Package: SynergyLMM (via r-universe)

October 29, 2026

Title Statistical Framework for in Vivo Drug Combination Studies

Version 1.1.4.9000

Description A framework for evaluating drug combination effects in preclinical in vivo studies. 'SynergyLMM' provides functions to analyze longitudinal tumor growth experiments using mixed-effects models, perform time-resolved analyses of synergy and antagonism, evaluate model diagnostics and performance, and assess both post-hoc and a priori statistical power. The calculation of drug combination synergy follows the statistical framework provided by Demidenko and Miller (2019, <doi:10.1371/journal.pone.0224137>). The implementation and analysis of linear mixed-effect models is based on the methods described by Pinheiro and Bates (2000, <doi:10.1007/b98882>), and Gałecki and Burzykowski (2013, <doi:10.1007/978-1-4614-3900-4>).

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports magrittr (>= 2.0.3), rlang, dplyr (>= 1.1.4), ggplot2 (>= 3.5.1), cowplot (>= 1.1.3), nlme (>= 3.1-165), nlmeU, fBasics, car, MASS, performance, lattice, marginaleffects, clubSandwich

Suggests lme4, knitr, rmarkdown, testthat (>= 3.1.5)

Config/testthat/edition 3

Depends R (>= 4.0)

LazyData true

VignetteBuilder knitr

URL https://github.com/RafRomB/SynergyLMM

BugReports https://github.com/RafRomB/SynergyLMM/issues

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libicu-dev

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

Date/Publication 2026-08-30 16:24:05 UTC

RemoteUrl https://github.com/rafromb/synergylmm

RemoteRef HEAD

RemoteSha f1a9a16d02daae7f84eeaa6d351c51b3e6286497

Contents

APrioriPwr . . . 2
CookDistance . . . 5
getRTV . . . 7
grwth_data . . . 8
lmmModel . . . 9
lmmModel_estimates . . . 13
lmmSynergy . . . 15
logLikSubjectDisplacements . . . 21
ObsvsPred . . . 23
plot_lmmModel . . . 24
plot_lmmSynergy . . . 26
plot_ObsvsPred . . . 27
plot_ranefDiagnostics . . . 28
plot_residDiagnostics . . . 29
plot_SynergyLMM . . . 30
PostHocPwr . . . 31
Pwr . . . 33
Pwr.lme . . . 34
PwrSampleSize . . . 35
PwrTime . . . 38
ranefDiagnostics . . . 40
residDiagnostics . . . 42
simulateTumorGrowth . . . 45

Index **47**

`APrioriPwr` A Priori *Synergy Power Analysis Based on Variability and Drug Effect*

Description

A priori power calculation for a hypothetical two-drugs combination study of synergy using linear-mixed models with varying drug combination effect and/or experimental variability.

Usage

```
APrioriPwr(
  npg = 5,
  time = c(0, 3, 5, 10),
  grwrControl = 0.08,
  grwrA = 0.07,
  grwrB = 0.06,
  grwrComb = 0.03,
  sd_ranef = 0.01,
  sgma = 0.1,
  sd_eval = NULL,
  sgma_eval = NULL,
  grwrComb_eval = NULL,
  method = "Bliss",
  vF = NULL,
  plot_exmpDt = FALSE,
  ...
)
```

Arguments

npg	Number of subjects per group.
time	Vector with the times at which the tumor volume measurements have been performed.
grwrControl	Coefficient for Control treatment group tumor growth rate.
grwrA	Coefficient for Drug A treatment group tumor growth rate.
grwrB	Coefficient for Drug B treatment group tumor growth rate.
grwrComb	Coefficient for Combination (Drug A + Drug B) treatment group tumor growth rate.
sd_ranef	Random effects standard deviation (between-subject variance) for the model.
sgma	Residuals standard deviation (within-subject variance) for the model.
sd_eval	A vector with values representing the standard deviations of random effects, through which the power for synergy calculation will be evaluated.
sgma_eval	A vector with values representing the standard deviations of the residuals, through which the power for synergy calculation will be evaluated.
grwrComb_eval	A vector with values representing the coefficients for Combination treatment group tumor growth rate, through which the power for synergy calculation will be evaluated.
method	String indicating the method for synergy calculation. Possible methods are "Bliss" and "HSA", corresponding to Bliss and highest single agent, respectively.
vF	An optional nlme::varFunc object or one-sided formula describing the within-group heteroscedasticity structure. If given as a formula, it is used as the argument to nlme::varFixed, corresponding to fixed variance weights. See the documentation on nlme::varClasses for a description of the available nlme::varFunc

	classes. Defaults to NULL, corresponding to homoscedastic within-group errors.
plot_exmpDt	Logical indicating if a plot representing the hypothetical data should also be returned.
...	Additional parameters to be passed to nlmeU::Pwr.lme method.

Details

APrioriPwr allows for total customization of an hypothetical drug combination study and allows the user to define several experimental parameters, such as the sample size, time of measurements, or drug effect, for the power evaluation of synergy for Bliss and HSA reference models. The power calculation is based on F-tests of the fixed effects of the model as previously described (Helms, R. W. (1992), Verbeke and Molenberghs (2009), Gałecki and Burzykowski (2013)).

The focus the power analysis with APrioriPwr is on the **drug combination effect** and the **variability** in the experiment. For other *a priori* power analysis see also PwrSampleSize() and PwrTime().

- npg, time, grwrControl, grwrA, grwrB, grwrComb, sd_ranef and sgma are parameters referring to the initial exemplary data set and corresponding fitted model. These values can be obtained from a fitted model, using lmmModel_estimates(), or be defined by the user.
- sd_eval, sgma_eval, and grwrComb_eval are the different values that will be modified in the initial exemplary data set to fit the corresponding model and calculate the power.

Value

The functions returns several plots:

- A plot showing the values of the power calculation depending on the values assigned to sd_eval and sgma_eval. The power result corresponding to the values assigned to sd_ranef and sgma is shown with a red dot.
- A plot showing the values of the power calculation depending on the values assigned to grwrComb_eval. The vertical dashed line indicates the value of grwrComb. The horizontal line indicates the power of 0.80. If plot_exmpDt = TRUE, a plot representing the hypothetical data, with the regression lines for each treatment group according to grwrControl, grwrA, grwrB and grwrComb values is also plotted. The values assigned to sd_ranef and sgma are also shown.

The statistical power for the fitted model for the initial data set according to the values given by npg, time, grwrControl, grwrA, grwrB, grwrComb, sd_ranef and sgma is also shown in the console.

References

- Helms, R. W. (1992). *Intentionally incomplete longitudinal designs: I. Methodology and comparison of some full span designs*. Statistics in Medicine, 11(14–15), 1889–1913. https://doi.org/10.1002/sim.47801114
- Verbeke, G. & Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data*. Springer New York. https://doi.org/10.1007/978-1-4419-0300-6
- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

PostHocPwr,PwrSampleSize(), PwrTime().

Examples

```
APrioriPwr(
sd_eval = seq(0.01, 0.2, 0.01),
sgma_eval = seq(0.01, 0.2, 0.01),
grwrComb_eval = seq(0.01, 0.1, 0.005)
)
```

CookDistance *Cook's distance for individual subjects*

Description

CookDistance allows the user to identify those subjects with a greater influence in the predicted values or in the estimation of the fixed effects for the treatment group, based in the calculation of Cook's distances.

Usage

```
CookDistance(
  model,
  type = "fitted",
  cook_thr = NA,
  label_angle = 0,
  maxIter = 1000,
  verbose = TRUE
)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by lmmModel().
type	Type of Cook's distance to calculated. Possible options are fitted, to calculte Cook's distances based on the change in fitted values, or fixef to calculate Cook's distances based on the change in the fixed effects. See Details section for more information.
cook_thr	Numeric value indicating the threshold for the Cook's distance. If not specified, the threshold is set to three times the mean of the Cook's distance values.
label_angle	Numeric value indicating the angle for the label of subjects with a Cook's distance greater than cook_thr.
maxIter	Limit of maximum number of iterations for the optimization algorithm. Default to 1000.
verbose	Logical indicating if the subjects with a Cook's distance greater than cook_thr should be printed to the console.

Details

The identification of influential subjects is based on the calculation of Cook's distances. The Cook's distances can be calculated based on the change in fitted values or fixed effects.

- **Cook's distances based on the change in fitted values**

When `type = "fitted"`, the Cook's distances are calculated as the normalized change in fitted response values due to the removal of a subject from the model. Firts, a leave-one-subject-out model is fitted, removing individually each subject to fit the model. Then, the Cook's distance for subject i, (D_i), is calculated as:

$$D_i = \frac{\sum_{j=1}^{n}\left(\hat{y}_j - \hat{y}_{j_{(-i)}}\right)^2}{p \cdot MSE}$$

where $\hat{y}_j$ is the j^{th} fitted response value using the complete model, and $\hat{y}_{j_{(-i)}}$ is the j^{th} fitted response value obtained using the model where subject i has been removed.

The denominator of the expression is equal to the number of the fixed-effects coefficients, p, which, under the assumption that the design matrix is of full rank, is equivalent to the rank of the design matrix, and the Cook distance is normalized by the mean square error (MSE) of the model.

- **Cook's distances based on the change in fixed effects values**

The identification of the subjects with a greater influence in the estimated fixed effects is based on the calculation of Cook's distances, as described in Gałecki and Burzykowsk (2013). To compute the Cook's distance for the fixed effect estimates (i.e., the contribution to each subject to the coefficients of its treatment group), first a matrix containing the leave-one-subject-out estimates or the fixed effects is calculated. Then, the Cook's distances are calculated according to:

$$D_i \equiv \frac{(\hat{\beta} - \hat{\beta}_{(-i)})[\widehat{Var(\hat{\beta})}]^{-1}(\hat{\beta} - \hat{\beta}_{(-i)})}{p}$$

where β represents the vector of fixed effects and $\hat{\beta}_{(-i)}$ is the estimate of the parameter vector β obtained by fitting the model to the data with the i-th subject excluded. The denominator of the expression is equal to the number of the fixed-effects coefficients, p, which, under the assumption that the design matrix is of full rank, is equivalent to the rank of the design matrix.

Value

A plot of the Cook's distance value for each subject, indicating those subjects whose Cook's distance is greater than `cook_thr`.

If saved to a variable, the function returns a vector with the Cook's distances for each subject.

References

- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

Examples

