the Runs folder, the folder name of a previous run that you wish to use for the npde, which will typically be a number, e.g. 1.
outeq	The number of the output equation to simulate/test. Default is missing, which will test all output equations.
nsim	The number of simulations per subject in the data file. We recommend 1000 (the default) to return valid npde results. More may result in excessive simulation times.
...	Other parameters to be passed to SIMrun.

Details

This function is a Pmetrics wrapper to the autonpde function in the npde package of Comets et al that will generate normalized prediction distribution errors. Output from a loaded NPAG or IT2B run will be used as the population model supplied to the simulator. The function will iterate through the .csv file, using each subject as a template to simulate nsim new individuals. It is HIGHLY recommended to use the default value of 1000 for nsim for the most valid calculation of npde. More than this could take a long time to execute. The mean population values will be used for each parameter and the covariance matrix. Errors may arise if extreme or negative concentrations are simulated from excessively large covariance matrices.

Value

The output of makeNPDE is a list of class PMnpde with objects of NpdeObject class. Additionally, two objects with run numbers appended will be saved to the output directory of the run for subsequent loading with PMload: npde and sim. *npde* is the PMnpde object, and *sim* is a PMsim object of all simulations combined which can be used for visual predictive checks (see plot.PMsim).

Author(s)

Michael Neely

References

Brendel K, Comets E, Laffont CM, Laveille C, Mentre F (2006) Metrics for external model evaluation with an application to the population pharmacokinetics of gliclazide. *Pharmaceutical Research*, 23:2036-49

Mentre F, Escolano S (2006) Prediction discrepancies for the evaluation of nonlinear mixed-effects models. *J Pharmacokinet Pharmacodyn*, 33:345-67

See Also

SIMrun, autonpde, plot.PMnpde

---

makeOP *Generated observed vs. predicted data*

Description

Generates an observed vs. predicted data.frame from an *NPAG* or *IT2B* object

Usage

```
makeOP(data)
```

Arguments

data	A suitable data object of the *NPAG* or *IT2B* class (see NPparse or ITparse).

Details

makeOP will parse the output of NPparse or ITparse to generate a data.frame suitable for analysis and plotting of observed vs. population or or individual predicted outputs.

Value

The output of makeOP is a data frame of class *PMop*, which has a population and posterior prediction object (also class *PMop*) for each output equation. Each of these has 13 columns:

id	subject identification
time	observation time in relative hours
obs	observation
pred	prediction
pred.type	Population predictions based on Bayesian prior parameter value distribution, or individual predictions based on Bayesian posterior parameter value distributions
icen	Predictions based on mean or median of Bayesian pred.type parameter values
outeq	output equation number
block	dosing block number for each subject, as defined by dose resets (evid=4).

obsSD	standard deviation of the observation based on the assay error polynomial
d	prediction error, pred-obs
ds	squared prediction error
wd	weighted prediction error, which is the prediction error divided by the obsSD
wds	weighted squared prediction error

A plot method exists in plot for *PMop* objects.

Author(s)

Michael Neely

See Also

NPparse, ITparse, plot.PMop, summary.PMop

Examples

```
data(PMex1)
op <- makeOP(NPdata.1)
op
names(op)
summary(op)
```

---

makePop *Individual Bayesian population prior predictions at short intervals*

Description

Returns the Bayesian population prior predictions at short intervals specified during the NPAG run, up to 12 minutes.

Usage

```
makePop(run, NPdata)
```

Arguments

run	The number of the folder that contains the relevant run. If missing, NPdata will be used.
NPdata	Optional name of NPdata object if run is missing.

Value

A dataframe of class *PMpop* with columns:

id	Subject id
time	Time of predictions in decimal hours
icen	Prediction based on mean or median of Bayesian posterior parameter distribution
pred	Predicted output for each outeq
outeq	Output equation number
block	Observation blocks within subjects as defined by EVID=4 dosing events

Author(s)

Michael Neely

`makePost` *Individual Bayesian posterior predictions at short intervals*

Description

Returns the Bayesian posterior predictions at short intervals specified during the NPAG run, up to 12 minutes. These results are contained separately from the main output of NPAG, in the PRTBxxxx file.

Usage

```
makePost(run, NPdata)
```

Arguments

`run`	The number of the folder that contains the relevant run. If missing will be set to current working directory.
`NPdata`	Optional name of NPdata object if run is missing.

Value

A dataframe of class *PMpost* with columns:

`id`	Subject id
`time`	Time of predictions in decimal hours
`icen`	Prediction based on mean or median of Bayesian posterior parameter distribution
`pred`	Predicted output for each outeq
`outeq`	Output equation number
`block`	Observation blocks within subjects as defined by EVID=4 dosing events

Author(s)

Michael Neely

makePTA *Calculation of PTAs*

Description

Calculates the Percent Target Attainment (PTA)

Usage

```
makePTA(simdata, simlabels, targets, target.type, success, outeq = 1,
  free.fraction = 1, start, end)
```

Arguments

simdata	A vector of simulator output filenames, e.g. c("simout1.txt","simout2.txt"), with wildcard support, e.g. "simout*" or "simout?", or a list of PMsim objects made by SIMparse with suitable simulated regimens and observations. The number and times of simulated observations does not have to be the same in all objects.
simlabels	Optional character vector of labels for each simulation. Default is c(``Regimen 1'', ``Regimen 2'
targets	A vector of pharmacodynamic targets, such as Minimum Inhibitory Concentrations (MICs), e.g. c(0.25, 0.5,1,2,4,8,16,32). This can also be a sampled distribution using makePTAtarget.
target.type	A numeric or character vector, length 1. If numeric, must correspond to an observation time common to all PMsim objects in simdata, rounded to the nearest hour. In this case, the target statistic will be the ratio of observation at time target.type to target. This enables testing of a specific timed concentration (e.g. one hour after a dose or C1) which may be called a peak, but is not actually the maximum drug concentration. Be sure that the time in the simulated data is used, e.g. 122 after a dose given at 120. Character values may be one of "time", "auc", "peak", or "min", for, respectively, percent time above target within the time range specified by start and end, ratio of area under the curve within the time range to target, ratio of peak concentration within the time range to target, or ratio of minimum concentration within the time range to target.
success	A single value specifying the success statistic, e.g. 0.4 for proportion time (end-start) above target, or 100 for peak:target.
outeq	An integer specifying the number of the simulated output equation to use. Default is 1.
free.fraction	Proportion of free, active drug. Default is 1, i.e. 100% free drug or 0% protein binding.
start	Specify the time to begin PTA calculations. Default is a vector with the first observation time for subjects in each element of simdata, e.g. dose regimen. If specified as a vector, values will be recycled as necessary.
end	Specify the time to end PTA calculations so that PTA is calculated from start to end. Default for end is the maximum observation time for subjects in each element of simdata, e.g. dose regimen. If specified as a vector, values will be recycled as necessary. Subjects with insufficient data (fewer than 5 simulated observations) for a specified interval will trigger a warning. Ideally then, the simulated datset should contain sufficient observations within the interval specified by start and end.

Details

makePTA will calculate the PTA for any number of simulations, targets and definitions of success. Simulations typically differ by dose, but may differ by other features such as children vs. adults.

Value

The output of makePTA is a list of class *PMpta*, which has 2 objects:

results	A data frame with the following columns: simnum, id, target, pdi. *simnum* is the number of the simulation; *id* is the simulated profile number within each simulation; *target* is the specified target; and *pdi* is the target pharmacodynamic index, e.g. time > target, auc:target, etc.
outcome	A data frame summarizing the results with the following columns: simnum, target, prop.success, pdi.mean, and pdi.sd. If targets was specified via makePTAtarget to be a sampled distribution, then the target column will be missing from the outcome table. *simnum* and *target* are as for results. The *prop.success* column has the proportion with a pdi > success, as specified in the function call. The *pdi.mean* and *pdi.sd* columns have the mean and standard deviation of the target pharmacodynamic index (e.g. proportion end-start above target, ratio of Cmax to target) for each simulation and target.

Author(s)

Michael Neely

See Also

plot.PMpta, SIMparse

---

makePTAtarget *Make PTA target object*

Description

Make a Percent Target Attainment (PTA) Target

Usage

```
makePTAtarget(x)
```

Arguments

x	A data.frame or name of .csv file in working directory whose first two columns are targets and the number of samples for each target. An example can be seen for Staphylococcus aureus susceptibility to vancomycin at the EUCAST website at http://mic.eucast.org/Eucast2/regShow.jsp?Id=1214.

Details

makePTAtarget generates an object of class *PMpta.targ* which can be used in the makePTA command for targets sampled from a distribution.

Value

A data frame with two columns named targets and n, of class *PMpta.targ*.

See Also

makePTA

---

mic1 *Example MIC data*

Description

Example MIC data

Usage

```
mic1
```

Format

An R data frame containing example MIC distribution data in two columns:

- mic Minimum inhibitory concentration
- n Number of organisms with the given MIC

Details

This data frame contains MIC data for cefepime against E. coli. It was obtained from the EUCAST website at http://mic.eucast.org. Select the organism or drug, and then select the desired row of the resulting table to see a histogram (top) and table (bottom) of MIC distributions.

Copy the table into excel, save as a .csv file, and read into R using read.csv. Then use makePTAtarget.

Author(s)

Michael Neely

MMopt *Compute MM-optimal Sample Times*

Description

Computes 1 to 4 MM-optimal sampling times.

Usage

```
MMopt(poppar, model = "model.txt", data = "data.csv", nsamp = 1,
  weight = c("none", "AUC"), predInt = 0.5, outeq = 1, ...)
```

Arguments

poppar	An object of class *PMfinal* (see makeFinal)
model	Name of a suitable model file template in the working directory. The default is "model.txt". This file will be converted to a fortran model file. If it is detected to already be a fortran file, then the simulation will proceed without any further file conversion.
data	Either a PMmatrix object previously loaded with (PMreadMatrix) or character vector with the filename of a Pmetrics matrix file that contains template regimens and observation times. The value for outputs can be coded as any number(s) other than -99. The number(s) will be replaced in the simulator output with the simulated values.
nsamp	The number of MM-optimal sample times to compute; default is 1, but can be up to 4. Values >4 will be capped at 4.
weight	Character label: • none The default. MMopt times will be chosen to maximally discriminate all responses at all times. • AUC MMopt times will be chosen to maximally discriminate AUC, regardless of the shape of the response profile.
predInt	The interval in fractional hours for simulated predicted outputs at times other than those specified in the template data. The default is 0.5, which means there will be simulated outputs every 30 minutes from time 0 up to the maximal time in the template file. You may also specify predInt as a vector of 3 values, e.g. c(1,4,1), similar to the R command seq, where the first value is the start time, the second is the stop time, and the third is the step value. Outputs for times specified in the template file will also be simulated. To simulate outputs *only* at the output times in the template data (i.e. EVID=0 events), use predInt=0. Note that the maximum number of predictions total is 594, so the interval must be sufficiently large to accommodate this for a given number of output equations and total time to simulate over. If predInt is set so that this cap is exceeded, predictions will be truncated.
outeq	Output equation to optimize
...	Other parameters to pass to SIMrun, which are not usually necessary.

Details

Based on the mulitple-model optimization algorithm developed by David Bayard and presented at the 2012 American College of Clinical Pharmacology Meeting and the 2013 International Association of Therapeutic Drug Monitoring and Clinical Toxicology meeting. A manuscript is in preparation.

Value

A object of class *MMopt* with 3 items.

sampleTime	The MM-optimal sample times
bayesRisk	The Bayesian risk of mis-classifying a subject based on the sample times. This is more useful for comparisons between sampling strategies, with minimization the goal.
simdata	A *PMsim* object with the simulated profiles

Author(s)

Michael Neely

See Also

SIMrun, plot.MMopt, print.MMopt

---

mtsknn.eq *Multivariate two-sample test based on k-nearest neighbors*

Description

Compare discrete distributions

Usage

```
mtsknn.eq(x, y, k, clevel = 0.05, getpval = TRUE, print = TRUE)
```

Arguments

x	A matrix or data frame.
y	A matrix or data frame.
k	k An integer.
clevel	The confidence level. Default value is 0.05.
getpval	Logic value. If it is set to be TRUE the p value of test will be calcuated and reported; if it is set to be false the p value will not be calculated.
print	Boolean value. If it is set to be TRUE the test result will be reported; if it is set to be FALSE the test result will not be reported.

Details

This function tests whether two samples share the same underlying distribution based on k-nearest-neighbors approach. Matrices or data frames x and y are the two samples to be tested. Each row consists of the coordinates of a data point. The integer k is the number of nearest neighbors to choose in the testing procedure. This approach is robust in the unbalanced case.

Value

A list consists of the test statistics, normalized Z score and corresponding P value.

Author(s)

Lisha Chen (Yale), Peng Dai (Stonybrook) and Wei Dou (Yale)

References

Schilling, M. F. (1986). Multivariate two-sample tests based on nearest neighbors. *J. Amer. Statist. Assoc.*, 81 799-806. Henze, N. (1988). A multivariate two-sample test based on the number of nearest neighbor type coincidences.*Ann. Statist.*, 16 772-783. Chen, L. and Dou W. (2009). Robust multivariate two-sample tests based on k nearest neighbors for unbalanced designs. *manuscripts*.

Examples

```
## Example of two samples from the same multivariate t distribution:
 n <- 100
 x <- matrix(rt(2*n, df=5),n,2)
 y <- matrix(rt(2*n, df=5),n,2)
 mtsknn.eq(x,y,3)
 ## Example of two samples from different distributions:
 n <- 100
 x <- matrix(rt(2*n, df=10),n,2)
 y <- matrix(rnorm(2*n),n,2)
 mtsknn.eq(x,y,3)
```

NM2PM *Convert NONMEM to Pmetrics Data Files*

Description

NM2PM will convert NONMEM .csv data files to Pmetrics csv data files.

Usage

```
NM2PM(data, ctl)
```

Arguments

data	The name and extension of a NONMEM data (e.g. .csv) file in the working directory, or the full path to a file.
ctl	The name and extension of a NONMEM control (e.g. .ctl) file in the working directory, or the full path to a file.

Details

The format of NONMEM and Pmetrics data .csv files are similar, but not quite identical. A major difference is that the order of the columns are fixed in Pmetrics (not including covariates), while they are user-determined in NONMEM, and specified in a control (.ctl) file.

A list of other differences follows by data item.

