Package ‘vachette’

May 8, 2026

Title A Method for Visualization of Pharmacometric Models

Version 0.40.1

Description A method to visualize pharmacometric analyses which are impacted by covariate effects. Variability-aligned covariate harmonized-effects and time-transformation equivalent ('vachette') facilitates intuitive overlays of data and model predictions, allowing for comprehensive comparison without dilution effects. 'vachette' improves upon previous methods Lommerse et al. (2021) <doi:10.1002/psp4.12679>, enabling its application to all pharmacometric models and enhancing Visual Predictive Checks (VPC) by integrating data into cohesive plots that can highlight model misspecification.

URL `https://github.com/certara/vachette`

BugReports `https://github.com/certara/vachette/issues`

Depends R (>= 4.0)

License LGPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, mrgsolve, tidyvpc, testthat (>= 3.0.0), learnr

Imports ggplot2, tidyr, dplyr, magrittr, prospectr, Hmisc, purrr, minpack.lm, progress, rlang, utils, photobiology

Config/testthat/edition 3

NeedsCompilation no

Author Jos Lommerse [aut],
Anna Largajolli [aut],
James Craig [aut, cre],
Nele Mueller-Plock [ctb],
Jeff Sachs [ctb],
Amy Cheung [ctb],
Certara USA, Inc. [cph, fnd]

Maintainer James Craig <`james.craig@certara.com`>

Repository CRAN
Date/Publication 2024-09-19 05:50:02 UTC

Contents

`apply_transformations` *Apply vachette transformations*

Description

Apply vachette transformations

Usage

```
apply_transformations(vachette_data, ...)

## S3 method for class 'vachette_data'
apply_transformations(
  vachette_data,
  tol.end = 0.001,
  tol.noise = 1e-08,
  step.x.factor = 1.5,
  ngrid.fit = 100,
  window = 17,
  log_file = NULL,
  ...
)
```

Arguments

`vachette_data`	object of class `vachette_data`
`...`	Additional arguments
`tol.end`	numeric; Relative tolerance to determine last x open end reference
`tol.noise`	numeric; Relative tolerance for landmark determination typical curves
`step.x.factor`	numeric; x-axis extension factor to search for last x, i.e., to determine where close enough to asymptote
`ngrid.fit`	numeric; number of grid points in last query segment for matching last reference segment
`window`	integer; size (gridpoints) of Savitzky Golay smoothing window for landmark position determination
`log_file`	character; File path to direct console output e.g., `"log.txt"`

Value

`vachette_data` object containing a list of vachette-transformed `data.frames`

Examples

```
obs <- read.csv(system.file(package = "vachette", "examples", "iv-obs.csv"))
typ <- read.csv(system.file(package = "vachette", "examples", "iv-typ-minmax.csv"))

vd <- vachette_data(
  obs.data = obs,
  typ.data = typ,
  covariates = c(WT = 70),
  mappings = c(OBS = "DV", x = "time"),
  model.name  = "IV"
 )

 vd <- apply_transformations(vd)
```

`p.add.distances` *Plot Distances Before and After* `vachette` *Transformation Assuming Additive Error Model*

Description

This function generates a ggplot2 visualization of the distances before and after a `vachette` transformation between observations and the relative typical curves. It is useful for assessing the impact of additive error model assumption on the distances.

Usage

```
p.add.distances(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function starts by setting the x-axis range based on the minimum and maximum values of the original and scaled distances. It then plots the distances before and after transformation, with a reference line indicating where the original and transformed distances are equal.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used.

Value

A `ggplot2` object representing the plot of distances. The plot compares the original distances to the transformed distances, with segments color-coded.

`p.obs.cov` *Plot Observations and Typical Curves Faceted by Covariate*

Description

This function generates a ggplot2 visualization of observations and typical curves within a pharmacometric model, faceted by unique covariate values (ucov). The plot compares the observed data against the typical reference and query curves, with each facet representing a different covariate value.

Usage

```
p.obs.cov(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function begins by ensuring that the input `vachette_data` is of the correct class. It then constructs a plot that displays the reference and query data, differentiating them using distinct colors, and faceting the plot by unique covariate values. Observations are represented by points, while the typical curves are depicted as lines.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling. If the x-axis is logarithmic, the axis label is adjusted accordingly.