```
#' # Load the example data
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Calulate Cook's distances for each subject
CookDistance(model = lmm)
# Change the Cook's distance threshold
CookDistance(model = lmm, cook_thr = 0.15)
```

`getRTV` *Helper function to calculate the relative tumor volume from an imput data frame of tumor growth*

Description

`getRTV` is a helper function used inside `lmmModel()` to obtain a dataframe with a column *RTV* corresponding to the relative tumor volume to time `time_start`, and a column *logRTV* with the logarithm of *RTV*.

Usage

```
getRTV(data, time_start)
```

Arguments

`data`	Data frame with the tumor growth data. The input data frame columns have to have the following names: • `SampleID`: Column with the identifiers for each sample. • `Time`: Column with the time for each measurement. • `TV`: Column with the tumor volume measurement.
`time_start`	Numeric value indicating the time at which the treatment started.

Value

The function returns the original data frame of tumor growth data, with 3 additional columns, corresponding to:

- RTV: Relative tumor volume to the tumor volume at `start_time`.
- logRTV: Logarithm of RTV column.
- TV0: Tumor volume at `start_time`.

Examples

```
# Load example dataset
data("grwth_data")
# Change column names
colnames(grwth_data) <- c("SampleID", "Time", "Treatment", "TV")
# Calculate relative tumor volume
getRTV(data = grwth_data, time_start = 0)
```

---

grwth_data *Example Tumor Growth Data*

Description

A long format data frame with example tumor growth data, generated with `simulateTumorGrowth()`, representing a simulated *in vivo* two-drugs combination experiment, involving 8 subjects per group, with 30 days of follow-up, taking measurement every 3 days.

Usage

```
data(grwth_data)
```

Format

A data frame with 352 rows and 4 columns:

subject Subject IDs.

Time Time for each measurement in an arbitrary unit, for example, days.

Treatment Treatment group for each subject (Control, DrugA, DrugB, and Combination).

TumorVolume Measurement representing the tumor volume in an arbitrary unit, for example, mm^3.

`lmmModel` *Linear Mixed Effect Model for Tumor Growth*

Description

`lmmModel()` fits a mixed effect model from a tumor growth dataset. The input data frame must be in long format and include at least the following columns: column with the sample ids, column with the time at which each measurement has been done, a column indicating the treatment group, and a column with the tumor measurement (e.g. tumor volume).

Usage

```
lmmModel(
  data,
  grwth_model = "exp",
  sample_id = "SampleID",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TV",
  trt_control = "Control",
  drug_a = "Drug_A",
  drug_b = "Drug_B",
  drug_c = NA,
  combination = "Combination",
  time_start = NULL,
  time_end = NULL,
  min_observations = 1,
  show_plot = TRUE,
  tum_vol_0 = "ignore",
  start_values = c(0.05, 0.01),
  ...
)
```

Arguments

`data`	A data frame with the tumor growth data, in long format. It must contain at least the following columns: mice IDs, time of follow-up (numeric number), treatment and tumor volume (numeric number).
`grwth_model`	Tumor growth model to use. Possible options are "exp", for exponential tumor growth model, or "gompertz" for Gompertz tumor growth model.
`sample_id`	String indicating the name of the column with the mice IDs.
`time`	String indicating the name of the column with the time of follow-up.
`treatment`	String indicating the name of the column with the treatment corresponding to each sample.
`tumor_vol`	String indicating the name of the column with the tumor volume (or any other measurement representing the tumor growth).

`trt_control`	String indicating the name assigned to the 'Control' group.
`drug_a`	String indicating the name assigned to the 'Drug A' group.
`drug_b`	String indicating the name assigned to the 'Drug B' group.
`drug_c`	String indicating the name assigned to the 'Drug C' group (if present).
`combination`	String indicating the name assigned to the Combination ('Drug A' + 'Drug B', or 'Drug A' + 'Drug B' + 'Drug C') group.
`time_start`	Numeric value indicating the time point at which the treatment started. If not specified, the minimum value in the `time` column is used as the starting time point.
`time_end`	Numeric value indicating the last time point to be included in the analysis. If not specified, the maximum value in the `time` column is used as the final time point.
`min_observations`	Minimum number of observation for each sample to be included in the analysis.
`show_plot`	Logical indicating if a plot for the log of the relative tumor volume (RTV) vs Time for each sample, and the model calculated marginal slope for each treatment, should be produced.
`tum_vol_0`	Select the behavior of the function regarding measurements in which the tumor measurement is 0, and therefore the logarithmic transformation is not possible. Possible options are 'ignore' (default), to ignore these measurements, or 'transform', to add 1 unit to all measurements before the log transformation.
`start_values`	Numeric vector of length 2 with initial estimates for the fixed effects of the Gompertz model (r_0, ρ). It can be set to "selfStart", in which case initial values will be derived from a Gompertz model derived from a call to `stats::nls`
`...`	Additional arguments to be passed to `nlme::lme` or `nlme::nlme`.

Details

`lmmModel()` fits a mixed effect model describing the tumor growth. Currently, two growth kinetics models are available: exponential growth, and Gompertz growth.

Exponential

`lmmModel()` will fit a linear mixed-effect model (LMM) assuming that the tumor growth follows an exponential kinetics. Any departure from this assumption can be tested using the diagnostics functions `ranefDiagnostics()`, `residDiagnostics()`, and `ObsvsPred()`.

The model formula for the LMM following the **exponential tumor growth** is:

$$\log RTV_i(t) = \beta_{T_i} \cdot t + b_i \cdot t + \varepsilon_i(t).$$

- $\log RTV_i(t)$ denotes the value of the logarithm of the relative tumor volume measured for subject i at time t.
- β_{T_i} represents the fixed effects for each treatment T_i, where $T_i \in \{Control, DrugA, DrugB, Combination\}$ in the case of two-drugs combination experiments, or $T_i \in \{Control, DrugA, DrugB, DrugC, Combination\}$ in the case of three-drugs combination experiments, and indicates the tumor-specific growth rate for each treatment group.

- $b_i \cdot t$ corresponds to the subject-specific random slope that takes into account the longitudinal nature of the data, where $b_i \sim \mathcal{N}(0, \sigma_b^2)$ is the random effect for subject i.
- $\varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2)$ is the residual error term.

Gompertz

`lmmModel()` will fit a non-linear mixed effect (NLME) model assuming the tumor growth follows a Gompertz growth kinetics. Any departure from this assumption can be tested using the diagnostics functions `ranefDiagnostics()`, `residDiagnostics()`, and `ObsvsPred()`.

The model formula for the non-linear mixed-effect model following the **Gompertz tumor growth** is:

$$\log RTV_i(t) = \frac{r_{0_{T_i}} + b_{0_i}}{\rho_{T_i} + b_{1_i}}(1 - e^{-(\rho_{T_i} + b_{1_i}) \cdot t}) + \varepsilon_i(t).$$

- $\log RTV_i(t)$ denotes the value of the logarithm of the relative tumor volume measured for subject i at time t.
- $T_i \in \{Control, DrugA, DrugB, Combination\}$ in the case of two-drugs combination experiments, or $T_i \in \{Control, DrugA, DrugB, DrugC, Combination\}$ in the case of three-drugs combination experiments, indicates the treatment groupt of individual i.
- $r_{0_{T_i}}$ is the fixed effect for the initial growth rate for treatment group T_i.
- ρ_{T_i} is the fixed effect for the constant accounting for the reduction in the tumor growth rate for treatment group T_i.
- $b_{0_i} \sim \mathcal{N}(0, \sigma_{r_0}^2)$ is the random effect on r_0 for individual i.
- $b_{1_i} \sim \mathcal{N}(0, \sigma_\rho^2)$ is the random effect on ρ for individual i.
- $\varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2)$ is the residual error term.

The implementation of the linear mixed model in `lmmModel()` is done using `nlme::lme` for the exponential model, or `nlme::nlme` for the Gompertz model. Both also allow for the specification of within-group correlations structures and/or unequal variances. These, and additional parameters, can be passed to the `nlme::lme` or `nlme::nlme` functions through the `...` argument for fitting the model (see examples below).

Value

An object of class "lme" (see `nlme::lme` for details) or "nlme" (see `nlme::nlme` for details) representing the linear mixed-effects model fit. If `show_plot = TRUE`, the plot of the tumor growth data obtained with `plot_lmmModel()` is also shown.

References

- Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in S and S-PLUS*. Springer, New York. doi:10.1007/b98882 doi:10.1007/b98882.
- Pinheiro J, Bates D, R Core Team (2024). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-166, https://CRAN.R-project.org/package=nlme.
- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

nlme::lmeControl, nlme::nlmeControl, nlme::varClasses.

Examples