- ID This item is the same in both formats and is required.
- EVID This is the same in both formats but is not required in NONMEM. Doses have an EVID of 1 and observations 0. EVID=4 (dose/time reset) is the same in Pmetrics and NONMEM. EVID=2 (other event) and EVID=3 (dose reset) are not directly supported in Pmetrics, but if included in a NONMEM file, will be converted into covariate values. Specifically the value in the CMT variable will be the covariate value for EVID=2, while for EVID=3, the covariate will be 1 at the time of the EVID=3 entry and 0 othewise. This allows for handling of these events in the Pmetrics model file using conditional statements.
- DATE Pmetrics does not use dates, but will convert all NONMEM dates and times into relative times.
- TIME Pmetrics uses relative times (as does NONMEM), but the NONMEM pre-processor will convert clock times to relative times, as does NM2PM.
- RATE NONMEM RATE items are converted by this function to Pmetrics DURation values.

- AMT becomes DOSE in Pmetrics
- ADDL is supported in both formats. However, if NONMEM files contain an SS flag, it will be incorporated as ADDL=-1 according to Pmetrics style.
- II is the same in both formats.
- INPUT in Pmetrics is similar to CMT in NONMEM for doses.
- DV in NONMEM becomes OUT in Pmetrics. Ensure that the units of OUT are consistent with the units of DOSE.
- OUTEQ In Pmetrics, this is roughly equivalent to CMT in NONMEM for observation events. The lowest CMT value for any observation becomes OUTEQ=1; the next lowest becomes OUTEQ=2, etc.
- SS Steady state dosing is incorporated into Pmetrics as ADDL=-1.
- MDV Missing DV in NONMEM become OUT=-99 in Pmetrics.
- Covariates These are copied from NONMEM to Pmetrics. Note that Pmetrics does not allow missing covariates at time 0 for each subject.
- DROP Items marked as DROP in the NONMEM control file will not be included in the Pmetric data file.

It is strongly suggested to run PMcheck on the returned object for final adjusting.

Value

A Pmetrics style PMmatrix data.frame.

Author(s)

Michael Neely

See Also

PMcheck, PMwriteMatrix, PMwrk2csv

NPparse *Parse Pmetrics NPAG Output*

Description

NPparse processes the output from an NPAG run into a list.

Usage

```
NPparse(outfile = "NP_RF0001.TXT")
```

Arguments

outfile	This is the filename of the output from NPAG. Typically, the file will be called NP_RF0001.txt, and this is the default.

Details

This function can take some time to process the RFILE, depending on the number of subjects, doses, observations, etc. Typical wait times are a few seconds up to 5 minutes. When processing is complete a summary of the extracted data will be reported on the console.

Value

The output of `NPparse` is a list with the following objects and of the class *NPAG*.

`nsub`	Number of subjects
`nactve`	Number of active grid points at the final cycle
`nvar`	Number of random variables or parameters in the model
`nofix`	Number of fixed variables or parameters in the model
`par`	Names of random parameters
`parfix`	Names of fixed parameters
`covnames`	Names of covariates
`ab`	Initial boundaries for each random parameter
`valfix`	Values for fixed parameters
`ndim`	Number of differential equations in model, or 0 for only output equation, or -1 for analytic solution (algabraic)
`indpts`	Index for the initial number of gridpoints in the model
`icycst`	Starting cycle number
`icycmax`	Maximum number of cycles specified by the user
`icyctot`	Number of cycles run. If less than `icycmax`, convergence occurred.
`converge`	Boolean value if convergence occurred.
`ODEtol`	Ordindary Differential Equation solver tolerance.
`prior`	Prior density for the run, either "UNIFORM" or the name of the user-specified density file, typically "DEN0001".
`ERRmod`	Assay error model: 1 for SD; 2 for SD*gamma; 3 for additive lambda model; and 4 for gamma only
`numeqt`	Number of output equations
`ndrug`	Number of drug inputs
`salt`	Vector of values of the salt fraction for each `ndrug`
`ndose`	Vector of the number of doses for each subject in the population
`ncov`	Number of covariates in the model
`nobs`	Vector of the number of observations for each subject in the population
`nobsmax`	Maximum number of observation in any individual subject
`numt`	Vector of the number of time points for each subject at which a prediction is generated for each *numeqt* output equation
`corden`	Final cycle joint population density of parameter estimates
`postden`	Array of posterior parameter value distributions for the first 100 subjects at each observation time point. *postden[nsub,nactvepost,density]* where *nactvepost* is the posterior grid point

pyjgx	Matrix of posterior probability of each *nactve* point for each subject, given that subject's data
ypredpop	Array of population model predictions for each subject at each observation time point. *ypredpop[nsub,numeqt,time,type]* where *type* is 1=mean, 2=median, 3=mode of the population prior used to calculate ypredpop
ypredbay	Array of Bayesian posterior model predictions for each subject at each observation time point. *ypredbay[nsub,numeqt,time,type]* where *type* is 1=mean, 2=median, 3=mode of the population prior used to calculate ypredbay
ttpred	Matrix of the prediction time points for each subject, with *nsub* rows and max(*numt*) columns
exx	Array of the mean, median, and mode of the posterior marginal distribution for each parameter in each subject, of the form *exx[nvar,type,nsub]*
ypredpopt	Array of population model predictions for each subject at each *ttpred* time point, of the form *ypredpopt[nsub, numeqt, time, type]*, where type is 1=mean, 2=median, 3=mode of the population prior used to calculate *ypredpopt*
ilog	Matrix of cycle number and associated log-likelihood
iic	Matrix with cycle number and Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) for each cycle
imean	Matrix of cycle numbers and associated means for each random parameter
isd	Matrix of cycle numbers and associated standard deviations for each random parameter
iaddl	Array of additional information for each random parameter in each cycle, of the form *iaddl[info, nvar, cycle]*, where info is a value from 1 to 12: 1= mode; 2= skewness; 3= kurtosis; 4-8 give percentiles of the distribution (4=2.5%; 5=25%; 6=50% [median], 7=75%; 8=97.5%); 9= the standard deviation of a normal distribution with the same interquartile range; 10=the standard deviation of a normal distribution with the same 95% range; 11=the average of 9 and 10; 12=the % scaled information
igamlam	Matrix of cycle number and associated gamma or lambda
blog	Vector of each subject's Bayesian posterior log-likelihood
bmean	Matrix of subject numbers and associated Bayesian posterior means for each random parameter
bsd	Matrix of subject numbers and associated Bayesian posterior standard deviations for each random parameter
baddl	Array of Bayesian posterior additional information for each random parameter for each subject, of the form *baddl[info, nvar, nsub]*, where info is the same as for *iaddl*.
bauc	Matrix of AUC blocks for each subject with 5 columns: [nsub, numeqt, nblock, tau, auc]; *nsub* and *numeqt* are as previously defined; *nblock* is the AUC block as defined by successive dose reset (evid=4) events; *tau* is the time interval for that block; *auc* is the AUC for that block
sdata	Subject data consisting of 5 columns: [id, nsub, age, sex, ht], *id* is the original identification number in the .csv matrix file; *nsub* is the sequential subject number in the NPAG run; *age*, *sex* and *ht* will be missing for .csv input and present if included in .wrk input files
dosecov	Matrix with all dosing information for each subject, including times, routes, amounts, and associated covariate values

outputs	Matrix with measured outputs for each subject and associated assay error polynomials. The order of the columns is nsub, time, numeqt, observation, c0, c1, c2, c3, where the last four columns are the coefficients of the assay error polynomial for that observation, such that SD[obs] = c0 + c1*[obs] + c2*[obs]**2 + c3*[obs]**3
negflag	A flag indicating that some negative predictions were changed to missing. This means that the model may be misspecified.
mdata	The filename of the data used in the run.

Author(s)

Michael Neely

NPrun *Execute an NPAG run.*

Description

Runs NPAG

Usage

```
NPrun(model = "model.txt", data = "data.csv", run, include, exclude,
  ode = -4, tol = 0.01, salt, cycles = 100, indpts, icen = "median",
  aucint, idelta = 12, prior, auto = T, intern = F, silent = F,
  overwrite = F, nocheck = F, parallel = NA)
```

Arguments

model	Name of a suitable model file template in the working directory or an existing (previous) run number corresponding to a folder in the current working directory that used the same model file as will be used in the current run. If this is supplied, then the model file will be copied into the current working directory for convenience. If not supplied, the default is "model.txt". This file will be converted to a fortran model file. If it is detected to already be a fortran file, then the analysis will proceed without any further file conversion.
data	Name of a suitable data file (see PMwriteMatrix) or an existing (previous) run number corresponding to a folder in the current working directory that used the same data file as will be used in the current run. If this is supplied, then previously made '.ZMQ' files will be copied into the current working directory, bypassing the need to re-convert the .csv file and speeding up the run..
run	Specify the run number of the output folder. Default if missing is the next available number.
include	Vector of subject id values in the data file to include in the analysis. The default (missing) is all.
exclude	A vector of subject IDs to exclude in the analysis, e.g. c(4,6:14,16:20)
ode	Ordinary Differential Equation solver log tolerance or stiffness. Default is -4, i.e. 0.0001. Higher values will result in faster runs, but parameter estimates may not be as accurate.

tol	Tolerance for convergence of NPAG. Smaller numbers make it harder to converge. Default value is 0.01.
salt	Vector of salt fractions for each drug in the data file, default is 1 for each drug. This is not the same as bioavailability.
cycles	Number of cycles to run. Default is 100.
indpts	Index of starting grid point number. Default is missing, which allows NPAG to choose depending on the number of random parameters: 1 or 2 = index of 1; 3 = 3; 4 = 4, 5 = 6, 6 or more is 10+number of multiples for each parameter greater than 5, e.g. 6 = 101; 7 = 102, up to 108 for 13 or more parameters.
icen	Summary of parameter distributions to be used to calculate predictions in HTML report. Default is "median", but could be "mean". Predictions based on both summaries will be available in objects loaded by PMload.
aucint	Maintained for backwards compatibility and not used currently. Interval for AUC calculations. Default is 24 hours if the number of intervals is not greater than 48; otherwise it defaults to the interval which allows for <= 48 intervals.
idelta	Interval in 1/60 time unit, typically minutes, for predictions at times other than observations. Default is 12.
prior	Name of a suitable NPAG output object from a prior run loaded with PMload, i.e. the *NPdata* object. A prior may be specified if the user wishes to start from a non-uniform prior distribution for the NPAG run. The default value is -99, which translates in NPAG to a uniform prior distribution. An alternative is to include a DEN0001 file from the prior NPAG run in the working directory of the new run, and specify this as the value for prior, e.g. prior = 'DEN0001'.
auto	If auto is False you can answer all questions about the run environment manually. This might be helpful for beginners. Default is True.
intern	MacOSX only: Run NPAG in the R console without a batch script. Default is false. This will be ignored if on Windows systems. On the latter, the behavior of cmd.exe (aka the "DOS" window) with R is poor - it does not update until the end of execution, so you cannot see any output that indicates that NPAG is running. If intern=T the HTML summary page will not be automatically loaded at the end of the run, but all post-run processing will occur normally, and you can find the HTML summary page in the /outputs folder: NPAGreport.html.
silent	Boolean operator controlling whether a model summary report is given. Default is TRUE.
overwrite	Overwrite existing run result folders. Default is FALSE.
nocheck	Suppress the automatic checking of the data file with PMcheck. Default is FALSE.
parallel	Run NPAG in parallel. Default is NA, which will be set to TRUE for models that use differential equations, and FALSE for algebraic/explicit models. The majority of the benefit for parallelization comes in the first cycle, with a speed-up of approximately 80% of the number of available cores on your machine, e.g. an 8-core machine will speed up the first cycle by 0.8 * 8 = 6.4-fold. Subsequent cycles approach about 50%, e.g. 4-fold increase on an 8-core machine. Overall speed up for a run will therefore depend on the number of cycles ru and the number of cores.

Details

NPrun will execute an NPAG run.

If all function arguments are default, the simplest execution of this command is `NPrun()`. This will result in generation of a batch file. On Unix (Mac) systems will be launched automatically in a terminal window. On Windows systems, the user must execute the batch file from the current working directory, which will launch NPAG in a command prompt (DOS-like) window. In either case, NPAG will run independently of R so that R can be used for other purposes if desired.

Value

A successful NPAG run will result in creation of a new folder in the working directory. This folder will be named with a date-time stamp in the format "out-YYYYMMMDD-hhmm", e.g. out-2011Apr10-1015. Under this folder will be four subfolders: etc, inputs, outputs, and wrkcopy, described below.

- **etc** Control files for NPAG generally not needed by the user after a completed run.
- **inputs** This folder will contain the .csv data file and the model file.
- **outputs** This folder will contain the output from the NPAG run. These files will be prefixed by DEN, ILOG, OUT, OUTT, PRTB and RFILE, with appended numbers, usually 0001. DEN is the density file which can be used to specifiy a non-uniform prior parameter value distribution for a subsequent NPAG run of the same model via the `prior` argument above. ILOG is a summary of cycle objective function values, gamma/lambda, and gridpoints. OUT and OUTT are full and truncated textfiles containing all output of NPAG. OUTT is missing density file. PRTB contains Bayesian posterior individual predictions for each subject and output at timepoints specified in the NPAG instructions (e.g. every 2, 4, 8, 12 minutes) as well as predictions at each observation time. RFILE contains NPAG output formatted for easy import into R, and is the file read by the `NPparse` command. Finally, there will also be an nplog.txt file containing additional run information.
- **wrkcopy** The working copy format which is used by NPAG. Invisibly to the user, the .csv input file is converted to these text files, one file per subject.

Author(s)

Michael Neely

See Also

`NPparse`, `ITrun`

`plot.MMopt` *Plot Pmetrics Multiple-Model Optimal Sampling Objects*

Description

Plots *MMopt* objects

Usage

```
## S3 method for class 'MMopt'
plot(x, mm.col = "red", mm.lty = 2, mm.lwd = 2, ...)
```

Arguments

x	The name of an *MMopt* data object generated by MMopt
mm.col	Color of the optimal sample time reference lines. Default is red.
mm.lty	Type of the optimal sample time reference lines. Default is dashed.
mm.lwd	Width of the optimal sample time reference lines. Default is 2.
...	Other parameters to pass to plot.PMsim.

Details

Simulated observations are plotted on the y-axis vs. time on the x.axis. Optimal sampling times are indicated as vertical lines.

Value

Plots the simulation profiles with MMoptimal times indicated as vertical lines.

