Package 'ncar'

July 20, 2026

Version 0.7.0

Date 2026-06-20

Title Noncompartmental Analysis for Pharmacokinetic Report

Description Conduct a noncompartmental analysis with industrial strength.
Some features are
1) 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) Produce pdf, rtf, text report files.
6) Produce Installation and Operational Qualification (IQ/OQ) reports in pdf.
After installation, qualify the package in your own environment:
run pdfIQ() for Installation Qualification and pdfOQ() for Operational
Qualification. Run writeMD5() 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 addSigField(),
to add click-to-sign fields), instead of printing and scanning; or use
signPDF()/verifyPDF() for a scriptable signature.
* Reference: Gabrielsson J, Weiner D. Pharmacokinetic and Pharmacodynamic Data Analysis - Concepts and Applications. 5th ed. 2016. (ISBN:9198299107).

Depends R (>= 3.5.0)

Imports NonCompart (>= 0.8.0), R.oo, R.methodsS3, tools

Suggests openssl

Author Kyun-Seop Bae [aut, cre],
Michael E. Schaffer [ctb, cph] (Original author of the 'rtf' functions vendored in R/rtf_utils.R)

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

Copyright 2016-, Kyun-Seop Bae; portions Copyright (c) Michael E. Schaffer (vendored from the 'rtf' package, GPL (>= 2))

License GPL-3

NeedsCompilation no

LazyLoad yes

URL https://cran.r-project.org/package=ncar
Repository CRAN
Date/Publication 2026-07-20 12:20:24 UTC

Contents

`ncar-package` *Noncompartmental Analysis for Pharmacokinetic Report*

Description

It can report a noncompartmental analysis (NCA) with industrial strength.

Details

The main functions are

```
pdfNCA to produce PDF file format NCA.
rtfNCA to produce rtf file format NCA.
```

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

# Output to PDF file
#pdfNCA(fileName="NCA-Theoph.pdf", Theoph, key="Subject", colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#pdfNCA(fileName="NCA-Theoph.pdf", Theoph, key=c("Subject", "Wt"), colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#pdfNCA(fileName="NCA-Indometh.pdf", Indometh, key="Subject", colTime="time",
#       colConc="conc", adm="Infusion", dur=0.5, dose=25, doseUnit="mg",
#       timeUnit="h", concUnit="mg/L")


# Output to RTF file
#rtfNCA(fileName="NCA-Theoph.rtf", Theoph, key="Subject", colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#rtfNCA(fileName="NCA-Theoph.rtf", Theoph, key=c("Subject", "Wt"), colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#rtfNCA(fileName="NCA-Indometh.rtf", Indometh, key="Subject", colTime="time",
#       colConc="conc", adm="Infusion", dur=0.5, dose=25, doseUnit="mg",
#       timeUnit="h", concUnit="mg/L")
```

`addSigField` *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 `pdfIQ` and `pdfOQ`.

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

```
addSigField(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 `pdfIQ`/`pdfOQ` 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

pdfIQ, pdfOQ, signPDF

Examples

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

---

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

Description

Generate a self-contained PDF report that documents whether the ncar / NonCompart packages are correctly installed, intact, loadable, and operational in the user's own R environment. It is intended as Installation Qualification (IQ) evidence, in the spirit of the WinNonlin validation suite. The report uses only base R and the package's own pdf helpers, so it requires no LaTeX, pandoc, or other external tools.

