

# Package ‘nonmem2R’

April 5, 2018

**Type** Package

**Title** Loading NONMEM Output Files and Simulate with Parameter Uncertainty

**Version** 0.1.9

**Author** Magnus Astrand

**Maintainer** Magnus Astrand <magnus.astrand@astrazeneca.com>

**Description** Loading NONMEM (NONlinear Mixed-Effect Modeling, <<http://www.iconplc.com/innovation/solutions/nonmem/>> ) output files and simulate with parameter uncertainty.

**License** Unlimited

**Depends** R (>= 3.0.0), mvtnorm, lattice, latticeExtra, MASS, ggplot2

**LazyData** TRUE

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-04-05 17:04:15 UTC

## R topics documented:

covload . . . . .	2
extload . . . . .	3
extToTable . . . . .	4
grid.sim . . . . .	5
sumoR . . . . .	6
systemPSN . . . . .	7
test.grid.sim . . . . .	7
vpcfig . . . . .	8
vpcfig2 . . . . .	10

<b>Index</b>	<b>14</b>
--------------	-----------

---

covload	<i>Load the covariance matrix from a nonmem .cov output file.</i>
---------	---

---

### Description

Load the covariance matrix from a nonmem .cov output file. Either the covariance matrix of all THETA parameters (default) or the covariance matrix of all parameters, THETA, OMEGA and SIGMA.

For cov-files with multiple table results either only the last table result is loaded (`last.table.only=TRUE`, default) or all table results are loaded (`last.table.only=FALSE`). The result is then a named list of covariance matrix's, `LastTable` is the last table in the cov-files.

### Usage

```
covload(model, use.model.path = TRUE, last.table.only = TRUE,  
        theta.only = TRUE)
```

### Arguments

<code>model</code>	name of the cov file with or without the .cov extension. <code>model</code> may include full or relative path to the cov file. See examples.
<code>use.model.path</code>	Load file from a global defined model library ( <code>TRUE=</code> default). If so will look for a global character vector named <code>model.path</code>
<code>last.table.only</code>	Include only the last table result for cov files with multiple table results
<code>theta.only</code>	return covariance matrix of theta's only (default)

### Value

The covariance matrix or a named list of covariance matrix's

### Examples

```
##### Load the .cov file "run001.cov"  
# 1) Get path to the example file included in nonmem2R package  
file1 <- system.file("extdata", "run001.cov", package = "nonmem2R")  
# 2) Load the file using the covload function  
covload(file1)
```

---

extload	<i>Load final parameter values from a nonmem .ext output file.</i>
---------	--

---

### Description

Load parameter values from a nonmem .ext output file. For output files from FO, FOCE and IMP only the final parameter values are loaded. For output tables of MCMC and SAEM all parameter values from ITERATIONS>0 will be loaded unless positive.iterations=FALSE in case all the output from all iterations are loaded.

For ext-files with multiple table results either only the last table result is loaded (last.table.only=TRUE, default) or all table result are loaded (last.table.only=FALSE). All but the last table results are then returned as sub-lists to the last table result.

### Usage

```
extload(model, use.model.path = TRUE, positive.iterations.only = TRUE,
        last.table.only = TRUE)
```

### Arguments

model	name of the ext file with or without the .ext extension. model may include full or relative path to the ext file. See examples.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
positive.iterations.only	Include only rows with ITERATIONS>0 from MCMC and SEAM table results (default=TRUE)
last.table.only	Include only the last table result for ext files with multiple table results

### Value

Named list including theta, theta.sd, omega, omega.sd, sigma, sigma.sd, and ofv. Here .sd is the vector(matrix) with standard errors estimated parameter values in theta(omega & sigma).

For MCMC output files each object are matrixes.

### Examples

```
##### Load the .ext file "run001.ext"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.ext", package = "nonmem2R")
# 2) Load the file using the extload function
extload(file1)
```

---

 extToTable

*Compile parameter table suitable for reports*


---

### Description

Load parameter values from a nonmem .ext output file and compile to table suitable for reports. Format can be "wide" (wide=TRUE, default) with a similar look as from sumoR, or similar a 3 column layout (wide=FALSE).

