Package: nmpilot (via r-universe)

November 26, 2026

Title R Tools for Direct NONMEM Workflow Automation

Version 0.9.0

Description Direct execution of NONMEM via nmfe (no Wings, no PsN), data-driven suggestion of structural models before any NONMEM run (number of compartments, input, elimination and residual error models, initial values, expected precision of the sampling design), writing of control streams (compartmental PK models, MU referencing, inter-occasion variability, M3 method for BQL data) with a static check, parsing of NONMEM output files (.lst, .ext, .phi, .fit), VPC/pc-VPC generation, dataset preparation and visualization, and run comparison tools for pharmacometric workflows.

License MIT + file LICENSE

URL https://github.com/nskoch/nmpilot, https://nskoch.github.io/nmpilot/

BugReports https://github.com/nskoch/nmpilot/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports dplyr, ggplot2, graphics, lubridate, magrittr, parallel, purrr, rlang, stats, stringr, tibble, tidyr, tools, utils

Suggests flextable, knitr, officer, patchwork, rmarkdown, testthat (>= 3.0.0), transport

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

Config/pak/sysreqs libicu-dev

Repository https://nskoch.r-universe.dev

Date/Publication 2026-11-26 06:56:24 UTC

RemoteUrl https://github.com/nskoch/nmpilot

RemoteRef HEAD

RemoteSha 2f9a3b58ddc7f9eefeb79177543c138ff949b1cf

Contents

nm_add_rate . . . 3
nm_bootstrap . . . 4
nm_check_dataset . . . 6
nm_compare . . . 7
nm_compare_runs . . . 10
nm_convert_ctl . . . 11
nm_cores . . . 12
nm_ctl_check . . . 13
nm_ctl_iov . . . 14
nm_ctl_m3 . . . 15
nm_ctl_mu . . . 17
nm_data_previsual . . . 18
nm_derivatives_ctl . . . 20
nm_export_data . . . 21
nm_export_plot . . . 22
nm_extract_run_table . . . 23
nm_extract_runs . . . 24
nm_filter_runs . . . 25
nm_gof . . . 26
nm_guide . . . 27
nm_identify_directory . . . 28
nm_identify_parameters . . . 29
nm_jackknife . . . 30
nm_model_diagram . . . 32
nm_optimize_base . . . 33
nm_plot_dv . . . 36
nm_profile . . . 37
nm_read_csv . . . 39
nm_run . . . 40
nm_runs_table . . . 43
nm_suggest_covariates . . . 44
nm_suggest_design . . . 46
nm_suggest_model . . . 47
nm_suggest_run . . . 50
nm_summary . . . 52
nm_vpc . . . 52
nm_vpc_ctl . . . 55
nm_wasserstein . . . 57
nm_write_ctl . . . 59
plot.nm_compare . . . 62
plot.nm_jackknife . . . 63
plot.nm_model_suggestion . . . 64

plot.nm_wasserstein 64

Index **66**

`nm_add_rate` *Add a RATE column to a NONMEM dataset*

Description

Reads the CSV verbatim (all columns as character) so that nothing is reformatted, adds a RATE column based on the dosing records, and writes a new file. The original columns (values, format, '#ID' header) are kept unchanged.

Usage

```
nm_add_rate(
  file,
  output_file = NULL,
  sep = ",",
  dose_value = -2,
  base_value = 0,
  after = "AMT",
  dose_cmt = NULL,
  cmt_col = NULL,
  evid_col = NULL,
  amt_col = NULL,
  overwrite = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`file`	Path to the original CSV.
`output_file`	Name (or path) of the output CSV. `NULL` (default) -> `"<base>_rate.csv"` in the same directory as `file`. If only a file name is given (no directory), it is placed next to the source file. (Formerly `out_file`, still accepted with a warning.)
`sep`	Column separator (default `","`).
`dose_value`	RATE value on the targeted dosing records: -2 (default, estimated duration) or -1 (estimated rate).
`base_value`	RATE value on all other records (default 0).
`after`	Name of the column after which RATE is inserted (default `"AMT"`). `NULL` = added as the last column.
`dose_cmt`	CMT value(s) identifying the route(s) that receive `dose_value`. `NULL` (default): all dosing records receive `dose_value` (single-route dataset). E.g. `dose_cmt = 1` to affect only the oral depot, leaving IV boluses (other CMT) at `base_value`.

cmt_col	Name of the CMT column (default: auto-detected). Required only when dose_cmt is supplied.
evid_col	Name of the EVID column (default: auto-detected).
amt_col	Name of the AMT column (default: auto-detected; used only when no EVID column is found).
overwrite	If TRUE, overwrite an existing RATE column (default FALSE: the function stops if RATE already exists).
verbose	Print a diagnostic report (default TRUE).
...	Former argument name, accepted with a warning until version 1.0.

Value

The path of the written file (invisibly).

Examples

```
## Not run:
# Single route: RATE=-2 on all doses -> data_nmol_rate.csv
nm_add_rate("C:/nm74g64/run/Bienvenu/data_nmol.csv")

# Estimated rate rather than duration
nm_add_rate("data.csv", dose_value = -1)

# Multiple routes: -2 only on the oral depot (CMT=1),
# IV boluses (CMT=2) keep RATE=0
nm_add_rate("data.csv", dose_cmt = 1)

# Chosen output name (placed next to the source file)
nm_add_rate("data.csv", output_file = "data_dur.csv")

## End(Not run)
```

---

nm_bootstrap *Nonparametric (case) bootstrap of a NONMEM model*

Description

Nonparametric (case) bootstrap of a NONMEM model

Usage

```
nm_bootstrap(
  ctl,
  nmfe_path = NULL,
  n = 500L,
  ci = 0.95,
  seed = NULL,
```

```
  id_col = "ID",
  strata = NULL,
  comment_char = NULL,
  sep = ",",
  output_dir = NULL,
  keep_datasets = FALSE,
  keep_outputs = FALSE,
  strip_cov = TRUE,
  strip_tables = TRUE,
  add_warnings = TRUE,
  warnings_line = "$WARN WARN=NONE",
  include_rounding = FALSE,
  resume = TRUE,
  run = TRUE,
  verbose = TRUE,
  n_cores = 1L,
  ...
)
```

Arguments

ctl	Path to the base control stream (.ctl).
nmfe_path	Path to the NONMEM launcher: nmfe*.bat on Windows (e.g. "C:/nm74g64/run/nmfe74.bat"), the nmfe script on Linux or macOS (e.g. "/opt/nm760/run/nmfe76").
n	Number of bootstrap replicates (default 500).
ci	Confidence level(s) for the percentile CI (default 0.95; e.g. c(0.95,0.90)).
seed	RNG seed for reproducibility (default: a random seed, stored in the result).
id_col	Label of the subject ID column, as it appears in $INPUT (default "ID").
strata	Optional $INPUT column to stratify the resampling on (constant within ID).
comment_char	Data-file comment/ignore character. NULL = auto-derive from the single-char IGNORE in $DATA ("@", "#", a letter...); "" = none.
sep	Data-file field separator (default ",").
output_dir	Working directory (default "<ctl_dir>/_bootstrap"). Formerly run_dir, still accepted with a warning.
keep_datasets	Keep each replicate data file? (default FALSE = space-optimal).
keep_outputs	Keep each replicate's NONMEM outputs? (default FALSE; .ext is read then removed).
strip_cov	Remove $COVARIANCE (default TRUE; useless for a bootstrap).
strip_tables	Remove $TABLE records (default TRUE; fixed FILE= names would collide). Formerly strip_table, which still works.
add_warnings	Insert a warning-suppression record (default TRUE).
warnings_line	Exact record inserted (default "$WARN WARN=NONE"; canonical form "$WARNINGS").

include_rounding
: Count "minimisation successful, but rounding errors" runs as successful (default FALSE).

resume
: Skip replicates already in bootstrap_estimates.csv (default TRUE).

run
: FALSE = only build the optimised ctl + resample plan (dry run, no NONMEM).

verbose
: Progress messages (default TRUE).

n_cores
: Number of replicates run at the same time (default 1). A whole number, "auto" (physical cores minus one, see nm_cores()) or "ask" (interactive sessions: the cores are shown and the number is asked once per R session, then reused; options(nmpilot.n_cores = NULL) asks again). Each run is then started in the background in its own directory; if R is interrupted, the runs already started go on in the background and a later call with the same arguments picks up their results. In parallel, each run uses a working subdirectory slot<k>/ of output_dir; the estimates are appended to bootstrap_estimates.csv as the replicates end (the file is sorted by replicate when the result is assembled).

...
: Former argument names, accepted with a warning until version 1.0: run_dir (now output_dir). The former name strip_table still works (R matches it to strip_tables).

Value

An object of class "nm_bootstrap" (see print()/plot()).

---

nm_check_dataset *Check that a dataset is compatible with NONMEM*

Description

Reads a dataset as NONMEM would (every value as text, . or an empty cell as missing) and reports the problems that make NM-TRAN refuse the data or NONMEM misread them. The data are never modified.

Usage

```
nm_check_dataset(data, cols = NULL, sep = ",", outlier_k = 4, verbose = TRUE)
```

Arguments

data
: Path of a CSV file, or a data frame (for example the data element of nm_read_csv()).

cols
: Optional named character vector giving the column of a standard item when its name differs, e.g. c(ID = "SUBJ", DV = "CONC"). Names are otherwise matched without regard to case.

sep
: Column separator of the CSV file (default ",").

outlier_k
: Number of median absolute deviations (log DV) beyond which an observation is reported as an outlier (default 4).

verbose
: Print the report (default TRUE).

Details

Checks:

- **columns**: ID, TIME and DV present (error); AMT absent (warning: no dose record);
- **missing ID or TIME** values (error);
- **non-numeric values** in a column (error; NONMEM reads numbers only, unless the column is dropped in `$INPUT`);
- **times**: TIME decreasing within a subject (error), except at a reset record (EVID 3 or 4);
- **EVID / MDV / AMT**: EVID outside 0 to 4 (error); dose record (EVID 1 or 4) without a positive AMT (error); observation (EVID 0) with a positive AMT (error); MDV 0 with a missing DV (error); MDV 0 on a record that is not an observation (error); observation with MDV 1 but a DV (warning: the value is ignored); without EVID, a record with AMT > 0 is a dose and MDV must be 1;
- **duplicates**: identical records (warning); two observations of the same subject, time and compartment with different DV (warning);
- **outliers** in the observed DV: negative values (warning) and values far from the observations taken at the same time after the last dose (log scale, more than `outlier_k` median absolute deviations from their median; per compartment when CMT exists; groups of at least 10 observations) (warning).

The header line may start with `#` or `@` (`#ID,TIME,...`), as read with `IGNORE=#` or `IGNORE=@` in `$DATA`.

Value

An object of class `nm_dataset_check`: a list with `findings` (one row per problem found: `check`, `severity` "error" or "warning", `column`, `n_rows`, `rows` (row numbers in the data, header excluded), `ids` and `message`), `n_rows`, `n_subjects`, `columns` (standard item -> column used) and `ok` (`TRUE` when no error was found). Printed (invisibly returned) when `verbose = TRUE`.

Examples

```
ex <- system.file("extdata", "example", package = "nmpilot")
chk <- nm_check_dataset(file.path(ex, "example_data.csv"))
chk$ok
chk$findings
```

---

nm_compare *Compare two NONMEM runs*

Description

Paired comparison of two NONMEM runs at the observation level (CWRES, NPDE) and at the subject level (individual OFV from `.phi`). Provides a summary of residual diagnostics, per-subject iOFV deltas, and identifies which subjects drive the OFV difference.

Usage

```
nm_compare(
  run1,
  run2,
  labels = c("Run 1", "Run 2"),
  join_keys = c("ID", "TIME"),
  n_par = NULL,
  nested = FALSE,
  covariates = NULL,
  data = NULL,
  n_boot = 2000L,
  seed = 2026L
)

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

## S3 method for class 'nm_compare'
summary(object, ...)
```

Arguments

run1, run2	Path to a run directory or run prefix (without extension). Each location must contain at least a sdtab* table. .phi and .lst files are used if available for iOFV and global OFV respectively.
labels	Character vector of length 2. Display labels for the two runs.
join_keys	Character vector. Columns used to pair observations between runs. Default c("ID", "TIME"). Add "OCC" for BOV models, "CMT" for multi-DV models.
n_par	Numbers of estimated parameters of the two runs (length 2). Default: read from the .ext files (parameters not fixed).
nested	TRUE if one model is nested in the other: a likelihood ratio test is added to the Vuong and Clarke tests.
covariates	Covariates (columns of the data set) against which the per-subject delta iOFV is examined.
data	Data set (path or data.frame) holding the covariates. Default: the $DATA file of run 1.
n_boot	Number of subject resamples for the bootstrap of the delta iOFV.
seed	Seed of the bootstrap.
x	An nm_compare object.
...	Unused.
object	An nm_compare object.

Details

Observations are merged on join_keys (default c("ID", "TIME")). If the two datasets differ (e.g. different BLQ handling), only the intersection is retained and a warning is issued.

Tests on the individual contributions. With `d_i = (iOFV_1,i - iOFV_2,i)/2`, the difference of the individual log-likelihoods, the Vuong (1989) statistic is `z = (sum(d_i) - c) / (sqrt(N) * sd(d_i))` and the Clarke (2007) test is a sign test on `d_i - c/N`, where `N` is the number of subjects and `c` the correction for the number of parameters: none, `k_2 - k_1` (AIC-like) or `(k_2 - k_1) log(N) / 2` (BIC-like). Both apply to non-nested models fitted to the same data; the subjects are the independent units. The individual contributions are those of the estimation method (FOCE: the iOFV of the `.phi` file).

Bootstrap. The subjects are resampled with their delta iOFV, without re-estimation: the distribution measures how much the difference depends on a few subjects, conditionally on the estimates (optimistic for the larger model).

Influence. `n_reverse` is the number of subjects with the largest contributions whose removal reverses the sign of dOFV (as `dOFV.vs.id` in xpose4).

Value

An object of class `nm_compare`, a list with:

`data_obs` Observation-level paired data.frame with `_1`/`_2` suffixed columns and delta columns (`dCWRES`, `dAbsCWRES`, `dNPDE`, `dAbsNPDE`).

`data_subj` Per-subject summary including `iOFV_1`, `iOFV_2`, `diOFV`, and residual aggregates.

`metrics` Global metrics table (OFV, RMSE(CWRES), Mean |CWRES|, proportions of outliers, NPDE moments). Wilcoxon paired p-value stored as attribute `wilcox_absCWRES_p`.

`meta` Labels, global OFV, numbers of parameters, sample sizes, dataset compatibility check.

`tests` Vuong, Clarke (and LRT) tests, subject bootstrap and influence (`NULL` without `.phi` files).

`parameters` Estimates, SE, RSE and shrinkage of the two runs matched by parameter name, with their ratio.

`etas` Paired EBEs of the ETAs present in both runs.

`covariate_tests` Delta iOFV versus each covariate.

`run_info` Run-level criteria (RSE, condition number, shrinkage, status).

`summary()`: a data.frame of decision criteria (class `nm_compare_summary`), one row per criterion with the value for each run and the run it favours.

References

Vuong QH. Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica 1989;57:307-33.

Clarke KA. A simple distribution-free test for nonnested model selection. Political Analysis 2007;15:347-63.

See Also

`nm_extract_runs()`, `plot.nm_compare()`, `print.nm_compare()`

Examples