```
data("grwth_data")
# Most simple model
lmmModel(
 data = grwth_data,
 sample_id = "subject",
 time = "Time",
 treatment = "Treatment",
 tumor_vol = "TumorVolume",
 trt_control = "Control",
 drug_a = "DrugA",
 drug_b = "DrugB",
 combination = "Combination"
 )
# Changing the last time point of follow-up
lmmModel(
 data = grwth_data,
 sample_id = "subject",
 time = "Time",
 treatment = "Treatment",
 tumor_vol = "TumorVolume",
 trt_control = "Control",
 drug_a = "DrugA",
 drug_b = "DrugB",
 combination = "Combination",
 time_end = 21
 )
# Adding additional parameters for model fitting
lmmModel(
 data = grwth_data,
 sample_id = "subject",
 time = "Time",
 treatment = "Treatment",
 tumor_vol = "TumorVolume",
 trt_control = "Control",
 drug_a = "DrugA",
 drug_b = "DrugB",
 combination = "Combination",
 # Adding variance function to represent a different variance per subject
 weights = nlme::varIdent(form = ~1|SampleID),
 # Specifiying control values for lme Fit (useful when convergence problems appear)
 control = nlme::lmeControl(maxIter = 1000, msMaxIter = 1000, niterEM = 100, msMaxEval = 1000)
 )

# Fit a model specifying a different variance per Time
lmmModel(
 data = grwth_data,
 sample_id = "subject",
```

```
 time = "Time",
 treatment = "Treatment",
 tumor_vol = "TumorVolume",
 trt_control = "Control",
 drug_a = "DrugA",
 drug_b = "DrugB",
 combination = "Combination",
 # Adding variance function to represent a different variance per Time
 weights = nlme::varIdent(form = ~1|Time)
 )

# Fit a model using the Gompertz model

lmmModel(
data = grwth_data,
grwth_model = "gompertz", # Selecting Gompertz model
start_values = "selfStart", # Using self-starting values
sample_id = "subject",
time = "Time",
treatment = "Treatment",
tumor_vol = "TumorVolume",
trt_control = "Control",
drug_a = "DrugA",
drug_b = "DrugB",
combination = "Combination"
)
```

`lmmModel_estimates` *Get estimates from a linear mixed model of tumor growth data*

Description

`lmmModel_estimates` allows the user to easily extract some of the interesting model estimates for further use in other functions, such as for power calculation.

Usage

```
lmmModel_estimates(model, robust = FALSE, type = "CR2")

## S3 method for class 'explme'
lmmModel_estimates(model, robust = FALSE, type = "CR2")

## S3 method for class 'gompertzlme'
lmmModel_estimates(model, robust = FALSE, type = "CR2")
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by lmmModel().
robust	If TRUE, sandwich-based robust estimators of the standard error of the regression coefficient estimates by clubSandwich::conf_int() are provided. Sandwich-based robust estimators are only available for exponential growth models ('explme').
type	Character string specifying which small-sample adjustment should be used when robust = True. Available options are "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of clubSandwich::vcovCR() for further information.

Details

The model estimates provided by lmmModel_estimates include:

- Fixed effect coefficients for the Control, Drug A, Drug B, (Drug C, if present), and Combination groups, respectively.
- The standard error (se) corresponding to each of the fixed effect coefficients.
- Standard deviation of the random effects (between-subject variance). Column sd_ranef.
- Standard deviation of the residuals (within-subject variance). Column sd_resid.

Value

A data frame with the estimated values for the coefficients of the tumor growth for each treatment, their standard error, the standard deviation of the random effects, and the standard deviation of the residuals of the model. These values can be useful for the power analysis of the model using APrioriPwr().

Examples

```
data("grwth_data")
# Fit example model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Get the estimates
lmmModel_estimates(lmm)
```

lmmSynergy *Synergy calculation using linear-mixed and non-linear mixed-effect models*

Description

lmmSynergy allows for the calculation of synergy using 3 different references models: Bliss independence, highest single agent and response additivity. The calculation of synergy is based on hypothesis testing on the coefficient estimates from the model fitted by lmmModel().