Author(s)

Michael Neely

See Also

plot.PMsim, plot, par, axis

---

plot.PMcov *Plot Pmetrics Covariate objects*

Description

Plot PMcov objects

Usage

```
## S3 method for class 'PMcov'
plot(x, formula, icen = "median", include, exclude,
  mult = 1, log = F, square = F, ref = F, lowess = F, grid = F,
  ident = F, reg = F, ci = 0.95, cex = 1, cex.lab = 1.2,
  x.stat = 0.6, y.stat = 0.1, col.stat = "black", cex.stat = 0.8,
  lwd = 2, col = "red", xlim, ylim, xlab, ylab, out = NA, ...)
```

Arguments

x	The name of an *PMcov* data object generated by makeCov
formula	This is a mandatory formula of the form y ~ x, where y and x are the two data parameters to plot.
icen	A character vector to summarize covariate and parameter values. Default is "median", but can also be one of "none", "mean". If time is a variable in formula, the value will be set to "none" and the y values will be aggregated by subject ID vs. time.

include	A vector of subject IDs to include in the plot, e.g. c(1:3,5,15)
exclude	A vector of subject IDs to exclude in the plot, e.g. c(4,6:14,16:20)
mult	Multiplication factor for y axis, e.g. to convert mg/L to ng/mL
log	Boolean operator to plot in log-log space; the default is False
square	Boolean operator to force a square plot with equal x and y limits; the default is True
ref	Boolean operator to draw a unity line; the default is True unless "time" is the x value in formula in which case this is ignored
lowess	Boolean operator to draw a lowess regression line; the default is False and this is ignored if "time" is the x value in formula
grid	Either a boolean operator to plot a reference grid, or a list with elements x and y, each of which is a vector specifying the native coordinates to plot grid lines; the default is False. For example, grid=list(x=seq(0,24,2),y=1:10). Defaults for missing x or y will be calculated by axTicks.
ident	Boolean operator to plot points as ID numbers; the default is False. This option is useful to identify outliers.
reg	Boolean operator to draw a linear regression line; the default is True unless "time" is the x value in formula in which case this is ignored. If this option is selected, regression statistics will be printed on the plot if at least 3 subjects are included.
ci	The confidence interval for the linear regression parameter estimates; the default is 0.95.
cex	Size of the plot symbols.
cex.lab	Size of the plot labels.
x.stat	Horizontal position to plot the linear regression statistics; the units are relative to the origin, i.e. extreme left is 0 and extreme right is 1.
y.stat	Vertical position to plot the linear regression statistics; the units are relative to the origin, i.e. extreme bottom is 0 and extreme top is 1.
col.stat	Color of the text for the regression statistics.
cex.stat	Size of the text for the regression statistics.
lwd	Width of the various regression or reference lines (unity, linear regression, or lowess regression)
col	This parameter will be applied to the plotting symbol and is "red" by default.
xlim	Limits of the x-axis as a vector, e.g. c(0,1). It does not need to be specified, but can be.
ylim	Analogous to xlim
xlab	Label for the x-axis. If missing, will default to the name of the x-variable.
ylab	Label for the y-axis. If missing, will default to the name of the y-variable.
out	Direct output to a PDF, EPS or image file. Format is a named list whose first argument, type is one of the following character vectors: "pdf", "eps" (maps to postscript), "png", "tiff", "jpeg", or "bmp". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is NA which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.

...	Other parameters as found in plot.default.

Details

This method will plot any two columns, specified using a formula, of a PMcov object, which contains covariate and Bayesian posterior parameter information for each subject. Specifiying any two variables that do not include time will result in a scatter plot with optional regression and reference lines. If time is included as the x variable, the y variable will be plotted vs. time, aggregated by subject. This can be useful to see time varying parameters, although a formula within formula approach may be required, e.g. plot(cov.1,I(cl_0*wt**0.75)~time) in order to see the change in cl over time according to the change in wt over time, even though cl_0 is constant for a given subject.

Value

Plots the object.

Author(s)

Michael Neely

See Also

makeCov, plot, par, axis

Examples

```
data(PMex1)
plot(cov.1,V~wt)
```

---

plot.PMcycle *Plot NPAG Cycle Information*

Description

plot.PMcycle plots *PMcycle* objects

Usage

```
## S3 method for class 'PMcycle'
plot(x, x.leg = 0, y.leg = 1, cex.leg = 1.2, omit, col,
  out = NA, ...)
```

Arguments

x	The name of an *PMcycle* data object generated by makeCycle
x.leg	Porportionate location along the X-axis to place legend; 0 (default) is at left, 1 at right.
y.leg	Porportionate location along the X-axis to place legend; 0 is at bottom, 1 (default) at top.
cex.leg	Porportionate size of legend text.

omit	Deceimal between 0 and 1 specifying the proportion of "burn-in" cycles to omit from the plots. If missing, the first 20% will be omitted.
col	A vector of colors for the curves, which will be recycled if too short. Not mandatory.
out	Direct output to a PDF, EPS or image file. Format is a named list whose first argument, type is one of the following character vectors: "pdf", "eps" (maps to postscript), "png", "tiff", "jpeg", or "bmp". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is NA which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.
...	Additional R plotting parameters.

Value

Plots a panel with the following windows: -2 times the log-likelihood at each cycle, gamma/lambda at each cycle; Akaike Information Criterion at each cyle and Bayesian (Schwartz) Information Criterion at each cycle, the mean parameter values at each cycle (normalized to starting values); the normalized standard deviation of the population distribution for each parameter at each cycle; and the normalized median parameter values at each cycle.

Author(s)

Michael Neely

See Also

makeCycle, plot, par, axis

Examples

```
data(PMex1)
plot(cycle.1)
plot(cycle.1,omit=0)
```

---

plot.PMfinal *Plot Pmetrics Final Cycle Parameter Value Distributions*

Description

Plot PMfinal objects

Usage

```
## S3 method for class 'PMfinal'
plot(x, formula, include, exclude, ref = T, cex.lab = 1.2,
  col, col.ref, alpha.ref = 0.5, pch, cex, lwd, lwd.ref, density = F,
  scale = 20, bg, standard = F, probs = c(0.05, 0.25, 0.5, 0.75, 0.95),
  legend = T, grid = T, xlab, ylab, xlim, ylim, out = NA, ...)
```

Arguments

x	The name of an *PMfinal* data object generated by makeFinal
formula	An optional formula of the form y ~ x, where y and x are two model parameters to plot in a 3-dimensional bivariate plot. See details.
include	A vector of subject IDs to include in a Bayesian posterior marginal parameter distribution plot, e.g. c(1:3,5,15). Only relevant for Bayesian posterior plots generated by formula values of the form *prob~par*, where *par* is a parameter in the model.
exclude	A vector of subject IDs to exclude in a Bayesian posterior marginal parameter distribution plot, e.g. c(4,6:14,16:20). Only relevant for Bayesian posterior plots generated by formula values of the form *prob~par*, where *par* is a parameter in the model.
ref	Boolean operator to include (if TRUE which is the default) the population marginals in posterior marginal plot as reference.
cex.lab	Size of the plot labels for any univariate or bivariate marginal plot.
col	This parameter will be applied to the histogram lines of a univariate marginal plot, or the central point of a bivariate plot and is "red" by default for the former, and "white" for the latter.
col.ref	Color of reference population marginals included in posterior marginal plots.
alpha.ref	Alpha value for transparency of reference marginals. Default is 0.5, with 0=invisible and 1=opaque.
pch	The plotting character for points in bivariate plots. Default is a cross (pch=3).
cex	The size of the points in bivariate plots
lwd	Width of the histogram lines in the univariate marginal parameter distributions or the thickness of the central points and lines around points in bivariate NPAG plots or around quantiles in the bivariate IT2B plots.
lwd.ref	Width of histogram lines for population marginals included in posterior marginal plots.
density	Boolean operator to plot a kernel density function overlying the histogram of a univarite marginal parameter distribution from NPAG; the default is False. See density. Ignored for IT2B output.
scale	How large to scale the points in a bivariate NPAG plot, relative to their probability. Ignored for IT2B output.
bg	Background fill for points in bivariate NPAG plot. Ignored for IT2B output.
standard	Standardize the normal parameter distribution plots from IT2B to the same scale x-axis. Ignored for NPAG output.
probs	Vector of quantiles to plot on bivariate IT2B plot. Ignored for NPAG plot.
legend	Boolean operator for default if True or list of parameters to be supplied to legend function to plot quantile legend on bivariate IT2B plot. Ignored for NPAG plot.
grid	Boolean operator to plot a grid on either a bivariate NPAG or IT2B plot.
xlab	Define x-axis label for bivariate NPAG or IT2B plot. Default is the name of the plotted x-variable.
ylab	Define y-axis label for bivariate NPAG or IT2B plot. Default is the name of the plotted y-variable.
xlim	Limits for the x-axis in a bivariate NPAG or IT2B plot. Default is the range of the x-variable.

ylim	Limits for the y-axis in a bivariate NPAG or IT2B plot. Default is the range of the y-variable.
out	Direct output to a PDF, EPS or image file. Format is a named list whose first argument, type is one of the following character vectors: "pdf", "eps" (maps to postscript), "png", "tiff", "jpeg", or "bmp". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is NA which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.
...	Other parameters as found in plot.default.

Details

If formula is omitted, this will generate a marginal plot for each parameter. For NPAG data, this will be a histogram of marginal values for each parameter and the associated probability of that value. For IT2B, this will be a series of normal distributions with mean and standard deviation equal to the mean and standard deviation of each parameter marginal distribution, and the standard deviation and 95 indicated at the bottom of each plot. IF formula IS specified, this will generate one of two plots. Specifying "prob" as the y-value vs. a parameter will generate a marginal plot of Bayesian posterior parameter distributions for included/excluded subjects. For example, prob~CL will plot Bayesian posterior distributions for CL for each included/excluded subject.

On the other hand, if formula is two parameters, e.g. CL~V, this will generate a bivariate plot. For NPAG data, it will be support point with size proportional to the probability of each point. For IT2B, it will be an elliptical distribution of a bivariate normal distribution centered at the mean of each plotted variable and surrounding quantiles of the bivariate distribution plotted in decreasing shades of grey.

Value

Plots the object.

Author(s)

Michael Neely

See Also

makeFinal, plot, par, axis

Examples

```
data(PMex1)
plot(final.1)
```

plot.PMmatrix *Plot PMmatrix Time-Output Data*

Description

plot.PMmatrix plots *PMmatrix* objects

Usage

```
## S3 method for class 'PMmatrix'
plot(x, include, exclude, pred = NULL, icen = "median",
  mult = 1, outeq, group, block = 1, layout = c(3, 3), log = F,
  pch = NA, errbar = F, doses = F, tad = F, join = T, grid,
  ident = F, overlay = T, main, xlim, ylim, xlab = "Time (h)",
  ylab = "Observation", col, col.pred, cex = 1, legend, out = NA, ...)
```

Arguments

x	The name of an *PMmatrix* data object read by PMreadMatrix
include	A vector of subject IDs to include in the plot, e.g. c(1:3,5,15)
exclude	A vector of subject IDs to exclude in the plot, e.g. c(4,6:14,16:20)
pred	The name of a population or posterior prediction object read by makePop or makePost, respectively
icen	Only relevant for PMpost or PMpop objects which have predictions based on median or mean of each subject's Bayesian posterior parameter distribution. Default is "median", but could be "mean".
mult	Multiplication factor for y axis, e.g. to convert mg/L to ng/mL
outeq	A vector of output equation(s) to plot; if missing, plot all. E.g. outeq=1, outeq=2, outeq=c(1,3).
group	Quoted name of a covariate in data by which to distinguish groups with color in the plot. Note that if covariates do not have values on observation rows, those observations will be unable to be grouped. Grouping is only applicable if outeq is specified; otherwise there would be a confusing mix of colors for groups and output equations.
block	Which block to plot, where a new block is defined by dose resets (evid=4); default is 1.
layout	If overlay is False, this parameter specifies the number of plots per page.
log	Boolean operator to plot in log-log space; the default is False
pch	Controls the plotting symbol for observations; default is NA which results in no symbol. Use 0 for open square, 1 for open circle, 2 for open triangle, 3 for cross, 4 for X, or 5 for a diamond. Other alternatives are "*" for asterisks, "." for tiny dots, or "+" for a smaller, bolder cross. These plotting symbols are standard for R (see par).
errbar	Either boolean (true/false) or a list. If assay error coefficients are included in the data file, setting this to True will plot error bars around each observation according to the standard deviation calculated from C0, C1, C2 and C3 in the data file. If C0, C1, C2, and C3 are missing in the data file, you can specify errbar to be a

named list, i.e. `list(c0=,c1=,c2=,c3=)`, where each value is a vector of length equal to the number of output equations. For example, with two output equations having coefficients of 0.1, 0.15, 0, 0 and 0.2, 0.1, -0.001, and 0, specify as `errbar=list(c0=c(0.1,0.2),c1=c(0.15,0.1),c2=c(0,-0.001),c3=c(0,0))`.

`doses` Boolean operator to include doses as small lines at the bottom of the plot. Infusions are correctly represented according to their duration. The default is `False`. This parameter is ignored if `overlay` is `True`.

`tad` Boolean operator to use time after dose rather than time after start. Default is `False`.

`join` Boolean operator to join observations by a straight line; the default is `True`.

`grid` Either a boolean operator to plot a reference grid, or a list with elements x and y, each of which is a vector specifying the native coordinates to plot grid lines; the default is `False`. For example, grid=list(x=seq(0,24,2),y=1:10). Defaults for missing x or y will be calculated by `axTicks`.

`ident` Boolean operator to plot points as ID numbers in overlay plots; the default is `False`. Ignored if `overlay` is false. This option is useful to identify outliers.#'

`overlay` Boolean operator to overlay all time concentration profiles in a single plot. The default is `True`.

`main` An optional parameter to specify the title for plot(s). If `overlay` is `False`, the default will be the subject identification. If `overlay` is `True`, the default is blank. To omit a title from a non-overlaid plot, use the syntax `main=""`.

`xlim` Optional to specify the limits for the x axis.

`ylim` Optional to specify the limits for the y axis.

`xlab` Label for the x axis. Default is "Time (h)"

`ylab` Label for the y axis. Default is "Observation"

`col` A vector of color names to be used for output equation or group coloring. If the length of `col` is too short, values will be recycled.

`col.pred` A vector of color names to be used for prediction (post or pop) coloring. Default is the same as `col`.

