Package 'NonCompart'

November 15, 2026

Version 0.8.5

Date 2026-11-15

Title Noncompartmental Analysis for Pharmacokinetic Data

Description Conduct a noncompartmental analysis with industrial strength.
Some features are
1) Use of CDISC SDTM terms
2) Automatic or manual slope selection
3) Supporting both 'linear-up linear-down' and 'linear-up log-down' method
4) Interval(partial) AUCs with 'linear' or 'log' interpolation method
5) Steady-state analysis over the dosing interval (AUCTAU, CAVG, CL and Vz from AUCTAU)
6) Installation/Operational Qualification (IQ/OQ) reports in pdf.
After installation, qualify the package in your own environment:
run IQNCA() for Installation Qualification and OQNCA() for Operational
Qualification. Run writeMD5NCA() once after installation so the IQ
file-integrity check passes. To approve a report, sign it digitally in
Adobe Acrobat Reader (generate with sigField=TRUE, or run
addSigFieldNCA(), to add click-to-sign fields), instead of printing and
scanning; or use signPDFNCA()/verifyPDFNCA() for a scriptable signature.
* Reference: Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016. (ISBN:9198299107).

Author Kyun-Seop Bae [aut, cre]

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

Copyright 2016-, Kyun-Seop Bae

License GPL-3

Depends R (>= 3.5.0)

Imports utils, graphics, grDevices, stats, tools

Suggests openssl

NeedsCompilation no

LazyLoad yes

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

Repository CRAN

Date/Publication 2026-11-15 12:10:02 UTC

Contents

NonCompart-package . . . 2
addSigFieldNCA . . . 3
AUC . . . 5
BestSlope . . . 6
DetSlope . . . 7
gAUC . . . 8
gIntAUC . . . 9
gInterpol . . . 10
IntAUC . . . 11
Interpol . . . 12
IQNCA . . . 13
LinAUC . . . 14
LogAUC . . . 15
OQNCA . . . 16
PQNCA . . . 18
signPDFNCA . . . 20
Slope . . . 21
sNCA . . . 22
tblAUC . . . 26
tblNCA . . . 28
Unit . . . 31
UnitUrine . . . 32
writeMD5NCA . . . 33

Index **35**

`NonCompart-package` *Noncompartmental Analysis for Pharmacokinetic Data*

Description

It conducts a noncompartmental analysis(NCA) with industrial strength.

Details

The main functions are

`tblNCA` to perform NCA for many subjects.

`sNCA` to perform NCA for one subject.

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

1. Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016.
2. Shargel L, Yu A. Applied Biopharmaceutics and Pharmacokinetics. 7th ed. 2015.
3. Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics - Concepts and Applications. 4th ed. 2011.
4. Gibaldi M, Perrier D. Pharmacokinetics. 2nd ed. revised and expanded. 1982.

Examples

```
# Theoph and Indometh data: dose in mg, conc in mg/L, time in h
tblNCA(Theoph, key="Subject", colTime="Time", colConc="conc", dose=320,
       adm="Extravascular", doseUnit="mg", concUnit="mg/L")

tblNCA(Indometh, key="Subject", colTime="time", colConc="conc", dose=25,
       adm="Infusion", dur=0.5, doseUnit="mg", concUnit="mg/L", R2ADJ=0.9)

# For individual NCA
iAUC = data.frame(Name=c("AUC[0-12h]","AUC[0-24h]"), Start=c(0,0), End=c(12,24)) ; iAUC

x = Theoph[Theoph$Subject=="1","Time"]
y = Theoph[Theoph$Subject=="1","conc"]

sNCA(x, y, dose=320, doseUnit="mg", concUnit="mg/L", timeUnit="h", iAUC=iAUC)
sNCA(x, y, dose=320, concUnit="mg/L", iAUC=iAUC)
```

---

addSigFieldNCA *Add Acrobat-Reader signature fields to a report PDF*

Description

Insert empty digital-signature form fields (AcroForm `/Sig` fields) into a PDF so it can be signed in Adobe Acrobat Reader (the free reader) with one click: open the PDF, click a field, and sign with a Digital ID. The field is added with a pure base-R PDF incremental update, with no external tools or packages, so it works wherever R does. It is designed for the simple PDFs produced by `IQNCA` and `OQNCA`.

Acrobat Reader can also sign a PDF that has no field at all ("All tools" > "Use a certificate" > "Digitally sign", then drag a rectangle). This function only makes the workflow click-to-sign by pre-placing labelled fields.

Usage

```
addSigFieldNCA(pdf, out = pdf, page = 1L,
            fieldNames = c("Performed_by", "Reviewed_by"), rects = NULL)
```

Arguments

pdf	path to the input PDF.
out	path to write the result; defaults to overwriting pdf.
page	1-based page number to place the field(s) on (the signature page is page 1 of IQNCA/OQNCA reports).
fieldNames	character vector of field names; one signature field is added per name. The names appear in Acrobat's Signature panel.
rects	optional list of numeric length-4 rectangles c(x0,y0,x1,y1) in PDF points (origin at the lower-left of the page), one per field. If NULL, sensible stacked rectangles are used within the page margins.

Details

The function appends an incremental-update section that adds an /AcroForm to the document catalog (with /SigFlags 3), one /Widget signature annotation per field, and the field references to the page's /Annots. The original content is left untouched. The resulting fields are unsigned; the actual cryptographic signature is applied by Acrobat Reader using the signer's Digital ID. Only classic cross-reference-table PDFs are supported (those written by R's pdf device); the function stops on cross-reference-stream PDFs.

Value

Invisibly, the output path. A PDF with signature fields is written as a side effect.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

IQNCA, OQNCA, signPDFNCA

Examples

```
#IQNCA("NonCompart-IQ-Report.pdf", performedBy = "Kyun-Seop Bae")
#addSigFieldNCA("NonCompart-IQ-Report.pdf")       # adds Performed_by / Reviewed_by fields
## or in one step:
#IQNCA("NonCompart-IQ-Report.pdf", performedBy = "Kyun-Seop Bae", sigField = TRUE)
## then open in Acrobat Reader and click each field to sign with your Digital ID.
```

AUC *Calculate Area Under the Curve (AUC) and Area Under the first Moment Curve (AUMC) in a table format*

Description

Calculate Area Under the Curve(AUC) and the first Moment Curve(AUMC) in two ways; 'linear trapezoidal method' or 'linear-up and log-down' method. Return a table of cumulative values.

Usage

```
AUC(x, y, down = "Linear")
```

Arguments

x	vector values of independent variable, usually time
y	vector values of dependent variable, usually concentration
down	either of "Linear" or "Log" to indicate the way to calculate AUC and AUMC

Details

down="Linear" means linear trapezoidal rule with linear interpolation. down="Log" means linear-up and log-down method. A segment that comes down to a zero concentration cannot take the log rule and is calculated by the linear trapezoidal rule. Up to version 0.8.4 such a segment added nothing to AUC and AUMC with down="Log".

Value

Table with two columns, AUC and AUMC; the first column values are cumulative AUCs and the second column values cumulative AUMCs.