Usage

```
lmmSynergy(
  model,
  method = "Bliss",
  min_time = 0,
  conf_level = 0.95,
  padj = "none",
  robust = FALSE,
  type = "CR2",
  nsim = 1000,
  set_seed = TRUE,
  show_plot = TRUE,
  ...
)

## S3 method for class 'explme'
lmmSynergy(
  model,
  method = "Bliss",
  min_time = 0,
  conf_level = 0.95,
  padj = "none",
  robust = FALSE,
  type = "CR2",
  nsim = 1000,
  set_seed = TRUE,
  show_plot = TRUE,
  ...
)

## S3 method for class 'gompertzlme'
lmmSynergy(
  model,
  method = "Bliss",
  min_time = 0,
  conf_level = 0.95,
```

```
  padj = "none",
  robust = FALSE,
  type = "CR2",
  nsim = 10000,
  set_seed = TRUE,
  show_plot = TRUE,
  ...
)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by `lmmModel()`.
method	String indicating the method for synergy calculation. Possible methods are "Bliss", "HSA" and "RA", corresponding to Bliss, highest single agent, and response additivity, respectively.
min_time	Minimun time for which to start calculating synergy.
conf_level	Numeric value between 0 and 1. Confidence level to use to build a confidence interval and obtain p-values. The default value is 0.95.
padj	String indicating the correction method for adjusting p-values. Possible options are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". More details in `stats::p.adjust()`.
robust	If TRUE, uncertainty is estimated using sandwich-based robust estimators of the variance-covariance matrix of the regression coefficient estimates provided by clubSandwich::vcovCR.lme. Sandwich-based robust estimators are only available for exponential growth models ('explme').
type	Character string specifying which small-sample adjustment should be used if 'robust = TRUE', with available options "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of `clubSandwich::vcovCR()` for further information.
nsim	Number of random sampling to calculate the synergy for Response Additivity model, and for synergy assessment in Gompertz growth models.
set_seed	Logical indicating if the seed for those methods based on simulations (RA synergy and Gompertz growth models) should be fixed for reproducible results. The seed can be also set before running `lmmSynergy()` using `set.seed()` function.
show_plot	Logical indicating if a plot with the results of the synergy calculation should be generated.
...	Additional arguments to be passed to `marginaleffects::hypotheses()`.

Details

`lmmSynergy` uses the statistical description provided by Demidenko and Miller (2019) for the calculation of synergy. It is based on hypothesis testing on the coefficients estimates from the model fitted by `lmmModel()`.

Exponential Growth Model:

Estimated coefficients $\hat{\beta}_C$, $\hat{\beta}_A$, $\hat{\beta}_B$, $\hat{\beta}_{AB}$, which represent the estimated specific growth rates for the Control, Drug A, Drug B and Combination groups, respectively, are to used to calculate synergy.

Bliss Independence

For Bliss model, `lmmSynergy` test the following null hypothesis:

Two-drugs combination experiment:

$$H_0 : \beta_{combination} = \beta_A + \beta_B - \beta_{control}$$

Three-drugs combination experiment:

$$H_0 : \beta_{combination} = \beta_A + \beta_B + \beta_C - 2\beta_{control}$$

Highes Single Agent (HSA)

For the HSA model, `lmmSynergy` test the following null hypothesis:

Two-drugs combination experiment:

$$H_0 : \beta_{combination} = \min(\beta_A, \beta_B)$$

Three-drugs combination experiment:

$$H_0 : \beta_{combination} = \min(\beta_A, \beta_B, \beta_C)$$

Response Additivity (RA)

For the RA model, `lmmSynergy` test the following null hypothesis:

Two-drugs combination experiment:

$$H_0 : e^{\beta_{combination} t} = e^{\beta_A t} + e^{\beta_B t} - e^{\beta_{control} t}$$

Three-drugs combination experiment:

$$H_0 : e^{\beta_{combination} t} = e^{\beta_A t} + e^{\beta_B t} + e^{\beta_C t} - 2e^{\beta_{control} t}$$

For **Bliss** and **HSA** models, `lmmSynergy` uses `marginaleffects::hypotheses()` to conduct hypothesis tests on the estimated coefficients of the model.

In the case of the **RA** model, the null hypothesis is tested comparing the area under the curve (i.e. cumulative effect from the beginning of a treatment to a time point of interest) obtained from each side of the equation for the null hypothesis, based on `ra_sim` random samplings from the distribution of the coefficients.

Gompertz Growth Model:

Estimated coefficients $r_{0_{T_i}}$ and ρ_{T_i} are to used to calculate synergy.

- $r_{0_{T_i}}$ is the fixed effect for the initial growth rate for treatment group T_i.
- ρ_{T_i} is the fixed effect for the constant accounting for the reduction in the tumor growth rate for treatment group T_i.

The following expressions are simplified with these symbols:

$$\gamma = \frac{r0_{Control}}{\rho_{Control}} \cdot (1 - e^{-\rho_{Control} \cdot t})$$

$$A = \frac{r0_{DrugA}}{\rho_{DrugA}} \cdot (1 - e^{-\rho_{DrugA} \cdot t})$$

$$B = \frac{r0_{DrugB}}{\rho_{DrugB}} \cdot (1 - e^{-\rho_{DrugB} \cdot t})$$

$$C = \frac{r0_{DrugC}}{\rho_{DrugC}} \cdot (1 - e^{-\rho_{DrugC} \cdot t})$$

$$\Lambda = \frac{r0_{Combination}}{\rho_{Combination}} \cdot (1 - e^{-\rho_{Combination} \cdot t})$$

Bliss Independence

For Bliss model, `lmmSynergy` test the following null hypothesis:
Two-drugs combination experiment:

$$H_0 : \Lambda = A + B - \gamma$$

Three-drugs combination experiment:

$$H_0 : \Lambda = A + B + C - 2\gamma$$

Highes Single Agent (HSA)

For the HSA model, `lmmSynergy` test the following null hypothesis:
Two-drugs combination experiment:

$$H_0 : \Lambda = \min(A, B)$$

Three-drugs combination experiment:

$$H_0 : \Lambda = \min(A, B, C)$$

Response Additivity (RA)

For the RA model, `lmmSynergy` test the following null hypothesis:
Two-drugs combination experiment:

$$H_0 : e^{\Lambda} = e^{A} + e^{B} - e^{\gamma}$$

Three-drugs combination experiment:

$$H_0 : e^{\Lambda} = e^{A} + e^{B} + e^{C} - 2e^{\gamma}$$

For the **Gompertz models**, the null hypothesis is tested comparing the area under the curve (i.e. cumulative effect from the beginning of a treatment to a time point of interest) obtained from each side of the equations for the null hypothesis, based on `nsim` random samplings from the distribution of the coefficients.

Combination Index and Synergy Score:

The results obtained by `lmmSynergy` include the synergy score (SS) and combination index (CI) for the model, for each time point, together with their confidence interval, and the corresponding p-value. The values of SS and CI provided by `lmmSynergy` follow previous definitions of these metrics so they have the same interpretation:

- The SS has been defined as the excess response due to drug interaction compared to the reference model (Ianevski et al. (2017), Ianevski, Giri, and Aittokallio (2022), Mao and Guo (2023)). Following this definition, a $SS > 0$, $SS = 0$, and $SS < 0$, represent synergistic, additive and antagonistic effects, respectively.
- According to the common definition of the CI, a $CI < 1$, $CI = 1$, and $CI > 1$ represent synergistic, additive and antagonistic effects, respectively (Yadav et al. (2015), Demidenko and Miller (2019), Mao and Guo (2023)), and provides information about the observed drug combination effect versus the expected additive effect provided by the reference synergy model. A drug combination effect larger than the expected ($CI < 1$) would indicate synergism, a drug combination effect equal to the expected ($CI = 1$) would indicate additivity, and a lower drug combination effect than the expected ($CI > 1$) would indicate antagonism.

As mentioned above, the results include the synergy results for **each day**. This means that `lmmSynergy` refits the model using the data from `time_start` defined in `lmmModel()` until each time point, providing the synergy results for each of these models and for that specific time point.

Uncertainty estimation using robust estimators

If `robust = TRUE`, `lmmSynergy` deals with possible model misspecifications, allowing for cluster-robust variance estimation using clubSandwich::vcovCR.lme. When using `robust = TRUE`, setting `type = "CR2"` is recommended. See more details in `clubSandwich::vcovCR()`. This option is only available for 'explme' exponential tumor growth models.

Note: When a variance structure has been specified in the model it is recommended to use always `robust = TRUE` to get a better estimation.

Value

The function returns a list with two elements:

- `Constrasts`: List with the outputs of the linear test for the synergy null hypothesis obtained by `marginaleffects::hypotheses()` for each time. See `marginaleffects::hypotheses()` for more details.
- `Synergy`: Data frame with the synergy results, indicating the model of synergy ("Bliss", "HSA" or "RA"), the metric (combination index and synergy score), the value of the metric estimate (with upper and lower confidence interval bounds) and the p-value, for each time.
- `Estimates`: Data frame with the estimates from each model at each time point, obtained with `lmmModel_estimates()` function. For 'explme' exponential tumor growth models, if `robust=TRUE`, sandwich-based robust estimators for the standard errors of the estimated coefficients are reported.

If `show_plot = TRUE`, a plot with the synergy results obtained with `plot_lmmSynergy()` is also shown.

References

- Demidenko, Eugene, and Todd W. Miller. 2019. *Statistical Determination of Synergy Based on Bliss Definition of Drugs Independence.* PLoS ONE 14 (November). https://doi.org/10.1371/journal.pone.0224137.

- Yadav, Bhagwan, Krister Wennerberg, Tero Aittokallio, and Jing Tang. 2015. *Searching for Drug Synergy in Complex Dose–Response Landscapes Using an Interaction Potency Model.* Computational and Structural Biotechnology Journal 13: 504–13. https://doi.org/10.1016/j.csbj.2015.09.001.
- Ianevski, Aleksandr, Liye He, Tero Aittokallio, and Jing Tang. 2017. *SynergyFinder: A Web Application for Analyzing Drug Combination Dose–Response Matrix Data.* Bioinformatics 33 (August): 2413–15. https://doi.org/10.1093/bioinformatics/btx162.
- Ianevski, Aleksandr, Anil K Giri, and Tero Aittokallio. 2022. *SynergyFinder 3.0: An Interactive Analysis and Consensus Interpretation of Multi-Drug Synergies Across Multiple Samples.* Nucleic Acids Research 50 (July): W739–43. https://doi.org/10.1093/nar/gkac382.
- Mao, Binchen, and Sheng Guo. 2023. *Statistical Assessment of Drug Synergy from in Vivo Combination Studies Using Mouse Tumor Models.* Cancer Research Communications 3 (October): 2146–57. https://doi.org/10.1158/2767-9764.CRC-23-0243.
- Vincent Arel-Bundock, Noah Greifer, and Andrew Heiss. Forthcoming. How to Interpret Statistical Models Using marginaleffects in R and Python. Journal of Statistical Software. https://marginaleffects.com

Examples

```
# Load the example data
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Most simple use with default values (Bliss reference model)
syn <- lmmSynergy(lmm)
# Accessing to synergy results data frame
syn$Synergy
# Selecting different reference models:
## HSA
lmmSynergy(lmm, method = "HSA")
## RA
lmmSynergy(lmm, method = "RA", nsim = 10) # 10 simulations for shorter computing time

# Only calculate synergy from Time 21 onwards
lmmSynergy(lmm, min_time = 21)
```

logLikSubjectDisplacements
Likelihood displacements for the model

Description

logLikSubjectDisplacements allows the user to evaluate the log-likelihood displacement for each subject, indicating the influence of every subject to the model.

Usage

```
logLikSubjectDisplacements(
  model,
  disp_thrh = NA,
  label_angle = 0,
  var_name = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by lmmModel().
disp_thrh	Numeric value indicating the threshold of log-likelihood displacement. If not specified, the threshold is set to three times the mean of the log-likelihood displacement values.
label_angle	Numeric value indicating the angle for the label of subjects with a log-likelihood displacement greater than disp_thrh.
var_name	Name of the variable for the weights of the model in the case that a variance structure has been specified using nlme::varIdent(). (See examples in lmmModel()).
verbose	Logical indicating if subjects with a log-likelihood displacement greater than disp_thrh should be printed to the console.
...	Extra arguments, if any, for lattice::panel.xyplot.

Details

The evaluation of the log-likelihood displacement is based in the analysis proposed in Verbeke and Molenberghs (2009) and Gałecki and Burzykowski (2013). First, a list of models fitted to leave-one-subject-out datasets are obtained. Then, for each model, the maximum likelihood estimate obtained by fitting the model to all data and the maximum likelihood estimate obtained by fitting the model to the data with the i-th subject removed are obtained and used for the log-likelihood displacement calculation. The likelihood displacement, LDi , is defined as twice the difference between the log-likelihood computed at a maximum and displaced values of estimated parameters (Verbeke and Molenberghs (2009), Gałecki and Burzykowsk (2013)):

$$LD_i \equiv 2\times\left[\ell_{\text{Full}}(\widehat{\Theta};\mathrm{y}) - \ell_{\text{Full}}(\widehat{\Theta}_{(-i)};\mathrm{y})\right]$$

where $\widehat{\Theta}$ is the maximum-likelihood estimate of Θ obtained by fitting the model to all data, while $\widehat{\Theta}_{-i}$ is the maximum-likelihood estimate obtained by fitting the model to the data with the i-subject excluded.