```
## Not run:
  cmp <- nm_compare("runs/run42", "runs/run43",
                    labels = c("Base", "+ CLcr"))
  print(cmp)
  plot(cmp)
  plot(cmp, type = "iofv_caterpillar")
  summary(cmp)
  plot(cmp, type = c("gain_map", "overlay_gof"))

## End(Not run)
```

---

nm_compare_runs *Compare the parameters of several runs side by side*

Description

Compare the parameters of several runs side by side

Usage

```
nm_compare_runs(
  x,
  what = c("value", "rse_pct", "se", "shrinkage"),
  types = NULL,
  format = FALSE,
  digits = 3,
  align_by = c("name", "index"),
 run_filter = c("all", "successful", "successful_no_warning", "successful_no_comment",
    "clean")
)
```

Arguments

x	An nm_runs object or a params tibble.
what	"value", "rse_pct", "se" or "shrinkage".
types	Parameter types to keep. NULL = all.
format	If TRUE, cells are formatted as "value (RSE%)".
digits	Significant digits when format = TRUE.
align_by	"name" (default) for semantic alignment; "index" for a strict comparison by NONMEM index.
run_filter	Run filter: "all" (default), "successful", "successful_no_warning", "successful_no_comment" or "clean".

Value

A wide tibble with one row per parameter and one column per run.

Examples

```
## Not run:
nm_compare_runs(res, types = "THETA", format = TRUE)
nm_compare_runs(res, types = "THETA", format = TRUE, run_filter = "successful")
nm_compare_runs(res, types = "THETA", format = TRUE, run_filter = "clean")
res_clean <- nm_filter_runs(res, "clean")
res_clean$runs

## End(Not run)
```

---

nm_convert_ctl *Convert a Wings for NONMEM extended control stream to standard NONMEM*

Description

Wings for NONMEM (WFN) supports an extended control stream format where parameter names (e.g. POP_CL, PPV_CL, RUV_PROP) are used directly in model records ($PK, $PRED, $ERROR, $DES) instead of THETA(n), ETA(n), ERR(n). This function parses the parameter names from $THETA, $OMEGA and $SIGMA comment fields and substitutes them back into standard NONMEM syntax, producing a control stream compatible with direct nmfe execution.

Usage

```
nm_convert_ctl(ctl, output_file = NULL, overwrite = FALSE, verbose = TRUE, ...)
```

Arguments

ctl	Path to the Wings extended control stream (.ctl). (Formerly ctl_file, still accepted with a warning.)
output_file	Output path for the converted control stream. If NULL (default), the output file is named <run>_converted.ctl in the same directory as ctl. (Formerly out_file, still accepted with a warning.)
overwrite	Logical. If TRUE, overwrite output_file if it already exists (default FALSE).
verbose	Logical. If TRUE (default), print a summary of substitutions made.
...	Former argument name, accepted with a warning until version 1.0: ctl_file (now ctl).

Details

The conversion follows the WFN specification:

- Parameter names are extracted from the first non-numeric word after the ; comment delimiter in each $THETA, $OMEGA and $SIGMA line.
- Names are substituted by THETA(n), ETA(n) and ERR(n) in model records ($PK, $PRED, $ERROR, $DES, $AES).
- Once a name appears on the left-hand side of an assignment, it is no longer substituted on subsequent right-hand sides (WFN rule).
- $OMEGA BLOCK structures are handled correctly.

Value

Invisibly returns the path to the converted control stream.

See Also

nm_run

Examples

```
## Not run:
# Convert a WFN extended control stream
out <- nm_convert_ctl("C:/analyses/run001_wfn.ctl")

# Then run with nmpilot
log <- nm_run(
  ctl          = out,
  nmfe_path    = "C:/nm76g64/run/nmfe76.bat",
  analysis_dir = "C:/analyses"
)

## End(Not run)
```

---

nm_cores *Number of processor cores available for parallel NONMEM runs*

Description

Reports the logical and physical cores of the computer, and the number of parallel runs that n_cores = "auto" would use (physical cores minus one, at least 1). Several runs share the memory and the disk: on a laptop, or with large data sets, fewer runs than cores can be faster.

Usage

```
nm_cores(verbose = TRUE)
```

Arguments

verbose	Print the result.

Value

Invisibly, a list with logical, physical and auto.

See Also

nm_run(), nm_bootstrap(), nm_jackknife(), nm_profile()

Examples

```
nm_cores()
```

---

nm_ctl_check *Static check of a NONMEM control stream*

Description

Looks for inconsistencies that can be detected without running NM-TRAN: THETA(n), ETA(n) and EPS(n)/ERR(n) used without a matching $THETA, $OMEGA or $SIGMA declaration (and declared parameters that are never used), data items used in the code but absent from $INPUT, an active RATE with first-order absorption, and the requirements of the M3 method (BQL records not ignored, LLOQ defined, Laplace estimation). It also reports four errors refused by NM-TRAN: TOL missing with a general ADVAN (6, 8, 9, 13-18), an unknown METHOD (FO is written METHOD=ZERO), a THETA initial estimate of 0 that is not fixed, a $TABLE item written after an option of the same line (items written on a line below the options are accepted), and Y with EPS/ERR defined only in IF blocks testing ICALL.EQ.4 (simulation code for NM-TRAN).

Usage

```
nm_ctl_check(ctl, reference = NULL)
```

Arguments

ctl	Path to a control stream, or its lines (character vector or nm_ctl object).
reference	Optional original control stream (same forms as ctl). When given, the line differences from reference to ctl are added.

Details

Declarations are counted over all $THETA, $OMEGA and $SIGMA records, including values written on the record line, BLOCK(n), DIAGONAL(n), SAME(k) and $ABBR REPLACE lists and ranges (ETA(4,6,8), ETA(,3:32)).

This check does not replace NM-TRAN: it catches common mistakes early.

Value

An object of class nm_ctl_check: a data frame with columns level ("ERROR", "WARNING" or "OK"), category and message. With reference, attribute "diff" holds the line differences (" " unchanged, "- " removed, "+ " added).

Examples

```
ctl <- nm_write_ctl()
nm_ctl_check(ctl)
nm_ctl_check(nm_ctl_mu(ctl), reference = ctl)
```

---

nm_ctl_iov *Add inter-occasion variability to a NONMEM control stream*

Description

Adds an inter-occasion (between-occasion) random effect to the selected parameters with $ABBR REPLACE (NONMEM 7.3 or later): one ETA per parameter and occasion, selected through the occasion data item, and an $OMEGA BLOCK(p) followed by $OMEGA BLOCK(p) SAME(n_occ - 1). A parameter with IIV becomes P = TVP*EXP(ETA(m)+ETA(OCC_BOVP)); a parameter without IIV (P = TVP) becomes P = TVP*EXP(ETA(OCC_BOVP)).

Usage

```
nm_ctl_iov(
  ctl,
  parameters,
  n_occ,
  occ = "OCC",
  cv = 20,
  structure = c("independent", "correlated"),
  rho = 0.1,
  data = NULL,
  output_file = NULL,
  overwrite = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

ctl	Path to a control stream, or its lines (character vector or nm_ctl object).
parameters	Names of the parameters receiving IOV.
n_occ	Number of occasions (at least 2).
occ	Name of the occasion data item; it must be active in $INPUT and take the values 1 to n_occ. Letters and digits only.

cv	Initial IOV, as CV% (one value, or one per parameter, possibly named).
structure	"independent" (diagonal block) or "correlated" (common initial correlation rho).
rho	Initial correlation for structure = "correlated".
data	Optional dataset (data frame or CSV path) used to check that occ holds the integers 1 to n_occ.
output_file	Output path. If NULL (default), nothing is written. (Formerly file, still accepted with a warning.)
overwrite	Logical. Replace an existing file.
verbose	Logical. Print a message when the file is written.
...	Former argument name, accepted with a warning until version 1.0.

Value

An object of class nm_ctl (control stream lines), invisibly when file is given.

Examples

```
ctl <- nm_write_ctl(input = c("ID", "TIME", "DV", "AMT", "EVID", "MDV", "CMT", "OCC"))
nm_ctl_iov(ctl, parameters = c("CL", "V"), n_occ = 3)
```

---

nm_ctl_m3 *Implement the M3 method for BQL data in a NONMEM control stream*

Description

Rewrites $ERROR so that quantifiable observations keep their usual likelihood and BQL observations contribute the probability P(DV < LLOQ) = PHI((LLOQ - IPRED)/SD) (Beal 2001, method M3), with F_FLAG. The dataset is not modified: comments listing its requirements are inserted after $DATA (BQL flag coded 0/1, BQL records kept, EVID=0 and MDV=0 on BQL records). LAPLACE is added to $ESTIMATION (EM, Bayesian and FO methods are replaced by FOCE-I Laplace), and M3LLOQ, M3SD, M3PBQL are added to the first $TABLE.

Usage

```
nm_ctl_m3(
  ctl,
  bql = "BQL",
  lloq = NULL,
  lloq_col = NULL,
  method_col = NULL,
  method_lloq = NULL,
  residual_sd = NULL,
  npde_mode = FALSE,
  output_file = NULL,
```

```
  overwrite = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

ctl	Path to a control stream, or its lines (character vector or nm_ctl object).
bql	Name of the BQL flag data item (0 = quantifiable, 1 = BQL).
lloq	Constant LLOQ, on the DV scale.
lloq_col	Name of a dataset column holding the LLOQ.
method_col	Name of a dataset column holding a numeric code of the analytical method.
method_lloq	Named numeric vector of LLOQs by method code, e.g. c("1" = 0.1, "2" = 0.05).
residual_sd	NONMEM expression of the residual SD on the DV scale. NULL (default): inferred from the Y = line for additive, proportional, combined and IPRED + W*EPS(1) models.
npde_mode	Logical. Add NPDE_MODE and DV_LOQ handling for the NONMEM NPDE diagnostics.
output_file	Output path. If NULL (default), nothing is written. (Formerly file, still accepted with a warning.)
overwrite	Logical. Replace an existing file.
verbose	Logical. Print a message when the file is written.
...	Former argument name, accepted with a warning until version 1.0.

Details

The LLOQ comes from exactly one of: a constant (lloq), a dataset column (lloq_col), or a code of the analytical method (method_col and method_lloq), for studies pooling assays with different LLOQs.

Value

An object of class nm_ctl (control stream lines), invisibly when file is given.

References

Beal SL. Ways to fit a PK model with some data below the quantification limit. J Pharmacokinet Pharmacodyn. 2001;28(5):481-504.

Examples

```
ctl <- nm_write_ctl(input = c("ID", "TIME", "DV", "AMT", "EVID", "MDV", "CMT", "BQL"),
                    error = "combined")
nm_ctl_m3(ctl, lloq = 0.1)
```

nm_ctl_mu *MU referencing of a NONMEM control stream*

Description

Rewrites log-normal individual parameters, `P = THETA(n)*EXP(ETA(m))` or `P = TVP*EXP(ETA(m))` with `TVP = THETA(n)`, as `MU_m = LOG(THETA(n))` and `P = EXP(MU_m + ETA(m))`, as required by the EM and Bayesian methods (SAEM, IMP, BAYES). The former `TVP` line is kept as a comment. Covariate effects on the typical value are not handled: only parameters written exactly in one of the two forms above are transformed.

Usage

```
nm_ctl_mu(
  ctl,
  parameters = NULL,
  mode = c("readable", "linear"),
  output_file = NULL,
  overwrite = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`ctl`	Path to a control stream, or its lines (character vector or `nm_ctl` object).
`parameters`	Names of the parameters to transform. `NULL` (default): every parameter written in a supported form.
`mode`	`"readable"` (`MU_m = LOG(THETA(n))`, THETAs unchanged) or `"linear"` (`MU_m = THETA(n)`, the THETA initial value and bounds are log-transformed; requires one value per line in `$THETA`; an initial value of 1, whose log is 0, is written 0.0001 because NM-TRAN refuses an initial estimate of 0 unless it is fixed).
`output_file`	Output path. If `NULL` (default), nothing is written. (Formerly `file`, still accepted with a warning.)
`overwrite`	Logical. Replace an existing `file`.
`verbose`	Logical. Print a message when the file is written.
`...`	Former argument name, accepted with a warning until version 1.0.

Value

An object of class `nm_ctl` (control stream lines), invisibly when `file` is given.

Examples

```
ctl <- nm_write_ctl()
nm_ctl_mu(ctl)
nm_ctl_mu(ctl, parameters = c("CL", "V"), mode = "linear")
```

`nm_data_previsual` *Pre-visualise a NONMEM-like dataset*

Description

Pre-visualise a NONMEM-like dataset

Usage

```
nm_data_previsual(
  file,
  sep = ",",
  na = c("", "NA", "."),
  time_unit = c("h", "d"),
  reconstruct_time = FALSE,
  clock_col = NULL,
  date_type = NULL,
  dv_col = NULL,
  strat = NULL,
  ci_probs = c(0.05, 0.95),
  dose_norm = FALSE,
  scales = c("linear", "log", "both"),
  cmt = NULL,
  xlim = NULL,
  blq_handling = c("show", "exclude", "half_lloq", "keep"),
  lloq = NULL,
  blq_col = NULL,
  export_jpg = FALSE,
  export_csv = FALSE,
  output_dir = NULL,
  jpg_width = 9,
  jpg_height = 5,
  jpg_dpi = 300,
  verbose = TRUE,
  ...
)
```

Arguments

`file`	Path to the CSV file (or an already loaded data.frame).
`sep`	Column separator (default `","`).
`na`	Strings to be treated as missing values.
`time_unit`	Output unit for TIME: `"h"` (default) or `"d"`.