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics - Concepts and Applications. 4th ed. pp687-689. 2011.

See Also

LinAUC, LogAUC

Examples

```
AUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"])
AUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"], down="Log")
```

`BestSlope` *Choose the best-fit slope for the log(y) and x regression by the criteria of adjusted R-square.*

Description

It sequentially fits (log(y) ~ x) from the last point of x to the previous points with at least 3 points. It chooses a slope the highest adjusted R-square. If the difference is less then 1e-4, it pickes longer slope.

Usage

```
BestSlope(x, y, adm = "Extravascular", TOL=1e-4, excludeDelta = 1)
```

Arguments

`x`	vector values of x-axis, usually time
`y`	vector values of y-axis, usually concentration
`adm`	one of `"Bolus"` or `"Infusion"` or `"Extravascular"` to indicate drug administration mode
`TOL`	tolerance. Adjusted R-squared values that differ by less than this are considered equal, and the fit with more points is then chosen.
`excludeDelta`	Improvement of R2ADJ larger than this value could exclude the last point. Default value 1 is for the compatibility with other software.

Details

Choosing the best terminal slope (y in log scale) in pharmacokinetic analysis is somewhat challenging, and it could vary by analysis performer. Phoenix WinNonlin chooses a slope with highest adjusted R-squared and the longest one. The difference of adjusted R-Squared less than TOL considered to be 0. This function uses ordinary least square method (OLS). Author recommends to use `excludeDelta` option with about 0.3.

Value

`R2`	R-squared
`R2ADJ`	adjusted R-squared
`LAMZNPT`	number of points used for slope
`LAMZ`	negative of the slope, lambda_z
`b0`	intercept of the regression line
`CORRXY`	correlation of log(y) and x
`LAMZLL`	earliest x for lambda_z
`LAMZUL`	last x for lambda_z
`CLSTP`	predicted y value at the last point, predicted concentration for the last time point

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

Slope

Examples

```
BestSlope(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"])
BestSlope(Indometh[Indometh$Subject==1, "time"], Indometh[Indometh$Subject==1, "conc"],
          adm="Bolus")
```

DetSlope *Determine slope for the log(y) and x regression manually*

Description

You choose a slope for terminal half-life.

Usage

```
DetSlope(x, y, SubTitle="", sel.1=0, sel.2=0)
```

Arguments

x	vector values of x-axis, usually time
y	vector values of y-axis, usually concentration
SubTitle	subtitle to be shown on the plot
sel.1	default index of the first element to use
sel.2	default index of the last element to use

Details

Sometimes BestSlope cannot find terminal slope satisfactorily. Then you can use this function to choose manually. It returns the same format result with BestSlope with an attribute indicating used points.

Value

R2	R-squared
R2ADJ	adjusted R-squared
LAMZNPT	number of points used for the slope
LAMZ	negative of the slope, lambda_z
b0	intercept of the regression line

CORRXY	correlation of log(y) and x
LAMZLL	earliest x for lambda_z
LAMZUL	last x for lambda_z
CLSTP	predicted y value at the last point, predicted concentration for the last time point

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

Slope

Examples

```
DetSlope(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"])
DetSlope(Indometh[Indometh$Subject==2, "time"], Indometh[Indometh$Subject==2, "conc"])
```

---

gAUC *General Area Under the Curve*

Description

General AUC function for Emax, TEmax and AUCs

Usage

```
gAUC(x, y, Ymax = "Emax", XofYmax = "TEmax", AUCname = "AUEClast", iAUC = "",
     Outer = "NEAREST")
```

Arguments

x	usually time
y	usually concentration or effect. This can be negative/
Ymax	usually Cmax or Emax
XofYmax	usually Tmax or TEmax
AUCname	usually AUClast or AUEClast
iAUC	a data.frame to calculate interval AUCs
Outer	indicates how to do the out of x range point

Details

This is a general purpose AUC function. It calculates only Cmax(Emax), Tmax(TEmax) and AUCs(AUECs). This can be used for effect(pharmacodynamic) data which has negative values. For concentration data, use IntAUC.

Value

Column names can vary according to the options.

Emax	maximum y value
TEmax	x value at the maximum y value
AUEClast	Area under the y versus x curve
iAUCs	Columns from iAUC input

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
# For one subject
x = Theoph[Theoph$Subject=="1", "Time"]
y = Theoph[Theoph$Subject=="1", "conc"]
gAUC(x, y)

iAUC = data.frame(Name=c("AUC[0-12h]","AUC[0-24h]"), Start=c(0,0), End=c(12,24))
gAUC(x, y, iAUC=iAUC)
```

---

gIntAUC *Calculate interval AUC of general form*

Description

It calculates interval AUC of general form. This is useful for pharmacodynamic data.

Usage

```
gIntAUC(x, y, t1, t2, Outer = "NEAREST")
```

Arguments

x	vector values of independent variable, usually time
y	vector values of dependent variable, usually concentration
t1	start time for AUC
t2	end time for AUC
Outer	indicates how to do the out of x range point

Details

This calculates an interval (partial) AUC (from t1 to t2) with the given series of x and y. If t1 and/or t2 cannot be found within x vector, it interpolates. If t1 and/or t2 are out of x range, it uses the nearest value. For concentration data, use IntAUC.

Value

return interval AUC value (scalar)

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

gAUC, gInterpol, tblAUC

Examples

```
gIntAUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"], t1=0.5, t2=11)
```

---

gInterpol *Interpolate y value for general y value not for concentration*

Description

It interpolates y value when a corresponding x value (xnew) does not exist within x vector

Usage

```
gInterpol(x, y, xnew, Outer="NEAREST")
```

Arguments

x	vector values of x-axis, usually time
y	vector values of y-axis, usually concentration
xnew	new x point to be interpolated, usually new time point
Outer	indicates how to do the out of x range point

Details

This function interpolate y value, if xnew is not in x vector. If xnew is in the x vector, it just returns the given x and y vector. This function usually is called by gIntAUC function. Returned vector is sorted in the order of increasing x values.

Value

new x and y vector containing xnew and ynew point

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

gIntAUC

Examples

```
x = 1:10 + 0.1
y = -2*x + 40.2
gInterpol(x, y, 1.5)
gInterpol(x, y, 0.5) # Out of range, Left
gInterpol(x, y, 11) # Out of range, Right
```

IntAUC *Calculate interval AUC*

Description

It calculates interval AUC

Usage