Usage

```
pdfIQ(fileName = "ncar-IQ-Report.pdf", pkgs = c("ncar", "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. Their declared dependencies are checked automatically.
`functional`	if `TRUE`, run a small operational check: a known NCA computation (Theophylline subject 1) is compared with an external WinNonlin reference within a relative tolerance of 1e-3, plus two control tests that verify the comparator itself (a known-match must pass, a known-differ must fail).
`performedBy`	name of the person performing the qualification, printed in the report header. Defaults to the login name.
`paper`	paper size: `"letter"`, `"a4"`, or `"auto"` (the default), which selects `"letter"` in a United States locale and `"a4"` elsewhere. The report uses 1 inch margins on every side and places the signature (approval) page first.
`sigField`	if `TRUE`, add Adobe Acrobat Reader signature fields to the finished report (via `addSigField`) so it can be signed with one click in Acrobat Reader.

Details

The report contains: (1) the test environment (R version, platform, OS, locale, library paths); (2) installed package versions, locations and declared-dependency version satisfaction; (3) file integrity via `checkMD5sums` (PASS when files match the MD5 manifest, FAIL on mismatch, WARN when no manifest is present, e.g. for a local source install rather than a CRAN install); (4) namespace load and presence of core exported functions; (5) the optional functional verification; an overall QUALIFIED / NOT QUALIFIED verdict; a signature block; and a `sessionInfo` appendix.

Value

Invisibly returns a list with elements `fileName`, `qualified` (logical), `checks` (a data frame of every check with its Section, Item, Result and Status), and the counts `nPass`, `nFail`, `nWarn`. A PDF file is written to `fileName` as a side effect.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

`pdfNCA`, `txtNCA`, `checkMD5sums`, `sessionInfo`

Examples

```
#pdfIQ()
#pdfIQ(fileName = "MyInstall-IQ.pdf", performedBy = "Jane Doe")
#res <- pdfIQ()
#res$qualified
#res$checks
```

---

pdfNCA *NCA output to pdf file*

Description

This output NCA result in a pdf file.

Usage

```
pdfNCA(fileName = "Temp-NCA.pdf", 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, iAUC = "", excludeDelta = 1, UsePoints = NULL,
       performedBy = "")
```

Arguments

fileName	file name to save
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	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 drug
SS	if steady-state, this should be TRUE. AUCLST (AUClast) is used instead of AUCIFO (AUCinf) for the calculation of Vz (VZFO, VZO), CL (CLFO, CLO), and Vdss (VSSO).

iAUC	interval AUC information in a dataframe with "Name", "Start", and "End" columns
excludeDelta	Improvement of R2ADJ larger than this value could exclude the last point. Default value 1 is for the compatibility with other software. Author recommends to use excludeDelta option with about 0.3.
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.
performedBy	name of the person performing the analysis, printed on the first (title) page above the Performed by / Reviewed by approval lines.

Value

CMAX	maximum concentration, Cmax
CMAXD	dose normalized Cmax, CMAX / Dose, Cmax / Dose
TMAX	time of maximum concentration, Tmax
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 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 bolus intravascular 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
MRTIVLST	mean residence time (MRT) to TLST, for intravascular administration
MRTIVIFO	mean residence time (MRT) infinity using CLST, for intravascular administration
MRTIVIFP	mean residence time (MRT) infinity using CLSTP, for intravascular administration
MRTEVLST	mean residence time (MRT) to TLST, for extravascular administration
MRTEVIFO	mean residence time (MRT) infinity using CLST, for extravascular administration
MRTEVIFP	mean residence time (MRT) infinity using CLSTP, for extravascular administration
VZO	volume of distribution determined by LAMZ and AUCIFO, for intravascular administration
VZP	volume of distribution determined by LAMZ and AUCIFP, for intravascular administration
VZFO	VZO for extravascular administration, VZO/F, F is bioavailability
VZFP	VZP for extravascular administration, VZP/F, F is bioavailability
CLO	clearance using AUCIFO, for intravascular administration
CLP	clearance using AUCIFP, for intravascular administration
CLFO	CLO for extravascular administration, CLO/F, F is bioavailability
CLFP	CLP for extravascular administration, CLP/F, F is bioavailability
VSSO	volume of distribution at steady state using CLST, for intravascular administration only
VSSP	volume of distribution at stead state using CLSTP, for intravascular administration only

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

help, txtNCA, rtfNCA

Examples

```
#pdfNCA(fileName="NCA-Theoph.pdf", Theoph, key="Subject", colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#pdfNCA(fileName="NCA-Theoph.pdf", Theoph, key=c("Subject", "Wt"), colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#pdfNCA(fileName="NCA-Indometh.pdf", Indometh, key="Subject", colTime="time",
#       colConc="conc", adm="Infusion", dur=0.5, dose=25, doseUnit="mg",
#       timeUnit="h", concUnit="mg/L")
```

`pdfOQ` *Operational Qualification (OQ) report to a pdf file*

Description

Generate a self-contained PDF report that documents whether the **NonCompart** noncompartmental analysis engine (as used through **ncar**) 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, in the spirit of the WinNonlin computational-engine verification. The report uses only base R, **NonCompart**, and the package's own pdf helpers, so it requires no LaTeX, pandoc, or other external tools.

Usage