Value

Returns a plot of the log-likelihood displacement values for each subject, indicating those subjects whose contribution is greater than `disp_thrh`.

References

- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1
- Verbeke, G. & Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data*. Springer New York. https://doi.org/10.1007/978-1-4419-0300-6

Examples

```
# Load the example data
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Obtain log-likelihood displacement for each subject
logLikSubjectDisplacements(model = lmm)
# Modifying the threshold for log-likelihood displacement
logLikSubjectDisplacements(model = lmm, disp_thrh = 1)

# Calculating the log-likelihood contribution in a model with a variance structure specified
lmm_var <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination",
```

```
  weights = nlme::varIdent(form = ~ 1|SampleID)
  )
# Calculate the log-likelihood contribution
logLikSubjectDisplacements(model = lmm, var_name = "SampleID")
```

`ObsvsPred` *Observed vs predicted values and performance of the model*

Description

`ObsvsPred` allows the user to have a straight forward idea about how the model is fitting the data, providing plots of the predicted regression lines versus the actual data points.

Usage

```
ObsvsPred(model, nrow = 4, ncol = 5, ...)
```

Arguments

`model`	An object of class "lme" representing the linear mixed-effects model fitted by `lmmModel()`.
`nrow`	Number of rows of the layout to organize the observed vs predicted plots.
`ncol`	Number of columns of the layout to organize the observed vs predicted plots.
`...`	Additional arguments to be passed to `performance::model_performance()`.

Details

The function provides visual and quantitative information about the performance of the model:

- A layout of the observed and predicted values of *log*(relative tumor volume) vs Time for each SampleID (i.e. subject), with the actual measurements, the regression line for each SampleID, and the marginal, treatment-specific, regression line for each treatment group.
- Performance metrics of the model obtain calculated using `performance::model_performance()`. The maximum likelihood-based Akaike's Information Criterion (AIC), small sample AIC (AICc), and Bayesian Information Criterion, and the Nakagawa's r-squared root mean squared error (RMSE) of the residuals, and the standard deviation of the residuals (sigma) are provided.

Value

Performance metrics of the model obtain calculated using `performance::model_performance()` and a layout of plots of the observed vs predicted values for each SampleID.

References

- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1
- Lüdecke et al., (2021). *performance: An R Package for Assessment, Comparison and Testing of Statistical Models*. Journal of Open Source Software, 6(60), 3139. https://doi.org/10.21105/joss.03139
- Sakamoto, Y., M. Ishiguro, and G. Kitagawa. 1984. *Akaike Information Criterion Statistics*. Mathematics and Its Applications. Reidel.
- Nakagawa, Shinichi, and Holger Schielzeth. 2013. *A General and Simple Method for Obtaining r2 from Generalized Linear Mixed-effects Models*. Methods in Ecology and Evolution 4 (February): 133–42. https://doi.org/10.1111/j.2041-210x.2012.00261.x.
- Johnson, Paul C. D. 2014. *Extension of Nakagawa & Schielzeth's r 2 GLMM to Random Slopes Models*. Methods in Ecology and Evolution 5 (September): 944–46. https://doi.org/10.1111/2041-210X.12225.
- Nakagawa, Shinichi, Paul C. D. Johnson, and Holger Schielzeth. 2017. *The Coefficient of Determination r2 and Intra-Class Correlation Coefficient from Generalized Linear Mixed-Effects Models Revisited and Expanded*. Journal of The Royal Society Interface 14 (September): 20170213. https://doi.org/10.1098/rsif.2017.0213.

Examples

```
# Load the example data
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Obtain Observed vs Predicted plots, and model performance metrics
ObsvsPred(model = lmm, nrow = 4, ncol = 8)
```

`plot_lmmModel` *Plotting of tumor growth data from a fitted model*

Description

Vizualization of tumor growth data and linear mixed model fitted regression line for the fixed effects. This functions returns a ggplot2 plot, allowing for further personalization.

Usage

```
plot_lmmModel(
  model,
  trt_control = "Control",
  drug_a = "Drug_A",
  drug_b = "Drug_B",
  drug_c = NA,
  combination = "Combination"
)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by lmmModel().
trt_control	String indicating the name assigned to the 'Control' group.
drug_a	String indicating the name assigned to the 'Drug A' group.
drug_b	String indicating the name assigned to the 'Drug B' group.
drug_c	String indicating the name assigned to the 'Drug C' group (if present).
combination	String indicating the name assigned to the Combination ('Drug A' + 'Drug B', or 'Drug A' + 'Drug B' + 'Drug C') group.

Value

A ggplot2 plot (see ggplot2::ggplot() for more details) showing the tumor growth data represented as log(relative tumor volume) versus time since treatment initiation. The regression lines corresponding to the fixed effects for each treatment group are also plotted.

Examples

```
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination",
  show_plot = FALSE
  )
# Default plot
plot_lmmModel(lmm,
trt_control = "Control",
drug_a = "DrugA",
drug_b = "DrugB",
combination = "Combination"
```

```
)
# Adding ggplot2 elements
plot_lmmModel(lmm,
trt_control = "Control",
drug_a = "DrugA",
drug_b = "DrugB",
combination = "Combination"
) + ggplot2::labs(title = "Example Plot") + ggplot2::theme(legend.position = "top")
```

`plot_lmmSynergy` *Plotting synergy results*

Description

Visualization of synergy results obtained by `lmmSynergy()`. This functions returns a ggplot2 plot, allowing for further personalization.

Usage

```
plot_lmmSynergy(syn_data)
```

Arguments

`syn_data`	Object obtained by `lmmSynergy()` with the results of synergy calculation using linear mixed models.

Details

`plot_lmmSynergy` produces a ggplot2 plot with the results of the synergy calculation. Each dot represents the estimated combination index or synergy score, and the gray lines represent the 95% confidence intervals, for each day. Each dot is colored based on the $-\log_{10}(p-value)$, with purple colors indicating a $-\log_{10}(p-value) < 1.3; (p-value > 0.05)$, and green colors indicating a $-\log_{10}(p-value) > 1.3; (p-value < 0.05)$.

Value

A list with ggplot2 plots (see `ggplot2::ggplot()` for more details) with the combination index (CI) and synergy score (SS) estimates, confidence intervals and p-values for the synergy calculation using linear mixed models.

Examples

```
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
```

```
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Obtain synergy results
lmmSyn <- lmmSynergy(lmm)
# Plot synergy results
plot_lmmSynergy(lmmSyn)
# Accessing to the combination index plot
plot_lmmSynergy(lmmSyn)$CI
# Accessing to only synergy score plot
plot_lmmSynergy(lmmSyn)$SS
# Accessing to the grid of both plots side by side
plot_lmmSynergy(lmmSyn)$CI_SS
# Adding ggplot2 elements
plot_lmmSynergy(lmmSyn)$CI +
ggplot2::labs(title = "Synergy Calculation for Bliss") +
ggplot2::theme(legend.position = "top")
```

`plot_ObsvsPred` *Plots of Observed vs Predicted Values*

Description

Visualization of observed vs predicted values by a fitted linear mixed model of tumor growth data.

Usage

```
plot_ObsvsPred(model, nrow = 4, ncol = 5)
```

Arguments

`model`	An object of class "lme" representing the linear mixed-effects model fitted by `lmmModel()`.
`nrow`	Number of rows of the layout to organize the observed vs predicted plots.
`ncol`	Number of columns of the layout to organize the observed vs predicted plots.

Value

A layout (arranged in `nrow` rows and `ncol` columns) of the observed and predicted values of *log*(relative tumor volume) vs Time for each SampleID (i.e. subject), with the actual measurements, the regression line for each SampleID, and the marginal, treatment-specific, regression line for each treatment group.

Examples

```
#' data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination",
  show_plot = FALSE
  )
# Obtain the plots
plot_ObsvsPred(lmm, nrow = 4, ncol = 8)
```

`plot_ranefDiagnostics` *Plots for random effects diagnostics*

Description

Visualization of random effects diagnostics for a fitted linear mixed model of tumor growth data.

Usage

```
plot_ranefDiagnostics(model)
```

Arguments

`model`	An object of class "lme" representing the linear mixed-effects model fitted by `lmmModel()`.

Value

A list with different plots for evaluating the normality and homoscedasticity of the random effects, including:

- A normal Q-Q plot of the random effects of the model.
- A normal Q-Q plot of the residuals by sample.
- Boxplots of the "raw" residuals (observed - fitted) by sample.
- Scatter plots of the normalized residuals (standardized residuals pre-multiplied by the inverse square-root factor of the estimated error correlation matrix, see nlme::residuals.lme) vs fitted values by sample. Observations with absolute standardized (normalized) residuals greater than the $1 - 0.05/2$ quantile of the standard normal distribution are identified in the plots labelled with the time point corresponding to the observation.

Examples

```
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination",
  show_plot = FALSE
  )
# Generate plots
plot_ranefDiagnostics(lmm)
# Access to specific plots
plot_ranefDiagnostics(lmm)$plots[[1]]
plot_ranefDiagnostics(lmm)$plots[[2]]
```

---

plot_residDiagnostics *Plots for residuals diagnostics*

Description

Visualization of residuals diagnostics for a fitted linear mixed model of tumor growth data.

Usage

```
plot_residDiagnostics(model)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by `lmmModel()`.