`reconstruct_time`
: If `TRUE`, rebuild a relative TIME (t0 = first dose per subject) from a clock-time column and, optionally, a DATE column. If `FALSE` (default), the TIME column is used as is (`as.numeric`). Detecting a DATE/DAT1/DAT2/DAT3 column NEVER triggers the reconstruction automatically.

`clock_col`
: Name of the column holding the clock time (`hh:mm:ss`, `hh:mm` or decimal hours), used when `reconstruct_time = TRUE`. `NULL` (default) = column `"TIME"`.

`date_type`
: How the DATE column is interpreted during the reconstruction (only if `reconstruct_time = TRUE`):

  - `NULL` (default): clock time only, anchored on a single day
  - `"continuous"`: DATE = integer day counter (1, 2, 3...)
  - `"DATE"`: calendar date month-day-year (mm-dd-yy)
  - `"DAT1"`: calendar date day-month-year (dd-mm-yy)
  - `"DAT2"`: calendar date year-month-day (yy-mm-dd)
  - `"DAT3"`: calendar date year-day-month (yy-dd-mm)

`dv_col`
: Name of the DV column to use (manual override).

`strat`
: Name of a stratification column for the plot (e.g. `"DOSE"`, `"OCC"`, `"STUDY"`). `NULL` = none.

`ci_probs`
: Two quantiles defining the interval band around the median (default `c(0.05, 0.95)`, i.e. a 90% interval).

`dose_norm`
: If `TRUE`, also compute DV_norm = DV / AMT (dose given to the subject) and produce the matching plot. Requires an identified AMT column.

`scales`
: `"linear"`, `"log"` or `"both"`: Y axes produced. `"both"` gives side-by-side linear and log panels (patchwork).

`cmt`
: CMT value(s) to keep for the plot. Requires an identified CMT column. `NULL` (default) = all observed compartments (CMT != 0 when EVID is available). E.g. `cmt = 2` for the central compartment.

`xlim`
: X-axis limits `c(min, max)`. `NULL` (default) = full data range.

`blq_handling`
: How BLQ values are handled:

  - `"show"` (default): BLQ drawn as grey crosses below a dashed line at LLOQ; excluded from the median and quantiles
  - `"exclude"`: BLQ removed from the plot and statistics
  - `"half_lloq"`: DV replaced by LLOQ/2 (requires `lloq`)
  - `"keep"`: nothing done, plotted as is

`lloq`
: Numeric LLOQ. Needed to identify BLQ records when no flag column is found, and for `"half_lloq"`. `NULL` = no threshold.

`blq_col`
: Name of the censoring flag column (1 = BLQ, 0 = quantified). If `NULL`, auto-detected among `"BLQ"`, `"BQL"`, `"LLQ"`, `"CENS"`.

`export_jpg`
: If `TRUE`, export the plots as JPG files in `output_dir`.

`export_csv`
: If `TRUE`, export the final data.frame (relative TIME, DV_norm) as CSV in `output_dir`.

`output_dir`
: Output directory (default: directory of the source file). Formerly `out_dir`, still accepted with a warning.

jpg_width	Width in inches of the JPG export (default 9).
jpg_height	Height in inches of the JPG export (default 5).
jpg_dpi	Resolution in dpi (default 300).
verbose	Print the identification report (default TRUE).
...	Former argument name, accepted with a warning until version 1.0: out_dir (now output_dir).

Value

An object of class nm_data_previsual (a list, printed as a short summary):

- data: data.frame with TIME (relative if rebuilt) and DV_norm
- mapping: reserved labels -> original column names
- plot_dv: ggplot of DV vs TIME (according to scales)
- plot_norm: ggplot of DV/dose vs TIME (NULL if dose_norm = FALSE)
- summary: diagnostic data.frame (N subjects, N obs, time range, % BLQ...)
- files: paths of the written files (NULL if nothing was exported)

Examples

```
## Not run:
# TIME already numeric/relative: used as is (default behaviour)
res <- nm_data_previsual("data.csv")

# Rebuild from a clock-time column TIME_R + DATE = day counter
res <- nm_data_previsual("data.csv", reconstruct_time = TRUE,
                         clock_col = "TIME_R", date_type = "continuous")

# Rebuild from a calendar DATE in day-month-year order (DAT1)
res <- nm_data_previsual("data.csv", reconstruct_time = TRUE,
                         clock_col = "TIME", date_type = "DAT1")

res <- nm_data_previsual("data.csv", strat = "DOSE", dose_norm = TRUE,
                         scales = "both", lloq = 0.5, export_jpg = TRUE)
print(res$plot_dv); print(res$summary)

## End(Not run)
```

---

nm_derivatives_ctl *Control stream of derivatives for the linearization of a base model*

Description

Writes a copy of a base control stream that evaluates the model once at the final estimates (MAXEVAL=0, all parameters FIX) and tabulates, for every record, the individual prediction and the derivatives of Y with respect to the ETAs (G_ETAk), the EPSs (H_EPSj) and the cross derivatives (D_EPSj_ETAk, interaction). The derivatives are read through verbatim code (G(k,1), HH(j,1), HH(j,k+1) in $ERROR; G and H in $PRED) placed at the end of the block ("LAST), as in the PsN linearize tool. The table is the input of level 2 of nm_optimize_base().

Usage

```
nm_derivatives_ctl(run, output_dir = NULL, name = NULL, extra = NULL, ...)
```

Arguments

run	Run directory of the base model (e.g. "run1.reg") or path prefix ("dir/run1") whose .ctl and .ext files are used.
output_dir	Directory of the new control stream (formerly out_dir, still accepted with a warning). Default: the analysis directory (parent of a .reg/.nm7 run directory), so that the $DATA path keeps the convention of nm_run().
name	Name of the new run. Default "<run>_deriv".
extra	Additional data items to tabulate (e.g. an occasion column "OCC" for inter-occasion variability). Items absent from $INPUT are dropped with a warning.
...	Former argument name, accepted with a warning until version 1.0: out_dir (now output_dir).

Details

Not testable without NONMEM: the verbatim code follows the NONMEM help (ERROR: G(i,1) is the partial derivative of F with respect to ETA(i); HH(j,1) with respect to EPS(j); HH(j,i+1) the second derivative with respect to EPS(j) and ETA(i)). Models with F_FLAG (M3 likelihood) or simulation-only records are not supported.

Value

Invisibly, the path of the control stream written. The table file is <name>.dtab.

See Also

nm_optimize_base()

---

nm_export_data *Export a data.frame to CSV (NONMEM convention: na = ".")*

Description

Export a data.frame to CSV (NONMEM convention: na = ".")

Usage

```
nm_export_data(data, output_file, ...)
```

Arguments

data	A data.frame, typically from nm_read_csv().
output_file	Output path (.csv). The directory is created if needed. (Formerly file, still accepted with a warning.)
...	Former argument name, accepted with a warning until version 1.0.

Value

The path of the written file (invisibly).

`nm_export_plot` *Export a ggplot to an image file*

Description

Single export entry point for all nmpilot plots: plot methods return the ggplot object and leave saving to this function.

Usage

```
nm_export_plot(
  plot,
  output_file,
  width = 9,
  height = 5,
  dpi = 300,
  verbose = TRUE,
  ...
)
```

Arguments

`plot`	A ggplot (or patchwork) object.
`output_file`	Output path. The file type is taken from the extension: `.jpg`/`.jpeg`, `.png`, `.pdf` or `.tif`/`.tiff`. The directory is created if needed. (Formerly `file`, still accepted with a warning.)
`width`	Width in inches (default 9).
`height`	Height in inches (default 5).
`dpi`	Resolution in dots per inch (default 300; ignored for PDF).
`verbose`	Print the path of the written file (default `TRUE`).
`...`	Former argument name, accepted with a warning until version 1.0.

Value

The path of the written file (invisibly), or `NULL` if `plot` is `NULL`.

Examples

```
## Not run:
p <- nm_plot_dv(nm_read_csv("data.csv")$data)
nm_export_plot(p, "figures/dv_vs_time.png")

## End(Not run)
```

`nm_extract_run_table` *Parameter table of one NONMEM run, exported to Word*

Description

Reads the `.ext` file of a run and writes a Word document with the parameter estimates, 95% confidence intervals and RSE. When there is no `.ext` file, or when it holds no estimates (empty file), the estimates printed in the `.lst` file are used, as in `nm_extract_runs()`; the table footer then says so. ETA and EPS standard deviations are derived from the variances (delta method for their SE); ETA correlations are derived from the OMEGA matrix.

Usage

```
nm_extract_run_table(
  run = NULL,
  run_name = NULL,
  run_path = NULL,
  output_file = "nonmem_results.docx",
  extensions = c("nm7", "reg"),
  ...
)
```

Arguments

`run`	Path of the run: run directory, control stream or stem.
`run_name`	Deprecated: with `run_name`, `run` is read as the parent directory (former argument `run_rep`) and `run_name` as the run name. A full path given here by mistake is converted automatically.
`run_path`	Deprecated: former name of `run`.
`output_file`	Name of the Word file, written in the run directory (default `"nonmem_results.docx"`). (Formerly `output_name`, still accepted with a warning.)
`extensions`	Run directory extensions to try first, in order (default `c("nm7", "reg")`), before the folder `<run>` and the flat layout.
`...`	Former argument name, accepted with a warning until version 1.0: `run_rep` (with `run_name`).

Details

The run is given by its path, `run`: its run directory (`.nm7`, `.reg`), its control stream, or its stem (e.g. `C:/project/run1` for `C:/project/run1.ext`). The `.ext` file is looked for in `<run>.nm7/`, `<run>.reg/`, a folder named `<run>`, then directly in the folder of the run (flat layout, as produced by a plain nmfe call). Wings for NONMEM is not required.

The former call forms, `run_rep` (parent directory) + `run_name`, and `run_path`, are still accepted with a warning until version 1.0.

Value

Invisibly, an object of class nm_run_table (a list, printed as a short summary) with flextable, params (parameter table), ext (parsed .ext, or estimates read in the .lst, with source = "lst"), output_file, ext_file (NA without .ext), lst_file, run_dir and extension.

Examples

```
## Not run:
res <- nm_extract_run_table("C:/nm76g64/run/para/run_1_prop_kaBbloc.reg")
res <- nm_extract_run_table("C:/nm76g64/run/para/run_1_prop_kaBbloc.ctl")
res$params

## End(Not run)
```

---

nm_extract_runs *Extract the results of several NONMEM runs*

Description

Extract the results of several NONMEM runs

Usage

```
nm_extract_runs(path, runs = NULL, verbose = TRUE)
```

Arguments

path	Analysis directory, or the directory of a single run.
runs	Optional names of the runs to keep.
verbose	Print progress.

Details

When the control stream of a run is found, parameters are named from the model code with nm_identify_parameters(): typical-value THETAs and ETAs after the PK parameter they act on (CL, V...), covariate-effect and residual-error THETAs after their $THETA comment label when it is a short name (CL_WT, PROP, ADD...). Other THETAs keep their NONMEM name (THETA6...).

Value

An object of class nm_runs with:

- $runs: one row per run
- $params: one row per parameter (same = TRUE for the OMEGA elements of an $OMEGA ... SAME record)
- $mixture: for $MIX runs with a .phm file, one row per subject and subpopulation (PMIX, assigned = most probable subpopulation)
- $path: the analysis directory
- $created: time of extraction

Several estimation steps, several problems, failed runs

With several $EST records, the OFV, the minimisation status, the shrinkage and the covariance step are those of the last estimation step (est_methods lists the steps, e.g. "ITS > SAEM > IMP(eval) > FOCEI"). With several $PROB, the last problem that has an estimation is read (n_problems). The estimates are read in the .ext; when it is empty, has no final row, or reports an OFV different from the .lst (an .ext left by another run), they are read in the .lst (params_source) and a warning is given. run_state is "completed", "interrupted" (the output stops before the end of the last step), "terminated" (PROGRAM TERMINATED), "nmtran_error" or "simulation" (simulation-only run, $SIMULATION ONLYSIM, finished: no OFV, min_ok is NA; subproblems are not counted in n_problems). n_omega and n_sigma count the estimated elements only (not FIX, not SAME, not the zeros outside the blocks); n_eta_same counts the ETAs declared SAME (inter-occasion variability); ETAs fixed to 0 are left out of max_eta_shrink. For $MIX runs, n_subpop is the number of subpopulations, the mixing-proportion THETAs are named MIX_P when they have no other name, the shrinkage of an ETA is its smallest value over the subpopulations, and mix_assign counts the subjects assigned to each subpopulation.

---

nm_filter_runs *Filter NONMEM runs and recompute relative criteria*

Description

Filter NONMEM runs and recompute relative criteria

Usage

```
nm_filter_runs(
  x,
 run_filter = c("all", "successful", "successful_no_warning", "successful_no_comment",
    "clean")
)
```

Arguments

x	An object of class nm_runs.
run_filter	Filter to apply: "all" (default), "successful", "successful_no_warning", "successful_no_comment" or "clean".

Value

A filtered nm_runs object. The $runs and $params tables are filtered together; dOFV, rank_ofv, dAIC and dBIC are recomputed on the remaining runs.

nm_gof *Goodness-of-fit diagnostic plots from a NONMEM $TABLE file*

Description

Reads a NONMEM output table (space/tab-delimited, possibly with repeated headers and multiple "TABLE NO." blocks), keeps the observation rows, and builds the standard goodness-of-fit diagnostics: observed vs predictions and residual plots (CWRES / NPDE / IWRES). Plots are only built when the required columns are present. Optionally writes all plots to a Word (.docx) file in the analysis directory.