```
pdfOQ(fileName = "ncar-OQ-Report.pdf", cases = NULL, tol = 0.001,
      refDir = system.file("OQ", package = "ncar"), 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 eight WinNonlin-referenced scenarios is used (Theophylline and Indometacin, extravascular / IV bolus / IV infusion, each with linear-up-linear-down and linear-up-log-down AUC). Each case is a list with elements `id`, `desc`, `dataset`, `key`, `colTime`, `colConc`, `args` (a list of further `tblNCA` arguments), and `ref` (the reference csv file name in `refDir`).
`tol`	acceptance tolerance: a parameter passes when the symmetric relative difference `abs(R - W) / ((abs(R) + abs(W))/2)` between the computed value `R` and the reference value `W` is at most `tol`. An absolute fallback is used when both values are essentially zero.
`refDir`	directory holding the reference csv files. Defaults to the `OQ` folder installed with **ncar**.
`performedBy`	name of the person performing the qualification, printed in the report header. Defaults to the login name.

paper	paper size: "letter", "a4", or "auto" (the default), which selects "letter" in a United States locale and "a4" elsewhere. The report uses 1 inch margins on every side and places the signature (approval) page first.
sigField	if TRUE, add Adobe Acrobat Reader signature fields to the finished report (via addSigField) 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 WinNonlin parameter names using the internal RptCfg table, 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.

The 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. Independent (Tier A) anchors (hand-calculated edge cases, published textbook values) are a planned addition.

Value

Invisibly returns a list with elements fileName, qualified (logical), tol, nCases, nFailCases, and results (a per-case list with status, comparison counts, maximum relative difference, and any failed comparisons). A PDF file is written to fileName.

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

pdfIQ, pdfNCA, tblNCA

Examples

```
#pdfOQ()
#res <- pdfOQ(fileName = "MyEngine-OQ.pdf", performedBy = "Jane Doe")
#res$qualified
#res$results[[1]]$maxRD
```

---

pdfPQ *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 pdfOQ 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

```
pdfPQ(fileName = "ncar-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 addSigField).

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

`pdfIQ`, `pdfOQ`, `tblNCA`

Examples

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

Res2Txt *Convert sNCA output table to text form*

Description

This converts the table output of `sNCA` to text form output.

Usage

```
Res2Txt(ResNCA, x, y, dose = 0, adm = "Extravascular", dur = 0, doseUnit = "mg",
        down = "Linear")
```

Arguments

`ResNCA`	Output table from `sNCA`
`x`	usually time, a vector
`y`	usually concentration, a vector
`dose`	given amount, 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
down	either of "Linear" or "Log" to indicate the way to calculate AUC and AUMC

Value

Text form output from the coversion of table form output

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

txtNCA, pdfNCA, rtfNCA

Examples

```
x = Theoph[Theoph$Subject=="1","Time"]
y = Theoph[Theoph$Subject=="1","conc"]
z = NonCompart::sNCA(x, y, dose=320, doseUnit="mg", concUnit="mg/L", timeUnit="h")
Res2Txt(z, x, y)
```

Round *Round Half Away from Zero*

Description

This is an ordinary rounding function, so called round half away from zero

Usage

```
Round(x, n = 0)
```

Arguments

x	numeric to be rounded
n	indicating decimal digits

Details

The function round in R base rounds to the even number, i.e. round(0.5) is 0 not 1. If you want rounding 0.5 be 1, you can use this Round function. This function is for the consistency with other software like MS-Excel, SAS.

Value

ordinarily rounded value

Author(s)

Kyun-Seop Bae <k@acr.kr>

References

See wikipedia subject "Rounding"

Examples

```
(x = 1:10 - 0.5)
Round(x)
round(x) # compare with the above
```

RptCfg *NCA Report Configuration Table*

Description

Contains the names and order of column of return table/text in outputs

Usage