```
IntAUC(x, y, t1, t2, Res, down = "Linear")
```

Arguments

x	vector values of independent variable, usually time
y	vector values of dependent variable, usually concentration
t1	start time for AUC
t2	end time for AUC
Res	result from sNCA function
down	either of "Linear" or "Log" to indicate the way to calculate AUC

Details

This calculates an interval (partial) AUC (from t1 to t2) with the given series of x and y. If t1 and/or t2 cannot be found within x vector, it interpolates according to the down option.

The rules are as follows. Within the observations the concentration at t1 or t2 is interpolated, and the area is calculated according to down. A trailing zero concentration is an observation like any other, so a time between the last positive concentration and that zero is interpolated linearly. After the last observation the concentration is predicted from the terminal regression, exp(b0 - LAMZ * t), and that part is integrated by the log trapezoidal rule whatever down is. Without LAMZ the result is NA. If both t1 and t2 are at or after the time of the last positive concentration (TLST), the whole interval is integrated by the log trapezoidal rule. A segment with a zero concentration at either end, or with the same concentration at both ends, is integrated by the linear trapezoidal rule in any case. Up to version 0.8.4 an interval starting after TLST returned NA.

Value

return interval AUC value (scalar)

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

1. Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016.
2. Shargel L, Yu A. Applied Biopharmaceutics and Pharmacokinetics. 7th ed. 2015.
3. Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics - Concepts and Applications. 4th ed. 2011.
4. Gibaldi M, Perrier D. Pharmacokinetics. 2nd ed. revised and expanded. 1982.

See Also

AUC, Interpol

Examples

```
Res = sNCA(Theoph[Theoph$Subject==1,"Time"], Theoph[Theoph$Subject==1, "conc"],
           dose=320, concUnit="mg/L")
IntAUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"], t1=0.5, t2=11, Res)
```

Interpol *Interpolate y value*

Description

It interpolates y value when a corresponding x value (xnew) does not exist within x vector

Usage

```
Interpol(x, y, xnew, Slope, b0, down = "Linear")
```

Arguments

x	vector values of x-axis, usually time
y	vector values of y-axis, usually concentration
xnew	new x point to be interpolated, usually new time point
Slope	slope of regression log(y) ~ x
b0	y value of just left point of xnew
down	either of "Linear" or "Log" to indicate the way to interpolate

Details

This function interpolate y value, if xnew is not in x vector. If xnew is in x vector, it just returns the given x and y vector. This function usually is called by `IntAUC` function. Returned vector is sorted in the order of increasing x values.

Value

new x and y vector containing xnew and ynew point

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

`IntAUC`

Examples

```
x = 10:1 + 0.1
y = -2*x + 40.2
Interpol(x, y, 1.5)
Interpol(x, y, 1.5, down="Log")
```

IQNCA *Installation Qualification (IQ) report to a pdf file*

Description

Generate a self-contained PDF report documenting whether the **NonCompart** package is correctly installed, intact, loadable, and operational in the user's own R environment. It is intended as Installation Qualification (IQ) evidence. The report uses only base R and the package's own pdf helpers, so it requires no LaTeX, pandoc, or other external tools. The signature (approval) page is placed first and the report uses 1 inch margins on every side, with Letter paper in a United States locale and A4 elsewhere.

Usage

```
IQNCA(fileName = "NonCompart-IQ-Report.pdf", pkgs = "NonCompart",
      functional = TRUE, performedBy = "", paper = "auto", sigField = FALSE)
```

Arguments

`fileName`	file name to save the PDF report.
`pkgs`	character vector of package names to qualify; defaults to `"NonCompart"`. Declared dependencies are checked automatically.

functional	if TRUE, run a small operational check: a known NCA computation (Theophylline subject 1) via sNCA is compared with an external WinNonlin reference within a relative tolerance of 1e-3, plus two control tests that verify the comparator itself.
performedBy	name of the person performing the qualification, printed on the signature page. Defaults to the login name.
paper	paper size: "letter", "a4", or "auto" (the default; Letter in a United States locale, A4 elsewhere).
sigField	if TRUE, add Adobe Acrobat Reader signature fields to the finished report (via addSigFieldNCA) so it can be signed with one click in Acrobat Reader.

Details

The report contains: the test environment (R version, platform, OS, locale, library paths); the installed package version, location and declared-dependency satisfaction; file integrity via checkMD5sums (PASS / FAIL / WARN when no manifest is present - see writeMD5NCA); namespace load and core exports; an optional functional verification; an overall QUALIFIED / NOT QUALIFIED verdict; a sessionInfo appendix; and a per-file md5 checksum appendix.

Value

Invisibly, a list with fileName, qualified, paper, checks (a data frame of every check), and the counts nPass, nFail, nWarn. A PDF file is written to fileName.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

OQNCA, writeMD5NCA, sNCA, tblNCA

Examples

```
#IQNCA()
#res <- IQNCA(performedBy = "Jane Doe"); res$qualified
```

LinAUC *Area Under the Curve(AUC) and Area Under the first Moment Curve(AUMC) by linear trapezoidal method*

Description

It calculates AUC and AUMC using the linear trapezoidal method

Usage

```
LinAUC(x, y)
```

Arguments

x	vector values of the independent variable, usually time
y	vector values of the dependent variable, usually concentration

Details

This function returns AUC and AUMC by the linear trapezoidal method.

Value

AUC	area under the curve
AUMC	area under the first moment curve

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

1. Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016.
2. Shargel L, Yu A. Applied Biopharmaceutics and Pharmacokinetics. 7th ed. 2015.
3. Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics - Concepts and Applications. 4th ed. 2011.
4. Gibaldi M, Perrier D. Pharmacokinetics. 2nd ed. revised and expanded. 1982.

See Also

LogAUC, AUC

Examples

```
LinAUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"])
AUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"]) # compare the last line
```

LogAUC *Area Under the Curve(AUC) and Area Under the first Moment Curve(AUMC) by linear-up log-down method*

Description

It calculates AUC and AUMC using the linear-up log-down method

Usage

```
LogAUC(x, y)
```

Arguments

x	vector values of the independent variable, usually time
y	vector values of the dependent variable, usually concentration

Details

This function returns AUC and AUMC by the linear-up log-down method.

Value

AUC	area under the curve
AUMC	area under the first moment curve

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

1. Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016.
2. Shargel L, Yu A. Applied Biopharmaceutics and Pharmacokinetics. 7th ed. 2015.
3. Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics - Concepts and Applications. 4th ed. 2011.
4. Gibaldi M, Perrier D. Pharmacokinetics. 2nd ed. revised and expanded. 1982.

See Also

LinAUC,AUC

Examples

```
LogAUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"])
# Compare the last line with the above
AUC(Theoph[Theoph$Subject==1, "Time"], Theoph[Theoph$Subject==1, "conc"], down="Log")
```

OQNCA *Operational Qualification (OQ) report to a pdf file*

Description

Generate a self-contained PDF report that documents whether the **NonCompart** noncompartmental analysis engine reproduces pre-specified reference results on the user's own machine, within a pre-specified relative tolerance. It is intended as Operational Qualification (OQ) evidence. The report lists every computed value next to its reference value, with the absolute and relative difference and a pass flag, then an overall verdict. It uses only base R and the package's pdf helpers (no LaTeX). The signature page is first; 1 inch margins; Letter paper in a United States locale and A4 elsewhere.

Usage