`cex` Size of the plot symbols.

`legend` Either a boolean operator or a list of parameters to be supplied to the `legend` function (see its documentation). If `False` or missing, a legend will not be plotted. If `True`, the default legend parameters will be used, as documented in that function, with exceptions as noted in *Details*.

`out` Direct output to a PDF, EPS or image file. Format is a named list whose first argument, `type` is one of the following character vectors: "pdf", "eps" (maps to `postscript`), "png", "`tiff`", "`jpeg`", or "`bmp`". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is `NA` which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.

`...` Other parameters as found in `plot.default`.

Details

This function will plot raw and fitted time and concentration data with a variety of options. For the legend, defaults that are different that the standard are:

- x Default "topright"
- legend Default will be factor label names if `group` is specified and valid; otherwise "Output 1, Output 2,...Output n", where *n* is the number of output equations. This default can be overridden by a supplied character vector of output names.
- fill The color of each group/output as specified by the default color scheme or `col`
- bg Default "white"

Value

Plots the object.

Author(s)

Michael Neely

See Also

`PMreadMatrix`, `plot`, `par`, `axis`

Examples

```
data(PMex1)
plot(mdata.1)
```

---

plot.PMnpde *Plot Pmetrics normalized prediction distribution errors*

Description

Plots PMnpde objects

Usage

```
## S3 method for class 'PMnpde'
plot(x, outeq = 1, ...)
```

Arguments

`x`	The name of an *PMnpde* list object made by `makeNPDE`
`outeq`	Plot the NPDE or VPC for which output equation. Default is 1.
`...`	Other non-standard and standard R graphical parameters to pass to plot.NpdeObject (see details).

Details

This function is wrapper around the plot.NpdeObjects invisible method of Comets et al in the npde package for R. Full documentation is available at http://www.npde.biostat.fr.

Plot arguments which may be passed on to the npde plot function via the ... argument include:

- `plot.type` Control the type of plot. The default is "default".
  - `default` Combines 4 plots below: QQ-plot, hist, x.scatter, pred.scatter
  - `data` Plots the observed data in the dataset
  - `x.scatter` Scatterplot of the npde versus the predictor X (e.g. time)
  - `pred.scatter` Scatterplot of the npde versus the population predicted values
  - `vpc` Plots a Visual Predictive Check
  - `ecdf` Empirical distribution function of the npde (optionally pd or npd)
  - `hist` Histogram of the npde (optionally pd or npd)
  - `qqplot` QQ-plot of the npde versus its theoretical distribution (optionally pd or npd)
- `frame.plot` If TRUE, a box is drawn around the current plot. Default is TRUE.
- `xlog` If TRUE, x axis will be log scale. Default FALSE.
- `ylog` If TRUE, y axis will be log scale. Default FALSE.
- `ilist` List of subject numbers to include in the individual plots. Default is 1:N.
- `box` If TRUE, boxplots are produced instead of scatterplots. Default is FALSE.
- `pch.pobs` Plot character for observations. Default 20 (dot).
- `col.pobs` Color for observations. Default is steelblue4.
- `col.lobs` Color for lines joining observations. Default is steelblue4.
- `lty.lobs` Type for lines joining observations. Default is 1 (solid).
- `lwd.lobs` Width for lines joining observations. Default is 1.
- `col.abline` Color of the horizontal/vertical lines added to the plots. Default is DarkBlue.
- `lty.abline` Type of the lines added to the plots. Default is 2 (dashed).
- `wd.abline` Width of the lines added to the plots. Default is 2.
- `col.fillpi` Color used to fill histograms and prediction bands. Default is slategray1.
- `col.fillmed` Color used to fill prediction band on the median (VPC, npde). Default is pink.
- `col.lmed` Color used to plot the predicted median (VPC, npde). Default is indianred4.
- `col.lpi` Color used to plot lower and upper quantiles. Default is slategrey4.
- `lty.lmed` Line type used to plot the predicted median (VPC, npde). Default is 2 (dashed).
- `lty.lpi` Line type used to plot lower and upper quantiles. Default is 2 (dashed).
- `lwd.lmed` Line width used to plot the predicted median (VPC, npde). Default is 1.
- `lwd.lpi` Line width used to plot lower and upper quantiles. Default is 1.
- `bands` Whether prediction intervals should be plotted. Default is TRUE.
- `approx.pi` If TRUE, samples from N (0, 1) are used to plot prediction intervals, while if FALSE, prediction bands are obtained using npde computed for the simulated data. Default is TRUE.
- `vpc.method` Method used to bin points (one of "equal", "width", "user" or "optimal"); at least the first two letters of the method need to be specified. Default is "equal".
- `vpc.bin` Number of binning intervals. Default is 10.
- `vpc.interval` Size of interval. Default is 0.95.

- vpc.breaks Vector of breaks used with user-defined breaks when vpc.method="user"). Default is NULL.
- vpc.extreme Can be set to a vector of 2 values to fine-tune the behaviour of the binning algorithm at the boundaries; specifying c(0.01,0.99) with the "equal" binning method and vpc.bin=10 will create 2 extreme bands containing 1 X-interval, then divide the region within the two bands into the remaining 8 intervals each containing the same number of data; in this case the intervals will all be equal except for the two extreme intervals, the size of which is fixed by the user; complete fine-tuning can be obtained by setting the breaks with the vpc.method="user". Default is NULL.
- pi.size Width of the prediction interval on the quantiles. Default is 0.95.
- vpc.lambda Value of lambda used to select the optimal number of bins through a penalised criterion. Default is 0.3.
- vpc.beta Value of beta used to compute the variance-based criterion (Jopt,beta(I)) in the clustering algorithm. Default is 0.2.
- bands.rep Number of simulated datasets used to compute prediction bands. Default is 200.

Value

Plots the object.

Author(s)

Michael Neely

See Also

makeNPDE, plot, autonpde, par

---

plot.PMop *Plot Pmetrics Observed vs. Predicted Objects*

Description

Plot PMop objects

Usage

```
## S3 method for class 'PMop'
plot(x, include, exclude, pred.type = "post",
  icen = "median", outeq = 1, mult = 1, resid = F, log = F,
  square = T, ref = T, lowess = F, reg = T, grid, ident = F,
  ci = 0.95, cex = 1, cex.lab = 1.2, x.stat = 0.4, y.stat = 0.1,
  col.stat = "black", cex.stat = 1.2, lwd = 2, col = "red", xlim, ylim,
  xlab, ylab, out = NA, ...)
```

Arguments

x	The name of an *PMop* data object generated by makeOP.
include	A vector of subject IDs to include in the plot, e.g. c(1:3,5,15)
exclude	A vector of subject IDs to exclude in the plot, e.g. c(4,6:14,16:20)
pred.type	Either 'post' for a posterior object or 'pop' for a population object. Default is 'post'.
icen	Can be either "median" for the predictions based on medians of pred.type parameter value distributions, or "mean". Default is "median".
outeq	Output equation number. Default is 1.
mult	Multiplication factor for x and y axes, e.g. to convert mg/L to ng/mL. Ignored for residual plots.
resid	Boolean operator to generate a plot of weighted prediction error vs. prediction, a plot of weighted prediction error vs. time, and histogram plot of the weighted prediction errors, with overlying normal distribution of the same mean and variance if ref is true, and a P-value for the Kolmogorov-Smirnov test for non-normality if reg is true. The default is False.
log	Boolean operator to plot in log-log space. This parameter is ignored for residual plots. The default is False
square	Boolean operator to force a observed vs. predicted plots to be square with equal x and y limits. This parameter is ignored for residual plots. The default is True
ref	Boolean operator to draw a reference line of slope 1 in observed vs. predicted plots and slope 0 in residual plots, or a reference normal distribution in residual histogram; the default is True
lowess	Boolean operator to draw a lowess regression line in observed vs. predicted or residual plots; the default is False
reg	Boolean operator to draw a linear regression line and print regression statistics on the plot. For weighted residual plots, it will print the mean weighted prediction error with P value for difference from 0, and the standard deviation of the weighted prediction errors, as well as the probability that the distribution of weighted residuals is not different from normal by the Kolmogorov-Smirnov test. The default is True.
grid	Either a boolean operator to plot a reference grid, or a list with elements x and y, each of which is a vector specifying the native coordinates to plot grid lines; the default is False. For example, grid=list(x=seq(0,24,2),y=1:10). Defaults for missing x or y will be calculated by axTicks. For residual plots, list values for grid will be interpreted as True, i.e. custom grid lines are not allowed.
ident	Boolean operator to plot points as ID numbers; the default is False. This option is useful to identify outliers.
ci	The confidence interval for the linear regression parameter estimates; the default is 0.95.
cex	Size of the plot symbols.
cex.lab	Size of the plot labels.
x.stat	Horizontal position to plot the regression or residual statistics; the units are relative to the origin, i.e. extreme left is 0 and extreme right is 1.
y.stat	Vertical position to plot the regression or residual statistics; the units are relative to the origin, i.e. extreme bottom is 0 and extreme top is 1.

col.stat	Color of the text for the regression or residual statistics.
cex.stat	Size of the text for the regression or residual statistics
lwd	Width of the various regression or reference lines (reference, linear regression, or lowess regression)
col	This parameter will be applied to the plotting symbol and is "red" by default.
xlim	Limits of the x-axis as a vector, e.g. c(0,1). It does not need to be specified, but can be.
ylim	Analogous to xlim
xlab	Label for the x-axis. If missing, will default to "Observed".
ylab	Label for the y-axis. If missing, will default to "Predicted".
out	Direct output to a PDF, EPS or image file. Format is a named list whose first argument, type is one of the following character vectors: "pdf", "eps" (maps to postscript), "png", "tiff", "jpeg", or "bmp". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is NA which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.
...	Other parameters as found in plot.default.

Value

Plots the object.

Author(s)

Michael Neely

See Also

makeOP, plot, par, axis

Examples

```
data(PMex1)
plot(op.1)
```

---

plot.PMpta *Plot PMpta Percent Target Attainment objects*

Description

Plots PMpta objects

Usage

```
## S3 method for class 'PMpta'
plot(x, include, exclude, plot.type = "pta", log = T, pch,
  grid, xlab, ylab, col, lty, lwd = 4, legend = T, ci = 0.9, out = NA,
  ...)
```

Arguments

x	The name of an *PMpta* data object read by makePTA
include	A vector of simulations (regimens) to include in the plot, e.g. c(1,3)
exclude	A vector of simulations (regimens) in the plot, e.g. c(2,4:6)
plot.type	Character vector controlling type of plot. Default is "pta", which plots proportion with success on the y-axis and target on the x-axis. The other choice is "pdi", which plots the median pdi (pharmacodynamic index), e.g. AUC/MIC, on the y-axis, and target on the x-axis.
log	Boolean operator to plot x-axis in logarithmic scale; the default is True
pch	Vector of integers which control the plotting symbol for each regimen curve; the default is 1:nsim. NA results in no symbol. Use 0 for open square, 1 for open circle, 2 for open triangle, 3 for cross, 4 for X, or 5 for a diamond. Other alternatives are "*" for asterisks, "." for tiny dots, or "+" for a smaller, bolder cross. These plotting symbols are standard for R (see par).
grid	Either a boolean operator to plot a reference grid, or a list with elements x and y, each of which is a vector specifying the native coordinates to plot grid lines; the default is False. For example, grid=list(x=seq(0,24,2),y=1:10). Defaults for missing x or y will be calculated by axTicks.
xlab	Label for the x axis. Default is "MIC"
ylab	Label for the y axis. Default is "Proportion with success"
col	A vector of color names to be used for each regimen plotted. If the length of col is too short, values will be recycled.
lty	A vector of line types to be used for each regimen plotted. If the length of lty is too short, values will be recycled.
lwd	Line width, with default of 4.
legend	Either a boolean operator or a list of parameters to be supplied to the legend function (see its documentation). If False, a legend will not be plotted. If True (the default), the default legend parameters will be used, as documented in that function, with exceptions as noted in *Details*.
ci	Confidence interval around curves on pdi plot, on scale of 0 to 1. Default is 0.9.
out	Direct output to a PDF, EPS or image file. Format is a named list whose first argument, type is one of the following character vectors: "pdf", "eps" (maps to postscript), "png", "tiff", "jpeg", or "bmp". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is NA which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.
...	Other parameters as found in plot.default.

Details

This function will plot the percent target attainment for objects made with the makePTA function. For the legend, defaults that are different that the standard are:

- x Default "topright"
- legend Default will be the labeled regimen names supplied during makePTA, or if missing, "Regimen 1, Regimen 2,...Regimen n", where *n* is the number of regimens in the PMpta object. This default can be overridden by a supplied character vector of regimen names.
- col The color of each Regimen plot as specified by the default color scheme or `col`
- pch The plotting character for each Regimen plot as specified by the default plotting characters or `pch`
- lty The line type of each Regimen plot as specified by the default line types or `lty`
- bg Default "white"

Value

Plots the object.