Usage

```
nm_gof(
  file,
  pred = c("PRED", "CPRED"),
  time = c("TIME", "TAD"),
  by = NULL,
  loess = TRUE,
  reg = TRUE,
  se = FALSE,
  alpha = 0.5,
  log_pred = FALSE,
  stem = NULL,
  save_docx = FALSE,
  output_file = NULL,
  width = 6.5,
  height = 5,
  dpi = 150,
  ...
)
```

Arguments

file	Path to a NONMEM table file, OR a run directory (.reg / .nm7 / any folder) inside which a NONMEM table is auto-detected. Since the table is not necessarily named "sdtab", detection is by content ("TABLE NO." header), optionally filtered by stem.
pred	Preference for the population prediction column used on x-axes. The first available name is used. Default c("PRED","CPRED"). Pass e.g. pred = "CPRED" to force one.
time	Preference for the time column used on residual-vs-time plots. Default c("TIME","TAD").
by	Optional character vector of column names for faceting (facet_wrap), e.g. c("SEX","CMT").
loess	Add a loess trend (default TRUE).
reg	Add a linear regression (default TRUE).

se	Show confidence band around the linear regression (default FALSE).
alpha	Point transparency (default 0.5).
log_pred	Log10 axes for the DV-vs-prediction plots (default FALSE).
stem	Optional filename stem to disambiguate auto-detection in a run directory (e.g. "sdtab", "patab"). Not required to be "sdtab".
save_docx	If TRUE, write all plots to a .docx in the analysis directory (default FALSE).
output_file	Output .docx path. If NULL and save_docx=TRUE, a name is derived from the table file, written next to it. (Formerly docx_file, still accepted with a warning.)
width, height, dpi	Image size (inches) and resolution for the docx export.
...	Former argument name, accepted with a warning until version 1.0.

Value

An object of class "nm_gof": a list with plots (named list of ggplot objects), data (the filtered observation rows), file, columns (resolved column mapping), skipped (plots that could not be built and why), and by.

Examples

```
## Not run:
g <- nm_gof("C:/nm74g64/run/lopi_04.reg/lopi_04_03_04.fit")
g                        # summary
plot(g)                  # all panels
plot(g, type = "dv_pred")
nm_gof(".../run.reg", pred = "CPRED", by = "SEX", save_docx = TRUE)

## End(Not run)
```

---

nm_guide *Interactive task guide*

Description

Builds and opens an HTML page that lists the steps of a population PK analysis with nmpilot. Buttons group the tasks by stage (data preparation, exploration, control streams, running, results, diagnostics, model selection, validation, reporting); selecting a task shows the matching nmpilot function(s), a short description and an example of R code that can be copied. Tasks whose function is not available yet are shown as "planned".

Usage

```
nm_guide(output_file = NULL, open = interactive(), ...)
```

Arguments

output_file	Output path of the HTML page. Default: nmpilot_guide.html in the temporary directory of the R session. (Formerly file, still accepted with a warning.)
open	Open the page in the web browser (default: in interactive sessions only).
...	Former argument name, accepted with a warning until version 1.0.

Details

The page is generated from the file guide/guide_tasks.csv installed with the package (one row per task: stage, task, functions, description, code, status), so that it follows the package. It is a single self-contained file: no internet connection is needed. The examples use the example project of the package, copied to a temporary folder (see the "Setup" block of the page); nmfe must be set to the path of your NONMEM launcher.

Value

The path of the HTML file, invisibly.

Examples

```
p <- nm_guide(open = FALSE)
file.exists(p)
## Not run:
nm_guide()     # opens the guide in the browser

## End(Not run)
```

---

nm_identify_directory *Identify THETA/ETA roles for all control streams of a directory*

Description

Applies nm_identify_parameters() to every .ctl file found in path and stacks the parameter tables. A file that cannot be parsed gives one row with its error message.

Usage

```
nm_identify_directory(path, recursive = TRUE, likely_pk_only = TRUE)
```

Arguments

path	Directory to search.
recursive	Search subdirectories (default TRUE).
likely_pk_only	Passed to nm_identify_parameters().

Value

A data.frame with a `file` column followed by the columns of `parameter_table` (or an `error` column for files that failed).

`nm_identify_parameters` *Identify the semantic role of THETA and ETA in a NONMEM control stream*

Description

Parses `$THETA`, `$OMEGA`, `$PK` and `$ERROR` of a control stream (or `$PRED` when there is no `$PK`) and links each THETA and ETA to the PK parameter(s) it acts on, with a semantic role: `typical_value`, `covariate_effect`, `iiv_scale`, `residual_error`, `iiv` or `structural_other`. Executable equations are the primary source; comments are only used as labels and consistency checks.

Usage

```
nm_identify_parameters(ctl, likely_pk_only = TRUE, ...)
```

Arguments

`ctl`	Path to a NONMEM control stream (`.ctl`). (Formerly `file`, still accepted with a warning.)
`likely_pk_only`	If `TRUE` (default), `mapping_detail` keeps only variables that look like PK parameters (CL, V, Q, KA, F, ALAG...). `mapping_detail_all` always keeps every variable.
`...`	Former argument name, accepted with a warning until version 1.0: `file` (now `ctl`).

Details

Covariate effects are recognised in power, exponential, linear, saturable and conditional forms (e.g. `(WT/70)**THETA(4)`, `EXP(THETA(4)*WT)`, `THETA(1)*(1 + THETA(4)*(WT-70))`, `C_RES/(THETA(4) + C_RES)`, `IF(SEX.EQ.1) TVCL = TVCL*THETA(4)`, including `IF ... THEN ... ELSE ... ENDIF` blocks). A variable defined several times keeps the THETA/ETA of all its definitions. When a variable is only defined under conditions (e.g. `IF(SEX.EQ.1) CL = THETA(1)` / `IF(SEX.EQ.0) CL = THETA(2)`), the THETA of its first definition in the file is the reference (`typical_value`) and the others are covariate effects. A covariate is any data item declared in `$INPUT` that is not a NONMEM-defined item (`ID`, `TIME`, `DV`, `AMT`, `MDV`, `EVID`, `CMT`...), whatever its name; items dropped with `=DROP` are ignored. Without `$INPUT`, any symbol used in `$PK` but not defined there is taken as a data item.

Wings for NONMEM extended control streams, where model records use the labels of the `$THETA`, `$OMEGA` and `$SIGMA` comments (e.g. `CL = POPCL*EXP(eta_cl)`), are handled: such labels are first replaced by `THETA(n)`, `ETA(n)` or `ERR(n)`, as in `nm_convert_ctl()`. A label is replaced only if it is used but never assigned in a model record and is not a `$INPUT` item.

Value

An object of class nm_parameter_map, a list with, among others: parameter_table (one row per PK parameter with its THETA/ETA by role; derived_from names the parameters of the table that a derived parameter is computed from, e.g. "V" for S2 = V or "CL, V" for KE = CL/V, and is NA for the others), mapping_detail and mapping_detail_all (one row per PK parameter / THETA or ETA association; depth is the number of assignment steps between the THETA/ETA and the parameter, 0 when it appears directly in its definition), residual_error, special_theta (THETAs of model event times MTIME(n) in $PK and of mixture probabilities P(n) in $MIX, with their role model_event_time or mixture_probability; they are not counted in unmapped_theta), theta, omega, omega_elements (variances and BLOCK covariances), unmapped_theta, unmapped_omega, missing_theta_declaration, missing_omega_declaration and label_substitutions (the labels replaced and their THETA(n)/ETA(n)/ERR(n) reference; empty for a standard control stream).

See Also

nm_identify_directory(), nm_extract_runs()

Examples

```
## Not run:
x <- nm_identify_parameters("run.ctl")
x$parameter_table
x$mapping_detail
x$omega_elements

## End(Not run)
```

---

nm_jackknife *Leave-one-subject-out jackknife of a NONMEM model*

Description

Removes one subject at a time from the data set, re-estimates the model with NONMEM and measures how much each subject influences the fit. The primary metric is the likelihood influence $|\Delta OFV| = |OFV_{(-i)} - OFV_{ref}|$; the secondary metrics are the relative THETA shifts $(\theta_{(-i)} - \theta_{ref})/\theta_{ref}$, their L2 norm, their maximum, and an L2 norm standardised by the reference standard errors when the reference covariance step is usable. Subjects are ranked by $|\Delta OFV|$. The analysis is descriptive (an influence diagnostic, not a formal outlier test).

Usage

```
nm_jackknife(
  ctl,
  nmfe_path = NULL,
  lst = NULL,
  reference_data = NULL,
  id_col = "ID",
```

```
  comment_char = NULL,
  sep = ",",
  output_dir = NULL,
  overwrite_existing_runs = FALSE,
  strip_tables = TRUE,
  strip_cov = FALSE,
  dtheta_threshold = 0.01,
  save_figures = TRUE,
  run = TRUE,
  verbose = TRUE,
  n_cores = 1L
)

## S3 method for class 'nm_jackknife'
print(x, digits = 4, ...)
```

Arguments

ctl	Path to the reference control stream (.ctl), already run.
nmfe_path	Path to the nmfe launcher (e.g. "C:/nm76g64/run/nmfe76.bat" on Windows, "/opt/nm760/run/nmfe76" on Linux or macOS). Needed when run = TRUE.
lst	Reference output. Default: the .lst of the same name next to ctl.
reference_data	Data file. NULL (default): read from $DATA.
id_col	Label of the subject column in $INPUT (default "ID").
comment_char	Data comment/ignore character. NULL (default): the single-character IGNORE of $DATA ("#", "@", a letter...); "": none.
sep	Field separator of the data file (default ",").
output_dir	Working directory (default "<ctl_dir>/<run>_jackknife"). Holds jk_data/, jk_ctl/, jk_run/ (one subdirectory per run), figures/ and the two CSV files.
overwrite_existing_runs	FALSE (default): skip the runs already minimised (resume).
strip_tables	Remove $TABLE from the jackknife control streams (default TRUE).
strip_cov	Remove $COVARIANCE from the jackknife runs (default FALSE). TRUE is faster and does not change the influence metrics (the reference SE come from the reference run).
dtheta_threshold	Relative THETA shift flagged as large (default 0.01).
save_figures	Write the three figures to output_dir/figures (default TRUE).
run	FALSE: only write the data sets and control streams (no NONMEM).
verbose	Progress messages (default TRUE).
n_cores	Number of jackknife runs at the same time (default 1). A whole number, "auto" (physical cores minus one, see nm_cores()) or "ask" (interactive sessions: the cores are shown and the number is asked once per R session, then reused; options(nmpilot.n_cores = NULL) asks again). Each run is then started in

	the background in its own directory; if R is interrupted, the runs already started go on in the background and a later call with the same arguments picks up their results.
x	An nm_jackknife object.
digits	Significant digits in the printed table (default 4).
...	Ignored.

Details

The method is that of the standalone script nonmem-jackknife-influence (Simon N, doi:10.5281/zenodo.22257994). The data lines are copied verbatim (positional reading through $INPUT), so number formatting and rows removed by a conditional IGNORE=(...) are preserved. Runs already minimised are skipped on a second call (resume), unless overwrite_existing_runs = TRUE.

Value

An object of class nm_jackknife: the reference OFV, THETA and SE, all_runs (every replicate, including the failed ones), summary (successful runs ranked by $|\Delta OFV|$, with rank, rank_dtheta and, when available, rank_dtheta_z) and the settings. jackknife_all_runs.csv and jackknife_summary.csv are written to output_dir.

See Also

nm_bootstrap(), nm_profile()

Examples

```
## Not run:
jk <- nm_jackknife("C:/study/run2.reg/run2.ctl",
                   nmfe_path = "C:/nm76g64/run/nmfe76.bat")
jk
plot(jk, "bar")
plot(jk, "scatter")

## End(Not run)
```

---

nm_model_diagram *Diagram of the models tested in a project*

Description

Draws the tree of the runs of a project: each run is linked to the run it was derived from (its parent), and shows its OFV, the change of OFV from its parent (dOFV) and the change in the number of estimated parameters, coloured by convergence state.

Usage

```
nm_model_diagram(x, parents = NULL, label_width = 40)
```

Arguments

x	Result of nm_extract_runs(), or the path of an analysis directory, which is then read with nm_extract_runs(x).
parents	Optional links that complete or replace those read in the control streams: a named character vector (c(run2 = "run1")) or a data frame with columns run and parent. NA removes a link.
label_width	Maximum width of the description (text of $PROBLEM) in the table; longer texts are cut.

Details

The parent of a run is read in its control stream, in a comment written with the PsN and Pirana convention, e.g. ;; 1. Based on: run3 or ;; Based on: 3 (the number alone stands for run3). The parents argument completes or corrects these links. A run whose parent is not found is drawn as a root.

Value

An object of class nm_model_diagram: a list with tree, a data frame with one row per run (run, parent, parent_source: "control stream", "argument" or "none", ofv, dofv_parent, n_par, dpar_parent, status, description, and the layout coordinates x and y; rows in depth-first order, each run right after its parent), and plot, the ggplot2 tree. plot() returns it (shown when not assigned); it can be saved with nm_export_plot().

See Also

nm_extract_runs(), nm_compare_runs()

Examples

```
ex <- system.file("extdata", "example", package = "nmpilot")
d <- nm_model_diagram(ex, parents = c(run2 = "run1"))
d
plot(d)
```

---

nm_optimize_base *Proposals to improve a base model without iterative NONMEM runs*

Description

Examines a base run and proposes changes to the residual error model, the random effects and the covariate model.

Usage

```
nm_optimize_base(
  run,
  covariates = NULL,
  deriv = NULL,
  nmfe_path = NULL,
  data = NULL,
  occ = NULL,
  level = 2L,
  cov_form = c("power", "exponential"),
  p_forward = 0.05,
  p_backward = 0.01,
  max_levels = 5L,
  objective = c("focei", "fast"),
  verbose = TRUE
)

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

## S3 method for class 'nm_optimize_base'
plot(x, type = c("all", "residual", "random", "covariate"), ...)
```

Arguments

run	Run directory (e.g. "run1.reg") or path prefix of the base run.
covariates	Covariates to test (columns of the data set). NULL: no covariate analysis.
deriv	Table written by the run of nm_derivatives_ctl() (<run>_deriv.dtab). NULL: searched for next to the run, otherwise the function runs level 1 only (unless nmfe_path is given).
nmfe_path	Path of nmfe: if given and no derivatives table exists, the derivatives control stream is written and run with nm_run().
data	Data set (path or data.frame) holding the covariates. Default: the $DATA file of the run.
occ	Occasion column for inter-occasion variability (level 2; it must be in the derivatives table: nm_derivatives_ctl(extra = "OCC")).
level	1 or 2 (2 runs both levels when the table is available).
cov_form	Form of continuous covariate effects: "power" ((COV/median)^theta) or "exponential" (exp(theta*(COV-median))).
p_forward, p_backward	Significance levels of the forward inclusion and backward elimination of covariates (linearized model).
max_levels	A numeric covariate with at most this number of distinct values is treated as categorical.

objective	Objective of the linearized model. "focei" (default): the FOCE-I objective of NONMEM with interaction (individual modes of the full individual objective, residual-variance term in the determinant); at the NONMEM estimates it reproduces the OFV of the base run. "fast": an approximation (modes given by the linear predictor with the residual variance held fixed), about three times faster, whose OFV differs from NONMEM's by a few units when the residual error depends on the ETAs; the dOFV are close.
verbose	Print progress.
x	An nm_optimize_base object.
...	Ignored.
type	For plot(): "all", "residual", "random" or "covariate". Formerly area, still accepted with a warning.