```
RptCfg
```

Format

A data frame with 48 observations on the following 10 variables.

`PPTESTCD` a character vector of CDISC SDTM PPTESTCD

`SYNONYM` a character vector of CDISC SDTM PPTESTCD Synonym

`NCI` a character vector of NCI peferred terms

`WNL` a character vector of WinNonlin(R) software variables

`ExtravascularDefault` a numeric vector of ordering in report for extravascular administration, Zero means exclusion in the report.

`ExtravascularWNL` a numeric vector of WinNonlin(R) style ordering in report for extravascular administration, Zero means exclusion in the report.

`BolusDefault` a numeric vector of ordering in report for extravascular administration, Zero means exclusion in the report.

`BolusWNL` a numeric vector of WinNonlin(R) style ordering in report for extravascular administration, Zero means exclusion in the report.

`InfusionDefault` a numeric vector of ordering in report for extravascular administration, Zero means exclusion in the report.

`InfusionWNL` a numeric vector of WinNonlin(R) style ordering in report for extravascular administration, Zero means exclusion in the report.

Details

This table should exist in this package.

`rtfNCA` *NCA output to rtf file*

Description

This output NCA result in a rtf file.

Usage

```
rtfNCA(fileName = "Temp-NCA.rtf", 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, iAUC = "", excludeDelta = 1, UsePoints = NULL)
```

Arguments

`fileName`	file name to save
`concData`	concentration data table
`key`	column names of concData to be shown in the output
`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`	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 drug
`SS`	if steady-state, this should be TRUE. AUCLST (AUClast) is used instead of AUCIFO (AUCinf) for the calculation of Vz (VZFO, VZO), CL (CLFO, CLO), and Vdss (VSSO).
`iAUC`	interval AUC information in a dataframe with "Name", "Start", and "End" columns
`excludeDelta`	Improvement of R2ADJ larger than this value could exclude the last point. Default value 1 is for the compatibility with other software. Author recommends to use `excludeDelta` option with about 0.3.
`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.

Value

CMAX	maximum concentration, Cmax
CMAXD	dose normalized Cmax, CMAX / Dose, Cmax / Dose
TMAX	time of maximum concentration, Tmax
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 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 bolus intravascular 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
MRTIVLST	mean residence time (MRT) to TLST, for intravascular administration
MRTIVIFO	mean residence time (MRT) infinity using CLST, for intravascular administration

MRTIVIFP	mean residence time (MRT) infinity using CLSTP, for intravascular administration
MRTEVLST	mean residence time (MRT) to TLST, for extravascular administration
MRTEVIFO	mean residence time (MRT) infinity using CLST, for extravascular administration
MRTEVIFP	mean residence time (MRT) infinity using CLSTP, for extravascular administration
VZO	volume of distribution determined by LAMZ and AUCIFO, for intravascular administration
VZP	volume of distribution determined by LAMZ and AUCIFP, for intravascular administration
VZFO	VZO for extravascular administration, VZO/F, F is bioavailability
VZFP	VZP for extravascular administration, VZP/F, F is bioavailability
CLO	clearance using AUCIFO, for intravascular administration
CLP	clearance using AUCIFP, for intravascular administration
CLFO	CLO for extravascular administration, CLO/F, F is bioavailability
CLFP	CLP for extravascular administration, CLP/F, F is bioavailability
VSSO	volume of distribution at steady state using CLST, for intravascular administration only
VSSP	volume of distribution at stead state using CLSTP, for intravascular administration only

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

help, txtNCA, pdfNCA

Examples

```
#rtfNCA(fileName="NCA-Theoph.rtf", Theoph, key="Subject", colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#rtfNCA(fileName="NCA-Theoph.rtf", Theoph, key=c("Subject", "Wt"), colTime="Time",
#       colConc="conc", dose=320, doseUnit="mg", timeUnit="h", concUnit="mg/L")
#rtfNCA(fileName="NCA-Indometh.rtf", Indometh, key="Subject", colTime="time",
#       colConc="conc", adm="Infusion", dur=0.5, dose=25, doseUnit="mg",
#       timeUnit="h", concUnit="mg/L")
```

signPDF *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. `signPDF` 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`'). `verifyPDF` 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

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

verifyPDF(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 (`signPDF`) or read (`verifyPDF`).
`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

`signPDF` 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. `verifyPDF` 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

pdfIQ, pdfOQ, writeMD5

Examples