```
OQNCA(fileName = "NonCompart-OQ-Report.pdf", cases = NULL, tol = 0.001,
      refDir = system.file("OQ", package = "NonCompart"), performedBy = "",
      paper = "auto", sigField = FALSE)
```

Arguments

`fileName`	file name to save the PDF report.
`cases`	a list of OQ case definitions. If `NULL`, the built-in set of fourteen scenarios is used: eight concordance (Tier B) ones (Theophylline and Indometacin; extravascular / IV bolus / IV infusion; linear-up-linear-down and linear-up-log-down AUC) and six Tier A cases on synthetic fixtures whose terminal phase is exactly mono-exponential, so their reference values are known in closed form. Each case is a list with elements `id`, `desc`, either `dataset` (the name of a **datasets** object) or `data` (an inline data frame), `key`, `colTime`, `colConc`, `args` (further `tblNCA` arguments), `ref` (the reference csv file name in `refDir`), and optionally `roundTrip` (re-run the analysis with the `UsedPoints` the first run reported and require every parameter to be unchanged).
`tol`	acceptance tolerance: a parameter passes when the symmetric relative difference `abs(R - W) / ((abs(R) + abs(W))/2)` is at most `tol`; an absolute fallback is used when both values are essentially zero.
`refDir`	directory holding the reference csv files and the '`RptCfg.csv`' PPTESTCD-to-reference-column name map. Defaults to the `OQ` folder installed with **NonCompart**.
`performedBy`	name of the person performing the qualification, printed on the signature page. Defaults to the login name.
`paper`	paper size: `"letter"`, `"a4"`, or `"auto"` (the default; Letter in a United States locale, A4 elsewhere).
`sigField`	if `TRUE`, add Adobe Acrobat Reader signature fields to the finished report (via `addSigFieldNCA`) so it can be signed with one click in Acrobat Reader.

Details

For each case the function runs `tblNCA` on the (frozen) input data, maps the CDISC PPTESTCD result columns to the column names of the reference tables using '`RptCfg.csv`' in `refDir`, matches comparable parameters to the reference table (robust to `%` vs `.` naming), and compares every parameter for every subject. A case passes only when all of its comparisons pass.

Most bundled reference values are WinNonlin 6.4 NCA outputs. These establish *concordance* with the de-facto reference implementation (Tier B); they are not, by themselves, an independent verification.

Since 0.8.3 the set also contains three Tier A cases, whose reference values are derived from first principles rather than from another implementation. They run a synthetic single-subject fixture carrying a lag-phase zero, an embedded below-the- limit-of-quantification sample inside the lambda z window, and trailing BLQ, and they cover automatic slope selection, an explicit `UsePoints` fit, and a round trip that feeds the reported `UsedPoints` back in. Neither `Theoph` nor `Indometh` contains

a trailing or an embedded BLQ sample or exercises `UsePoints`, so the eight Tier B cases alone cannot reach those paths.

Since 0.8.4 two further Tier A cases qualify the steady-state path (`SS = TRUE` with `tau`): a single dosing interval whose concentrations are small integers, so that AUCTAU is the hand-summable trapezoid 66, CAVG is 66/12 and CL/F is the dose over 66, once with tau given in hours and once with the same tau given in days. Every reference value in that file was written out from the definition of the parameter, not produced by this package or by any other software.

Since 0.8.5 a sixth Tier A case qualifies the intravascular steady-state path: a one-compartment bolus at steady state (dose 100 every 12 h, volume 20, half-life 4 h) run with the log-down method, for which every area is exact. Its reference row is closed form throughout; in particular the MRT to infinity is 1/k and Vss equals the volume, 20. A case now also fails when a parameter or a subject of its reference table is absent from the result, or when a subject occurs twice; such rows used to be left out of the comparison.

Value

Invisibly, a list with `fileName`, `qualified`, `tol`, `nCases`, `nFailCases`, `paper`, and `results` (a per-case list whose `comp` element is the full table of every comparison). A PDF is written to `fileName`.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

`IQNCA`, `writeMD5NCA`, `tblNCA`

Examples

```
#OQNCA()
#res <- OQNCA(); res$qualified; res$results[[1]]$comp
```

`PQNCA` *Performance Qualification (PQ) report to a pdf file*

Description

Generate a self-contained PDF report documenting a Performance Qualification (PQ) of noncompartmental analysis for the user's own data and intended workflow. Whereas `OQNCA` compares the engine to fixed reference values, PQ runs the analysis the user actually intends to run, on the user's (or representative) data, and checks that the result is reproducible, scientifically plausible, and (optionally) matches the user's pre-specified expected values, with a documented, signed record including a fingerprint (md5) of the input data. The report uses the package's own pdf helpers (no LaTeX) and the signature/approval page is placed first.

Usage

```
PQNCA(fileName = "NonCompart-PQ-Report.pdf", concData, key = "Subject",
      colTime = "Time", colConc = "conc", ..., expected = NULL,
      acceptance = NULL, performedBy = "", paper = "auto", sigField = FALSE)
```

Arguments

fileName	file name to save the PDF report.
concData	the user's concentration data set (data frame) to qualify.
key	column name(s) identifying each profile.
colTime	column name for time.
colConc	column name for concentration.
...	further arguments passed to tblNCA (e.g. dose, adm, dur, down, R2ADJ, concUnit), i.e. the analysis options the user intends to use in routine work.
expected	optional data frame of the user's pre-specified expected parameter values (a key column plus PPTESTCD columns such as CMAX, AUCLST); each is compared to the computed value within acceptance$tol. This is the strongest, user-owned PQ acceptance.
acceptance	a list of PQ acceptance criteria; if NULL, sensible defaults are used (maxPctExtrap=20, minR2adj=0.80, tol=1e-3, reproTol=0, and flags checkTmaxRange, checkCmaxGEClast, checkAUCINFgeAUClast, failOnNonEvaluable). Adjust to your intended use / SOP.
performedBy	name of the person performing the qualification, printed on the signature page. Defaults to the login name.
paper	paper size: "letter", "a4", or "auto" (the default; Letter in a United States locale, A4 elsewhere).
sigField	if TRUE, add Adobe Acrobat Reader signature fields under the Performed by / Reviewed by lines (via addSigFieldNCA).

Details

The report contains: the intended-use inputs and the input-data md5 fingerprint; the PQ acceptance criteria; a reproducibility check (the analysis is re-run on the same data and must reproduce identical results); per-profile performance/plausibility checks (Cmax > 0, Cmax >= Clast, Tmax within the observed time range, AUClast > 0, and for profiles with a terminal slope: AUCINF_obs >= AUClast, AUC %Extrap within limit, adjusted R-squared above limit, half-life positive and finite); an optional comparison to user-supplied expected values; an overall QUALIFIED / NOT QUALIFIED verdict; and a sessionInfo appendix.

PQ acceptance criteria and the choice of representative data are the user's responsibility; this function provides the tool and configurable defaults.

Value

Invisibly, a list with fileName, qualified, paper, results (the tblNCA table), reproducible, profiles (per-profile checks), fingerprint, nProfFail, nNonEval, and expected. A PDF is written to fileName.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

IQNCA, OQNCA, tblNCA

Examples