Details

Level 1 uses only the outputs of the base run (.phi, .ext, a table with CWRES, IPRED and TIME, and the data set): resmod-like models on the CWRES (power, time-varying, autocorrelated residuals, individual residual variance), EBE correlations, distribution and shrinkage, and an ETA-covariate screening.

Level 2 needs one extra NONMEM evaluation run written by nm_derivatives_ctl(). Its table of derivatives gives a first-order linearization of the model around the EBEs; every candidate is re-estimated in R under the FOCE-I objective of the linearized model: residual models (additive, proportional, combined, power, time-varying), OMEGA structure (full block, pairwise blocks, removal of an ETA, inter-occasion variability), covariates on the ETAs (univariate tests, then forward inclusion and backward elimination), and finally the combined proposal.

The changes in objective function (dOFV) are those of the linearized model: they approximate the changes of the real model, and the proposal must be confirmed by a NONMEM run of the modified model. The quality of the approximation is reported in check (linearized vs NONMEM OFV of the base model).

Value

An object of class nm_optimize_base: lists level1 (residual, random, pairs, covariates) and level2 (check, residual, random, covariates, scm, proposal), and proposal_text.

References

Khandelwal A, Harling K, Jonsson EN, Hooker AC, Karlsson MO. A fast method for testing covariates in population PK/PD models. AAPS J 2011;13:464-72. Ibrahim MMA, Nordgren R, Kjellsson MC, Karlsson MO. Model-based residual post-processing for residual model identification. AAPS J 2018;20:81.

See Also

nm_derivatives_ctl(), nm_wasserstein()

Examples

```
ex <- system.file("extdata", "example", "run1.reg", package = "nmpilot")
o <- nm_optimize_base(ex, covariates = c("WT", "GENO"), level = 1, verbose = FALSE)
o
```

`nm_plot_dv` *Plot DV vs TIME with median and interval band*

Description

Plot DV vs TIME with median and interval band

Usage

```
nm_plot_dv(
  data,
  y = c("DV", "DV_norm"),
  time_var = c("TIME", "TAD"),
  strat = NULL,
  ci_probs = c(0.05, 0.95),
  cmt = NULL,
  xlim = NULL,
  scales = c("linear", "log", "both"),
  blq_handling = c("show", "exclude", "keep"),
  verbose = TRUE
)
```

Arguments

`data`	A data.frame from `nm_read_csv()`.
`y`	Y variable: `"DV"` (default) or `"DV_norm"`.
`time_var`	X variable: `"TIME"` (default) or `"TAD"` (time after dose, computed by `nm_read_csv()` when EVID is present).
`strat`	Name of a stratification column (e.g. `"DOSE"`).
`ci_probs`	Quantiles of the interval band (default 90% interval, `c(0.05, 0.95)`).
`cmt`	CMT value(s) to keep (`NULL` = all).
`xlim`	X-axis limits `c(min, max)` (`NULL` = full range).
`scales`	`"linear"`, `"log"` or `"both"` (side-by-side panels).
`blq_handling`	BLQ handling: `"show"` (grey crosses, default), `"exclude"` or `"keep"`.
`verbose`	Print diagnostics (default `TRUE`).

Value

A ggplot (or a patchwork object if `scales = "both"`).

Examples

```
## Not run:
parsed <- nm_read_csv("data.csv", lloq = 0.5)
p <- nm_plot_dv(parsed$data, strat = "DOSE", cmt = 2, scales = "both")
print(p)

## End(Not run)
```

`nm_profile` *Profile the objective function over one model parameter*

Description

Runs a series of NONMEM estimations in which a single `$THETA` is fixed to each value of a grid while every other parameter is re-estimated, yielding the profile of the objective function value (OFV, *without* the additive constant) for that parameter. The minimum of the curve locates the conditional estimate; the width at `min(OFV) + ci_threshold` gives an approximate likelihood-based confidence interval.

Usage

```
nm_profile(
  ctl,
  nmfe_path,
  param,
  from,
  to,
  by,
  output_dir = NULL,
  strip_tables = TRUE,
  strip_cov = TRUE,
  overwrite = FALSE,
  ci_threshold = 3.84,
  digits = NULL,
  run = TRUE,
  verbose = TRUE,
  n_cores = 1L,
  ...
)

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

## S3 method for class 'nm_profile'
plot(x, type = c("ofv", "dofv"), ...)
```

Arguments

ctl	Path to the reference control file. (Formerly ctl_path, still accepted with a warning; the former order, nmfe_path first, is also recognised.)
nmfe_path	Path to the NONMEM launcher (e.g. nmfe76.bat on Windows, the nmfe76 script on Linux or macOS).
param	The parameter to profile: a $THETA label (matched whole-word, case-insensitive, e.g. "CL") or an index (3, "3", or "THETA(3)").
from, to, by	Numeric grid: first value, last value, and increment.
output_dir	Output directory. Defaults to <ctl_dir>/<ctl>_<param>_profile.
strip_tables, strip_cov	Logical; drop $TABLE / $COV from the generated controls. Both TRUE by default (recommended for a profile).
overwrite	Logical; if FALSE (default) a run whose .lst already exists and minimised successfully is skipped (resume after interruption).
ci_threshold	Numeric; delta-OFV cut-off for the interval. Default 3.84 (chi-square, 1 df, two-sided 95%). Set NA to disable. Note: for a variance component tested at its boundary ($\omega^2 = 0$) the appropriate one-sided cut-off is 2.71, not 3.84.
digits	Decimals used to format fixed values and run labels. Auto- derived from by when NULL.
run	Logical; if FALSE, generate the control files without launching NONMEM.
verbose	Logical; print progress messages (default TRUE). Formerly quiet (the opposite), still accepted with a warning.
n_cores	Number of grid values run at the same time (default 1). A whole number, "auto" (physical cores minus one, see nm_cores()) or "ask" (interactive sessions: the cores are shown and the number is asked once per R session, then reused; options(nmpilot.n_cores = NULL) asks again). Each run is then started in the background in its own directory; if R is interrupted, the runs already started go on in the background and a later call with the same arguments picks up their results.
...	Ignored.
x	An nm_profile object.
type	Which figure to draw: "ofv" (OFV vs value) or "dofv" (delta-OFV vs value, with the ci_threshold line).

Details

Only $DATA is remapped (to an absolute path) and $TABLE/$COV are stripped by default; every other record is preserved byte-for-byte. Each run is written to its own sub-directory under output_dir/runs/.

Value

An object of class nm_profile: a list with results (a data frame of run, value, ofv, dOFV, min_ok, cov_ok, cov_warning, dir, lst), plus ci, min_ofv, min_value, param, param_safe, target, ci_threshold, and output_dir. Use print() for a summary and plot() (with type = "ofv" or "dofv") for figures.

See Also

nm_run(), nm_extract_runs(), nm_compare()

Examples

```
## Not run:
prof <- nm_profile(
  ctl       = "C:/nm76g64/run/test/run_test.reg/run_test.ctl",
  nmfe_path = "C:/nm76g64/run/nmfe76.bat",
  param     = "CL", from = 0.1, to = 3, by = 0.05
)
prof                        # summary (print.nm_profile)
plot(prof)                  # OFV vs value
plot(prof, type = "dofv")  # dOFV vs value, with the 3.84 threshold

## End(Not run)
```

---

nm_read_csv *Read and normalise a NONMEM-like dataset*

Description

Read and normalise a NONMEM-like dataset

Usage

```
nm_read_csv(
  file,
  sep = ",",
  na = c("", "NA", "."),
  time_unit = c("h", "d"),
  lloq = NULL,
  blq_col = NULL,
  date_format = NULL,
  dv_col = NULL,
  verbose = TRUE
)
```

Arguments

file	Path to the CSV file.
sep	Column separator (default ",").
na	Strings to be treated as missing values.
time_unit	Output unit for TIME: "h" (default) or "d".
lloq	LLOQ value. If supplied and no BLQ/CENS column exists, records with DV < lloq are flagged as BLQ.

blq_col	Name of the BLQ flag column (1 = BLQ, 0 = quantified). By default, "BLQ" or "CENS" is auto-detected.
date_format	Date format forced in lubridate::parse_date_time() (e.g. "dmy" for DD/MM/YYYY). By default, chosen from the name of the DATE/DAT1-3 column.
dv_col	Name of the DV column if non-standard (e.g. "DV_nmol/L"). If NULL (default), a column whose name starts with "DV" is auto-detected. Required when several DV columns exist.
verbose	Print the identification report (default TRUE).

Value

A list:

- data: data.frame with relative TIME, BLQ (logical), and DV_norm when AMT is available. Reserved columns are renamed in upper case.
- mapping: reserved labels -> original column names.
- lloq: LLOQ value used (NULL if not supplied).

Examples

```
## Not run:
res <- nm_read_csv("data.csv", lloq = 0.5)
head(res$data)

## End(Not run)
```

---

nm_run *Run one or more NONMEM models*

Description

Sequentially launches a list of NONMEM control streams via the nmfe executable (direct call, no Wings for NONMEM, no PsN required). By default each run gets its own subdirectory <run>.reg inside analysis_dir, holding a copy of the control stream and all NONMEM output files; with run_subdir = "" the run is made directly in the folder of its control stream, as when nmfe is called by hand. The data file is not copied. A relative $DATA path is accepted: it is looked up from the folder of the control stream ($DATA data.csv), then from the run directory ($DATA ../data.csv, Wings style), and written as an absolute path in the copied control stream.

Usage

```
nm_run(
  ctl,
  nmfe_path,
  analysis_dir,
  skip_terminated = TRUE,
  accept_failed = TRUE,
```

```
  overwrite_ctl = TRUE,
  run_subdir = ".reg",
  verbose = TRUE,
  n_cores = 1L,
  ...
)
```

Arguments

ctl	Character vector of paths to .ctl files. May be absolute paths or file names relative to analysis_dir. (Formerly ctl_files, still accepted with a warning; the former order, nmfe_path first, is also recognised.)
nmfe_path	Full path to the nmfe executable (e.g. "C:/nm76g64/run/nmfe76.bat" on Windows, "/opt/nm760/run/nmfe76" on Linux or macOS, where it is run directly and must be executable).
analysis_dir	Root directory of the analysis. One run subdirectory is created here for each run (see run_subdir). A warning is issued when it lies in a folder synchronised by a cloud client (Dropbox, OneDrive, Google Drive, iCloud, Box): the client can lock the files written by NM-TRAN and make a run fail (silence it with options(nmpilot.warn_synced = FALSE)).
skip_terminated	Logical. If TRUE (default), runs whose .lst file already exists and reports a terminal state are skipped.
accept_failed	Logical. Only used when skip_terminated = TRUE. If TRUE (default), runs that terminated with an error are also skipped (not re-launched). If FALSE, only runs with run_ok = TRUE are skipped.
overwrite_ctl	Logical. If TRUE (default), the .ctl file is overwritten in the .reg directory if already present.
run_subdir	Suffix of the run subdirectories: ".reg" (default, Wings-compatible layout run001.reg/), any other suffix (e.g. ".run"), or "" to run in the folder of the control stream itself, without copying it (outputs next to the .ctl). Control streams in Wings label syntax need a subdirectory, because the converted copy is written there.
verbose	Logical. If TRUE (default), progress messages are printed to the console.
n_cores	Number of runs launched at the same time (default 1: one after the other, NONMEM output shown in the console). A whole number, "auto" (physical cores minus one, see nm_cores()) or "ask" (interactive sessions: the cores are shown and the number is asked once per R session, then reused; options(nmpilot.n_cores = NULL) asks again). In parallel, each run is started in the background from its run directory; the console output of nmfe goes to <run>_nmpilot.log. Runs written in the same folder (run_subdir = "") are always run one after the other. If R is interrupted, the runs already started go on in the background.
...	Former argument names, accepted with a warning until version 1.0: ctl_files (now ctl).

Details

Wings for NONMEM is not needed. The default `.reg` layout mirrors the Wings output structure (a run named `run001.ctl` produces a directory `run001.reg/` containing `run001.ctl`, `run001.lst`, `run001.ext`, etc.), so that projects started with Wings can be continued; the other functions of the package read runs in any of these layouts.

Run status is parsed from the `.lst` file after completion. The `skip_terminated` mechanism reads any existing `.lst` before launching, allowing safe re-use of `nm_run()` on a partially completed batch.

Value

A run log of class `nm_run_log` (a `data.frame`, printed as a short summary), with one row per run and the following columns:

run Run name (without extension).

reg_dir Full path to the run directory (`<run>.reg`, or the folder of the control stream when `run_subdir = ""`).

action One of `"launched"`, `"skipped_done"`, or `"skipped_no_ctl"`.

exit_status Shell return code from `nmfe` (0 = success).

lst_exists Logical: does the `.lst` file exist after the run?

ofv Objective function value (OFV).

minimization_status Minimization status string: `"successful"`, `"terminated"`, `"rounding_errors"`, `"min_success_cov_failed"`, `"evaluation"` (run without minimization, `MAXEVAL=0`; `min_ok` is then `NA` and `run_ok` `TRUE`), `"simulation"` (simulation-only run, `$SIMULATION ONLYSIM`, finished; `ofv` and `min_ok` are then `NA` and `run_ok` `TRUE`), or `"unknown"`.

min_ok Logical: did minimization succeed?

cov_ok Logical: did the covariance step succeed?

run_ok Logical: is the run globally acceptable (minimization and covariance both successful)?

cov_status Detailed covariance step status: `"successful"`, `"aborted"`, `"not_requested"`, or `"unknown"`.

See Also

`nm_extract_runs()` for parsing NONMEM output files after a run.