### Usage

```
extToTable(model, use.model.path = TRUE, tableType = 2, wide = TRUE,
  format.estimate = "% -#6.4g", format.rse = "%#6.3g")
```

### Arguments

model	name of the ext file with or without the .ext extension. model may include full or relative path to the ext file. See examples.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
tableType	Table type for THETA's, OMEGA's and SIGMA's tableType=0: Present OMEGA and SIGMA as variance and covariances and display SE for THETA, OMEGA, SIGMA tableType=1: Present OMEGA and SIGMA as variance and covariances and display RSE for THETA, OMEGA, SIGMA tableType=2: Present OMEGA and SIGMA as standard-deviation and correlations and display RSE for THETA, OMEGA, SIGMA tableType=3: Present OMEGA and SIGMA as standard-deviation and correlations and display SE for THETA, OMEGA, SIGMA
wide	produce a wide 9-column table (wide=TRUE, default) or a thin 3-column table(wide=FALSE)
format.estimate	format for estimated value, passed to sprintf
format.rse	format for rse, passed to sprintf

### Value

a character-matrix

### Examples

```
##### Load the .ext file "run001.ext"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.ext", package = "nonmem2R")
# 2) Load the file using the extload function
extToTable(file1)
```

---

grid.sim	<i>Combine fix grid and simulated grid based on multivariate normal distribution</i>
----------	--

---

## Description

grid.sim produce comparable output to mvnorm, however for methods 2 to 4 grid.sim make use of center and scaling, and or use a fix grid of values for one column of the output. The intended use of grid.sim is for computing confidence intervals (CI) for model predictions. Method 1 is the same as rmvnorm, however can handle cases of zero variances in sigma. Method 2 is method 1 followed by centering and scaling of the simulated matrix. Thus output when using method 2 always have sample mean and covariance equal to input parameters means and sigma. Method 3 is the same as method 1 however with one column having a fixed range of values rather than a simulated range of values. Method 4 is method 3 followed by centering and scaling of the simulated matrix. Thus output when using method 4 always have sample mean and covariance equal to input parameters means and sigma.

As the number of simulations (n) goes to infinity all methods in grid.sim are identical to mvnorm in that the sample covariance (mean) of the output will converge to the input covariance matrix sigma (vector means).

The advantage with methods 2 to 4 (over method 1 and rmvnorm) is that they provides more stable results, hence number of simulations can be reduced and still have equally stable results when used to represent parameter uncertainty and or population variability in model predictions.

## Usage

```
grid.sim(n, means = NULL, sigma, grid.param = NULL, method = 4)
```

## Arguments

n	Number of simulations
means	vector of mean values
sigma	covariance matrix
grid.param	the index of the parameter for which a fix grid (from qnorm) is used instead of a simulated grid (rnorm). If grid.param=NULL (default) the fix grid will be used for the parameter with largest variance.
method	simulation method, default=4

## Value

row-matrix of parameters

## Examples

```
sigma<-matrix(c(1,0.5,0.5,2),ncol=2)
sim1<-grid.sim(1000,sigma=sigma)
pairs(sim1)
cov(sim1)
```

---

sumoR	<i>Compile summary information of for NONMEM model based on the lst file, ext file, and the cov file.</i>
-------	---

---

## Description

Compile summary information similar to that of the sumo PSN function, based on the NONMEM output files lst, ext, and if covariance setp was run, the cov file.

## Usage

```
sumoR(model, use.model.path = TRUE, tableType = 2,
      format.estimate = "% -#6.4g", format.rse = "%#6.3g")
```

## Arguments

model	name of the lst file with or without the .lst extension. model may include full or relative path to the lst file.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
tableType	Table type for THETA's, OMEGA's and SIGMA's tableType=0: Present OMEGA and SIGMA as variance and covariances and display SE for THETA, OMEGA, SIGMA tableType=1: Present OMEGA and SIGMA as variance and covariances and display RSE for THETA, OMEGA, SIGMA tableType=2: Present OMEGA and SIGMA as standard-deviation and correlations and display RSE for THETA, OMEGA, SIGMA tableType=3: Present OMEGA and SIGMA as standard-deviation and correlations and display SE for THETA, OMEGA, SIGMA
format.estimate	format for estimated value, passed to sprintf
format.rse	format for RSE or SE, passed to sprintf

## Value

named list of class sumoR

## Examples

```
##### Compile summary information from the .lst file "run001.lst"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.lst", package = "nonmem2R")
# 2) Compile summary information from "run001.lst"
sumoR(file1)
```

---

systemPSN	<i>Run qpsn system call</i>
-----------	-----------------------------

---

**Description**

Run qpsn system call on system where psn is available. If psn is not available systembase is used.