```
#key <- openssl::rsa_keygen()
#openssl::write_pem(key, "signer_key.pem")
#pdfIQ("ncar-IQ-Report.pdf", performedBy = "Kyun-Seop Bae")
#signPDF("ncar-IQ-Report.pdf", "signer_key.pem", signer = "Kyun-Seop Bae")
#verifyPDF("ncar-IQ-Report.pdf")
```

---

txtNCA *Text output of NCA for one subject*

Description

This is the text form output.

Usage

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

Arguments

x	usually time
y	usually concentration
dose	given amount, 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
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. AUCLST (AUClast) is used instead of AUCIFO (AUCinf) for the calculation of Vz (VZFO, VZO), CL (CLFO, CLO), and Vdss (VSSO).
excludeDelta	Improvement of R2ADJ larger than this value could exclude the last point. Default value 1 is for the compatibility with other software. Author recommends to use excludeDelta option with about 0.3.
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.

Value

CMAX	maximum concentration, Cmax
CMAXD	dose normalized Cmax, CMAX / Dose, Cmax / Dose
TMAX	time of maximum concentration, Tmax
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 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 bolus intravascular 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
MRTIVLST	mean residence time (MRT) to TLST, for intravascular administration
MRTIVIFO	mean residence time (MRT) infinity using CLST, for intravascular administration
MRTIVIFP	mean residence time (MRT) infinity using CLSTP, for intravascular administration
MRTEVLST	mean residence time (MRT) to TLST, for extravascular administration
MRTEVIFO	mean residence time (MRT) infinity using CLST, for extravascular administration
MRTEVIFP	mean residence time (MRT) infinity using CLSTP, for extravascular administration
VZO	volume of distribution determined by LAMZ and AUCIFO, for intravascular administration
VZP	volume of distribution determined by LAMZ and AUCIFP, for intravascular administration
VZFO	VZO for extravascular administration, VZO/F, F is bioavailability
VZFP	VZP for extravascular administration, VZP/F, F is bioavailability
CLO	clearance using AUCIFO, for intravascular administration
CLP	clearance using AUCIFP, for intravascular administration
CLFO	CLO for extravascular administration, CLO/F, F is bioavailability
CLFP	CLP for extravascular administration, CLP/F, F is bioavailability
VSSO	volume of distribution at steady state using CLST, for intravascular administration only
VSSP	volume of distribution at stead state using CLSTP, for intravascular administration only

Author(s)

Kyun-Seop Bae <k@acr.kr>

See Also

help, pdfNCA, rtfNCA

Examples

```
# For one subject
txtNCA(Theoph[Theoph$Subject=="1","Time"], Theoph[Theoph$Subject=="1","conc"],
       dose=320, doseUnit="mg", concUnit="mg/L", timeUnit="h")

# or equivalently
x = Theoph[Theoph$Subject=="1","Time"]
y = Theoph[Theoph$Subject=="1","conc"]
txtNCA(x, y, dose=320, doseUnit="mg", concUnit="mg/L", timeUnit="h")

# For all subjects
IDs = sort(as.numeric(unique(Theoph[,"Subject"])))
nID = length(IDs)
Res = vector()
for (i in 1:nID) {
  tRes = txtNCA(Theoph[Theoph[,"Subject"]==IDs[i],"Time"],
                Theoph[Theoph[,"Subject"]==IDs[i],"conc"],
                dose=320, concUnit="mg/L")
  tRes = c(paste("ID =", IDs[i]), tRes, "")
  Res = c(Res, tRes)
}
Res
```

`writeMD5` *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 `pdfIQ`) 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 `writeMD5()` once, immediately after installing the package, to record the trusted baseline; thereafter `checkMD5sums()` detects any later modification of the installed files.

Usage

```
writeMD5(pkg = "ncar", 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

`pdfIQ`, `checkMD5sums`, `md5sum`

Examples

```
#writeMD5("ncar")
#writeMD5("NonCompart")
#ncar::pdfIQ()          # the file-integrity check now reports PASS
```

Index