Examples

```
## Not run:
log <- nm_run(
  ctl             = c("run001.ctl", "run002.ctl", "run003.ctl"),
  nmfe_path       = "C:/nm76g64/run/nmfe76.bat",
  analysis_dir    = "C:/analyses/my_pk_project",
  skip_terminated = TRUE,
  accept_failed   = TRUE,
  verbose         = TRUE
)
```

```
# Show only failed runs
log[!log$run_ok, ]

## End(Not run)
```

`nm_runs_table` *Summary table of all the runs of a project*

Description

Builds one table with one row per run of a project and, when `output_file` is given, writes it to a Word document. Each run is shown with its description (text of `$PROBLEM`), the run it was derived from (its parent, read in the control stream as in `nm_model_diagram()`), its OFV, the change of OFV from its parent and from the best run, its number of estimated parameters, its AIC and BIC, its minimisation and covariance status, its largest ETA shrinkage and the share of parameters estimated with a relative standard error above 50%.

Usage

```
nm_runs_table(
  x,
  parents = NULL,
  sort = c("tree", "ofv", "run"),
  output_file = NULL,
  overwrite = FALSE,
  title = "Summary of the runs",
  verbose = TRUE
)
```

Arguments

`x`	Result of `nm_extract_runs()`, or the path of an analysis directory, which is then read with `nm_extract_runs(x)`.
`parents`	Optional links that complete or replace those read in the control streams, as in `nm_model_diagram()`: a named character vector (`c(run2 = "run1")`) or a data frame with columns `run` and `parent`.
`sort`	Order of the rows: `"tree"` (default; each run right after its parent, as in `nm_model_diagram()`), `"ofv"` (best OFV first) or `"run"` (run names, numbers sorted as numbers).
`output_file`	Path of the Word file (.docx). `NULL` (default): nothing is written.
`overwrite`	Logical. Replace an existing `output_file` (default `FALSE`).
`title`	Title written above the table in the Word file.
`verbose`	Print a message when the file is written.

Value

An object of class `nm_runs_table`: a data frame with one row per run and the columns `run`, `description`, `parent`, `ofv`, `dofv_parent` (OFV of the run minus OFV of its parent), `dofv_best` (minus the lowest OFV of the project), `n_par`, `aic`, `bic`, `minimization`, `covariance`, `max_eta_shrink` (%), `pct_rse_gt50` (% of the parameters with an RSE above 50%) and `status` (as in `nm_model_diagram()`). The Word file, when written, has the same rows in landscape format; its path is kept in the attribute `output_file`.

See Also

`nm_extract_runs()`, `nm_model_diagram()`, `nm_compare_runs()`, `nm_extract_run_table()` (table of the estimates of one run)

Examples

```
ex <- system.file("extdata", "example", package = "nmpilot")
tab <- nm_runs_table(ex, parents = c(run2 = "run1"))
tab

if (requireNamespace("flextable", quietly = TRUE) &&
    requireNamespace("officer", quietly = TRUE)) {
  f <- file.path(tempdir(), "runs_summary.docx")
  nm_runs_table(ex, parents = c(run2 = "run1"), output_file = f, overwrite = TRUE)
}
```

---

nm_suggest_covariates *Covariate screening on individual PK estimates before modelling*

Description

Applies the Wasserstein-distance comparison of `nm_wasserstein()` to the individual estimates obtained by `nm_suggest_model()` (NCA clearance, central volume and absorption parameter of the rich intervals, log scale), before any NONMEM run. For each covariate, subjects are split into groups (each level against the first for a categorical covariate; Q4 against Q1, or all pairs of quartiles, for a continuous one) and the distributions of the log estimates are compared with the 1-Wasserstein distance, a permutation test and a Welch t-test. It points to covariates worth testing in the model; it is no substitute for the covariate model (individual NCA estimates carry their own estimation noise, and only subjects with a rich interval are used).

Usage

```
nm_suggest_covariates(
  x,
  covariates,
  data = NULL,
  parameters = c("CL", "Vc"),
  n_perm = 2000L,
```

```
  seed = 2025L,
  max_levels = 5L,
  continuous = c("extremes", "pairwise")
)

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

Arguments

x	An nm_model_suggestion object (individual estimates are needed, i.e. at least 3 rich intervals).
covariates	Character vector of covariate columns.
data	Covariate data (data.frame or CSV path with an ID column). Default: the dataset analysed by nm_suggest_model().
parameters	Individual estimates to analyse among "CL", "Vc", "KA", "DUR", "MTT" (default c("CL", "Vc")).
n_perm	Number of permutations (default 2000).
seed	Random seed of the permutation tests; the user's random state is restored.
max_levels	A numeric covariate with at most this many distinct values is treated as categorical (default 5).
continuous	"extremes" (Q4 vs Q1, default) or "pairwise".
...	Ignored.

Value

A data.frame of class nm_suggest_covariates: parameter, covariate, type, comparison, n_1, n_0, ratio_median (median of group 1 / median of group 0), W, p_W, p_t, p_W_adj (Bonferroni over the comparisons of a covariate and parameter).

References

Simon N, Hulot JS, von Fabeck K. Distribution-based covariate assessment using Wasserstein distance in population pharmacokinetic models. *Front Pharmacol.* 2026;17:1804989. doi:10.3389/fphar.2026.1804989

See Also

nm_suggest_model(), nm_wasserstein()

Examples

```
ex <- system.file("extdata", "example", "example_data.csv", package = "nmpilot")
s <- nm_suggest_model(ex, verbose = FALSE, design_eval = FALSE, max_profiles = 15)
nm_suggest_covariates(s, covariates = c("WT", "GENO"), n_perm = 500)
```

`nm_suggest_design` *Expected precision of a suggested model for a sampling design*

Description

Computes, before any NONMEM run, the expected relative standard errors (RSE) of the parameters of a candidate model from `nm_suggest_model()` and the expected eta-shrinkage, for the sampling design of the dataset or for another design. The population Fisher information matrix is computed with the first-order (FO) approximation, block-diagonal between fixed effects and variance parameters (as in PopED's reduced FIM), with numerical derivatives; predictions use the analytic solution of the linear model, with superposition of all doses (`ADDL`/`II` expanded). The expected shrinkage of each ETA is `1 - sqrt(1 - mean(C_kk / omega_k))`, where `C` is the posterior covariance of the ETAs of each subject in the linearised model.

Usage

```
nm_suggest_design(x, candidate = 1L, times = NULL, n_subjects = NULL)

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

## S3 method for class 'nm_suggest_design'
plot(x, ...)
```

Arguments

`x`	An `nm_model_suggestion` object.
`candidate`	Candidate number (default 1).
`times`	Optional sampling times after the first dose, applied to every subject in place of the observed times (to compare designs).
`n_subjects`	Optional number of subjects: the information is scaled from the subjects of the dataset to this number.
`...`	Ignored.

Details

The values depend on the initial values of the candidate (THETA, IIV, SIGMA), which are approximate: they rank designs and parameters rather than predict the RSE reported by NONMEM. Michaelis-Menten candidates are not supported (no analytic solution).

Value

An object of class `nm_suggest_design`: `parameters` (parameter, type, value, se, rse_pct), `shrinkage` (parameter, shrinkage_pct), `n_subjects`, `n_obs`, `spec`, `times`.

See Also

nm_suggest_model()

Examples

```
ex <- system.file("extdata", "example", "example_data.csv", package = "nmpilot")
s <- nm_suggest_model(ex, verbose = FALSE, design_eval = FALSE, max_profiles = 15)
nm_suggest_design(s)
# a sparse design with 3 samples per subject
nm_suggest_design(s, times = c(1, 4, 12))
```

---

nm_suggest_model *Suggest structural PK models from a NONMEM dataset*

Description

Analyses a NONMEM dataset (rich or sparse data) without running NONMEM and ranks candidate structural models: number of compartments (1 to 3), input model (IV bolus or infusion, first-order, first-order with lag time, zero-order with estimated duration, transit compartments), elimination (linear or Michaelis-Menten) and residual error model (additive, proportional or combined). Each candidate comes with initial values for THETA, IIV and SIGMA in the parameter names of nm_write_ctl(), the parameters likely to be hard to estimate and, for linear models, the precision expected from the sampling design (nm_suggest_design()).

Usage

```
nm_suggest_model(
  data,
  cols = NULL,
  route = c("auto", "EV", "IV"),
  lloq = NULL,
  comment_char = c("#", "@"),
  sep = ",",
  dose_conc_factor = 1,
  min_obs_rich = 6L,
  n_bins = 10L,
  max_cpts = 3L,
  n_candidates = 6L,
  max_profiles = 60L,
  design_eval = TRUE,
  seed = 1L,
  verbose = TRUE
)

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

Arguments

data	Path to a NONMEM CSV file, or a data.frame.
cols	Optional named list mapping roles to column names: id, time, dv, amt, evid, mdv, cmt, rate, addl, ii, ss, blq. Missing roles are guessed from standard names.
route	"auto" (default: from the CMT of doses and observations, then from the shape of the profiles), "EV" or "IV".
lloq	Optional LLOQ (DV scale). Concentrations below it are BLQ. Without it, the lowest quantified concentration is used as LLOQ for the BLQ samples.
comment_char	Characters marking ignored lines (first field).
sep	Field separator of the CSV file.
dose_conc_factor	Unit factor between amount/volume and the DV units, such that CL = factor * Dose / AUC: 1 (default) when DV is in dose units per volume unit (mg and mg/L), 1000 for doses in mg and DV in ng/mL (volumes then in L). It is passed to nm_write_ctl() by from, which writes the scale S = V/factor, so that the initial values match the control stream.
min_obs_rich	Minimum number of quantified concentrations in a dosing interval for it to be analysed individually (default 6).
n_bins	Number of time-after-dose bins for the pooled analysis.
max_cpts	Maximum number of compartments tested (1 to 3).
n_candidates	Number of candidate models returned.
max_profiles	Maximum number of rich intervals analysed individually (a reproducible random subset is drawn above it).
design_eval	If TRUE (default), the expected precision (RSE) and eta-shrinkage of each linear candidate are computed for the sampling design of the dataset with nm_suggest_design().
seed	Seed of the subset of intervals; the user's random state is restored.
verbose	Print progress messages.
x	An nm_model_suggestion object.
...	Ignored.

Details

Method (heuristic; it chooses a starting model and does not replace model comparison in NONMEM):

1. Records are split into dosing intervals; ADDL/II doses are expanded. An interval is "rich" when it holds at least min_obs_rich quantified concentrations, with at least 3 points after the peak for extravascular data.
2. NCA on each rich interval (linear-up log-down AUC, lambda_z by best adjusted R2, CL = Dose/AUCinf after a first dose, Dose/AUCtau at steady state).

3. Number of compartments: sums of 1 to 3 exponentials fitted to the post-peak phase (log scale); BIC weights averaged over intervals. A phase is kept only when it is identifiable (rates at least 1.5-fold apart, each amplitude at least 1% of the total, fast phase decaying within the sampling range).
4. Input (extravascular data): first-order, first-order + lag, zero-order (duration) and gamma (transit) inputs, each convolved with the disposition retained for the interval and fitted to the whole profile; BLQ samples before the first quantified one enter as a censoring penalty (M3-like). BIC weights averaged.
5. Elimination: dose proportionality (power model on AUC, or on dose-normalised concentrations by time bin for sparse data) and curvature of the log-concentration decline (concave = saturable).
6. Residual error (rich data; in sparse data the error model is not assessable and the best structure is proposed with the three error models, with equal support, to be compared in NONMEM by `nm_suggest_run()`): the best structural fit of each interval is refitted by iteratively reweighted least squares under a combined variance `a^2 + b^2 f^2`; on these common, leverage-corrected (PRESS) residuals, the variance models `a^2` (additive), `b^2 f^2` (proportional) and `a^2 + b^2 f^2` (combined) are fitted by maximum likelihood and compared by BIC, the combined model being kept only when its BIC is better by more than 15 (threshold chosen by simulation; 10 until version 0.9.0). Points of leverage above 0.7 and predictions below twice the LLOQ are left out. The estimates give the initial SIGMA values (variances).
7. IIV: variance of the log of the individual estimates of CL, central volume and absorption parameter (median per subject), given as CV%. It includes the estimation noise of each individual value and tends to overestimate the true IIV, which is the safe side for initial values.
8. With fewer than 3 rich intervals, all observations are pooled by time after dose (naive pooled, dose-normalised, binned) and steps 2 to 5 are run on the binned geometric means (indicative only); the residual error model and the IIV cannot be separated in that case and default values are used.

The candidate score is the product of the relative supports of the four components; it sums to 1 over all combinations and is a ranking aid, not a probability.

Alerts (`$alerts`) flag features that none of the tested structures describes: double peaks (enterohepatic recycling, several absorption sites), a clearance that differs between first dose and steady state (auto-induction, inhibition, time-varying CL), an accumulation different from the one predicted by the first-dose terminal slope, and a less-than-proportional exposure (saturable absorption or bioavailability).

Value

An object of class `nm_model_suggestion`, a list with:

`candidates` one row per candidate: `rank`, `cpts`, `route`, `input`, `elimination`, `error`, `score`, `parameters`, `theta_init`, `iiv_cv`, `sigma_init`, `max_rse_pct`, `max_shrinkage_pct`, `hard_to_estimate`, `notes`.

`specs` one list per candidate with the arguments used by `nm_write_ctl(from = , candidate = )`.

`calls` the matching `nm_write_ctl()` calls, as text.

`support` relative support of each level of the four components.

`design` expected RSE and shrinkage of each candidate (`nm_suggest_design` objects, `NULL` for Michaelis-Menten).

`alerts` data.frame `type`, `message`.

`nca` one row per analysed interval.

`individual` individual estimates (CL, central volume, absorption parameter) used for the IIV and by `nm_suggest_covariates()`.

`evidence`, `profiles`, `data_summary`, `flags`, `notes`, `settings` details of the analysis.