Author(s)

Michael Neely

See Also

makePTA, plot, par, axis

---

plot.PMsim *Plot Pmetrics Simulation Objects*

Description

Plots *PMsim* objects with the option to perform a visual and numerical predictive check

Usage

```
## S3 method for class 'PMsim'
plot(x, mult = 1, log = T, probs = c(0.05, 0.25, 0.5,
  0.75, 0.95), binSize = 0, outeq = 1, pch = NA, join = T,
  x.qlab = 0.4, cex.qlab = 0.8, pos.qlab = 1, ci = 0.95,
  cex.lab = 1.2, xlab = "Time (h)", ylab = "Output", xlim, ylim, obs,
  grid, ocol = "blue", add = F, out = NA, ...)
```

Arguments

`x`	The name of an *PMsim* data object generated by SIMparse
`mult`	Multiplication factor for y axis, e.g. to convert mg/L to ng/mL
`log`	Boolean operator to plot in log-log space; the default is `False`
`probs`	Vector of quantiles to plot; if set to `NA`, all simulated profiles will be plotted, and numerical predictive checking will be suppressed

binSize	Width of binning interval for simulated concentrations, in time units, e.g. hours. For example, a binSize of 0.5 will pull all simulated concentrations +/- 0.5 hours into the same time. This is useful for plotting PMsim objects made during makeNPDE. The default is 0, i.e. no binning.
outeq	Which output equation to plot if more than 1
pch	Controls the plotting symbol for observations; default is NA which results in no symbol. Use 0 for open square, 1 for open circle, 2 for open triangle, 3 for cross, 4 for X, or 5 for a diamond. Other alternatives are "*" for asterisks, "." for tiny dots, or "+" for a smaller, bolder cross. These plotting symbols are standard for R (see par).
join	Boolean operator to join observations by a straight line; the default is True.
x.qlab	Proportionate value of x-axis at which to draw the quantile labels; 0 is left, 1 is right. The default is 0.4.
cex.qlab	Size of the quantile labels.
pos.qlab	This allows more refined positioning of the quantile labels. It takes standard R values: 1, below; 2, left; 3, above; 4, right.
ci	Width of confidence interval bands around simulated quantiles, from 0 to 1. If 0, or *nsim*<100, will not plot. Default is 0.95, i.e. 95th percentile with tails of 2.5 percent above and below excluded.
cex.lab	Size of the plot labels.
xlab	Label for x-axis; default is "Time"
ylab	Label for y-axis; default is "Output"
xlim	Limits of the x-axis as a vector, e.g. c(0,1). It does not need to be specified, but can be.
ylim	Analogous to xlim
obs	The name of an *makeOP* data object generated by makeOP. If specified, the observations will be overlaid upon the simulation plot enabling a visual predicitve check. In this case, a list object will be returned with two items: $npc containing the quantiles and probability that the observations are below each quantile (binomial test); and $simsum, the times of each observation and the value of the simulated quantile with upper and lower confidence intervals at that time.
grid	Either a boolean operator to plot a reference grid, or a list with elements x and y, each of which is a vector specifying the native coordinates to plot grid lines; the default is False. For example, grid=list(x=seq(0,24,2),y=1:10). Defaults for missing x or y will be calculated by axTicks.
ocol	Color for observations
add	Boolean operator, if True will add lines to existing plot
out	Direct output to a PDF, EPS or image file. Format is a named list whose first argument, type is one of the following character vectors: "pdf", "eps" (maps to postscript), "png", "tiff", "jpeg", or "bmp". Other named items in the list are the arguments to each graphic device. PDF and EPS are vector images acceptable to most journals in a very small file size, with scalable (i.e. infinite) resolution. The others are raster images which may be very large files at publication quality dots per inch (DPI), e.g. 800 or 1200. Default value is NA which means the output will go to the current graphic device (usually the monitor). For example, to output an eps file, out=list("eps") will generate a 7x7 inch (default) graphic.
...	Other parameters as found in plot.default.

Details

Simulated observations are plotted as quantiles on the y-axis vs. time on the x.axis. If measured observations are included, a visual and numerical predictive check will be performed.

Value

Plots the simulation object. If `obs` is included, a list will be returned with the folowing items:

`npc`	A dataframe with three columns: quantile, prop.less, pval. *quantile* are those specified by the `prob` argument to the plot call; *prop.less* are the proportion of simulated observations at all times less than the quantile; *pval* is the P-value of the difference in the prop.less and quantile by the beta-binomial test.
`simsum`	A dataframe with the quantile concentration at each simulated time, with lower and upper confidence intervals
`obs`	A dataframe similar to an PMop object made by `makeOP` with the addition of the quantile for each observation

Author(s)

Michael Neely

See Also

`SIMparse`, `plot`, `par`, `axis`

`PMbuild` *Build Pmetrics*

Description

`PMBuild` will ensure all dependent packages are installed and compile Fortran source code for permanent Pmetrics modules

Usage

```
PMbuild()
```

Author(s)

Michael Neely

PMcheck *Check Pmetrics Inputs for Errors*

Description

This function will check a .csv file or a data frame containing a previously loaded .csv file (the output of PMreadMatrix for errors which would cause the analysis to fail. If a model file is provided, and the data file has no errors, it will also check the model file for errors.

Usage

```
PMcheck(data, model, fix = F, quiet = F)
```

Arguments

data	The name of a Pmetrics .csv matrix file in the current working directory, the full path to one not in the current working directory, or a data.frame containing the output of a previous PMreadMatrix command.
model	The filename of a Pmetrics model file in the current working directory. This parameter is optional. If specified, and the data object has no errors, the model file will be evaluated.
fix	Boolean operator; if TRUE, Pmetrics will attempt to fix errors in the data file. Default is FALSE.
quiet	Boolean operator to suppress printed output. Default is false.

Details

Either a filename or a data object in memory are accepted as data. The format of the .csv matrix file is fairly rigid. It must have the following features. Text is case-sensitive.

- A header in row 1 with the appropriate version, currently "POPDATA DEC_11"
- Column headers in row 2. These headers are: #ID, EVID, TIME, DUR, DOSE, ADDL, II, INPUT, OUT, OUTEQ, C0, C1, C2, C3.
- No cell should be empty. It should either contain a value or "." as a placeholder.
- Columns after C3 are interpreted as covariates.
- All subject records must begin with TIME=0.
- All dose events (EVID=1) must have entries in ID, EVID, TIME, DUR, DOSE and INPUT. ADDL and II are optional, but if ADDL is not 0 or missing, then II is mandatory.
- All observation events (EVID=0) must have entries in ID, EVID, TIME, OUT, OUTEQ. If an observation is missing, use *-99*; otherwise use a "." as a placeholder in cells that are not required (e.g. INPUT for an observation event).
- If covariates are present in the data, there must be an entry for every covariate at time 0 for each subject.
- All covariates must be numeric.
- All times within a subject ID must be monotonically increasing.
- All subject IDs must be contiguous.
- All rows must have EVID and TIME values.

- All columns must be numeric except ID which may be alpha-numeric.
- All subjects must have at least one observation, which could be missing, i.e. -99.

To use this function, see the example below.

After running PMcheck and looking at the errors in the errors.xlsx file, you can fix the errors manually directly in the errors.xlsx file and resave it as a .csv file. Alternatively, you could then try to fix the problem(s) with `mdata2 <- PMcheck(mdata,fix=T)`. Note that we are now returning a PMmatrix data object called mdata2 (hopefully cleaned of errors) rather than the PMerr object returned when `fix=FALSE`. Pmetrics handles each of the errors in the following ways.

- If the columns are simply out of order, they will be reordered. If some are missing, the fix must be done by the user, i.e. manually.
- All id and covariate values are truncated to 11 characters.
- Missing observations are set to -99 (not ".").
- Incomplete dose records are flagged for the user to fix manually.
- Incomplete observation records are flagged for the user to fix manually.
- Subjects without an EVID=1 as first event are flagged for the user to fix manually.
- Subjects with TIME != 0 as first event have dummy dose=0 events inserted at time 0.
- Subjects with a missing covariate at time 0 are flagged for the user to fix manually.
- Non-numeric covariates are converted to numeric (via factor).
- Non-ordered times are sorted within a subject if there are no EVID=4 events; otherwise the user must fix manually.
- Non-contiguous subject ID rows are combined and sorted if there are no EVID=4 events; otherwise the user must fix manually.
- Rows missing an EVID are assigned a value of 0 if DOSE is missing, 1 otherwise.
- Rows missing a TIME value are flagged for the user to fix manually.
- Columns that are non-numeric which must be numeric are flagged for the user to fix manually. These are EVID, TIME, DUR, DOSE, ADDL, II, INPUT, OUT, OUTEQ, C0, C1, C2, and C3. Covariate columns are fixed separately (see above).

Value

If `fix=TRUE`, then `PMcheck` returns a PMmatrix data object which has been cleaned of errors as much as possible, displaying a report on the console. If `fix=FALSE`, then `PMcheck` creates a file in the working directory called "errors.xlsx". This file can be opened by Microsoft Excel or any other program that is capable of reading .xlsx files. This file contains highlighted areas that are erroneous, with clarifying comments. You can correct the errors in the file and then re-save as a .csv file.

When `fix=FALSE`, the function also returns a list of objects of class *PMerr*. Each object is itself a list whose first object (`$msg`) is a character vector with "OK" plus a brief description if there is no error, or the error. The second object (`$results`) is a vector of the row numbers that contain that error.

`colorder`	The first 14 columns must be named id, evid, time, dur, dose, addl, ii, input, out, outeq, c0, c1, c2, and c3 in that order.
`maxcharCol`	All column names should be less than or equal to 11 characters.
`maxcharID`	All id values should be less than or equal to 11 characters.
`missEVID`	Ensure that all rows have an EVID value.
`missTIME`	Ensure that all rows have a TIME value.

doseDur	Make sure all dose records are complete, i.e. contain a duration.
doseDose	Make sure all dose records are complete, i.e. contain a dose.
doseInput	Make sure all dose records are complete, i.e. contain an input number.
obsOut	Make sure all observation records are complete, i.e. contain an output.
obsOuteq	Make sure all observation records are complete, i.e. contain and outeq number.
T0	Make sure each subject's first time=0.
covT0	Make sure that there is an non-missing entry for each covariate at time=0 for each subject.
timeOrder	Ensure that all times within a subject ID are monotonically increasing.
contigID	Ensure that all subject IDs are contiguous.
nonNum	Ensure that all columns except ID are numeric.
noObs	Ensure that all subjects have at least one observation, which could be missing, i.e. -99.

Author(s)

Michael Neely and Patrick Nolain

See Also

PMwriteMatrix, PMreadMatrix

Examples

```
## Not run:
data(PMex3)
err <- PMcheck(badData)
#look at the errors.xlsx file in the working directory
#try to automatically fix what can be fixed
goodData <- PMcheck(badData,fix=T)
PMcheck(goodData)
#you have to fix manually problems which require data entry

## End(Not run)
```

PMcheckMatrix *Deprecated functions.*

Description

The following functions are deprecated in Pmetrics.

Usage

```
PMcheckMatrix()
```

Author(s)

Michael Neely

PMcode *Pmetrics GUI Tutor*

Description

Learn Pmetrics R code with user friendly graphical interfaces in the default browser.

Usage

```
PMcode(func)
```

Arguments

func	Quoted name of a function family used in Pmetrics. Currently, these are limited to "run", for NPrun, ITrun and "plot". For the first two, make sure that the model and data files are in your working directory before calling the function.

Details

PMcode provides a graphical user interface to learn many of the Pmetrics functions and their arguments using the Shiny package. A graphical user interface will launch in the default browser. This GUI enables a point and click approach to generating Pmetrics code (which can be pasted into the R script) and plot previews. The idea is for users to learn the R code in an intuitive and easier manner. There are more options available for Pmetrics functions that are served by the GUI, but it is sufficiently powerful to serve basic needs. To stop the shiny browser GUI, click the stop buttton in Rstudio (upper left corner of console window) or ESC or CTRL-C may work when the R window is active.

Value

Nothing is returned, but the user interface is launched in the default browser. Appropriate R code to execute Pmetrics commands is generated depending on defaults and user-selected input. For plotting, the resulting plot is previewed directly in the browser.

Author(s)

Michael Neely

PMcompare *Compare NPAG or IT2B runs*

Description

Compare NPAG or IT2B runs

Usage

```
PMcompare(x, y, ..., icen = "median", outeq = 1, plot = F)
```

Arguments

x	The run number of the first object you wish to compare. This should be a folder in your working directory. To avoid confusion, this function does not use objects already loaded with PMload. This will serve as the reference output for P-value testing (see details).
y	The run number of the second object to compare.
icen	Can be either "median" for the predictions based on medians of pred.type parameter value distributions, or "mean". Default is "median".#'
outeq	Number of the output equation to compare; default is 1
plot	Boolean operator selecting whether to generate observed vs. predicted plots for each data object as in plot.PMop
...	Additional run numbers to compare. See details. Also, parameters to be passed to plot.PMop if plot is true as well as to mtsknn.eq. Order does not matter.

Details

Objects can be specified separated by commas, e.g. PMcompare(1,2,3) followed by any arguments you wish to plot.PMop, mtsknn.eq. P-values are based on comparison using the nearest neighbors approach if all models are non-parametrics. Models may only be compared on parameters that are included in the first model. The P-value is the comparison between each model and the first model in the list. Missing P-values are when a model has no parameter names in common with the first model, and for the first model compared to itself, or when models from IT2B runs are included. Significant P-values indicate that the null hypothesis should be rejected, i.e. the joint distributions between the two compared models are significantly different.

Value

A data frame with the following objects for each model to analyze:

run	The run number of the data
type	NPAG or IT2B data
nsub	Number of subjects in the model
nvar	Number of random parameters in the model
par	Names of random parameters
cycles	Number of cycles run
converge	Boolean value if convergence occurred.
ll	Final cycle -2*Log-likelihood
aic	Final cycle Akaike Information Criterion
bic	Final cycle Bayesian (Schwartz) Information Criterion
popBias	Bias, or mean weighted prediction error of predictions based on population parameters minus observations
popImp	Imprecision, or bias-adjusted mean weighted squared error of predictions based on population parameters minus observations
popPerRMSE	Percent root mean squared error of predictions based on population parameters minus observations
postBias	Bias, or mean weighted prediction error of predictions - observations based on posterior parameters

postImp	Imprecision, or bias-adjusted mean weighted squared error of predictions - observations based on posterior parameters
postPerRMSE	Percent root mean squared error of predictions based on posterior parameters minus observations
pval	P-value for each model compared to the first. See details.

Author(s)

Michael Neely

See Also

PMload, plot.PMop, mtsknn.eq

PMex1 *Example NPAG Output*

Description

Example dataset from an NPAG run.

Usage

```
PMex1
```

Format

An R data file containing the output generated at the end of a successful NPAG run.

- NPdata.1 made by NPparse
- final.1 made by makeFinal
- cycle.1 made by makeCycle
- op.1 made by makeOP
- cov.1 made by makeCov
- pop.1 made by makePop
- post.1 made by makePost
- mdata.1 the original data file as read by PMreadMatrix

Details

The run consisted of a model with an absorptive compartment and a central compartment. There were 4 parameters in the model: lag time of absorption (Tlag1), rate constant of absorption (Ka), volume (V) and rate constatn of elmination (Ke). Parameters were log transformed. There were 20 subjects in the dataset. The run was 100 cycles long and did not converge.

The input files for this run (ex.csv and model.txt) can be downloaded as a zip file from http://www.lapk.org/Pmetrics_install.php#examples.

Author(s)

Michael Neely

PMex2 *Example IT2B Output*

Description

Exmaple dataset from an IT2B run.

Usage

```
PMex2
```

Format

An R data file containing the output generated at the end of a successful IT2B run.