Value

A list with different plots for evaluating the normality and homoscedasticity of the normalized residuals (standardized residuals pre-multiplied by the inverse square-root factor of the estimated error correlation matrix, see nlme::residuals.lme), including:

- A normal Q-Q plot of the normalized residuals of the model.
- A normal Q-Q plot of the normalized residuals of the model by Time.
- A normal Q-Q plot of the normalized residuals of the model by Treatment.
- A dotplot of normalized residuals vs fitted values.
- A dotplot of the normalized residuals by Time and Treatment.

Examples

```
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination",
  show_plot = FALSE
  )
# Generate plots
plot_residDiagnostics(lmm)
# Access to specific plots
plot_residDiagnostics(lmm)[[1]]
plot_residDiagnostics(lmm)[[2]]
```

---

plot_SynergyLMM *Plotting SynergyLMM results*

Description

Generic function to generate different types of plots based on SynergyLMM outputs.

Usage

```
plot_SynergyLMM(object, plot_type = "lmmModel", ...)
```

Arguments

`object`	An object generated by `lmmModel()` or `lmmSynergy()` functions.
`plot_type`	String indicating the type of plot to generate. Possible options include: • "lmmModel" for plotting of tumor growth data from a fitted model. • "lmmSynergy" for plotting synergy results. • "ObsvsPred" for generating plots of Observed vs Predicted Values. • "ranefDiagnostics" for plots of random effects diagnostics. • "residDiagnostics" for plots of residual diagnostics.
`...`	Additional arguments passed to the specific plot function.

Value

Different output plots are produced depending on the `plot_type` selected.

See Also

plot_lmmModel(), plot_lmmSynergy(), plot_ObsvsPred(), plot_ranefDiagnostics(), plot_residDiagnostics().

Examples

```
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination",
  show_plot = FALSE
  )

# Plot lmmModel
plot_SynergyLMM(lmm, plot_type = "lmmModel",
trt_control = "Control",
drug_a = "DrugA",
drug_b = "DrugB",
combination = "Combination"
)

# Plot ObsvsPred
plot_SynergyLMM(lmm, plot_type = "ObsvsPred")

# Plot ranefDiagnostics
plot_SynergyLMM(lmm, plot_type = "ranefDiagnostics")

# Plot residDiagnostics
plot_SynergyLMM(lmm, plot_type = "residDiagnostics")

# Plot lmmSynergy
lmmSyn <- lmmSynergy(lmm)
plot_SynergyLMM(lmmSyn, plot_type = "lmmSynergy")
```

PostHocPwr *Post hoc power calculation based on simulations of the synergy evaluation using LMM.*

Description

PostHocPwr allows for the *post hoc* power analysis of the synergy hypothesis testing for Bliss and HSA refence models for a given tumor growth data fitted model.

Usage

```
PostHocPwr(model, nsim = 1000, method = "Bliss", pvalue = 0.05, time = NA, ...)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by lmmModel().
nsim	Number of simulations to perform.
method	String indicating the method for synergy calculation. Possible methods are "Bliss" and "HSA", corresponding to Bliss and highest single agent, respectively.
pvalue	Threshold for the p-value of synergy calculation to be considered statistically significant.
time	Time point for which to calculate the statistical power. If not specified, the last time point is used by default.
...	Additional parameters to be passed to nlmeU::simulateY:

Details

The *post hoc* power calculation relies on simulation of the dependent variable, using nlmeU::simulateY.

1. For a given fitted model of the tumor growth data, nsim simulations of the dependent variable ($\log(RTV)$) are done, based on the marginal distribution implied by the fitted model.
2. The model is then fitted to the new values of the dependant variable.
3. For each simulation, the new estimates from each model are then used for the synergy hypothesis testing as explained in lmmSynergy, and the p-values stored.
4. The power is returned as the proportion of simulations resulting in a significant synergy hypothesis testing (p-value < pvalue).

When time is specified, PostHocPwr refits the model using the data from the time_start time point defined in lmmModel() until time, and report the statistical power for that model. If time is not specified, the model fitted using all data points is used for statistical power calculation.

Value

Returns a numeric value of the power for the synergy calculation for the model using the method specified in method. The power is expressed as the proportion of simulations that provides a p-value below the threshold specified in pvalue.

References

Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

APrioriPwr(), PwrSampleSize(), PwrTime().

Examples

```
#' data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
 PostHocPwr(lmm, nsim = 10) # 10 simulations for shorter computing time
 # Using a seed to obtain reproducible results
 PostHocPwr(lmm, seed = 123, nsim = 10)
 # Calculating the power for an specific day
 PostHocPwr(lmm, nsim = 10, time = 6)
```

Pwr *Calculates power based on a model fit*

Description

This function is generic; method functions can be written to handle specific classes of objects.

Usage

```
Pwr(object, ...)
```

Arguments

`object`	an object containing the results returned by a model fitting function (e.g., `lme`).
`...`	some methods for this generic function may require additional arguments.

Value

numeric scalar value.

Author(s)

Andrzej Galecki and Tomasz Burzykowski

See Also

`Pwr.lme`

Examples

```
data("grwth_data")
lmm <- lmmModel(data = grwth_data,
                sample_id = "subject",
                time = "Time",
                treatment = "Treatment",
                tumor_vol = "TumorVolume",
                trt_control = "Control",
                drug_a = "DrugA",
                drug_b = "DrugB",
                combination = "Combination")
Pwr(lmm,
    L = c("Time:TreatmentControl" = 1,
    "Time:TreatmentDrugA" = -1,
    "Time:TreatmentDrugB" = -1,
    "Time:TreatmentCombination" = 1))
```

Pwr.lme *Performs power calculations*

Description

This is method for `Pwr()` generic function. It is a modified version from the one described by Galecki and Burzykowski implemented in `nlmeU` package (nlmeU::Pwr.lme).

Usage

```
## S3 method for class 'lme'
Pwr(
  object,
  ...,
  type = c("sequential", "marginal"),
  Terms,
  L,
  verbose = FALSE,
  sigma,
  ddf = numeric(0),
  alpha = 0.05,
  altB = NULL,
  tol = 1e-10
)
```

Arguments

`object`	an object containing `lme` fit, which provides information needed for power calculations
`...`	some additional arguments may be required.

type	an optional character string specifying the type of sum of squares to be used in F-tests needed for power calculations. Syntax is the same as for anova.lme() in nlme package.
Terms	an optional integer or character vector specifying which terms in the model should be jointly tested to be zero using a Wald F-test. See anova.lme in nlme package for details.
L	an optional numeric vector or array specifying linear combinations of the coefficients in the model that should be tested to be zero. See anova.lme in nlme package for details.
verbose	an optional logical value. See anova.lme in nlme package for details.
sigma	numeric scalar value.
ddf	numeric scalar value. Argument can be used to redefine default number of denominator degrees of freedom
alpha	numeric scalar value. By default 0.05.
altB	matrix/vector containing alternative values for beta parameters
tol	numeric scalar value.

Value

a data frame inheriting from class Pwr.lme

References

- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

nlme::anova.lme, nlmeU::Pwr.lme

PwrSampleSize A Priori *Synergy Power Analysis Based on Sample Size*

Description

A priori power calculation for a hypothetical two-drugs combination study of synergy evaluation using linear-mixed models depending on the sample size per group.

Usage