Value

A `ggplot2` object representing the observations and typical curves, faceted by covariate. The plot displays both the reference and query curves and their corresponding observations, color-coded for easy comparison.

`p.obs.excluded` *Plot Excluded Observations with Reasons for Exclusion*

Description

This function generates a ggplot2 visualization of observations that were excluded from the transformation process within a pharmacometric model. The plot includes both the included and excluded observations, with the excluded ones color-coded by reason for exclusion.

Usage

```
p.obs.excluded(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function plots the reference and query curves along with the observations. Observations that were excluded from the transformation process are highlighted and labeled with the reason for exclusion, while included observations are shown in light grey.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values from the original observations. If the x-axis is logarithmic, the axis label is adjusted accordingly.

Value

A `ggplot2` object representing the excluded observations and their corresponding reasons for exclusion. The plot also displays the typical reference and query curves for comparison.

`p.obs.ref.query` *Plot Observations and Typical Curves for Query and Reference Data*

Description

This function generates a ggplot2 visualization of the observations and typical curves for both query and reference data within a pharmacometric model. It is designed to compare the observed data against the typical reference and query curves.

Usage

```
p.obs.ref.query(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function plots the query and reference data, differentiating them using distinct colors. Observations are represented by points, while the typical curves are depicted as lines.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling. If the x-axis is logarithmic, the axis label is adjusted accordingly.

Value

A `ggplot2` object representing the observations and typical curves. The plot displays the query and reference curves and their corresponding observations, color-coded for easy comparison.

`p.prop.distances` *Plot Distances Before and After* `vachette` *Transformation Assuming Proportional Error Model*

Description

This function creates a ggplot2 visualization of the distances before and after the `vachette` transformation between observations and the relative typical curves. It is useful for assessing the impact of proportional error model assumption on the distances.

Usage

```
p.prop.distances(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The plot includes a reference line with a slope of 1, indicating where the original and transformed distances are equal. The title of the plot is dynamically generated based on the model name provided in the `vachette_data` object.

The subtitle of the plot indicates whether an additive or proportional error transformation was applied. The caption provides the reference covariate(s) used in the model.

Value

A `ggplot2` object representing the plot of distances. The plot compares the log of the original distances to the log of the distances after transformation, color-coded by segment.

`p.scaled.observation.curves`

Plot `vachette` *Transformed Observation Curves*

Description

This function generates a ggplot2 visualization of transformed observation curves, including their transformations and overlays with reference curves. The plot is useful for comparing the original and the `vachette` transformed observation data against typical reference curves.

Usage

```
p.scaled.observation.curves(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function constructs a plot that overlays observation curves with the corresponding reference curves. Query observations and their transformed curves are highlighted, and if applicable, an extrapolated segment of the reference curve is shown as a dashed line.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling.

Value

A `ggplot2` object representing the `vachette` transformed observation curves. The plot includes different layers for query observations, their transformed counterparts, and the reference curves, with appropriate color coding and line styles.

`p.scaled.observation.curves.by.id`

Plot `vachette` *Transformed Observation Curves by Individual*

Description

This function generates a ggplot2 visualization of transformed observation curves facetted by individual, including their transformations and overlays with reference curves. The plot is useful for comparing the original and the `vachette` transformed observation data against typical reference curves.

Usage

```
p.scaled.observation.curves.by.id(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function constructs a plot that overlays individual observation curves with the corresponding reference curves. Query observations and their transformed curves are highlighted, and if applicable, an extrapolated segment of the reference curve is shown as a dashed line.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling.

Value

A `ggplot2` object representing the `vachette` transformed observation curves. The plot includes different layers for query observations, their transformed counterparts, and the reference curves, with appropriate color coding and line styles.

`p.scaled.typical.curves`

Plot `vachette` *Transformed Typical Curves*

Description

This function generates a ggplot2 visualization of the `vachette` transformed typical curves for a given pharmacometric model. It distinguishes between reference curves and query curves, showing how the curves transform.

Usage

```
p.scaled.typical.curves(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function creates a plot that distinguishes between reference curves (in red) and query curves (in blue). The dashed lines represent the curves after the vachette transformation.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling.

