Package ‘Certara.DarwinReporter’

March 7, 2025

Title Data Visualization Utilities for 'pyDarwin' Machine Learning Pharmacometric Model Development

Version 2.0.1

Description Utilize the 'shiny' interface for visualizing results from a 'pyDarwin' (<https://certara.github.io/pyDarwin/>)
machine learning pharmacometric model search. It generates Goodness-of-Fit plots and summary tables for selected models,
allowing users to customize diagnostic outputs within the interface. The underlying R code for generating plots and
tables can be extracted for use outside the interactive session. Model diagnostics can also be incorporated into an
R Markdown document and rendered in various output formats.

URL https://certara.github.io/R-DarwinReporter/

Depends R (>= 4.1.0)

License LGPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, data.table, readr, testthat (>= 3.0.0)

Imports DT, colourpicker, shinyAce, shinymeta, utils, ggplot2, xpose,
Certara.Xpose.NLME, dplyr, jsonlite, tidyr, flextable,
shinyjqui, grDevices, plotly, scales, shiny (>= 1.7.0),
shinyjs, shinyWidgets, bslib (>= 0.7.0), shinyTree (>= 0.3.1),
sortable

Config/testthat/edition 3

NeedsCompilation no

Author James Craig [aut, cre],
Michael Tomashevskiy [aut],
Mike Talley [aut],
Certara USA, Inc [cph, fnd]

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

Repository CRAN

Date/Publication 2025-03-07 16:40:13 UTC

Contents

`darwinReportUI` *Generate and Report Model Diagnostics from NLME or NONMEM runs*

Description

Shiny application to generate, customize, and report diagnostic plots and tables from NLME or NONMEM output files. Create an Rmarkdown file of tagged model diagnostics and render into submission ready report.

Usage

```
darwinReportUI(darwin_data, tagged = NULL, settings = NULL, ...)
```

Arguments

`darwin_data`	Object of class `darwin_data`. Note, key_models xpose_data must be available.
`tagged`	List of tagged objects returned from previous `tagged <- darwinReportUI()` session.
`settings`	List of settings (e.g., settings.Rds) returned from previous Shiny session.
`...`	Additional arguments for Pirana integration.

Value

If `interactive()`, returns a list of tagged diagnostics from the Shiny application, otherwise returns `TRUE`.

Examples

```
if (interactive()) {
ddb <- darwin_data("./darwin_search_09") |>
   import_key_models("./darwin_search_09/key_models")

darwinReportUI(ddb)
}
```

`darwin_data` *Initialize darwin data structure.*

Description

Initialize darwin data structure.

Usage

```
darwin_data(
  project_dir,
  working_dir = NULL,
  output_dir = NULL,
  key_models_dir = NULL,
  ...
)
```

Arguments

`project_dir`	Directory containing input files for pyDarwin (e.g., options.json).
`working_dir`	Directory containing misc results folders generated from a pyDarwin search. This is the default location of the key_models, output, and temp folders.
`output_dir`	Directory containing output files such as "results.csv" and final control files. Default location is inside `working_dir/output`.
`key_models_dir`	Directory of the key_models folder. Default location is inside `working_dir/key_models`. Note, key models are not imported if argument is `NULL`, explicitly specify `key_models_dir` to import files for `darwinReportUI`.
`...`	Additional args.

Details

If `working_dir` and `output_dir` are sub directories of `project_dir`, these arguments can be ignored. The `key_models_dir` is not required to initialize the `darwin_data` object. If specified, however, key models data will be imported which may take time given the number of key models and size of output tables. See `import_key_models`.

Value

Object of class `darwin_data`.

`fitness_penalties_vs_iteration` *Plot minimum fitness by iteration with penalty composition.*

Description

Plot minimum fitness by iteration with penalty composition.