```
#Theoph2 <- as.data.frame(Theoph); Theoph2$Subject <- as.numeric(as.character(Theoph2$Subject))
#PQNCA("MyStudy-PQ.pdf", Theoph2, "Subject", "Time", "conc", dose = 320,
#      concUnit = "mg/L", performedBy = "Kyun-Seop Bae", sigField = TRUE)
```

---

signPDFNCA *Digitally sign and verify a PDF report*

Description

Apply a cryptographic digital signature to a (report) PDF, as an alternative to the print-sign-scan workflow, and verify it. `signPDFNCA` signs the PDF bytes with the signer's private key (RSA or EC) using SHA-256 and writes a detached signature sidecar '`<pdf>.sig`' (and, by default, the signer's public key '`<pdf>.pubkey.pem`'). `verifyPDFNCA` confirms, with the signer's public key or certificate, that the PDF is byte-for-byte intact and was signed by that key (tamper-evidence and non-repudiation). These functions require the **openssl** package.

This is a *detached* signature (a separate '`.sig`' file), not a PAdES signature embedded inside the PDF; embedding a visible in-viewer signature requires a dedicated PDF tool (e.g. Adobe Acrobat or 'pyhanko'). The detached signature is cryptographically equivalent for integrity and non-repudiation.

Usage

```
signPDFNCA(pdf, key, password = NULL, signer = "", role = "",
        sigFile = paste0(pdf, ".sig"), writePubkey = TRUE)

verifyPDFNCA(pdf, sigFile = paste0(pdf, ".sig"), pubkey = paste0(pdf, ".pubkey.pem"))
```

Arguments

`pdf`	path to the PDF file to sign or verify.
`key`	the signer's private key: a path to a PEM key file or an **openssl** key object. Create one once with, e.g., `openssl::rsa_keygen()` and `openssl::write_pem()`.
`password`	password for an encrypted private key, or `NULL`.
`signer`	name of the signer, recorded in the signature sidecar. Defaults to the login name.
`role`	optional role recorded in the sidecar, e.g. `"Performed by"` or `"Reviewed by"`.
`sigFile`	path of the signature sidecar to write (`signPDFNCA`) or read (`verifyPDFNCA`).

writePubkey	if TRUE, also write the signer's public key next to the PDF for convenience. For non-repudiation, verify against the signer's independently-trusted public key or certificate, not one shipped with the file.
pubkey	the signer's public key or certificate used to verify: a PEM path (public key or X.509 certificate) or an **openssl** pubkey/cert object.

Value

signPDFNCA invisibly returns a list with the sidecar path, the PDF sha256, the signer, and the public-key fingerprint; it writes the '.sig' sidecar as a side effect. verifyPDFNCA invisibly returns TRUE only if both the signature is valid and the recorded hash matches, and prints a human-readable result.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

IQNCA, OQNCA, writeMD5NCA

Examples

```
#key <- openssl::rsa_keygen()
#openssl::write_pem(key, "signer_key.pem")
#IQNCA("NonCompart-IQ-Report.pdf", performedBy = "Kyun-Seop Bae")
#signPDFNCA("NonCompart-IQ-Report.pdf", "signer_key.pem", signer = "Kyun-Seop Bae")
#verifyPDFNCA("NonCompart-IQ-Report.pdf")
```

Slope *Get the Slope of regression log(y) ~ x*

Description

It calculates the slope with linear regression of log(y) ~ x

Usage

```
Slope(x, y)
```

Arguments

x	vector values of the independent variable, usually time
y	vector values of the dependent variable, usually concentration

Details

With time-concentration curve, you frequently need to estimate slope in log(concentration) ~ time. This function is usually called by `BestSlope` function, and you seldom need to call this function directly.

Value

`R2`	R-squared
`R2ADJ`	adjusted R-squared
`LAMZNPT`	number of points used for slope
`LAMZ`	negative of the slope, lambda_z
`b0`	intercept of the regression line
`CORRXY`	correlation of log(y) and x
`LAMZLL`	earliest x for lambda_z
`LAMZUL`	last x for lambda_z

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

`BestSlope`

Examples

```
Slope(Indometh[Indometh$Subject==1, "time"], Indometh[Indometh$Subject==1, "conc"])
```

---

sNCA *Simplest NCA*

Description

This is the work-horse function for NCA.

Usage