Value

A `ggplot2` object representing the `vachette` transformed typical curves. The plot shows both the reference and query curves, and the curves after the vachette transformation.

`p.scaled.typical.curves.landmarks`

Plot `vachette` *Transformed Typical Curves with Landmarks*

Description

This function generates a ggplot2 visualization of the `vachette` transformed typical curves and their corresponding landmarks for a given pharmacometric model. The plot highlights the segments of the reference and query curves, with a special emphasis on the landmark positions.

Usage

```
p.scaled.typical.curves.landmarks(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function plots the typical curves along with their segments and marking the landmarks. If the x-axis is logarithmic, it adjusts the axis label accordingly.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling.

Value

A `ggplot2` object representing the `vachette` transformed typical curves with their landmarks. The plot displays different segments of the curves in different color, and landmarks marked with black crosses.

`p.scaled.typical.full.curves.landmarks`
Plot Full `vachette` *Transformed Typical Curves with Landmarks*

Description

This function generates a ggplot2 visualization of the full `vachette` transformed typical curves and their corresponding landmarks for a given pharmacometric model. The plot highlights the segments of the reference and query curves, with a special emphasis on specific landmark positions.

Usage

```
p.scaled.typical.full.curves.landmarks(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function plots the full typical curves along with their segments, applying specific styling to the reference curve and marking the landmarks. If the x-axis is logarithmic, it adjusts the axis label accordingly.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling.

Value

A `ggplot2` object representing the full `vachette` transformed typical curves with their landmarks. The plot displays different segments of the curves in different colors, and landmarks marked with black crosses.

`p.scaling.factor` *Plot Scaling Factors for Typical Curves*

Description

This function generates a ggplot2 visualization of the scaling factors applied to the typical curves within a pharmacometric model. The plot differentiates between reference and query curves and illustrates how scaling factors vary across segments.

Usage

```
p.scaling.factor(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function creates a plot that displays the x-scaling factors for the curves, with different segments color-coded. The plot is faceted by curve type (reference or query).

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the minimum and maximum x values, before and after scaling.

Value

A `ggplot2` object representing the scaling factors for the curves. The plot includes facets for each curve type (reference or query), and it shows the scaling factors as a function of the x-axis values.

`p.vachette` *Main Vachette Plot*

Description

This function generates the main Vachette plot, which visualizes all transformed data points within a pharmacometric model. The plot makes a clear distinction between reference data points (those that were not moved, shown in red) and transformed query data points (those that were moved, shown in purple).

Usage

```
p.vachette(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function plots all the data points, highlighting reference points in red and transformed query points in purple. The plot also overlays reference curves, with extrapolated segments displayed as dashed lines if available. The subtitle provides information about the error model used (additive or proportional) and the number of excluded observations.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the scaled x values. If the x-axis is logarithmic, the axis label is adjusted accordingly.

Value

A `ggplot2` object representing, which includes all observations, with reference data points shown in red, and transformed query data points shown in purple. The plot also includes reference curves and, if applicable, extrapolated reference curves as dashed lines.

`p.vachette.arrow` *Plot Observations and* `vachette` *Transformed Observations with Connecting Arrows*

Description

This function generates a ggplot2 visualization of observations and their transformations within a pharmacometric model. The plot superimposes the observed query and reference data from original to `vachette` transformed values, showing direction of transformation. Reference and query curves are additionally overlaid for comparison.

Usage

```
p.vachette.arrow(vachette_data)
```

Arguments

`vachette_data`	An object of class `vachette_data`, which contains the necessary data for plotting.

Details

The function plots the original and `vachette` transformed observations, with arrows showing the transformation path. Reference and query curves are also plotted, with extrapolated reference curves displayed as dashed lines if available. The subtitle provides information about the error model used (additive or proportional) and the number of excluded observations.