See Also

`nm_write_ctl()`, `nm_suggest_design()`, `nm_suggest_run()`, `nm_suggest_covariates()`, `plot.nm_model_suggestion`

Examples

```
ex <- system.file("extdata", "example", "example_data.csv", package = "nmpilot")
s <- nm_suggest_model(ex, verbose = FALSE, max_profiles = 15)
s
s$candidates[, c("rank", "cpts", "input", "elimination", "error", "score")]
nm_write_ctl(from = s, candidate = 1, verbose = FALSE)
```

---

nm_suggest_run *Write, run and rank the candidates of a model suggestion*

Description

Writes the control streams of the chosen candidates of `nm_suggest_model()` with `nm_write_ctl()` (`from = x`), launches them with `nm_run()`, reads their objective function value and ranks them by BIC (OFV + number of estimated parameters x log(number of observations)). The rank by BIC is compared with the rank of the data-driven score (Spearman correlation), which shows whether the suggestion anticipated the NONMEM result. A lag time of a candidate is added after `$PK` as `ALAG1 = THETA(n)` with its initial value, since `nm_write_ctl()` writes none.

Usage

```
nm_suggest_run(
  x,
  analysis_dir,
  nmfe_path = NULL,
  candidates = 1:3,
  prefix = "sugg",
  data = NULL,
  run = TRUE,
  overwrite = FALSE,
  verbose = TRUE,
  n_cores = 1L,
  ...
)
```

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

Arguments

x	An nm_model_suggestion object.
analysis_dir	Analysis directory where the control streams are written and run. It must contain the dataset, or data must point to it.
nmfe_path	Path to the nmfe launcher (see nm_run()).
candidates	Candidate numbers (default: the first 3).
prefix	Prefix of the control stream names (<prefix><n>.ctl).
data	Dataset file name written in $DATA (default: the file analysed by nm_suggest_model()), relative to analysis_dir or absolute.
run	If FALSE, only the control streams are written.
overwrite	Replace existing control streams and adapted datasets, and run the candidates again even when a previous run of the same name has terminated (otherwise nm_run() skips it and its old results are read).
verbose	Print progress messages.
n_cores	Number of candidates run at the same time, passed to nm_run() (default 1).
...	Further arguments passed to nm_write_ctl() (e.g. method).

Details

Each candidate gets a dataset that matches its compartment numbering: doses in the depot (CMT 1) and observations in the central compartment (CMT 2) for first-order and transit absorption (ADVAN2/4/12 or ODE); doses and observations in the central compartment (CMT 1) for IV and zero-order input (ADVAN1/3/11), with RATE = -2 on the doses for a modelled zero-order duration (D1). When the original dataset does not match, a copy <data>_<prefix><n>.csv is written in analysis_dir (records copied verbatim, only CMT and RATE changed); the original file is never modified. Datasets with several observed compartments (parent and metabolite, PK and PD) or doses by several routes are not handled: the function stops.

Value

An object of class nm_suggest_run: a data.frame with one row per candidate (candidate, model, score, rank_score, ctl, data (dataset used), ofv, n_par, n_obs, bic, rank_bic, min_ok), with the Spearman correlation between score and BIC ranks in attribute spearman and the nm_run() log in attribute log.

See Also

nm_suggest_model(), nm_write_ctl(), nm_run()

Examples

```
## Not run:
s <- nm_suggest_model("C:/nm76g64/run/etude/data.csv")
r <- nm_suggest_run(s, analysis_dir = "C:/nm76g64/run/etude",
                    nmfe_path = "C:/nm76g64/run/nmfe76.bat", candidates = 1:4)
r

## End(Not run)
```

`nm_summary` *Diagnostic summary of a NONMEM-like dataset*

Description

Diagnostic summary of a NONMEM-like dataset

Usage

```
nm_summary(data, verbose = TRUE)
```

Arguments

`data`	A data.frame from `nm_read_csv()`.
`verbose`	Print the summary to the console (default `TRUE`).

Value

A one-row data.frame with: `n_subjects`, `n_records`, `n_observations`, `n_blq`, `pct_blq`, `time_min`, `time_max`, `time_unit`, `unique_doses` (distinct positive doses), `n_cmt_obs`.

`nm_vpc` *Visual Predictive Check (VPC or pc-VPC) from NONMEM simulation output*

Description

Reads the `_orig.fit` and `_sim.fit` files produced by `nm_vpc_ctl`, computes observed and simulated percentiles in time bins, and returns a `ggplot2` VPC or pc-VPC plot.

Usage

```
nm_vpc(
  run,
  analysis_dir = NULL,
  col_names,
  type = 1,
  pi_low = 0.05,
  pi_up = 0.95,
  ci_low = 0.025,
  ci_up = 0.975,
  bin_method = 4,
  TTIME_seq_manual = NULL,
  n_bins_quantile = 8,
  bin_rule = "FD",
  plot_title = NULL,
  output_file = NULL,
  show_hist = TRUE,
  ...
)
```

Arguments

run	Path of the estimation run, as given to nm_vpc_ctl: its control stream ("C:/project/run001.ctl"), its run directory or its stem ("C:/project/run001"). The function looks for run001_orig.fit and run001_sim.fit, looked for in run001_orig.reg/ (resp. _sim.reg/), .nm7/, a folder of the same name, then directly in the folder of the run.
analysis_dir	Deprecated (former form run_name + analysis_dir): when given, run is read as the run name in this directory, with a warning.
col_names	Character vector of column names matching the $TABLE output columns in the .fit files (e.g. c("NSIM","ID","TIME","AMT","MDV","DV","PRED","IPRED")).
type	Integer. 1 = standard VPC (default), 2 = pc-VPC (prediction-corrected, Bergstrand 2011).
pi_low	Lower bound of the prediction interval (default: 0.05).
pi_up	Upper bound of the prediction interval (default: 0.95).
ci_low	Lower bound of the confidence interval around simulated percentiles (default: 0.025).
ci_up	Upper bound of the confidence interval around simulated percentiles (default: 0.975).
bin_method	Integer. Binning method: 1 = manual, 2 = equiprobable quantiles, 3 = nominal time points, 4 = automatic (default: 4).
TTIME_seq_manual	Numeric vector of bin breaks. Required when bin_method = 1.
n_bins_quantile	Number of bins for quantile method (bin_method = 2). Default: 8.

bin_rule	Rule for automatic binning (bin_method = 4): "Sturges", "Scott", or "FD" (default).
plot_title	Character. Title of the VPC plot. If NULL (default), a title is generated from the run name and type.
output_file	Character. If provided, the plot is saved to this path (e.g. "vpc_run001.jpeg") through nm_export_plot (.jpg, .png, .pdf or .tiff). (Formerly plot_file, still accepted with a warning.)
show_hist	Logical. If TRUE (default), prints a diagnostic histogram of observed TIME with bin boundaries before the VPC plot.
...	Former argument name, accepted with a warning until version 1.0: run_name (now run).

Details

pc-VPC correction (Bergstrand 2011, Eq. 1, lb = 0):

$$DV_{pc} = DV \times \tilde{PRED}_{bin}/PRED$$

where $\tilde{PRED}_{bin}$ (MEDpred_ref) is the median of all simulated PRED values pooled across replicates within each bin. The same reference is applied to both observed and simulated data.

Required packages: ggplot2 and dplyr must be installed (listed in Imports).

Value

A ggplot2 object, displayed when not assigned.

See Also

nm_vpc_ctl for generating the simulation runs.

Examples

```
## Not run:
col_names <- c("NSIM", "ID", "TIME", "AMT", "MDV", "DV", "PRED", "IPRED")

# Standard VPC
p <- nm_vpc(
  run          = "C:/analyses/my_pk_project/run001.ctl",
  col_names    = col_names,
  type         = 1,
  bin_method   = 4
)

# pc-VPC with manual bins, saved to file
p <- nm_vpc(
  run              = "C:/analyses/my_pk_project/run001.ctl",
  col_names        = col_names,
  type             = 2,
  bin_method       = 1,
  TTIME_seq_manual = c(0, 1, 2, 4, 8, 12, 24),
```

```
  output_file      = "pcvpc_run001.jpeg"
)

## End(Not run)
```

`nm_vpc_ctl` *Generate VPC control streams and launch NONMEM runs*

Description

From an estimation run, generates two NONMEM control streams required for a VPC:

- `<run>_orig.ctl`: re-runs the model at `MAXEVAL=0` with final estimates fixed, producing observed predictions (`_orig.fit`).
- `<run>_sim.ctl`: simulates `n_sim` replicates using `$SIMULATION ONLYSIM SUBPROBLEMS=n_sim`, producing `_sim.fit`.

Both control streams are then launched via `nm_run`.

Usage

```
nm_vpc_ctl(
  run,
  analysis_dir = NULL,
  nmfe_path,
  n_sim = 500,
  seed = 12345,
  skip_terminated = TRUE,
  run_subdir = NULL,
  verbose = TRUE,
  n_cores = 1L,
  ...
)
```

Arguments

`run`	Path of the estimation run: its control stream (`"C:/project/run001.ctl"`), its run directory (`"C:/project/run001.reg"`) or its stem (`"C:/project/run001"`). Its `.ctl`, `.ext` and `.lst` files are looked for in `run001.reg/`, `run001.nm7/`, `run001/`, then directly in the folder of the run (flat layout); this folder is the analysis directory, where the VPC control streams are written.
`analysis_dir`	Deprecated (former form `run_name` + `analysis_dir`): when given, `run` is read as the run name in this directory, with a warning.
`nmfe_path`	Full path to the nmfe executable (e.g. `"C:/nm76g64/run/nmfe76.bat"` on Windows, `"/opt/nm760/run/nmfe76"` on Linux or macOS).
`n_sim`	Number of simulation replicates (default: 500).

seed	Random seed for $SIMULATION (default: 12345).
skip_terminated	Passed to nm_run. If TRUE (default), already completed VPC runs are skipped.
run_subdir	Layout of the two VPC runs, passed to nm_run. By default (NULL) the layout of the estimation run is reused: "" (flat) if its files were found directly in the analysis directory, ".reg" otherwise.
verbose	Passed to nm_run and used locally for messages.
n_cores	Passed to nm_run: 2 runs the _orig and _sim control streams at the same time (default 1).
...	Former argument name, accepted with a warning until version 1.0: run_name (now run).

Details

The $PK block of the source control stream must contain a line of the form NSIM = <value> which will be replaced by NSIM = 1 in _orig.ctl and NSIM = IREP in _sim.ctl.

Final parameter estimates are read from the .ext file of the estimation run (row -1000000000) and written as FIX values in both generated control streams.

The generated .ctl files are written to the analysis directory (not inside a run subdirectory). They can be inspected or edited before running if needed. Wings for NONMEM is not required.

Value

A data.frame run log (from nm_run) with one row per VPC run (_orig and _sim).

See Also

nm_run

Examples

```
## Not run:
log <- nm_vpc_ctl(
  run          = "C:/analyses/my_pk_project/run001.ctl",
  nmfe_path    = "C:/nm76g64/run/nmfe76.bat",
  n_sim        = 500,
  seed         = 12345
)
print(log)
# Then run the VPC:
# fig <- nm_vpc("C:/analyses/my_pk_project/run001.ctl")

## End(Not run)
```

`nm_wasserstein` *Distribution-based covariate assessment with the Wasserstein distance*

Description

Implements the method of Simon, Hulot and von Fabeck (2026). For each run and each covariate, the empirical Bayes estimates of one ETA (read from the `.phi` file) are split into two groups of subjects and compared with the 1-Wasserstein distance `W` and a permutation test, together with a Welch t-test for reference. A covariate that the model accounts for leaves no residual distributional signal in the ETA: comparing a base and a final model shows whether the signal has been removed. A Wasserstein-based R2 (`R2W`) compares the distance of each run's ETA distribution to 0 with the one of a reference run.

Usage

```
nm_wasserstein(
  runs,
  covariates,
  eta = "CL",
  data = NULL,
  ref = 1,
  n_perm = 10000L,
  seed = 2025L,
  max_levels = 5L,
  continuous = c("extremes", "pairwise")
)

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

Arguments

`runs`	Character vector of run directories (`.reg`, `.nm7` or any directory holding the `.phi` file) or run path prefixes. Names, if any, are used as labels; otherwise the run names are used.
`covariates`	Character vector of covariate columns of the data set.
`eta`	The ETA to analyse: a PK parameter name (e.g. `"CL"`, identified from the control stream with `nm_identify_parameters()`), `"ETA(n)"` or an integer index.
`data`	Covariate data: a data.frame or the path of a CSV file with an `ID` column (one row per subject is kept, the first one). If `NULL` (default), the data set named in `$DATA` of the first run's control stream is read.
`ref`	Reference run for `R2W`: index or label (default 1).
`n_perm`	Number of permutations (default 10000).
`seed`	Random seed for the permutation tests (default 2025). Each test uses `seed + k`, so results are reproducible and independent of the order of the other tests. The user's random state is restored.

max_levels	A numeric covariate with at most this many distinct values is treated as categorical (default 5).
continuous	How continuous covariates are split into quartiles (defined once on the subjects of the covariate data): "extremes" (default) compares the highest quartile Q4 with the lowest Q1; "pairwise" compares every pair of quartiles (6 comparisons), as in the pairwise analysis of the original article. Adjusted p-values (p_W_adj, p_t_adj) use the Bonferroni correction over the comparisons of the same covariate within a run.
x	An nm_wasserstein object.
...	Ignored.

Value

An object of class nm_wasserstein, a list with:

tests one row per run x covariate comparison: run, covariate, type, comparison, n_1, n_0, W, p_W, t_stat, p_t, and the Bonferroni-adjusted p_W_adj and p_t_adj (equal to p_W and p_t when a covariate gives a single comparison).

r2w one row per run: run, eta, n, W_to_zero, R2W (1 - W_to_zero / W_to_zero[ref]), omega and shrinkage_pct (eta-shrinkage 1 - sd(EBE)/sqrt(omega), when the .ext is found). A large difference in shrinkage between runs can make the ETA distribution narrower without any real gain.

data the EBE merged with the comparison groups (long format), used by plot.nm_wasserstein().

settings the arguments used.

References

Simon N, Hulot JS, von Fabeck K. Distribution-based covariate assessment using Wasserstein distance in population pharmacokinetic models. *Front Pharmacol.* 2026;17:1804989. doi:10.3389/fphar.2026.1804989

See Also

nm_identify_parameters(), nm_compare()

Examples

```
## Not run:
w <- nm_wasserstein(
  runs       = c(BASE = "C:/nm76g64/run/mtx/base.reg",
                 FINAL = "C:/nm76g64/run/mtx/final.reg"),
  covariates = c("CLCR", "CT01"),
  eta        = "CL"
)
w                       # tests and R2W
plot(w)                 # ECDF of the ETA by group, per run and covariate
plot(w, type = "r2w")  # Wasserstein-based R2