```
sNCA(x, y, dose = 0, adm = "Extravascular", dur = 0, doseUnit = "mg", timeUnit = "h",
    concUnit = "ug/L", iAUC = "", down = "Linear", R2ADJ = 0.7, MW = 0, SS = FALSE,
     tau = NA, unitTau = "h", Keystring="", excludeDelta = 1, UsePoints = NULL)
```

Arguments

x	usually time, a vector
y	usually concentration, a vector
dose	given amount, not amount per body weight, a scalar
adm	one of "Bolus" or "Infusion" or "Extravascular" to indicate drug administration mode
dur	duration of infusion, a scalar
doseUnit	unit of dose
timeUnit	unit of time
concUnit	unit of concentration
iAUC	interval AUCs to calculate: a data.frame with the columns Name, Start and End. See IntAUC for the rules; an interval may start and end after the last sample when LAMZ is available.
down	either of "Linear" or "Log" to indicate the way to calculate AUC and AUMC
R2ADJ	Minimum adjusted R-square value to determine terminal slope automatically
MW	molecular weight of the drug
SS	if steady-state, this should be TRUE and tau must be given. AUCTAU (AUC over one dosing interval, 0 to tau) is then used instead of AUCIFO (AUCinf) for the calculation of Vz (VZFO, VZO) and CL (CLFO, CLO), and AUCTAU and CAVG are added to the result. CMAX and TMAX are then taken from the samples within the dosing interval, and without a sample at time 0 the lowest concentration observed within the dosing interval is inserted there (0 after a single dose). The MRT to infinity (MRTEVIFO, MRTEVIFP, MRTIVIFO, MRTIVIFP) is (AUMC from 0 to tau + tau * (AUCinf - AUCTAU)) / AUCTAU, and Vdss (VSSO, VSSP) is that MRT times CLO. The Pred-based twins of Vz and CL (VZFP, CLFP, VZP, CLP) are NA at steady state. Up to version 0.8.4 the MRT to infinity and VSSP were NA at steady state and VSSO was MRTIVLST times CLO. Up to version 0.8.3 AUCLST (AUClast) was used in place of AUCTAU, which is only right when the last sample falls exactly at tau.
tau	dosing interval, a positive scalar in the unit of unitTau. Required when SS = TRUE; ignored, with a warning, otherwise.
unitTau	unit of tau: "h" (default), "min", "s", "d", "wk" or their common spellings. tau is converted into timeUnit before use.
Keystring	a text string to be shown at the plot in case of manual selection of terminal slope
excludeDelta	Improvement of R2ADJ larger than this value could exclude the last point. Default value 1 is for the compatibility with other software.
UsePoints	Indices of points to calculate terminal slope. Vaules of use points should not be zero. Use only when automatic determination is not satisfactory. If this is used, R2ADJ option is ignored.

Details

This replaced previous IndiNCA. Author recommends to use excludeDelta option with about 0.3.

Value

CMAX	maximum concentration, Cmax; within the dosing interval (0 to tau) when SS = TRUE
CMAXD	dose normalized Cmax, CMAX / Dose, Cmax / Dose
TMAX	time of maximum concentration, Tmax; within the dosing interval (0 to tau) when SS = TRUE
TLAG	time to observe the first non-zero concentration, for extravascular administration only
CLST	last positive concentration observed, Clast
CLSTP	last positive concentration predicted, Clast_pred
TLST	time of last positive concentration, Tlast
LAMZHL	half-life by lambda z, ln(2)/LAMZ
LAMZ	lambda_z negative of the best-fit terminal slope
LAMZLL	earliest time for LAMZ
LAMZUL	last time for LAMZ
LAMZNPT	number of points for LAMZ
CORRXY	correlation of log(concentration) and time
R2	R-squared
R2ADJ	R-squared adjusted
C0	back extrapolated concentration at time 0, for intravascular bolus administration only
AUCLST	AUC from 0 to TLST
AUCALL	AUC using all the given points, including trailing zero concentrations
AUCIFO	AUC infinity observed
AUCIFOD	AUCIFO / Dose
AUCIFP	AUC infinity predicted using CLSTP instead of CLST
AUCIFPD	AUCIFP / Dose
AUCPEO	AUC % extrapolation observed
AUCPEP	AUC % extrapolated for AUCIFP
AUCPBEO	AUC % back extrapolation observed, for bolus IV administration only
AUCPBEP	AUC % back extrapolation predicted with AUCIFP, for bolus IV administration only
AUMCLST	AUMC to the TLST
AUMCIFO	AUMC infinity observed using CLST
AUMCIFP	AUMC infinity determined by CLSTP
AUMCPEO	AUMC % extrapolated observed
AUMCPEP	AUMC % extrapolated predicted

AUCTAU	AUC over one dosing interval, from 0 to tau, calculated like an interval AUC of iAUC (see IntAUC): interpolated according to down when tau falls between two samples, extrapolated with LAMZ when tau is beyond the last sample. With SS = TRUE only
CAVG	average concentration over the dosing interval at steady state, AUCTAU / tau. With SS = TRUE only
MRTIVLST	mean residence time (MRT) to TLST, for intravascular administration
MRTIVIFO	mean residence time (MRT) infinity using CLST, for intravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFO - AUCTAU)) / AUCTAU, less half the infusion duration
MRTIVIFP	mean residence time (MRT) infinity using CLSTP, for intravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFP - AUCTAU)) / AUCTAU, less half the infusion duration
MRTEVLST	mean residence time (MRT) to TLST, for extravascular administration
MRTEVIFO	mean residence time (MRT) infinity using CLST, for extravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFO - AUCTAU)) / AUCTAU
MRTEVIFP	mean residence time (MRT) infinity using CLSTP, for extravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFP - AUCTAU)) / AUCTAU
VZO	volume of distribution determined by LAMZ and AUCIFO (AUCTAU when SS = TRUE), for intravascular administration
VZP	volume of distribution determined by LAMZ and AUCIFP, for intravascular administration; NA when SS = TRUE
VZFO	VZO for extravascular administration, VZO/F, F is bioavailability
VZFP	VZP for extravascular administration, VZP/F, F is bioavailability; NA when SS = TRUE
CLO	clearance using AUCIFO (dose / AUCTAU when SS = TRUE), for intravascular administration
CLP	clearance using AUCIFP, for intravascular administration; NA when SS = TRUE
CLFO	CLO for extravascular administration, CLO/F, F is bioavailability
CLFP	CLP for extravascular administration, CLP/F, F is bioavailability; NA when SS = TRUE
VSSO	volume of distribution at steady state using CLST, MRTIVIFO times CLO, for intravascular administration only
VSSP	volume of distribution at steady state using CLSTP, MRTIVIFP times CLP (times CLO when SS = TRUE), for intravascular administration only
units	An attribute to show units
UsedPoints	An attribute to show indices of used points

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

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

See Also

help, tblNCA

Examples

```
# For one subject
x = Theoph[Theoph$Subject=="1","Time"]
y = Theoph[Theoph$Subject=="1","conc"]

sNCA(x, y, dose=320, doseUnit="mg", concUnit="mg/L", timeUnit="h")
sNCA(x, y, dose=320, concUnit="mg/L")

iAUC = data.frame(Name=c("AUC[0-12h]","AUC[0-24h]"), Start=c(0,0), End=c(12,24))
sNCA(x, y, dose=320, doseUnit="mg", concUnit="mg/L", timeUnit="h", iAUC=iAUC)

MW = 180.164 # Molecular weight of theophylline

sNCA(x, y/MW, dose=320, doseUnit="mg", concUnit="mmol/L", timeUnit="h")
sNCA(x, y/MW, dose=320, doseUnit="mg", concUnit="mmol/L", timeUnit="h", MW=MW)
sNCA(x, y, dose=320/MW, doseUnit="mmol", concUnit="mg/L", timeUnit="h", MW=MW)
sNCA(x, y/MW, dose=320/MW, doseUnit="mmol", concUnit="mmol/L", timeUnit="h", MW=MW)

sNCA(x, y/MW, dose=320/MW, doseUnit="mmol", concUnit="mmol/L", timeUnit="h", MW=MW)
sNCA(x, y/MW, doseUnit="mmol", concUnit="mmol/L", timeUnit="h", MW=MW)
sNCA(x, y/MW, dose=as.numeric(NA), doseUnit="mmol", concUnit="mmol/L", timeUnit="h",
     MW=MW)

sNCA(x, y, dose=320, concUnit="mg/L", timeUnit="hr")
sNCA(x*60, y, dose=320, concUnit="mg/L", timeUnit="min")

# Steady state: one dosing interval of a q12h regimen, sampled from the pre-dose trough to
# the next dose. tau is the dosing interval; AUCTAU, CAVG, CLFO and VZFO are derived over it.
xs = c(0, 1, 2, 4, 8, 12)
ys = c(2, 6, 10, 8, 4, 2)
sNCA(xs, ys, dose=100, concUnit="mg/L", SS=TRUE, tau=12)
sNCA(xs, ys, dose=100, concUnit="mg/L", SS=TRUE, tau=0.5, unitTau="d") # the same result
```

---

tblAUC *Table output of gAUCs*

Description

Do multiple AUCs and returns a result table. See gAUC for more detail i.e. iAUC

Usage

```
tblAUC(Data, key = "Subject", colX = "Time", colY = "Y", iAUC = "",
      Ymax = "Emax", XofYmax = "TEmax", AUCname = "AUEClast", Outer = "NEAREST")
```

Arguments

Data	data table name
key	column names of Data to be shown in the output table
colX	column name for x axis
colY	column name for y axis
iAUC	a data.frame to calculate interval AUCs
Ymax	usually Cmax or Emax
XofYmax	usually Tmax or TEmax
AUCname	usually AUClast or AUEClast
Outer	indicates how to do the out of x range point

Details

Tabular output of AUC with many subjects. This calculates only Cmax(Emax), Tmax(TEmax), AUCs

Value

Basically same with gAUC

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

help, gAUC

Examples