The plot's title includes the model name, and the caption provides details about the reference covariate(s) used. The x-axis range is dynamically set based on the scaled x values. If the x-axis is logarithmic, the axis label is adjusted accordingly.

Value

A `ggplot2` object representing the observations, their transformations, and the corresponding reference and query curves. The plot also indicates if any extrapolated reference curves are present and how many observations were excluded from the transformation.

`print.vachette_data` *Print* `vachette_data`

Description

Print generic used to return information about `vachette_data` object

Usage

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

Arguments

`x`	`vachette_data` object.
`...`	additional args.

Value

`x` invisibly.

`summary.vachette_data` *Summarise* `vachette_data`

Description

Summary generic used to return information about `vachette_data` object

Usage

```
## S3 method for class 'vachette_data'
summary(object, verbose = FALSE, trim = TRUE, log_file = NULL, ...)
```

Arguments

`object`	`vachette_data` object.
`verbose`	logical; Set to `TRUE` to view additional warnings output
`trim`	logical; If `TRUE`, only the first 20 rows are printed for `data.frame` output. Set to `FALSE` to display all rows
`log_file`	character; File path to direct console output e.g., `"log.txt"`
`...`	additional args.

Value

`object` invisibly.

`vachette_data` *Initialize vachette object with required data*

Description

Initialize vachette object with required data

Usage

```
vachette_data(
  obs.data,
  typ.data,
  sim.data = NULL,
  covariates,
  ref.dosenr,
  log.x = FALSE,
  iiv.correction = FALSE,
  error.model = c("proportional", "additive"),
  model.name = NULL,
  mappings = NULL
)
```

Arguments

`obs.data`	data.frame; Observed data
`typ.data`	data.frame; Typical (population) curves
`sim.data`	data.frame; Simulated (VPC) data
`covariates`	named character vector; Covariate names with reference values in vachette transformation
`ref.dosenr`	integer; Dose number to use as the reference dose, corresponding to value in "dosenr" column in input data
`log.x`	logical; Apply log(x) conversion. Default `FALSE`
`iiv.correction`	logical; Apply inter-individual variability correction. Default `FALSE`
`error.model`	character; Applied error model, `"proportional"` or `"additive"`. Default `"proportional"`
`model.name`	character; Optional model name for plot output
`mappings`	named character vector; Optional mappings to be included if column names in input `data.frame` differ from required column names. See Required Columns section:

Details

If "dosenr" column is missing it will be automatically calculated using the priority of available columns:

- "EVID": If available in data, "dosenr" will be calculated using `cumsum(EVID==1)`
- "ADDL"/"II": If "ADDL" and "II" are available in data, "dosenr" will be calculated given additional dose number and interval
- "AMT": If only "AMT" column exists in data, "dosenr" will be calculated using `cumsum(AMT!=0)`

Value

`vachette_data`

Required columns `obs.data`

- `"ID"` - Subject ID
- `"x"` - Typically time
- `"PRED"` - Population prediction, required if `iiv.correction` = `TRUE`
- `"IPRED"` - Individual prediction, required if `iiv.correction` = `TRUE`
- `"OBS"` - DV
- `"dosenr"` - Dose number; unique dose number for ID/time point

Required columns `typ.data`

- `"ID"` - Subject ID
- `"x"` - Typically time
- `"PRED"` - Population prediction
- `"dosenr"` - Dose number; unique dose number for ID/time point

Required columns `sim.data`

- `"ID"` - Subject ID
- `"x"` - Typically time
- `"PRED"` - Population prediction, required if `iiv.correction` = `TRUE`
- `"IPRED"` - Individual prediction, required if `iiv.correction` = `TRUE`
- `"REP"` - Replicate number

Examples

```
obs <- read.csv(system.file(package = "vachette", "examples", "iv-obs.csv"))
typ <- read.csv(system.file(package = "vachette", "examples", "iv-typ-minmax.csv"))

vd <- vachette_data(
  obs.data = obs,
  typ.data = typ,
  covariates = c(WT = 70),
  mappings = c(OBS = "DV", x = "time"),
  model.name  = "IV"
  )
```

Index