**Usage**

```
systemPSN(cmd, use.model.path = TRUE, ...)
```

**Arguments**

cmd	qpsn cmd
use.model.path	Run cmd in a specified model library (TRUE=default). If so will look for a global character vector named model.path and run the system cmd the folder path as specified in model.path.
...	Further arguments

**Examples**

```
# (if the platform has who)
try(systemPSN("who", intern = TRUE))

try(systemPSN("ls fizzlipuzzli", intern = TRUE, ignore.stderr = TRUE))
# zero-length result since file does not exist, and will give warning.
```

---

test.grid.sim	<i>Function for testing grid.sim and compare with rmvnorm</i>
---------------	---

---

**Description**

Test grid.sim

**Usage**

```
test.grid.sim(n = 1000, k = 1:4)
```

**Arguments**

n	number of simulations
k	subset of parameters from a 4X4 sigma to use

**Value**

graphics

**Examples**

```
## Not run:
require(lattice)
test.grid.sim(n=1000)

## End(Not run)
```

---

vpcfig	<i>Visual Predictive Check (VPC) based on Perl-speaks-NONMEM (PsN) generated VPC files (lattice version).</i>
--------	---

---

**Description**

This function creates VPC using output files from the vpc command in Pearl Speaks NONMEM (PsN). Graphs are generated using the lattice package with many arguments for plot settings are passed directly to the xyplot function and have the same flexibility as when used in xyplot.

**Usage**

```
vpcfig(vpcdir = NULL, vpctab = NULL, vpcresult = NULL,
       use.model.path = TRUE, strata.names = NULL, strata.subset = NULL,
       percentile = 10, fy = function(y) { y }, fx = function(x) { x },
       xlab = NULL, ylab = NULL, col.data = 8, cex.data = 0.3,
       pch.data = 3, col.line = c("blue", "red", "blue"), lwd.line = c(1, 2,
       1), lty.line = c(1, 1, 1), col.segm = c("lightblue", "pink", "lightblue"),
       alpha.segm = c(0.5, 0.5, 0.5), type = 3, ...)
```

**Arguments**

vpcdir	Path of directory of the VPC files
vpctab	Path to the vpctab-file
vpcresult	Path to the vpcresult file
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
strata.names	Character vector for strata names. Must have length equal to number of strata's in vpc files, otherwise ignored.
strata.subset	Vector specifying subset of strata to use. Either a vector of index, e.g. strata.subset=c(1,3), or a character vector naming which strata to use. See details.
percentile	percentile to use, default=10 will display 10 Percentile argument must match columns included in the vpcresult file.
fy	transformation function for y-axis, default to identity function (f(y)=y).
fx	transformation function for x-axis, default to identity function (f(x)=x)
xlab	label for x-axis, default is IDV as found in vpcresultfile, passed to xyplot.
ylab	label for y-axis, default is DV as found in vpcresultfile, passed to xyplot.



col.data	color for data points, passed to xyplot, default=8
cex.data	cex for data points, passed to xyplot, default=0.3
pch.data	plot symbol (pch) for data points, passed to xyplot, default=3
col.line	line color for low quartile, median, and high quartile of data, passed to xyplot, default=c("blue","red","blue")
lwd.line	line width (lwd) for low quartile, median, and high quartile of data, passed to xyplot, default=c(1,2,1)
lty.line	line type (lty) for low quartile, median, and high quartile of data, passed to xyplot, default=c(1,1,1)
col.segm	Color of CI regions (low, median and high) for model, passed to xyplot, default=c("lightblue","pink","lightblue")
alpha.segm	alpha of CI regions (low, median and high) for model, passed to xyplot, default=c(0.5,0.5,0.5)
type	type of VPC plot, 1=display model regions only, 2=as 1 + lines (low, median and high) of data, 3= as 2 + points for data. type=0 can also be used and then no graph is produced but instead a list with 2 dataframes is returned. one for the vpcresult and one for the observed data in the vpctab file.
...	Further arguments, passed to xyplot, e.g xlim and ylim axis limits, main for title of plot, abline for adding reference lines, or scales for formatting axes, see help files for xyplot. Note: col, cex, pch, lty, lwd, and alpha should not be used. Instead use the corresponding arguments defined above.

## Details

The lattice package is used for creating the VPC and vpcfig have functionality for e.g. modify names of strata, change layout of panels, use log-scale.

The data used is either specified by the directory of the PsN generated files, or by specifying the file names of both the vpctab-file and the vpcresult file. See examples 1 and 2 below.