```
tblAUC(Theoph, key="Subject", colX="Time", colY="conc")

iAUC = data.frame(Name=c("AUC[0-12h]","AUC[0-24h]"), Start=c(0,0), End=c(12,24))
tblAUC(Indometh, key="Subject", colX="time", colY="conc", iAUC=iAUC)
```

tblNCA *Table output NCA*

Description

Do multiple NCA and returns a result table. See sNCA for more detail i.e. iAUC

Usage

```
tblNCA(concData, key = "Subject", colTime = "Time", colConc = "conc", dose = 0,
       adm = "Extravascular", dur = 0, doseUnit = "mg", timeUnit = "h",
       concUnit = "ug/L", down = "Linear", R2ADJ = 0, MW = 0, SS = FALSE,
       tau = NA, unitTau = "h", iAUC = "", excludeDelta = 1, UsePoints = NULL)
```

Arguments

concData	concentration data table
key	column names of concData to be shown in the output table
colTime	column name for time
colConc	column name for concentration
dose	administered dose, a scalar or a vector
adm	one of "Bolus" or "Infusion" or "Extravascular" to indicate drug administration mode
dur	duration of infusion, a scalar or a vector
doseUnit	unit of dose
timeUnit	unit of time
concUnit	unit of concentration
down	method to calculate AUC, "Linear" or "Log"
R2ADJ	Lowest threshold of adjusted R-square value to do manual slope determination
MW	molecular weight of drug
SS	if steady-state, this should be TRUE and tau must be given. AUCTAU (AUC over one dosing interval, 0 to tau) is then used instead of AUCIFO (AUCinf) for the calculation of Vz (VZFO, VZO) and CL (CLFO, CLO), and the columns AUCTAU and CAVG are added. CMAX and TMAX are then taken from the samples within the dosing interval, and without a sample at time 0 the lowest concentration observed within the dosing interval is inserted there (0 after a single dose). The MRT to infinity (MRTEVIFO, MRTEVIFP, MRTIVIFO, MRTIVIFP) is (AUMC from 0 to tau + tau * (AUCinf - AUCTAU)) / AUCTAU, and Vdss (VSSO, VSSP) is that MRT times CLO. The Pred-based twins of Vz and CL (VZFP, CLFP, VZP, CLP) are NA at steady state. Up to version 0.8.4 the MRT to infinity and VSSP were NA at steady state and VSSO was MRTIVLST times CLO. Up to version 0.8.3 AUCLST (AUClast) was used in place of AUCTAU, which is only right when the last sample falls exactly at tau.

tau	dosing interval in the unit of unitTau, a positive scalar or one value per NCA (subject). Required when SS = TRUE; ignored, with a warning, otherwise.
unitTau	unit of tau: "h" (default), "min", "s", "d", "wk" or their common spellings. tau is converted into timeUnit before use.
iAUC	data.frame for interval AUC. A name that equals a standard parameter name (for example "AUCTAU" together with SS = TRUE) is refused.
excludeDelta	Improvement of R2ADJ larger than this value could exclude the last point. Default value 1 is for the compatibility with other software.
UsePoints	A list of length equal to the number of subjects/NCAs. Each element is a vector of indices of points to calculate terminal slope for the corresponding subject. Values of use points should not be zero. Use only when automatic determination is not satisfactory. If this is used, R2ADJ option is ignored.

Details

Tabular output of NCA with many subjects. Author recommends to use excludeDelta option with about 0.3.

Value

CMAX	maximum concentration, Cmax; within the dosing interval (0 to tau) when SS = TRUE
CMAXD	dose normalized Cmax, CMAX / Dose, Cmax / Dose
TMAX	time of maximum concentration, Tmax; within the dosing interval (0 to tau) when SS = TRUE
TLAG	time to observe the first non-zero concentration, for extravascular administration only
CLST	last positive concentration observed, Clast
CLSTP	last positive concentration predicted, Clast_pred
TLST	time of last positive concentration, Tlast
LAMZHL	half-life by lambda z, ln(2)/LAMZ
LAMZ	lambda_z negative of the best-fit terminal slope
LAMZLL	earliest time for LAMZ
LAMZUL	last time for LAMZ
LAMZNPT	number of points for LAMZ
CORRXY	correlation of log(concentration) and time
R2	R-squared
R2ADJ	R-squared adjusted
C0	back extrapolated concentration at time 0, for intravascular bolus administration only
AUCLST	AUC from 0 to TLST
AUCALL	AUC using all the given points, including trailing zero concentrations

AUCIFO	AUC infinity observed
AUCIFOD	AUCIFO / Dose
AUCIFP	AUC infinity predicted using CLSTP instead of CLST
AUCIFPD	AUCIFP / Dose
AUCPEO	AUC % extrapolation observed
AUCPEP	AUC % extrapolated for AUCIFP
AUCPBEO	AUC % back extrapolation observed, for bolus IV administration only
AUCPBEP	AUC % back extrapolation predicted with AUCIFP, for bolus IV administration only
AUMCLST	AUMC to the TLST
AUMCIFO	AUMC infinity observed using CLST
AUMCIFP	AUMC infinity determined by CLSTP
AUMCPEO	AUMC % extrapolated observed
AUMCPEP	AUMC % extrapolated predicted
AUCTAU	AUC over one dosing interval, from 0 to tau, calculated like an interval AUC of iAUC (see IntAUC): interpolated according to down when tau falls between two samples, extrapolated with LAMZ when tau is beyond the last sample. With SS = TRUE only
CAVG	average concentration over the dosing interval at steady state, AUCTAU / tau. With SS = TRUE only
MRTIVLST	mean residence time (MRT) to TLST, for intravascular administration
MRTIVIFO	mean residence time (MRT) infinity using CLST, for intravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFO - AUCTAU)) / AUCTAU, less half the infusion duration
MRTIVIFP	mean residence time (MRT) infinity using CLSTP, for intravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFP - AUCTAU)) / AUCTAU, less half the infusion duration
MRTEVLST	mean residence time (MRT) to TLST, for extravascular administration
MRTEVIFO	mean residence time (MRT) infinity using CLST, for extravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFO - AUCTAU)) / AUCTAU
MRTEVIFP	mean residence time (MRT) infinity using CLSTP, for extravascular administration. When SS = TRUE, (AUMC from 0 to tau + tau * (AUCIFP - AUCTAU)) / AUCTAU
VZO	volume of distribution determined by LAMZ and AUCIFO (AUCTAU when SS = TRUE), for intravascular administration
VZP	volume of distribution determined by LAMZ and AUCIFP, for intravascular administration; NA when SS = TRUE
VZFO	VZO for extravascular administration, VZO/F, F is bioavailability
VZFP	VZP for extravascular administration, VZP/F, F is bioavailability; NA when SS = TRUE

CLO	clearance using AUCIFO (dose / AUCTAU when SS = TRUE), for intravascular administration
CLP	clearance using AUCIFP, for intravascular administration; NA when SS = TRUE
CLFO	CLO for extravascular administration, CLO/F, F is bioavailability
CLFP	CLP for extravascular administration, CLP/F, F is bioavailability; NA when SS = TRUE
VSSO	volume of distribution at steady state using CLST, MRTIVIFO times CLO, for intravascular administration only
VSSP	volume of distribution at steady state using CLSTP, MRTIVIFP times CLP (times CLO when SS = TRUE), for intravascular administration only
units	An attribute to show units
UsedPoints	An attribute to show indices of used points

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

help, sNCA

Examples