- ITdata.1 made by `ITparse`
- final.1 made by `makeFinal`
- cycle.1 made by `makeCycle`
- op.1 made by `makeOP`
- cov.1 made by `makeCov`
- mdata.1 the original data file as read by `PMreadMatrix`

Details

The run consisted of a model with an absorptive compartment and a central compartment. There were 4 parameters in the model: lag time of absorption (Tlag1), rate constant of absorption (Ka), volume (V) and rate constatn of elmination (Ke). Parameters were log transformed. There were 20 subjects in the dataset. The run was 20 cycles long and did converge.

The input files for this run (ex.csv and model.txt) can be downloaded as a zip file from http://www.lapk.org/Pmetrics_install.php#examples.

Author(s)

Michael Neely

PMex3 *Pmetrics data file with errors*

Description

Example dataset for an NPAG run, which has been corrupted with errors.

Usage

```
PMex3
```

Format

badData is a PMmatrix object as read by `PMreadMatrix`

Details

Errors include missing covariate on first line for subject 1, alphanumeric covariate for gender, and trailing dose for subject 1.

Author(s)

Michael Neely

PMFortranConfig *Read or define the Fortran compiler and command line template*

Description

`PMFortranConfig` will read or define the installed Fortran compiler and generate a command line template appropriate to the compiler.

Usage

```
PMFortranConfig(reconfig = F)
```

Arguments

`reconfig`	Default is `False`. If `True`, will allow user to change the previously specified compiler and template.

Details

Command line templates are defined for the following compilers: **gfortran**, **g95**, **Intel Visual**, and **Lahey**. Additionally, users may specify a custom command line template for any other compiler. Within the template <*exec*> is used as a placeholder for the filename of the executable file, and <*files*> as a placeholder for the files to compile and link, both of which will be defined at run time by the appropriate Pmetrics functions. The Pmetrics functions which use a Fortran compiler are `NPrun`, `ITrun`, `ERRrun`, and `SIMrun`.

Value

`PMFortranConfig` returns the compile command template specific to the chosen compiler.

Author(s)

Michael Neely

See Also

`NPrun`, `ITrun`,`ERRrun`, and `SIMrun`

PMgetCRCL *Add Jelliffe Creatinine Clearance*

Description

Gets creatinine clearance as estimated by the Jelliffe equation.

Usage

```
PMgetCRCL(mdata, idCol = "id", wtCol = "wt", maleCol = "male",
  ageCol = "age", scrCol = "scr", SI = F)
```

Arguments

mdata	A Pmetrics matrix data object
idCol	A character vector with the name of the id column in mdata. The default is "id".
wtCol	A character vector with the name of the weight column in data. The default is "wt".
maleCol	A character vector with the name of the gender column in mdata. Male should be 1 and female should be 0. The default is "male".
ageCol	A character vector with the name of the age column in mdata. The default is "age".
scrCol	A character vector with the name of the serum creatinine column in mdata. Default units are mg/dL, and the the default name is "scr".
SI	Boolean value, if true, will expect serum creatinine to be in micromol/L. Default is FALSE.

Details

The equation depends on age, sex, weight, and serum creatinine. ESS = wt * (29.3 - (0.203 * age)) for males ESS = wt * (25.1 - (0.175 * age)) for females scrAve = (Scr1 + Scr2) / 2 ESS_cor = ESS * (1.035 - (0.0337 * scrAve)) E = ESS_cor - 4 * wt * (Scr2 - Scr1) / (time2 - time1) CRCL = E / (14.4 * scrAve) in ml/min/1.73m^2

Value

A vector of length nrow(mdata) with Jelliffe CRCL values for every dose in mdata. Vector values for observation events in mdata are NA.

Author(s)

Michael Neely

PMload *Load Pmetrics NPAG or IT2B output*

Description

Loads all the data from an *NPAG* or *IT2B* run

Usage

```
PMload(run = 1, ...)
```

Arguments

run	The numerical value of the folder number containing the run results. This number will also be used to name objects uniquely by appending ".run", e.g. NPdata.1 or ITdata.1 if run=1. This parameter is 1 by default.
...	Additional runs to load if desired.

Value

The following objects are loaded into R.

NPdata/ITdata	List with all output from NPAG/IT2B
pop	NPAG only: Population predictions for each output equation
post	NPAG only: Individual posterior predictions for each output equation
final	Final cycle population support points and parameter summary statistics
cycle	Cycle log-likelihood, AIC, BIC, Gamma/lambda, and normalized parameter means, medians and SDs
op	List of observed vs. population and posterior predicted plots for each output equation
cov	Data frame of subject ID, covariate values, and Bayesian posterior parameter estimates
mdata	The original .csv data file used in the run
npde	If makeNPDE has been run after a run, this object will be added to the save data. It contains the information required to plot and analzye normalized prediction error discrepancies via the npde package of Comets et al
sim	If makeNPDE has been run after a run, this list object will be added to the save data. It contains the results of each subject in the dataset simulated n times (default 1000) using the final model population parameters. To plot the results of subject 3 from run 2, for example, use the form plot(sim.2[[3]])

.

Author(s)

Michael Neely

See Also

PMreport, NPparse, ITparse, makeFinal, makeCycle, makeOP, makeCov, makePop, makePost

PMmanual *Open user and function manuals.*

Description

Opens the Pmetrics User Manual and function libraries

Usage

```
PMmanual()
```

Details

Help for Pmetrics.

PMmatrixRelTime *Convert Absolute Dates and Times to Relative Hours*

Description

PMmatrixRelTime will convert absolute dates and times in a dataset into relative hours, suitable for Pmetrics analysis. Additionally, the user has the option to split subjects into pseudosubjects every time a dose reset (evid=4) is encountered.

Usage

```
PMmatrixRelTime(data, idCol = "id", dateCol = "date", timeCol = "time",
  evidCol = "evid", format = c("m/d/y", "h:m"), split = F)
```

Arguments

data	The name of an R data object.
idCol	A character vector with the name of the id column in data or the number of the id column, default is "id"
dateCol	A character vector with the name of the date column in data or the number of the date column, default is "date"
timeCol	A character vector with the name of the time column in data or the number of the time column, default is "time"
evidCol	A character vector with the name of the event id column in data or the number of the evid column, default is "evid"
format	Format of the date and time columns; default is m/d/y and h:m:s, as specified in the chron::chron function. Note the separators in each case (/ for dates and : for times). For dates, *m* is months in digits and can be one or two digits; *d* is the day of the month, again as one or two digits; *y* is the year in 2 or 4 digits. For times, all values can be one or two digits, but time is in 24-hour format, and *s* is required to avoid ambiguity.
split	If *true*, PMmatrixRelTime will split every id into id.block, where block is defined by a dose reset, or evid=4, e.g. id 1.1, 1.2, 1.3, 2.1, 3.1, 3.2.

Value

Returns a dataframe with columns [id, evid, relTime]. If `split=T` all evid values that were previously 4 will be converted to 1.

Author(s)

Michael Neely

See Also

`PMreadMatrix`

`PMmb2csv` *Convert Old .mb or USC*PACK Files to .csv Matrix File*

Description

`PMmb2csv` will convert old style, single drug .mb or USC*PACK files into a single .csv matrix file.

Usage

```
PMmb2csv(oldFiles, newFile = "data")
```

Arguments

`oldFiles`	A character vector of files in the current working directory to convert. This could be easily obtained with `list.files`.
`newFile`	A single character vector with the basename (without any file extension) of the new file to be created.

Details

IDs will be suffixed with .1 to .9 for <10 subjects, .01 to .99 for <100 subjects and .001 to .999 for <1000 subjects, as needed to ensure unique ID numbers.

Value

A new file will be created with the name equal to `newFile` and an extension of "csv".

Author(s)

Michael Neely

PMnews *Pmetrics changelog*

Description

See changelog for Pmetrics

Usage

```
PMnews(version = packageVersion("Pmetrics"))
```

Arguments

version	Default is the current version, otherwise a character string with the starting version you wish to see up to the current, e.g. "0.21". Use "all" for all versions.

Value

The changelog for the requested version.

Author(s)

Michael Neely

Examples

```
PMnews()
```

PMpatch *Download and install Pmetrics patches*

Description

Download and install Pmetrics patches from LAPK website

Usage

```
PMpatch()
```

Value

A Pmetrics patch which will be installed via source

Author(s)

Michael Neely

PMreadMatrix *Read a Pmetrics .csv Matrix Input File*

Description

PMreadMatrix reads an NPAG .csv matrix input file into R.

Usage

```
PMreadMatrix(file, skip = 1, sep = getPMoptions("sep"),
  dec = getPMoptions("dec"), quiet = F, ...)
```

Arguments

file	The name of the file to be loaded, including the full path if not in the current working directory (check with getwd).
skip	Skip *n* lines, with default set to 1.
sep	Delimiter between columns, which is a comma by default, but can be changed with setPMoptions.
dec	Decimal separator, which is a period by default, but can be changed with setPMoptions.
quiet	Default is *false*. If *true*, there will be no report to the console on the contents of file.
...	Other parameters to be passed to read.table

Details

The structure of a valid .csv file is fairly rigid. See PMcheckMatrix for details. Note that PMreadMatrix converts the column headers in the matrixfile from upper to lowercase for convenient referencing in R.

Value

PMreadMatrix returns a data.frame of class "PMmatrix" with one row per event and the following columns.

id	The id value for each event.
evid	The evid value for each event, with 0=observation, 1=dose, 4=dose reset, which resets the time to 0 and all compartment amounts to 0. Note that evid=2 and 3 are not currently implemented.
time	Relative time of the event in hours.
dur	Duration of the dose. If dose is instantaneous, e.g. an oral dose into an absorptive compartment, dur should be 0. Any values greater than 0 are interpreted to mean a constant infusion of that duration, equalling the dose.
dose	The dose. Be sure that the units are consistent with out.
addl	Optional number of additional doses to add at an interval specified in *ii*. The default if missing is 0. A value of -1 will cause steady state conditions to be approximated. Any value for *addl* other than 0 or missing requires input in *ii*.
ii	The interdose interval for *addl* doses or dosing at steady state.

input	The input number corresponding to dose.
out	The measured output, equivalent to "DV" in some other PK modeling software tools.
outeq	The number of the output equation specified in the model file which corresponds to the out value.
C0	Assay error polynomial coefficient, e.g. SD = C0 + C1*obs + C2*obs^2 + C3*obs^3
C1	See C0
C2	See C0
C3	See C0
...	Additional columns are interpreted to be covariates.

If the file is successfully read and quiet=F, the column headers of the scanned file will be reported to the console as a validation check.

Author(s)

Michael Neely

See Also

PMwriteMatrix, PMcheckMatrix, and plot.PMmatrix

PMreport *Summarize NPAG or IT2B Run*

Description

Generates a summary of a Pmetrics NPAG or IT2B run

Usage

```
PMreport(wd, icen = "median", type = "NPAG", parallel = F)
```

Arguments

wd	The working directory containing the NP_RFxxxx.TXT or IT_RFxxxx.TXT file
icen	Median (default), mean or mode of Bayesian posterior to be used to calculate predictions.
type	"NPAG" (default) or "IT2B" report type
parallel	Boolean parameter which indicates the type of run done. Default is FALSE for serial.

Details

Creates an HTML page and several files summarizing an NPAG or IT2B run. This report is generated automatically at the end of a successful run.

Value

Several files are placed in the wd

`NPAGreport.html or IT2Breport.html`
: An .html file containing a summary of all the results

`poppoints.csv`
: NPAG only: A .csv file containing the population support points and probabilities

`poparam.csv`
: A .csv file containing a summary of the population parameter values, including mean, standard deviation, coefficient of variation, variance, and median

`popcor.csv`
: A .csv file containing the population parameter correlation matrix

`popcov.csv`
: A .csv file containing the population parameter covariance matrix

`cycle.pdf`
: A .pdf file containing the run cycle information (see `plot.PMcycle`)

`cycle.png`
: A thumbnail of the run cycle information for the .html file

`final.pdf`
: A .pdf file containing the population final cycle information (see `plot.PMfinal`)

`final.png`
: A thumbnail of the population final cycle information for the .html file

`opx.pdf`
: One or more .pdf files, where *x* is the number of the output equation, each containing two observed vs. predicted plots: population and individual Bayesian posterior predictions (see `plot.PMop`)

`opx.png`
: One or more thumnails of the observed vs. predicted plots for the .html file

`NPAGout.Rdata or IT2Bout.Rdata`
: An R data file containing the output of `NPparse` or `ITparse`, `makeFinal`, `makeCycle`, `makeOP`, `makeCov`, `makePop`, `makePost`, and the data file for the run read by `PMreadMatrix`. This file can be loaded using `PMload`.

Author(s)

Michael Neely

`PMsave` *Save Pmetrics objects*

Description

Saves Pmetrics objects

Usage

```
PMsave(run, ..., quiet = F)
```

Arguments

`run`
: The numerical value of the run number of the objects to be saved. This parameter must be specified, as it also determines where to save the revised output.

`quiet`
: Suppress written report. Default is `FALSE`.

`...`
: Additional objects to be saved, which do not need to be suffixed with the run number, e.g. var1, var2, var3.

Details

Any objects that are made during the course of analysis in R can be added to the saved data that are automatically generated at the end of an NPAG or IT2B run and loaded with PMload. Objects with the same run number will be saved as a group. So if a user has made a new object called lm.1 that contains regressions related to run 1, it will be saved with any other object that also has .1 at the end.

Additionally, other objects can be saved via the ... argument. For exmaple PMsave(1,lm) will save any object with .1 at the end, plus an object named "lm". All objects will be suffixed with the run number when loaded back with PMload.

Author(s)

Michael Neely

See Also

PMload

PMstep *Stepwise covariate-parameter regressions*

Description

Perform a stepwise linear regression on all covariates and Bayesian posterior parameters

Usage

```
PMstep(x, icen = "median", direction = "backward")
```

Arguments

x	A PMcov object loaded by PMload or made by makeCov.
icen	A character vector to summarize covariate values. Default is "median", but can also be "mean".
direction	The direction for covariate elmination can be "backward", "forward", or "both". *backward* is the default.

Details

This function will perform stepwise linear regressions on a PMcov object loaded by PMload, or made by makeCov. Every covariate in the model will be tested in a stepwise linear regression for their relationships to each parameter in the model. Bayesian posterior parameters and individual covariates are used.

Value

A matrix with covariates in the rows and parameters in the columns. Values for the matrix are the multi-variate P-values. A value of NA indicates that the variable was not retained in the final model.