Usage

```
fitness_penalties_vs_iteration(
  darwin_data,
  group_penalties = TRUE,
  scale_ofv = TRUE,
  ...
)
```

Arguments

`darwin_data`	Object of class `darwin_data`.
`group_penalties`	Logical; defaults to `TRUE`.
`scale_ofv`	Set to `TRUE` to rescale OFV axis limit. Used to better observe penalty effects.
`...`	Additional arguments.

Value

Object of class `ggplot`.

`fitness_vs_iteration` *Plot best fitness by iteration.*

Description

Plot best fitness by iteration.

Usage

```
fitness_vs_iteration(darwin_data, ...)
```

Arguments

`darwin_data`	Object of class `darwin_data`.
`...`	Additional arguments.

Value

Object of class `ggplot`.

`get_eps_shk` *Get eps shrinkage values* `xpose_data` *object*

Description

This function returns eps shrinkage values from `xpose_data` object as a `data.frame`.

Usage

```
get_eps_shk(xpdb)
```

Arguments

`xpdb`	Object of class `xpose_data`.

Value

Returns an object of class `data.frame`.

`get_eta_shk` *Get eta shrinkage values from* `xpose_data` *object*

Description

This function returns eta shrinkage values from `xpose_data` object as a `data.frame`.

Usage

```
get_eta_shk(xpdb)
```

Arguments

`xpdb`	Object of class `xpose_data`.

Value

Returns an object of class `data.frame`.

`import_key_models` *Imports files from key model output folders*

Description

Use to create xpose data object from files in NLME or NONMEM key model output folders.

Usage

```
import_key_models(darwin_data, dir, ...)
```

Arguments

`darwin_data`	Object of class `darwin_data`.
`dir`	File path to key models directory.
`...`	Additional args.

Value

Object of class `darwin_data`.

Examples

```
if (interactive()) {
ddb <- darwin_data(project_dir = "./darwin_search_09") |>
   import_key_models(dir = "./darwin_search_09/key_models")
}
```

`summarise_fitness_by_iteration` *Summarise fitness by iteration*

Description

Summarise minimum, cumulative minimum, and mean fitness values by pyDarwin search iteration/generation.

Usage

```
summarise_fitness_by_iteration(darwin_data)
```

Arguments

`darwin_data`	Object of class `darwin_data`.

Value

`data.frame` with columns `iteration`, `min_fitness`, `mean_fitness`, and `min_cum_fitness`

`summarise_fitness_penalties_by_iteration`

Summarize minimum fitness and penalty values by iteration

Description

Summarise minimum fitness, ofv, and penalty values used in calculation of overall fitness values by pyDarwin search iteration/generation.

Usage

```
summarise_fitness_penalties_by_iteration(darwin_data, group_penalties = FALSE)
```

Arguments

`darwin_data`	Object of class `darwin_data`.
`group_penalties`	Logical. Set to `TRUE` to group penalties.

Value

`data.frame` of columns `"iteration"`, `"fitness"`, `"ofv"` and corresponding penalty columns.

`summarise_overall_by_key_models`

Summarise overall table by key models

Description

Generate a summary `data.frame` by key models, which includes columns such as condition number, number of parameters, -2LL, AIC, BIC, fitness, and penalty values.

Usage

```
summarise_overall_by_key_models(darwin_data)
```

Arguments

`darwin_data`	Object of class `darwin_data`.

Value

`data.frame`

`theme_certara` *A ggplot2 theme for Certara.*

Description

A ggplot2 theme for Certara.

Usage

```
theme_certara(
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22,
  grid = c("none", "horizontal", "both"),
  ...
)
```

Arguments

`base_size`	base font size, given in pts.
`base_family`	base font family
`base_line_size`	base size for line elements
`base_rect_size`	base size for rect elements
`grid`	Which grid lines should appear? Horizontal only, both horizontal and vertical, or none (default). `continuous_scale()`.
`...`	Additional args

Details

There are 3 variants of the theme: no grid `theme_certara()`, full grid `theme_certara(grid = "both")`, and horizontal grid lines only `theme_certara(grid = "horizontal")`.

Value

An object of class `theme()`.

Index