```
PwrSampleSize(
  npg = c(5, 8, 10),
  time = c(0, 3, 5, 10),
  grwrControl = 0.08,
  grwrA = 0.07,
  grwrB = 0.06,
  grwrComb = 0.03,
  sd_ranef = 0.01,
  sgma = 0.1,
  method = "Bliss",
  vF = NULL,
  plot_exmpDt = FALSE,
  ...
)
```

Arguments

npg	A vector with the sample size (number of subjects) per group to calculate the power of the synergy analysis.
time	Vector with the times at which the tumor volume measurements have been performed.
grwrControl	Coefficient for Control treatment group tumor growth rate.
grwrA	Coefficient for Drug A treatment group tumor growth rate.
grwrB	Coefficient for Drug B treatment group tumor growth rate.
grwrComb	Coefficient for Combination (Drug A + Drug B) treatment group tumor growth rate.
sd_ranef	Random effects standard deviation for the model.
sgma	Residuals standard deviation for the model.
method	String indicating the method for synergy calculation. Possible methods are "Bliss" and "HSA", corresponding to Bliss and highest single agent, respectively.
vF	An optional nlme::varFunc object or one-sided formula describing the within-group heteroscedasticity structure. If given as a formula, it is used as the argument to nlme::varFixed, corresponding to fixed variance weights. See the documentation on nlme::varClasses for a description of the available nlme::varFunc classes. Defaults to NULL, corresponding to homoscedastic within-group errors.
plot_exmpDt	Logical indicating if a plot representing the hypothetical data should also be returned.
...	Additional parameters to be passed to nlmeU::Pwr.lme method.

Details

PwrSampleSize allows the user to define an hypothetical drug combination study, customizing several experimental parameters, such as the sample size, time of measurements, or drug effect,

for the power evaluation of synergy for Bliss and HSA reference models. The power calculation is based on F-tests of the fixed effects of the model as previously described (Helms, R. W. (1992), Verbeke and Molenberghs (2009), Gałecki and Burzykowski (2013)).

The focus the power analysis with `PwrSampleSize` is on the **sample size per group**. The function allows for the evaluation of how the statistical power changes when the sample size per group varies while the other parameters are kept constant. For other *a priori* power analysis see also `APrioriPwr()` and `PwrTime()`.

- `time`, `grwrControl`, `grwrA`, `grwrB`, `grwrComb`, `sd_ranef` and `sgma` are parameters referring to the initial exemplary data set and corresponding fitted model. These values can be obtained from a fitted model, using `lmmModel_estimates()`, or be defined by the user.
- `npg` is a vector indicating the different sample sizes for which the statistical power is going to be evaluated, keeping the rest of parameters fixed.

Value

The functions returns a plot showing the values of the power calculation depending on the values assigned to `npg`. If `plot_exmpDt = TRUE`, a plot representing the hypothetical data, with the regression lines for each treatment group according to `grwrControl`, `grwrA`, `grwrB` and `grwrComb` values is also plotted. The values assigned to `sd_ranef` and `sgma` are also shown.

The function also returns the data frame with the power for the analysis for each sample size specified in `npg`.

References

- Helms, R. W. (1992). *Intentionally incomplete longitudinal designs: I. Methodology and comparison of some full span designs*. Statistics in Medicine, 11(14–15), 1889–1913. https://doi.org/10.1002/sim.47801114
- Verbeke, G. & Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data*. Springer New York. https://doi.org/10.1007/978-1-4419-0300-6
- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

PostHocPwr, `APrioriPwr()`, `PwrTime()`.

Examples

```
PwrSampleSize(npg = 1:20)
```

PwrTime A Priori *Synergy Power Analysis Based on Time*

Description

A priori power calculation for a hypothetical two-drugs combination study of synergy depending on the time of follow-up or the frequency of measurements.

Usage

```
PwrTime(
  npg = 5,
  time = list(seq(0, 9, 3), seq(0, 21, 3), seq(0, 30, 3)),
  type = "max",
  grwrControl = 0.08,
  grwrA = 0.07,
  grwrB = 0.06,
  grwrComb = 0.03,
  sd_ranef = 0.01,
  sgma = 0.1,
  method = "Bliss",
  vF = NULL,
  plot_exmpDt = FALSE,
  ...
)
```

Arguments

npg	Number of mouse per group.
time	A list in which each element is a vector with the times at which the tumor volume measurements have been performed. If type is set to "max", each vector in the list should represent measurements taken at the same interval and differ in the final time of follow-up. If type is set to "freq", each vector in the list should have the same final time of follow-up and differ in the intervals at which the measurements have been taken.
type	String indicating whether to calculate the power depending on the time of follow-up ("max"), or the frequency of measurements ("freq").
grwrControl	Coefficient for Control treatment group tumor growth rate.
grwrA	Coefficient for Drug A treatment group tumor growth rate.
grwrB	Coefficient for Drug B treatment group tumor growth rate.
grwrComb	Coefficient for Combination (Drug A + Drug B) treatment group tumor growth rate.
sd_ranef	Random effects standard deviation for the model.
sgma	Residuals standard deviation for the model.

method	String indicating the method for synergy calculation. Possible methods are "Bliss" and "HSA", corresponding to Bliss and highest single agent, respectively.
vF	An optional nlme::varFunc object or one-sided formula describing the within-group heteroscedasticity structure. If given as a formula, it is used as the argument to nlme::varFixed, corresponding to fixed variance weights. See the documentation on nlme::varClasses for a description of the available nlme::varFunc classes. Defaults to NULL, corresponding to homoscedastic within-group errors.
plot_exmpDt	Logical indicating if a plot representing the hypothetical data should also be returned.
...	Additional parameters to be passed to nlmeU::Pwr.lme method.

Details

PwrTime allows the user to define an hypothetical drug combination study, customizing several experimental parameters, such as the sample size, time of measurements, or drug effect, for the power evaluation of synergy for Bliss and HSA reference models. The power calculation is based on F-tests of the fixed effects of the model as previously described (Helms, R. W. (1992), Verbeke and Molenberghs (2009), Gałecki and Burzykowski (2013)).

The focus the power analysis with PwrTime is on the **time** at which the measurements are done. The function allows for the evaluation of how the statistical power changes when the time of follow-up varies while the frequency of measurements is keep constant. It also allows to how the statistical power changes when the time of follow-up is kept constant, but the frequency of measurements varies.

For other *a priori* power analysis see also APrioriPwr() and PwrSampleSize().

- npg, grwrControl, grwrA, grwrB, grwrComb, sd_ranef and sgma are parameters referring to the initial exemplary data set and corresponding fitted model. These values can be obtained from a fitted model, using lmmModel_estimates(), or be defined by the user.
- time is a list in which each element is a vector with the times at which the tumor volume measurements have been performed, and for which the statistical power is going to be evaluated, keeping the rest of parameters fixed.

Value

The functions returns a plot showing the values of the power calculation depending on the values assigned to Time. If type is set to "max", the plot shows how the power varies depending on the maximum time of follow-up. If type is set to "freq", the plot shows how the power varies depending on how frequently the measurements have been performed. If plot_exmpDt = TRUE, a plot representing the hypothetical data, with the regression lines for each treatment group according to grwrControl, grwrA, grwrB and grwrComb values is also plotted. The values assigned to sd_ranef and sgma are also shown.

The function also returns the data frame with the power for the analysis for each value specified in Time.

References

- Helms, R. W. (1992). *Intentionally incomplete longitudinal designs: I. Methodology and comparison of some full span designs*. Statistics in Medicine, 11(14–15), 1889–1913. https://doi.org/10.1002/sim.47801114
- Verbeke, G. & Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data*. Springer New York. https://doi.org/10.1007/978-1-4419-0300-6
- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

PostHocPwr, `APrioriPwr()`, `PwrSampleSize()`.

Examples

```
# Power analysis maintaining the frequency of measurements
# and varying the time of follow-up ('type = "max"')
PwrTime(time = list(seq(0, 9, 3),
                    seq(0, 12, 3),
                    seq(0, 15, 3),
                    seq(0, 21, 3),
                    seq(0, 30, 3)),
                    type = "max")

# Power analysis maintaining the time of follow-up
# and varying the frequency of measurements ('type = "freq"')
PwrTime(time = list(seq(0, 10, 1),
                    seq(0, 10, 2),
                    seq(0, 10, 5),
                    seq(0, 10, 10)),
                    type = "freq")
```

---

ranefDiagnostics *Diagnostics of random effects of the linear mixed model*

Description

`ranefDiagnostics` provides several plots as well as statistical test for the examination of the normality of the random effects of the input model.

Usage

```
ranefDiagnostics(model, norm_test = "shapiroTest", verbose = TRUE)
```

Arguments

model	An object of class "lme" or "nlme" representing the mixed-effects model fitted by lmmModel().
norm_test	String indicating the function for testing the normality of the random effects. A collection of functions from fBasics::normalTest is available. We recommend using one of "shapiroTest", "dagoTest", or "adTest" for performing Shapiro - Wilk, D'Agostino, or Anderson - Darling normality test, respectively.
verbose	Logical indicating if the normality and homoscedasticity tests results should be printed to the console.

Details

One of the assumptions of the model obtained with lmmModel() (as in any other linear mixed model) is that the random effects are normally distributed:

$$b_i = N(0, \psi)$$

For the evaluation of this assumption, ranefDiagnostics provides Q-Q plots of random effects, together with statistical assessment of their normality using Shapiro-Wilk, D'Agostini and Anderson-Darling normality tests. Additionally, Q-Q plots of the normalized residuals (standardized residuals pre-multiplied by the inverse square-root factor of the estimated error correlation matrix, see nlme::residuals.lme) by sample are provided to allow for the identification of subjects which could be notably different from the others and be affecting the adequacy of the model. Additionally, boxplots of the "raw" residuals (observed - fitted) by sample and scatter plots of the normalized residuals versus fitted values by sample are provided to give information about variability of the residuals by subject and possible outlier observations. Observations with absolute standardized (normalized) residuals greater than the $1 - 0.05/2$ quantile of the standard normal distribution are identified in the scatter plots labelled with the time point corresponding to the observation.

Value

A list with different elements for the diagnostics of the random effects are produced:

- plots: Different plots for evaluating the normality and homoscedasticity of the random effects are produced.
- Normality: Results from the test of the normality of the random effects.
- Levene.test: results from Levene homoscedasticity test (car::leveneTest()) of the normalized residuals by SampleID (i.e., by subject).
- Fligner.test: results from Fligner homoscedasticity test (stats::fligner.test()) of the normalized residuals by SampleID (i.e., by subject).

References

- Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in S and S-PLUS*. Springer, New York. doi:10.1007/b98882.
- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

See Also

plot_ranefDiagnostics()

Examples