Author(s)

Michael Neely

See Also

step

PMtree *Create a new Pmetrics folder tree*

Description

Sets up a directory tree for a new Pmetrics project

Usage

```
PMtree(project = "NewProject", folder = getwd())
```

Arguments

project	A character string of a new project name, e.g. "DrugX"
folder	The full path to the root folder for the new project. Default is the current working directory.

Details

This function will create a new project folder tree with appropriate subfolders and a skeleton R script.

Value

A new folder named project with the following subfolders:

Rscript	The folder for the Rscript containing all run instructions. Within this folder will be a skeleton R script for the project.
Runs	The folder for all Pmetrics runs. Put run files, i.e. a data file and a model file in this directory prior to each run.
Sim	The folder for all simulations related to the project.
src	The folder for source data files in their original format, to preserve integrity and for audit purposes.

Author(s)

Michael Neely

See Also

PMmanual

Examples

```
PMtree("DrugX")
```

PMupdate *Download and install Pmetrics updates*

Description

Download and install Pmetrics updates from LAPK website

Usage

```
PMupdate(force = F)
```

Arguments

force	Boolean operator to force downloading and installing. Default is false.

Value

The latest system-specific Pmetrics update will be downloaded to a temporary folder and then installed. You need to restart R (Rstudio) and then reload Pmetrics with the `library(Pmetrics)` command to complete the installation.

Author(s)

Michael Neely

PMwriteMatrix *Write a Pmetrics .csv Matrix File*

Description

`PMwriteMatrix` is the companion function to `PMreadMatrix`. It will write an appropriate R data object to a formatted .csv file.

Usage

```
PMwriteMatrix(data, filename, override = F, version = "DEC_11")
```

Arguments

data	Must be a data.frame with appropriate structure (see `PMcheck`.
filename	Name of file to create.
override	Boolean operator to write even if errors are detected. Default is `False`.
version	Which matrix data format version to write. Default is the current version.

Details

`PMwriteMatrix` will first run `PMcheck` to determine if there are any errors in the structure of `data`. If the error check fails, the file will not be written and a message will be printed on the console.

Value

Returns the error report (see PMcheck for details).

Author(s)

Michael Neely

See Also

PMcheck, PMreadMatrix

Examples

```
## Not run:
data <- PMreadMatrix(paste(.libPaths(),"/Pmetrics/example/NPAG/PMex1.csv",sep=""))
data
#write to the current directory
PMwriteMatrix(data,"PMex1.csv")

## End(Not run)
```

PMwrk2csv *Convert Old .wrk Files to .csv Matrix File*

Description

PMwrk2csv will convert old style, single drug working copy files into a single .csv matrix file.

Usage

```
PMwrk2csv(prefix, ext = NULL, nsub)
```

Arguments

prefix	The alphabetic prefix of the working copy files to be converted, as a character vector.
ext	The extension of the working copy files files, if it exists. Does not have to be specified.
nsub	The number of subjects, or working copy files to read.

Details

This function will determine if the working copy files are old and convert them. New, multi-drug working copy files will be ignored. IDs will be suffixed with .1 to .9 for <10 subjects, .01 to .99 for <100 subjects and .001 to .999 for <1000 subjects, as needed to ensure unique ID numbers.

Value

A new file will be created with the name equal to prefix and an extension of "csv".

Author(s)

Michael Neely

print.MMopt *Print Pmetrics Multiple-Model Optimal Sampling Objects*

Description

Print *MMopt* objects

Usage

```
## S3 method for class 'MMopt'
print(x)
```

Arguments

x	The name of an *MMopt* data object generated by MMopt

Details

Simulated observations are plotted on the y-axis vs. time on the x.axis. Optimal sampling times are indicated as vertical lines.

Value

Prints the optimal sampling times and Bayes Risk.

Author(s)

Michael Neely

See Also

MMopt

print.PMerr *Print Data Errors*

Description

Print a Pmetrics Error Object

Usage

```
## S3 method for class 'PMerr'
print(x, ...)
```

Arguments

x	A PMerr object made by PMcheckMatrix.
...	Other parameters which are not necessary.

Details

Print the errors in a Pmetrics data file or PMmatrix object.

Value

A printed object.

Author(s)

Michael Neely

See Also

PMcheckMatrix

---

print.summary.PMmatrix *Summarize Covariates and Bayesian Posterior Parameter Values*

Description

Print the summary of a Pmetrics PMmatrix object

Usage

```
## S3 method for class 'summary.PMmatrix'
print(x)
```

Arguments

x	A summary.PMmatrix object made by summary.PMmatrix

Details

Summarize the raw data used for a Pmetrics run.

Value

A formatted printing of a *summary.PMmatrix* object

Author(s)

Michael Neely

See Also

summary.PMmatrix

print.summary.PMop *Print Summary of Observations and Predictions*

Description

Print a Pmetrics Observed vs. Predicted Summary Object

Usage

```
## S3 method for class 'summary.PMop'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	A summary.PMop object made by summary.PMop.
digits	Integer, used for number of digits to print.
...	Other parameters which are not necessary.

Details

Print a summary of observations, predictions and errors in a summary.PMop object made by summary.PMop.

Value

A printed object.

Author(s)

Michael Neely

See Also

summary.PMop

qgrowth *Extract CDC pediatric growth charts*

Description

Will extract height and weight for boys, girls or both for a given range of ages in months and percentile. This can be useful for simulations in Pmetrics.

Usage

```
qgrowth(sex = c("M", "F", "B"), percentile = c("5", "10", "25", "50", "75",
  "90", "95"), agemos = (seq(0, 18) * 12))
```

Arguments

sex	A single quoted character: "M" for males, "F" for females, or "B" for both, in which case an average of the two sexes will be returned. Default is "M".
percentile	An integer of the percentile for each age/sex to return. Default is 5.
agemos	The ages in months to return. The default is `seq(0,18)*12`, i.e. 1 to 18 years.

Value

A dataframe with columns

age	Age in months
wt	Weight in kilograms
ht	Height or length in centimeters
sex	The selected sex
percentile	The selected percentile

Author(s)

Michael Neely

---

setPMoptions *Set Pmetrics User Options*

Description

Set user options for Pmetrics

Usage

```
setPMoptions(sep, dec)
```

Arguments

sep	The field separator character; "," by default, but could be ";"
dec	The decimal separator character; "." by default, but could be ","

Details

This function will set user options for Pmetrics.

Value

The user preferences file will be updated. This will persist from session to session.

Author(s)

Michael Neely

SIMparse *Parse Pmetrics Simulator Output*

Description

Parses the output of the Pmetrics simulator

Usage

```
SIMparse(file, include, exclude, combine = F, silent = F)
```

Arguments

file	An output file or files of the simulator in the current working directory, or the full pathname to the file. To load and combine multiple outputs, specify files separated by commas or using wild cards. See details.
include	A vector of files to include in the parsing. For example, if you used a wild card in the file argument, such as "simout?.txt", which returned four files: simout1.txt, simout2.txt, simout3.txt and simout4.txt, and you wished to only parse the first and fourth file, specify include=c(1,4).
exclude	See the discussion for include, but this will exclude specified files.
combine	Boolean parameter, default False, which specifies whether you wish to combine the parsed files into a single PMsim object. This can be useful for making visual predictive checks, for example. If combine=F, and multiple files are parsed, then the return object will be a list of PMsim objects, which can be plotted or otherwise accessed using standard list referencing, e.g. simlist[[1]], simlist[[2]], etc.
silent	Suppress messages

Details

For file specification "?" will be matched by just a single numeral or character; "*" will be matched by any number of consecutive alphanumeric characters. Examples include file='simout1.txt,simout2.txt, file='simout?.txt' and file='sim*.txt'. All three will find the files simout1.txt, simout2.txt, and simout3.txt in the working directory. The second example would also find simout4.txt, etc. The third example would also find sim_1.txt if that existed. Note that to combine simulator output files, the numbers of simulated profiles may differ. The number of outputs and times of observations also may differ, although combining these may lead to strange plots since not all profiles have the same observations.

Value

If one file is parsed or multiple files are parsed and combined, the return will be a list with five items, of class *PMsim*. If multiple files are parsed and not combined, then the return will be a list of *PMsim* objects.

obs	An data frame of simulated observations with 4 columns: id, time, out, outeq. *id* is the number of the simulated subject, which will have a unique ending appended if simulations are combined, such that *id* will become x.y with x being the simulated profile number and y being the simulation template number. *time* is the time of the simulated output, *out* of output equation number *outeq*.

`amt` An data frame of simulated amounts with 4 columns: id, time, out, comp. *id* is the number of the simulated subject, which will have a unique ending appended if simulations are combined, such that *id* will become x.y with x being the simulated profile number and y being the simulation template number. *time* is the time of the simulated amount, *out* in compartment number *comp*.

`parValues` A datframe of the simulated parameter values, combined across files as necessary

`totalSets` The total number of parmeter sets simulated, which may be greater than the number of rows in `parValues` if some sets were discarded for being outside specified limits. For more than one file parsed, this will the total number in all files.

`totalMeans` The means of each simulated parameter based on all profiles in a given file (even those discarded for exceeding limits). For more than one file parsed, this will be the weighted averages for all simulations.

`totalCov` The covariances of the simulated parameter sets based on all profiles in a given file (even those discarded for exceeding limits). For more than one file parsed, this will be the weighted averages for all simulations.

A plot method exists in `plot.PMsim` for *PMsim* objects.

Author(s)

Michael Neely

See Also

`SIMrun`

`SIMrun` *Run the Pmetrics Simulator*

Description

Runs the Pmetrics simulator

Usage