# All pairs of CLCR quartiles, with Bonferroni-adjusted p-values
```

```
wp <- nm_wasserstein(c(BASE = "base.reg", FINAL = "final.reg"),
                     covariates = "CLCR", eta = "CL", continuous = "pairwise")
wp$tests

## End(Not run)
```

---

nm_write_ctl *Write a NONMEM control stream for a population PK model*

Description

Builds a complete control stream ($PROBLEM to $TABLE) for a standard compartmental PK model: 1 to 3 compartments, IV or extravascular administration, first-order, zero-order (modelled rate or duration) or transit-compartment absorption, linear or Michaelis-Menten elimination, analytic (ADVAN/TRANS) or ODE solution. The $INPUT record follows the actual column order of the dataset when input is given.

Usage

```
nm_write_ctl(
  output_file = NULL,
  data = "data.csv",
  input = NULL,
  cols = NULL,
  cpts = 1L,
  route = c("EV", "IV"),
  absorption = c("first-order", "zero-order", "zero-order-d1", "transit"),
  transit_n = 2L,
  elimination = c("linear", "michaelis-menten"),
  dynamics = c("analytic", "ode"),
  ode_subroutine = c("ADVAN13", "ADVAN6"),
  error = c("additive", "proportional", "combined"),
 method = c("FOCEI", "FOCE", "FO", "LAPLACE", "LAPLACEI", "IMP", "ITS", "SAEM", "BAYES"),
  theta_init = NULL,
  theta_lower = NULL,
  theta_upper = NULL,
  theta_fix = NULL,
  iiv = NULL,
  iiv_scale = c("cv", "omega"),
  omega_structure = c("diagonal", "block"),
  omega_rho = 0.1,
  sigma_init = c(prop = 0.09, add = 0.01),
  sigma_fix = NULL,
  sigma_structure = c("auto", "diagonal", "block"),
  covariates = NULL,
  drop = NULL,
  ignore = NULL,
```

```
  npde_esample = 300L,
  npde_seed = 12334L,
  overwrite = FALSE,
  verbose = TRUE,
  from = NULL,
  candidate = 1L,
  conc_factor = 1,
  ...
)
```

Arguments

output_file	Output path. If NULL (default), nothing is written. (Formerly file, still accepted with a warning.)
data	File name written in $DATA.
input	Dataset column names, in file order, or the path to the dataset (its first line is read). If NULL, $INPUT lists the mapped columns only.
cols	Named list mapping NONMEM roles (id, time, dv, amt, evid, mdv, cmt, rate) to column names. Missing roles are guessed from input; without input, standard names are used.
cpts	Number of compartments (1, 2 or 3).
route	"EV" (extravascular) or "IV".
absorption	For route = "EV": "first-order", "zero-order" (rate K0 estimated), "zero-order-d1" (duration D1 estimated) or "transit".
transit_n	Number of transit compartments.
elimination	"linear" or "michaelis-menten".
dynamics	"analytic" or "ode". Analytic is used only when an ADVAN exists for the model; otherwise ODE is used.
ode_subroutine	"ADVAN13" or "ADVAN6", written with TOL=9.
error	"additive", "proportional" or "combined".
method	Estimation method: "FO", "FOCE", "FOCEI", "LAPLACE", "LAPLACEI", "IMP", "ITS", "SAEM" or "BAYES". $COVARIANCE is added for FO, FOCE and Laplace methods. FO is written METHOD=ZERO (NONMEM does not accept METHOD=FO).
theta_init	Named numeric vector of initial values; defaults are used for the other parameters.
theta_lower, theta_upper	Named numeric vectors of bounds. The lower bound defaults to 0 and the upper bound to none; NA removes a bound.
theta_fix	Names of the parameters to fix at their initial value.
iiv	Interindividual variability (exponential ETA). NULL (default): an ETA on every structural parameter (CV 20%, 30% for absorption parameters). A character vector: ETAs on these parameters with default values. A named numeric vector: ETAs on these parameters with these values. character(0): no ETA.

`iiv_scale` "cv" (values of `iiv` are CV%, converted exactly with OMEGA = log(1 + CV^2)) or "omega" (variances).

`omega_structure` "diagonal" or "block" (one `$OMEGA BLOCK`).

`omega_rho` Initial correlation of the `$OMEGA BLOCK`.

`sigma_init` Named numeric vector `c(prop = , add = )` of initial variances.

`sigma_fix` Components to fix: any of "prop", "add".

`sigma_structure` "auto", "diagonal" or "block" (combined error only).

`covariates` Covariate names written in the `cotab` table.

`drop` Column names written `=DROP` in `$INPUT`. For extravascular first-order or transit absorption, a mapped `RATE` column is dropped automatically.

`ignore` Extra `IGNORE` conditions, e.g. "BQL.EQN.1".

`npde_esample, npde_seed` `ESAMPLE` and `SEED` of the `sdtab` table (NPDE).

`overwrite` Logical. Replace an existing `file`.

`verbose` Logical. Print a message when the file is written.

`from` Optional `nm_model_suggestion` object from `nm_suggest_model()`. The structure (`cpts`, `route`, `absorption`, `transit_n`, `elimination`, `error`), the initial values (`theta_init`, `iiv`, `sigma_init`) and the dataset (`data`, `input`) of candidate `candidate` are used for every argument not given explicitly. A lag time of the candidate is not written (no `ALAG1` in this function): a message gives its value. When the structure is changed explicitly, only the initial values whose names exist in the new model are kept.

`candidate` Candidate number in `from` (default 1).

`conc_factor` Unit factor between amount/volume and the DV units: the scale is written `S = V/conc_factor` (e.g. 1000 for doses in mg, volumes in L and concentrations in ng/mL). Default 1 (`S = V`). With `from`, the `dose_conc_factor` of the suggestion is used, so that its initial values (volumes in L, KM in amount/L) match the DV units.

`...` Former argument name, accepted with a warning until version 1.0.

Details

Structural parameter names, used in `theta_init`, `theta_lower`, `theta_upper`, `theta_fix` and `iiv`:

- elimination: `CL` (linear) or `VMAX`, `KM` (Michaelis-Menten);
- central volume: `V` (1 compartment), `V1` (IV or zero-order input, 2-3 compartments), `V2` (first-order absorption, 2-3 compartments);
- peripheral compartments: `Q`, `V2` / `Q2`, `V3`, `Q3` / `V3`, `Q3`, `V4`, `Q4` (names of the matching `ADVAN`);
- absorption: `KA` (first-order), `K0` (zero-order rate, `RATE = -1` in the dataset), `DUR` (zero-order duration D1, `RATE = -2` in the dataset), `MTT` (mean transit time).

Combinations without an analytic solution (Michaelis-Menten elimination, transit compartments) are written with $DES and ode_subroutine, with TOL=9 on the $SUBROUTINES line (NM-TRAN requires TOL for these subroutines). Residual error uses ERR(1) (proportional) and ERR(2) (additive); the component absent from the chosen model is fixed to 0.

Value

An object of class nm_ctl (a character vector of control stream lines, printed as text), invisibly when file is given.

See Also

nm_ctl_check(), nm_ctl_mu(), nm_ctl_iov(), nm_ctl_m3(), nm_suggest_model()

Examples

```
# One-compartment oral model with combined error
nm_write_ctl(error = "combined")

# Two-compartment IV model, no IIV on Q, fixed V2, written to a file
f <- tempfile(fileext = ".ctl")
nm_write_ctl(f, route = "IV", cpts = 2,
             theta_init = c(CL = 3, V1 = 20), theta_fix = "V2",
             iiv = c(CL = 30, V1 = 25, V2 = 20), verbose = FALSE)
nm_ctl_check(f)

# From a data-driven suggestion
ex <- system.file("extdata", "example", "example_data.csv", package = "nmpilot")
s <- nm_suggest_model(ex, verbose = FALSE, design_eval = FALSE, max_profiles = 15)
nm_write_ctl(from = s, candidate = 1, verbose = FALSE)
```

---

plot.nm_compare *Diagnostic plots for* nm_compare *objects*

Description

Diagnostic plots for nm_compare objects

Usage

```
## S3 method for class 'nm_compare'
plot(x, type = "all", n_band = 200L, ...)
```

Arguments

x	An nm_compare object.

type	Character. One or several of "cwres_scatter", "npde_scatter", "iofv_caterpillar", "delta_density", "qq_npde", "gain_map" (Delta|CWRES| versus TIME and PRED, loess with a 95 % subject-bootstrap band), "overlay_gof" (diagnostics of both runs superimposed), "bootstrap" (subject bootstrap of dOFV), "iofv_covariate" (delta iOFV versus the covariates), "parameters" (estimates with 95 % Wald intervals), "eta_scatter" (EBEs of the two runs), or "all" (default). Formerly which, still accepted with a warning.
n_band	Number of subject resamples for the bands of "gain_map" and "overlay_gof".
...	Unused (except the former name which).

Value

A ggplot object (single plot) or a list of ggplot objects (multiple plots). Requires the ggplot2 package.

---

plot.nm_jackknife *Plots for* nm_jackknife *objects*

Description

Plots for nm_jackknife objects

Usage

```
## S3 method for class 'nm_jackknife'
plot(x, type = c("bar", "rank", "scatter"), ...)
```

Arguments

x	An nm_jackknife object.
type	"bar" (default): $\|\Delta OFV\|$ per subject; "rank": ranked $\|\Delta OFV\|$; "scatter": $\|\Delta OFV\|$ against the relative THETA shift norm.
...	Ignored.

Value

A ggplot object.

plot.nm_model_suggestion *Plot a model suggestion*

Description

Plot a model suggestion

Usage

```
## S3 method for class 'nm_model_suggestion'
plot(
  x,
  type = c("profiles", "support", "pooled", "design"),
  candidate = 1L,
  n_max = 12L,
  ...
)
```

Arguments

x	An nm_model_suggestion object.
type	"profiles" (observed concentrations of the analysed intervals with the best fit of each, log scale), "support" (relative support of each component), "pooled" (binned pooled data with their fit) or "design" (expected RSE and eta-shrinkage of a candidate).
candidate	Candidate shown by type = "design".
n_max	Maximum number of intervals shown by type = "profiles".
...	Ignored.

Value

A ggplot object.

plot.nm_wasserstein *Plots for* nm_wasserstein *objects*

Description

Plots for nm_wasserstein objects

Usage

```
## S3 method for class 'nm_wasserstein'
plot(x, type = c("ecdf", "r2w"), covariate = NULL, ...)
```

Arguments

x	An nm_wasserstein object.
type	"ecdf" (default): empirical cumulative distribution of the ETA in the two groups, one panel per run and comparison, annotated with W, its p-value, t and its p-value. "r2w": bar chart of the Wasserstein-based R2 per run.
covariate	Optional covariate(s) to show (ECDF plot only).
...	Ignored.

Value

A ggplot object.

Index

nm_add_rate, 3
nm_bootstrap, 4
nm_bootstrap(), *13*, *32*
nm_check_dataset, 6
nm_compare, 7
nm_compare(), *39*, *58*
nm_compare_runs, 10
nm_compare_runs(), *33*, *44*
nm_convert_ctl, 11
nm_convert_ctl(), *29*
nm_cores, 12
nm_cores(), *6*, *31*, *38*, *41*
nm_ctl_check, 13
nm_ctl_check(), *62*
nm_ctl_iov, 14
nm_ctl_iov(), *62*
nm_ctl_m3, 15
nm_ctl_m3(), *62*
nm_ctl_mu, 17
nm_ctl_mu(), *62*
nm_data_previsual, 18
nm_derivatives_ctl, 20
nm_derivatives_ctl(), *34, 35*
nm_export_data, 21
nm_export_plot, 22, *54*
nm_export_plot(), *33*
nm_extract_run_table, 23
nm_extract_run_table(), *44*
nm_extract_runs, 24
nm_extract_runs(), *9*, *23*, *30*, *33*, *39*, *42–44*
nm_filter_runs, 25
nm_gof, 26
nm_guide, 27
nm_identify_directory, 28
nm_identify_directory(), *30*
nm_identify_parameters, 29
nm_identify_parameters(), *24*, *28*, *57, 58*
nm_jackknife, 30
nm_jackknife(), *13*
nm_model_diagram, 32
nm_model_diagram(), *43, 44*
nm_optimize_base, 33
nm_optimize_base(), *20, 21*
nm_plot_dv, 36
nm_profile, 37
nm_profile(), *13*, *32*
nm_read_csv, 39
nm_read_csv(), *6*, *21*, *36*, *52*
nm_run, *12*, 40, *55, 56*
nm_run(), *13*, *21*, *34*, *39*, *50, 51*
nm_runs_table, 43
nm_suggest_covariates, 44
nm_suggest_covariates(), *50*
nm_suggest_design, 46
nm_suggest_design(), *47, 48*, *50*
nm_suggest_model, 47
nm_suggest_model(), *44–47*, *50, 51*, *61, 62*
nm_suggest_run, 50
nm_suggest_run(), *49, 50*
nm_summary, 52
nm_vpc, 52
nm_vpc_ctl, *52–54*, 55
nm_wasserstein, 57
nm_wasserstein(), *35*, *44, 45*
nm_write_ctl, 59
nm_write_ctl(), *47–51*

plot(), *38*
plot.nm_compare, 62
plot.nm_compare(), *9*
plot.nm_jackknife, 63
plot.nm_model_suggestion, 64
plot.nm_model_suggestion(), *50*
plot.nm_optimize_base
*(*nm_optimize_base*)*, 33
plot.nm_profile *(*nm_profile*)*, 37
plot.nm_suggest_design
*(*nm_suggest_design*)*, 46
plot.nm_wasserstein, 64

plot.nm_wasserstein(), *58*
print(), *38*
print.nm_compare *(*nm_compare*)*, 7
print.nm_compare(), *9*
print.nm_jackknife *(*nm_jackknife*)*, 30
print.nm_model_suggestion
*(*nm_suggest_model*)*, 47
print.nm_optimize_base
*(*nm_optimize_base*)*, 33
print.nm_profile *(*nm_profile*)*, 37
print.nm_suggest_covariates
*(*nm_suggest_covariates*)*, 44
print.nm_suggest_design
*(*nm_suggest_design*)*, 46
print.nm_suggest_run *(*nm_suggest_run*)*,
50
print.nm_wasserstein *(*nm_wasserstein*)*,
57

summary.nm_compare *(*nm_compare*)*, 7