```
# Load the example data
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )
# Run random effects diagnostics
ranef_diag <- ranefDiagnostics(lmm)

#Access to individual plots

ranef_diag$Plots[1]
ranef_diag$Plots[2]

# Access to normality tests

ranef_diag$Normality

# Access to homoscedasticity tests of residuals by subject

ranef_diag$Levene.test

ranef_diag$Fligner.test
```

---

residDiagnostics *Diagnostics of residuals of the linear mixed model*

Description

residDiagnostics provides several plots as well as statistical test for the examination of the normality and homoscedasticity of the residuals of the input model.

Usage

```
residDiagnostics(
  model,
  pvalue = 0.05,
  norm_test = "shapiroTest",
  verbose = TRUE
)
```

Arguments

model	An object of class "lme" representing the linear mixed-effects model fitted by lmmModel().
pvalue	Threshold for the p-value of outlier observations based on their normalized residuals.
norm_test	String indicating the function for testing the normality of the normalized residuals. A collection of functions from fBasics::normalTest is available. We recommend using one of "shapiroTest", "dagoTest", or "adTest" for performing Shapiro - Wilk, D'Agostino, or Anderson - Darling normality test, respectively.
verbose	Logical indicating if the normality and homoscedasticity tests results, and the list of potential outlier observations should be printed to the console.

Details

One of the assumption of the model fit by lmmModel() is that the residuals are normally distributed. For the evaluation of this assumption, residDiagnostics provides Q-Q plots of the normalized residuals (standardized residuals pre-multiplied by the inverse square-root factor of the estimated error correlation matrix, see nlme::residuals.lme), together with statistical assessment of their normality using Shapiro-Wilk, D'Agostini and Anderson-Darling normality tests. Additionally, Q-Q plots of the normalized residuals by time point and treatment group are provided to be able to detect time points or treatment groups which could be notably different from the others and be affecting the adequacy of the model.

Scatter plots of the normalized residuals versus fitted values and normalized residuals per time and per treatment are also provided to give information about variability of the residuals and possible outlier observations. These plots are accompanied by Levene and Fligner-Killend homogeneity of variance test results.

Observations with absolute standardized (normalized) residuals greater than the $1-0.05/2$ quantile of the standard normal distribution are identified and reported as potential outlier observations.

Value

A list with different elements for the diagnostics of the residuals are produced:

- plots: Different plots for evaluating the normality and homocedasticity of the residuals.
- outliers: Data frame with the identified outliers based on the Pearson residuals and the value of pval. The column resid.p contains the value of the Pearson residuals for each observation.
- Normality: Results from the test of the normality of the normalized residuals of the model.

- `Levene.test`: List with the Levene homoscedasticity test results of the normalized residuals by Time and Treatment.
- `Fligner.test`: List with the Fligner-Killeen homoscedasticity test results of the normalized residuals by Time and Treatment.

References

- Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in S and S-PLUS*. Springer, New York. doi:10.1007/b98882.
- Andrzej Galecki & Tomasz Burzykowski (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach* First Edition. Springer, New York. ISBN 978-1-4614-3899-1

Examples

```
# Load the example data
data(grwth_data)
# Fit the model
lmm <- lmmModel(
  data = grwth_data,
  sample_id = "subject",
  time = "Time",
  treatment = "Treatment",
  tumor_vol = "TumorVolume",
  trt_control = "Control",
  drug_a = "DrugA",
  drug_b = "DrugB",
  combination = "Combination"
  )

# Residuals diagnostics
resid_diag <- residDiagnostics(model = lmm, pvalue = 0.05)

# Access outliers data frame
resid_diag$Outliers

# Access individual plots
resid_diag$Plots[1]
resid_diag$Plots[2]

# Access results of normality tests
resid_diag$Normality

# Access to homoscedasticity test results

resid_diag$Levene.test

resid_diag$Fligner.test
```

`simulateTumorGrowth` *Helper function to simulate tumor growth data for a two-drug combination experiment.*

Description

Helper function to simulate tumor growth data for a two-drug combination experiment.

Usage

```
simulateTumorGrowth(
  npg = 5,
  timepoints = c(0, 3, 5, 10),
  initial_volume = 100,
  grwrControl = 0.08,
  grwrA = 0.07,
  grwrB = 0.06,
  grwrComb = 0.04,
  sd = 0.1
)
```

Arguments

`npg`	Number of samples per group.
`timepoints`	Vector with the time points at which the tumor volume measurements have been performed.
`initial_volume`	Initial volume for the tumor growth.
`grwrControl`	Coefficient for Control treatment group tumor growth rate.
`grwrA`	Coefficient for Drug A treatment group tumor growth rate.
`grwrB`	Coefficient for Drug B treatment group tumor growth rate.
`grwrComb`	Coefficient for Combination (Drug A + Drug B) treatment group tumor growth rate.
`sd`	Variability for the tumor growth.

Details

The function simulates the tumor growth following exponential kinetics, given by

$$TV(t) = TV_0 \cdot e^{\beta_i \cdot t}$$

where TV_0 is given by `initial_volume`, t is given by `timepoints` and β_i are the coefficients given by `grwrControl`, `grwrA`, `grwrB`, and `grwrComb`.

The variability is simulated using the `sd` argument to add random noise from a normal distribution $N(1, SD)$, with SD = `sd`.

Value

A data frame with tumor growth data in long format.

Examples

```
# This code generates the 'grwth_data' example dataset:
set.seed(123)
grwth_data <- simulateTumorGrowth(npg = 8, timepoints = seq(0,30,3),
                                  initial_volume = 200, grwrControl = 0.08,
                                  grwrA = 0.07, grwrB = 0.065,
                                  grwrComb = 0.04, sd = 0.2)
```

Index

∗ **datasets**
grwth_data, 8

APrioriPwr, 2
APrioriPwr(), *14*, *32*, *37*, *39, 40*

car::leveneTest(), *41*
clubSandwich::conf_int(), *14*
clubSandwich::vcovCR(), *14*, *16*, *19*
clubSandwich::vcovCR.lme, *16*, *19*
CookDistance, 5

getRTV, 7
ggplot2::ggplot(), *25, 26*
grwth_data, 8

lattice::panel.xyplot, *21*
lme, *10, 11*
lmeControl, *12*
lmmModel, 9
lmmModel(), *5*, *7*, *14–16*, *19*, *21*, *23*, *25*, *27–30*, *32*, *41*, *43*
lmmModel_estimates, 13
lmmModel_estimates(), *4*, *19*, *37*, *39*
lmmSynergy, 15, *32*
lmmSynergy(), *26*, *30*
logLikSubjectDisplacements, 21

marginaleffects::hypotheses(), *16, 17*, *19*

nlme, *10, 11*
nlme::anova.lme, *35*
nlme::residuals.lme, *28, 29*, *41*, *43*
nlme::varClasses, *3*, *36*, *39*
nlme::varFixed, *3*, *36*, *39*
nlme::varFunc, *3*, *36*, *39*
nlme::varIdent(), *21*
nlmeControl, *12*
nlmeU::Pwr.lme, *4*, *34–36*, *39*
nlmeU::simulateY, *32*
nls, *10*
normalTest, *41*, *43*

ObsvsPred, 23
ObsvsPred(), *10, 11*

performance::model_performance(), *23*
plot_lmmModel, 24
plot_lmmModel(), *11*, *31*
plot_lmmSynergy, 26
plot_lmmSynergy(), *19*, *31*
plot_ObsvsPred, 27
plot_ObsvsPred(), *31*
plot_ranefDiagnostics, 28
plot_ranefDiagnostics(), *31*, *42*
plot_residDiagnostics, 29
plot_residDiagnostics(), *31*
plot_SynergyLMM, 30
PostHocPwr, *5*, 31, *37*, *40*
Pwr, 33
Pwr.lme, *33*, 34
PwrSampleSize, 35
PwrSampleSize(), *4, 5*, *32*, *39, 40*
PwrTime, 38
PwrTime(), *4, 5*, *32*, *37*

ranefDiagnostics, 40
ranefDiagnostics(), *10, 11*
residDiagnostics, 42
residDiagnostics(), *10, 11*

simulateTumorGrowth, 45
simulateTumorGrowth(), *8*
stats::fligner.test(), *41*
stats::p.adjust(), *16*

varClasses, *12*