```
tblNCA(Theoph, key="Subject", dose=320, concUnit="mg/L")
tblNCA(Indometh, key="Subject", colTime="time", colConc="conc", dose=25,
       adm="Infusion", dur=0.5, concUnit="mg/L")

# Steady state, q12h: one dosing interval per subject from the pre-dose trough to the next
# dose. tau is the dosing interval; AUCTAU, CAVG, CLFO and VZFO are derived over it.
dSS = data.frame(ID = rep(1:2, each=6), Time = rep(c(0, 1, 2, 4, 8, 12), 2),
                 conc = c(2, 6, 10, 8, 4, 2, 3, 9, 15, 12, 6, 3))
tblNCA(dSS, key="ID", dose=100, concUnit="mg/L", SS=TRUE, tau=12)
```

Unit *Display CDISC standard units and multiplied factor of NCA results*

Description

It displays CDISC PP output units and multiplication factor for them.

Usage

```
Unit(code = "", timeUnit = "h", concUnit = "ng/mL", doseUnit = "mg", MW = 0)
```

Arguments

code	vector of PPTESTCD
timeUnit	unit of time
concUnit	unit of concentration
doseUnit	unit of dose
MW	molecular weight of drug

Value

row names	PPTESTCD
Unit	unit
Factor	internal mulitplication factor

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
Unit(concUnit="ug/L", doseUnit="mg")
Unit(concUnit="ng/L", doseUnit="mg")

Unit(concUnit="umol/L", doseUnit="mmol")
Unit(concUnit="nmol/L", doseUnit="mmol")

Unit(concUnit="mmol/L", doseUnit="mg", MW=500)
Unit(concUnit="umol/L", doseUnit="mg", MW=500)
Unit(concUnit="nmol/L", doseUnit="mg", MW=500)
Unit(concUnit="nmol/mL", doseUnit="mg", MW=500)

Unit(concUnit="ug/L", doseUnit="mmol", MW=500)
Unit(concUnit="ug/L", doseUnit="mol", MW=500)
Unit(concUnit="ng/L", doseUnit="mmol", MW=500)
Unit(concUnit="ng/mL", doseUnit="mmol", MW=500)

Unit(concUnit="nmol/L", doseUnit="mg")
Unit(concUnit="ug/L", doseUnit="mmol")
```

UnitUrine — *Retuns a conversion factor for the amount calculation from urine concentration and volume*

Description

You can get a conversion factor for the multiplication: conc * vol * factor = amount in the given unit.

Usage

```
UnitUrine(conU = "ng/mL", volU = "mL", amtU = "mg", MW = 0)
```

Arguments

conU	concentration unit
volU	volume unit
amtU	amount unit
MW	molecular weight

Value

Factor	conversion factor for multiplication with the unit in name

Author(s)

Kyun-Seop Bae <k@acr.kr>

Examples

```
UnitUrine()
UnitUrine("ng/mL", "mL", "mg")
UnitUrine("ug/L", "mL", "mg")
UnitUrine("ug/L", "L", "mg")

UnitUrine("ng/mL", "mL", "g")

UnitUrine("ng/mL", "mL", "mol", MW=500)
UnitUrine("ng/mL", "mL", "mmol", MW=500)
UnitUrine("ng/mL", "mL", "umol", MW=500)
```

---

writeMD5NCA *Write the MD5 integrity manifest for an installed package*

Description

Write the standard R 'MD5' manifest into an installed package directory so that checkMD5sums (and IQNCA) can verify file integrity. The manifest records the md5 checksum of every installed file. It is the same 'MD5' file that CRAN ships and that R's own installer writes; packages installed from CRAN already contain it, but a local source install usually does not (which is why the IQ integrity check then reports WARN). Run writeMD5NCA() once, immediately after installing the package, to record the trusted baseline; thereafter checkMD5sums() detects any later modification of the installed files.

Usage

```
writeMD5NCA(pkg = "NonCompart", lib.loc = NULL)
```

Arguments

pkg	name of the installed package whose 'MD5' manifest should be written.
lib.loc	character vector of library paths to search for pkg, passed to find.package. NULL uses the default libraries.

Details

The manifest is written in the format used by checkMD5sums: one line per file, <md5sum> *<relative-path>. The 'MD5' file itself is excluded. The function uses only the exported md5sum. The installed package directory must be writable.

Value

Invisibly, the path to the 'MD5' file that was written. Called for its side effect.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

IQNCA, checkMD5sums, md5sum

Examples

```
#writeMD5NCA("NonCompart")
#NonCompart::IQNCA()       # the file-integrity check now reports PASS
```

Index

∗ **AUC**
  AUC, 5
  gIntAUC, 9
  IntAUC, 11
  LinAUC, 14
  LogAUC, 15
∗ **Output Form**
  PQNCA, 18
  sNCA, 22
  tblNCA, 28
∗ **interpolation**
  gInterpol, 10
  Interpol, 12
∗ **interval AUC**
  gIntAUC, 9
  gInterpol, 10
  IntAUC, 11
  Interpol, 12
∗ **misc**
  IQNCA, 13
  OQNCA, 16
∗ **partial AUC**
  gIntAUC, 9
  gInterpol, 10
  IntAUC, 11
  Interpol, 12
∗ **slope**
  BestSlope, 6
  DetSlope, 7
  Slope, 21
∗ **utilities**
  addSigFieldNCA, 3
  signPDFNCA, 20
  writeMD5NCA, 33

addSigFieldNCA, 3, *14*, *17*, *19*
AUC, 5, *12*, *15*, *16*

BestSlope, 6, *22*

checkMD5sums, *14*, *33*, *34*

DetSlope, 7

find.package, *34*

gAUC, 8, *10*, *27*
gIntAUC, 9, *11*
gInterpol, *10*, 10

help, *26*, *27*, *31*

IntAUC, 11, *13*, *23*, *25*, *30*
Interpol, *12*, 12
IQNCA, *3*, *4*, 13, *18*, *20*, *21*, *33*, *34*

LinAUC, *5*, 14, *16*
LogAUC, *5*, *15*, 15

md5sum, *34*

NonCompart *(*NonCompart-package*)*, 2
NonCompart-package, 2

OQNCA, *3*, *4*, *14*, 16, *18*, *20*, *21*

pdf, *4*
PQNCA, 18

sessionInfo, *14*, *19*
signPDFNCA, *4*, 20
Slope, *7*, *8*, 21
sNCA, *14*, 22, *31*

tblAUC, *10*, 26
tblNCA, *14*, *17–20*, *26*, 28

Unit, 31
UnitUrine, 32

verifyPDFNCA *(*signPDFNCA*)*, 20

writeMD5NCA, *14*, *18*, *21*, 33