```
SIMrun(poppar, limits = NULL, model = "model.txt", data = "data.csv",
  split = F, include, exclude, nsim = 1000, predInt = 0, covariate,
  seed = -17, ode = -4, obsNoise, doseTimeNoise = rep(0, 4),
  doseNoise = rep(0, 4), obsTimeNoise = rep(0, 4), makecsv, outname,
  clean = T, silent = F, nocheck = F)
```

Arguments

`poppar` Either an object of class *PMfinal* (see `makeFinal`) or a list containing three items in this order, but of any name: vector of weights, vector of mean parameter values, and a covariance matrix. If only one distribution is to be specified the `weights` vector should be of length 1 and contain a 1. If multiple distributions are to be sampled, the `weights` vector should be of length equal to the number of distributions and its values should sum to 1, e.g. `c(0.25,0.05,0.7)`.

The means matrix may be a vector for a single distribution, or a matrix with `length(weights)` rows and number of columns equal to the number of parameters, *npar*. The covariance matrix will be divided by `length(weights)` and applied to each distribution.

`limits` If limits are specified, each simulated parameter set that contains a value outside of the limits will be ignored and another set will be generated. Four options exist for limits. 1) The default `NULL` indicates that no limits are to be applied to simulated parameters. 2) The second option is to set `limits` to `NA`. This will use the parameter limits on the primary parameters that are specified in the model file. 3) The third option is a numeric vector of length 1 or 2, e.g. 3 or c(0.5,4), which specifies what to multiply the columns of the limits in the model file. If length 1, then the lower limits will be the same as in the model file, and the upper limits will be multiplied by value specified. If length 2, then the lower and upper limits will be multiplied by the specified values. If this option is used, `popppar` must be a `PMfinal` object. 4) The fourth option for limits is a fully customized matrix of limits for simulated values for each parameter which will overwrite any limits in the model file. If specified, it should be a data.frame or matrix with number of rows equal to the number of random paramters and 2 columns, corresponding to the minimum and maximum values. For example, a final$ab object, or a directly coded matrix, e.g. matrix(c(0,5,0,5,0.01,100),nrow=3,ncol=2,byrow=T) for 3 parameters with limits of [0,5], [0,5] and [0.01,100], respectively. It is possible to convert a parameter to fixed by omitting the second limit. Means and covariances of the total number of simulated sets will be returned to verify the simulation, but only those sets within the specified limits will be used to generate output(s) and the means and covariances of the retained sets may (and likely will be) different than those specified by `poppar`.

`model` Name of a suitable model file template in the working directory. The default is "model.txt". This file will be converted to a fortran model file. If it is detected to already be a fortran file, then the simulation will proceed without any further file conversion.

`data` Either a PMmatrix object previously loaded with (`PMreadMatrix`) or character vector with the filename of a Pmetrics matrix file that contains template regimens and observation times. The value for outputs can be coded as any number(s) other than -99. The number(s) will be replaced in the simulator output with the simulated values.

`split` Boolean operator controlling whether to split an NPAG *PMfinal* object into one distribution per support point, with means equal to the vector of parameter values for that point, and covariance equal to the population covariance divided by the number of support points

`include` A vector of subject IDs in the `matrixfile` to iterate through, with each subject serving as the source of an independent simulation. If missing, all subjects in the datafile will be used.

`exclude` A vector of subject IDs to exclude in the simulation, e.g. c(4,6:14,16:20) If a *makecsv* filename is supplied, ID numbers will be of the form nsub.nsim, e.g. 1.001 through 1.1 for the first subject, 2.001 through 2.1 for the second subject, etc. if 1000 simulations are made from each subject.

`nsim` The number of simulated profiles to create, per subject. Default is 1000. Entering 0 will result in one profile being simulated from each point in the nonparametric prior (for NPAG final objects only).

`predInt` The interval in fractional hours for simulated predicted outputs at times other than those specified in the template `data`. The default is 0, which means there

will be simulated outputs only at times specified in the data file (see below). Values of predInt > 0 result in simulated outputs at the specified value of predInt, e.g. every 15 minutes for predInt = 0.25 from time 0 up to the maximal time in the template file, per subject if nsub > 1. You may also specify `predInt` as a vector of 3 values, e.g. `c(1,4,1)`, similar to the R command `seq`, where the first value is the start time, the second is the stop time, and the third is the step value. Finally, you can have multiple such intervals by specifying `predInt` as a list of such vectors, e.g. `list(c(0,24,1),c(72,96,1))`. Outputs for times specified in the template file will also be simulated. To simulate outputs *only* at the output times in the template data (i.e. EVID=0 events), use `predInt=0`, which is the default. Note that the maximum number of predictions total is 594, so the interval must be sufficiently large to accommodate this for a given number of output equations and total time to simulate over. If `predInt` is set so that this cap is exceeded, predictions will be truncated.

`covariate` If you are using the results of an NPAG or IT2B run to simulate, i.e. a *PMfinal* object as `poppar`, then you can also simulate with covariates. This argument is a list with the following names.

- `cov` The name of a PMcov object, such as that loaded with PMload. Pmetrics will use this object to calculate the correlation matrix between all covariates and Bayesian posterior parameter values.
- `mean` A named list that allows you to specify a different mean for one or more of the covariates. Each named item in the list is the name of a covariate in your data that is to have a different mean. If this argument is missing then the mean covariate values in the population will be used for simulation. The same applies to any covariates that are not named in this list. Example: `covariate=list(cov=cov.1,mean=list(wt=50))`.
- `sd` This functions just as the `mean` list argument does - allowing you to specify different standard deviations for covariates in the simulation. If it, or any covariate in the `sd` list is missing, then the standard deviations of the covariates in the population are used. Example: `covariate=list(cov=cov.1,sd=list(wt=10))`
- `limits` This is a bit different than the limits for population parameters above. Here, `limits` is similar to `mean` and `sd` for covariates in that it is a named list with the minimum and maximum allowable simulated values for each covariate. If it is missing altogether, then no limits will apply. If it is specified, then named covariates will have the indicated limits, and covariates not in the list will have limits that are the same as in the original population. If you want some to be limited and some to be unlimited, then specify the unlimited ones as items in this list with very large ranges. Example: `covariate=list(cov=cov.1,limits=list(wt=c(10,70)))`
- `fix` A character vector (not a list) of covariates to fix and not simulate. In this case values in the template data file will be used and not simulated. Example: c("wt", "age")

Whether you use the means and standard deviations in the population or specify your own, the covariance matrix in poppar will be augmented by the covariate covariances for any non-fixed covariates. The parameter plus covariate means and this augmented covariance matrix will be used for simulations. In effect, all non-fixed covariates are moved into the #Primary block of the model file to become parameters that are simulated. In fact, a copy of your model file is made with a "c" prepended to the model name (e.g. "model.txt" -> "c_model.txt").

`seed` The seed for the random number generator. For `nsub` > 1, should be a vector of length equal to `nsub`. Shorter vectors will be recycled as necessary. Default is

-17.

`ode` Ordinary Differential Equation solver log tolerance or stiffness. Default is -4, i.e. 0.0001. Higher values will result in faster runs, but simulated concentrations may not be as accurate.

`obsNoise` The noise added to each simulated concentration for each output equation, where the noise is randomly drawn from a normal distribution with mean 0 and SD = C0 + C1*conc + C2*conc^2 + C3*conc^3. Default values are 0 for all coefficients (i.e.) no noise. If present will override any other values in the data file or model file. Specify as a vector of length 4 times the number of output equations, e.g. c(0.1,0.1,0,0) for one output and c(0.1,0.1,0,0,0.01,0.2,-0.001,0) for two output equations. If specified as `NA`, values in the data file will be used (similar to `limits`, above). If they are missing, values in the model file will be used.

`doseTimeNoise` A vector of length four to specify dose time error polynomial coefficients. The default is 0 for all coefficients.

`doseNoise` A vector of length four to specify dose amount error polynomial coefficients. The default is 0 for all coefficients.

`obsTimeNoise` A vector of length four to specify observation timing error polynomial coefficients. The default is 0 for all coefficients.

`makecsv` A character vector for the name of the single .csv file to be made for all simulated "subjects". If missing, no files will be made.

`outname` The name for the output file(s) without an extension. Numbers 1 to `nsub` will be appended to the files. If missing, will default to "simout".

`clean` Boolean parameter to specify whether temporary files made in the course of the simulation run should be deleted. Defaults to `True`. This is primarily used for debugging.

`silent` Boolean operator controlling whether a model summary report is given. Default is `FALSE`.

`nocheck` Suppress the automatic checking of the data file with `PMcheck`. Default is `FALSE`.

Details

The Monte Carlo simulator in Pmetrics is a powerful tool for parametric or semi-parametric sampling. NPAG or IT2B final objects can easily be used as the prior distributions for sampling, or prior distributions may be manually specified. Prior distributions may be unimodal-multivariate (parametric sampling), or multimodal-multivariate (semi-parametric sampling). For priors from NPAG, this can easily be accomplished with the `split` argument.

It is also possible to simulate with covariates if they are included as part of the model. By specifying a covariate list argument, Pmetrics will first calculate the correlation matrix between the covariates and the Bayesian posterior parameter values for each subject in the population model. Using either the mean and standard deviation of each covariate in the population, or a user-specified mean and/or standard deviation, Pmetrics will then calculate an augmented covariance matrix to be used in simulations. Pmetrics will make a copy of the model file with all covariates moved into the primary block as parameters to be simulated.

Noise can be applied to the simulated observations. Noise may also be applied to the observation times, to the dose times, or to the dose amounts.

Limits on the simulated parameter sets can also be specified using the limits on primary parameters in the model file or by specifying them manually as an argument. Limits can also be applied to simulated covariates.

It is permissible to fix a parameter for simulation that was a random parameter in the model prior by changing the range in the model file to a single value for that parameter.

The same model and data file strutures are used for the simulator as for any other Pmetrics functions. In this case, the data file will serve as the template for the information regarding dosing, covariate values, and observations. Template data files may have more than one subject in them, in which case the simulator will use each subject specified by the `include` argument (default is all subjects) to generate `nsim` parameter sets and corresponding observations.

Simulator output is directed to text files, one for each template subject, which can be read back into R by `link{SIMparse}`. Output may also be directed to a new Pmetrics .csv data file using the `makecsv` argument.

Value

No value is returned, but simulated file(s) will be in the working directory.

Author(s)

Michael Neely

See Also

SIMparse

Examples

```
## Not run:
wd <- getwd()
#make 1 lognormal distribution for each parameter
weights <- 1
mean <- log(c(0.7,0.05,100))
cov <- matrix(rep(0,length(mean)**2),ncol=length(mean))
diag(cov) <- (c(0.15,0.15,0.15)*mean)**2
#make the prior for the simulation
poppar <- list(weights,mean,cov)
setwd(paste(normalizePath(get("PmetricsPath",envir=PMenv),winslash="/"),"/Pmetrics/example/Sim",sep=""))
#run simulation
SIMrun(poppar,"temp1.csv",nsim=15,model="model1.for",obsNoise=c(0.02,0.1,0,0),makecsv="PMex1.csv",outname="
#extract results of simulation
simout <- SIMparse("example1.txt")
file.remove("example1.txt")
#plot simulated profiles (use help(plot.PMsim) for more information)
plot(simout,ci=0,probs=NA,x.qlab=0.75,log=T,col="red",lwd=2,pch=NA,join=T)
setwd(wd)

## End(Not run)
```

---

ss.PK *Sample size calculations for Phase 1 PK study design*

Description

This function calculates sample size based on a desired standard error of the mean, to a specified confidence, for a given mean and standard deviation.

Usage

```
ss.PK(n, mean, sd, precision, ci = 0.95)
```

Arguments

n	Sample size. This value can be missing if sample size is desired, or specified to calculate the maximum sd for given mean, precision, and ci.
mean	Mean prameter value. User value is mandatory.
sd	Standard deviation of parameter values. If present, the function will return n. If missing and n is specified, will return the maximum sd as detailed above.
precision	Desired width of the standard error of the mean (SEM). Default is 0.2, i.e. 20% or 10% below and 10% above the mean. If missing, and mean, sd and n are specified, precision will be calculated.
ci	Confidence for the desired width of the SEM. Default is 0.95.

Details

The formula is n = qnorm((1+ci)/2)**2 * sd**2 / (precision*mean)**2

Value

The missing argument: n, sd or precision.

Author(s)

Michael Neely

---

summary.PMcov *Summarize Covariates and Bayesian Posterior Parameter Values*

Description

Summarize a Pmetrics Covariate object

Usage

```
## S3 method for class 'PMcov'
summary(x, icen = "median")
```

Arguments

x	A PMcov object made by makeCov.
icen	Summary function for covariates and posterior parameters. Default is "median", but can specify "mean".

Details

Summarize covariates and Bayesian posterior parameter values for each subject.

Value

A data frame with the summary of the PMcov object for each subject's covariates and Bayesian posterior parameter values.

Author(s)

Michael Neely

See Also

makeCov

---

summary.PMfinal *Summary Statistics for PMfinal Objects*

Description

Generates summary statistics of final population model parameters.

Usage

```
## S3 method for class 'PMfinal'
summary(x, lower = 0.025, upper = 0.975)
```

Arguments

x	The PMfinal object made after an NPAG or IT2B, e.g. final.1 after run 1.
lower	Desired lower confidence interval boundary. Default is 0.025. Ignored for IT2B objects.
upper	Desired upper confidence interval boundary. Default is 0.975. Ignored for IT2B objects.

Details

For NPAG runs, this function will generate weighted medians as central tendencies of the population points with a 95% confidence interval (95% CI) around the median, and the median absolute weighted deviation (MAWD) from the median as a measure of the variance, with its 95% CI. These estimates correspond to weighted mean, 95% CI of the mean, variance, and 95% CI of the variance, respectively, for a sample from a normal distribution. To estimate these non-parametric summaries, the function uses a Monte Carlo simulation approach, creating 1000 x npoint samples with replacement from the weighted marginal distribution of each parameter, where npoint is the number of support points in the model. As an example, if there are 100 support points, npoint = 100, and for Ka, there will be 1000 sets of 100 samples drawn from the weighted marginal distribution of the values for Ka. For each of the 1,000 sets of npoint values, the median and MAWD are calculated, with MAWD equal to the median absolute difference between each point and the median of that set. The output is npoint estimates of the weighted median and npoint estimates of the MAWD for each parameter, from which the median, 2.5th, and 97.5th percentiles can be found as point estimates and 95% confidence interval limits, respectively, of both the weighted median and MAWD.

For IT2B runs, the function will return the mean and variance of each parameter, and the standard errors of these terms, using SE (mean) = SD/sqrt(nsub) and SE (var) = var * sqrt(2/(nsub-1)).

Value

The output is a data frame. For NPAG this has 4 columns:

`value`	The value of the summary statistic
`par`	The name of the parameter
`type`	Either *WtMed* for weighted median, or *MAWD* for MAWD (see details)
`quantile`	Requested `lower`, 0.5 (median), and `upper` quantiles

For IT2B this has 5 columns:

`mean`	Parameter mean value
`se.mean`	Standard error of the mean
`cv.mean`	Error of the mean divided by mean
`var`	Variance of the parameter values
`se.var`	Standard error of the variance

Author(s)

Michael Neely

See Also

`makeFinal`, `ITparse`, `plot.PMfinal`

Examples

```
data(PMex1)
final <- makeFinal(NPdata.1)
summary(final)
```

`summary.PMmatrix` *Summarize PMmatrix objects*

Description

Summarize a Pmetrics PMmatrix object

Usage

```
## S3 method for class 'PMmatrix'
summary(x, formula, FUN, ..., include, exclude)
```

Arguments

x	A PMmatrix object loaded by PMreadMatrix or PMload.
formula	Optional formula for specifying custom summaries. See aggregate and formula for details on how to specify formulae in R. If, for example, the data contain a covariate for weight named 'wt', then to summarize the mean dose in mg/kg per subject specify formula=dose/wt~id, FUN=mean.
FUN	The summary function to apply to formula, if specified.
include	A vector of subject IDs to include in the summary, e.g. c(1:3,5,15)
exclude	A vector of subject IDs to exclude in the summary, e.g. c(4,6:14,16:20)
...	Additional arguments to FUN, e.g. na.rm=T

Details

Summarize the raw data used for a Pmetrics run.

Value

A list of class *summary.PMmatrix* with the summary of the PMmatrix object, containing the following items:

nsub	Number of subjects
ndrug	Number of drug inputs
numeqt	Number of outputs
nobsXouteq	Number of observations by outeq
missObsXouteq	Number of missing observations by outeq
ncov	Number of covariates
covnames	Covariate names
ndoseXid	Number of doses per input per subject
nobsXid	Number of observations per outeq per subject
doseXid	Doses per input per subject
obsXid	Observations per outeq per subject
formula	Results of including formula

Author(s)

Michael Neely

See Also

print.summary.PMmatrix, aggregate

summary.PMop *Summarize Observations and Predictions*

Description

Summarize a Pmetrics Observed vs. Predicted x

Usage

```
## S3 method for class 'PMop'
summary(x, digits = max(3, getOption("digits") - 3),
  pred.type = "post", icen = "median", outeq = 1, ...)
```

Arguments

x	A PMop object made by makeOP.
digits	Integer, used for number of digits to print.
pred.type	Either 'post' for a posterior object or 'pop' for a population object. Default is 'post'.
icen	Can be either "median" for the predictions based on medians of pred.type parameter value distributions, or "mean". Default is "median".
outeq	Output equation number. Default is 1.
...	Other parameters which can be passed to summary.

Details

Summarize observations, predictions and errors in a PMop x made by makeOP.

Value

A list with two xs. The first component of the list is a matrix with the minimum, first quartile, median, third quartile, maximum, mean and standard deviation for times, observations and predictions in x. The second contains the mean prediction error, the mean weighted prediction error (bias), the mean squared prediction error, root mean sqaured error (RMSE), percent root mean squared error (squared prediction error, the bias-adjusted mean squared prediction error, and the bias- adjusted mean weighted squared prediction error (imprecision).

Author(s)

Michael Neely

See Also

makeOP

`summary.PMpta` *Summarize Percent Target Attainment*

Description

Summarize a Pmetrics Percent Target Attainment Object

Usage

```
## S3 method for class 'PMpta'
summary(x, ci = 0.95, ...)
```

Arguments

`x`	A PMpta object made by `makePTA`.
`ci`	Confidence interval for pharmacodynamic index reporting. Default is 0.95.
`...`	Other parameters which can be passed to `summary`.

Details

Summarize target statistics and success proportions in a PMpta object made by `makePTA`.

Value

A list with two named objects: pta (probability of target attainment) and pti (pharmacodynamic index).

`pta`	A data frame with the following columns: simnum, target, prop.success, pdi.mean, and pdi.sd *simnum* is the number of the simulation; *target* is the specified target; *success* has the proportion with a ratio > `prop.success`; *pdi.mean* and *pdi.sd* are the mean and standard deviation of the pharmacodyamic index (e.g. AUC/MIC) for each simulation and target.
`pdi`	A data frame with the following columns: target, simnum, lowerCI, median, upperCI. *target* and *simnum* are as above. *lowerCI*, *median*, and *upperCI* are the lower limit, median, and upper limit of the confidence interval for the pdi whose width is specified by `ci`

Author(s)

Michael Neely

See Also

`makePTA`

Index