Names of strata can be changed with the strata.names argument, if strata.names is NULL the names as specified in the vpcresult file are used.

strata.subset can be used to select a subset of strata, and or to change the order of stratas. See example 3. NOTE: strata.subset is matched with strata.names unless strata.names=NULL. If strata.names=NULL then strata.subset is matched to names as specified in vpcresult file. Default is to use all strata's (strata.subset=NULL).

The fy and fx arguments can be used to alter the scale of data plotted. For example, using  $fy=function(y)\log(y+1)$  is a convenient way to get log-scale for y-axis but with an off-set to show any values==0. Proper y tick marks can then be set by using the xyplot argument scales.

## Value

lattice object of VPC plot

## Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "vpctab004.dat", package = "nonmem2R")
file2 <- system.file("extdata", "vpc_results.csv", package = "nonmem2R")

# Ex 1, produce VPC with default setting, here specifying both vpctab and vpcresult
vpcfig(vpctab=file1,vpcresult=file2)

## Not run:
# Ex 2, produce VPC with default setting, here specifying only directory of vpc files
path1<-gsub("vpctab004.dat","",file1)
vpcfig(vpcdir=path1)

## End(Not run)

# Ex 3, produce VPC with i) modifies strata names, ii) strata in reverse order, and
#      iii) labels
strata.names<-c("Group A","Group B")
xlab<-"Time after dose (hrs)"
ylab<-"Plasma Conc(mmol/L)"
vpcfig(vpctab=file1,vpcresult=file2,strata.names=strata.names,strata.subset=2:1,
       xlab=xlab,ylab=ylab)
```

---

vpcfig2

*Visual Predictive Check (VPC) based on Perl-speaks-NONMEM (PsN) generated VPC files (ggplot2-version).*

---

## Description

This function creates VPC using output files from the vpc command in Pearl Speaks NONMEM (PsN). Graphs are generated using the ggplot2 package and the return object is an ggplot class and can be further modified, see details below and documentation for ggplot2 for further details.

## Usage

```
vpcfig2(vpcdir = NULL, vpctab = NULL, vpcresult = NULL,
       use.model.path = TRUE, strata.names = NULL, strata.subset = NULL,
       percentile = 10, fy = function(y) { y }, fx = function(x) { x },
       xlab = NULL, ylab = NULL, col.data = "gray20", cex.data = 1,
       pch.data = 19, alpha.data = 0.5, col.line = "grey20", lwd.line = 1,
       lty.line = c(2, 1, 2), col.segm = c("steelblue", "gray50", "steelblue"),
       alpha.segm = 0.6, type = 3)
```

## Arguments

vpcdir            Path of directory of the VPC files

<code>vpctab</code>	Path to the vpctab-file
<code>vpcreresult</code>	Path to the vpcreresult file
<code>use.model.path</code>	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named <code>model.path</code>
<code>strata.names</code>	Character vector for strata names. Must have length equal to number of strata's in vpc files, otherwise ignored.
<code>strata.subset</code>	Vector specifying subset of strata to use. Either a vector of index, e.g. <code>strata.subset=c(1,3)</code> , or a character vector naming which strata to use. See details.
<code>percentile</code>	percentile to use, default=10 will display 10 Percentile argument must match columns included in the vpcreresult file.
<code>fy</code>	transformation function for y-axis, default to identity function ( $f(y)=y$ ).
<code>fx</code>	transformation function for x-axis, default to identity function ( $f(x)=x$ )
<code>xlab</code>	label for x-axis, default is IDV as found in vpcreresultfile, passed to labs.
<code>ylab</code>	label for y-axis, default is DV as found in vpcreresultfile, passed to labs.
<code>col.data</code>	color for data points, passed to <code>geom_point</code> , default="gray20"
<code>cex.data</code>	cex for data points, passed to <code>geom_point</code> , default=1
<code>pch.data</code>	plot symbol (pch) for data points, passed to <code>geom_point</code> , default=19
<code>alpha.data</code>	alpha for plotting of data points, passed to <code>geom_point</code> , default=0.5
<code>col.line</code>	line color for low quartile, median, and high quartile of data, passed to <code>geom_line</code> , default="gray20"
<code>lwd.line</code>	line width (lwd) for low quartile, median, and high quartile of data, passed to <code>geom_line</code> , default=1
<code>lty.line</code>	line type (lty) for low quartile, median, and high quartile of data, passed to <code>geom_line</code> , should be vector of length 3, default=c(2,1,2)
<code>col.segm</code>	Color of CI regions (low, median and high) for model, passed to <code>geom_polygon</code> , should be vector of length 3, default=c("steelblue", "gray50", "steelblue")
<code>alpha.segm</code>	alpha of CI regions (low, median and high) for model, passed to <code>geom_polygon</code> , default=0.6
<code>type</code>	type of VPC plot, 1=display model regions only, 2=as 1 + lines (low, median and high) of data, 3= as 2 + points for data. type=0 can also be used and then no graph is produced but instead a list with 2 dataframes is returned. one for the vpcreresult and on for the observed data in the vpctab file.

## Details

The `ggplot2` package is used for creating the VPC and `vpcfig2` have functionality for e.g. modify names of strata, order of and subset of strata's, and transform the x- and or y-axis before plotting.

The data used is either specified by the directory of the PsN generated files, or by specifying the file names of both the vpctab-file and the vpcreresult file. See examples 1 and 2 below.

Names of strata can be changed with the `strata.names` argument, if `strata.names` is NULL the names as specified in the vpcreresult file are used.

strata.subset can be used to select a subset of strata, and or to change the order of stratas. See example 3.

**Strata.subset** is matched with strata.names unless strata.names=NULL. If strata.names=NULL then strata.subset is matched to names as specified in vpcresult file. Default is to use all strata's (strata.subset=NULL).

The fy and fx arguments can be used to alter the scale of data plotted. For example, using `fy=function(y){log(y+1)}` is a convenient way to get log-scale for y-axis but with an off-set to show any values==0. Proper y tick marks can then be set using the `scale_y_continuous` ggplot2 function.

Since the returned object is a ggplot-class object it can be further modified to e.g. to log-scale for y.axis;

```
vpcfig2(...)+scale_y_log10(),
```

adding/changing labels & titles;

```
vpcfig2(...)+labs(y="Modified y-label", title="New title").
```

**Axis-limits** are preferably set using the ggplot2 function `coord_cartesian(...)`. This way data points outside the axis-limit are only hidden when plotting. Setting axis-limits using e.g. `scale_x_continuous`, `scale_x_log10`, or `xlim` may produce an incorrect vpcplot. This because data outside the axis-limits is deleted before plotting.

**Stratified** VPC's are created with `facet_wrap(~strata)` as default but can be modified to use `facet_grid` for setting the panel grid.

See examples below and documentation for ggplot2 for further details.

## Value

ggplot object of VPC plot

## Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "vpctab004.dat", package = "nonmem2R")
file2 <- system.file("extdata", "vpc_results.csv", package = "nonmem2R")

# Ex 1, produce VPC with default setting, here specifying both vpctab and vpcresult
vpcfig2(vpctab=file1,vpcresult=file2)

## Not run:
# Ex 2, produce VPC with default setting, here specifying only directory of vpc files
path1<-gsub("vpctab004.dat","",file1)
vpcfig2(vpcdir=path1)

## End(Not run)

# Ex 3, produce VPC with i) modifies strata names, ii) strata in reverse order, and
#      iii) labels
strata.names<-c("Group A","Group B")
xlab<-"Time after dose (hrs)"
ylab<-"Plasma Conc(mmol/L)"
```

```
vpcfig2(vpctab=file1,vpcresult=file2,strata.names=strata.names,strata.subset=2:1,
        xlab=xlab,ylab=ylab)

# Example using the fy argument to transform y-axis setting y-ticks using scale_y_continuous(...)
tmp<-c(1,2,3,4,6)
yticks<-c(0.1,tmp,tmp*10,tmp*100,tmp*1000)
vpcfig2(vpctab=file1,vpcresult=file2,fy=function(y){log(y+1)})+
  scale_y_continuous(breaks=log(yticks+1),labels=yticks,minor_breaks=NULL)

## Not run:
# Example changing to slog-scale using the ggplot2 function scale_y_log10
vpcfig2(vpctab=file1,vpcresult=file2)+scale_y_log10()

## End(Not run)
# Example changing y-axis label and adding figure title using ggplot2 function labs(...).
vpcfig2(vpctab=file1,vpcresult=file2)+labs(y="Modified y-label", title="New title")
```

# Index

`covload`, [2](#)

`extload`, [3](#)

`extToTable`, [4](#)

`grid.sim`, [5](#)

`sumoR`, [6](#)

`systemPSN`, [7](#)

`test.grid.sim`, [7](#)

`vpcfig`, [8](#)

`vpcfig2`, [10](#)