Package ‘glmmSeq’

August 12, 2022

Title  General Linear Mixed Models for Gene-Level Differential Expression

Version  0.4.0

Description  Using mixed effects models to analyse longitudinal gene expression can highlight differences between sample groups over time. The most widely used differential gene expression tools are unable to fit linear mixed effect models, and are less optimal for analysing longitudinal data. This package provides negative binomial and Gaussian mixed effects models to fit gene expression and other biological data across repeated samples. This is particularly useful for investigating changes in RNA-Sequencing gene expression between groups of individuals over time, as described in: Rivellese, F., Surace, A. E., Goldmann, K., Sciacca, E., Cubuk, C., Giorli, G., ... Lewis, M. J., & Pitzalis, C. (2022) Nature medicine <doi:10.1038/s41591-022-01789-0>.

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  true

biocViews

RoxygenNote  7.2.0

Language  en-gb

URL  https://github.com/KatrionaGoldmann/glmmSeq

BugReports  https://github.com/KatrionaGoldmann/glmmSeq/issues

Suggests  knitr, rmarkdown, kableExtra, DESeq2, edgeR, emmeans

VignetteBuilder  knitr

Depends  R (>= 3.6.0)

Imports  MASS, car, stats, ggplot2, ggpubr, graphics, lme4, lmerTest, methods, plotly, qvalue, pbapply, pbmcapply

NeedsCompilation  no

Author  Myles Lewis [aut] (<https://orcid.org/0000-0001-9365-5345>), Katriona Goldmann [aut, cre] (<https://orcid.org/0000-0002-9073-6323>), Elisabetta Sciacca [aut] (<https://orcid.org/0000-0001-7525-1558>), Cankut Cubuk [ctb] (<https://orcid.org/0000-0003-4646-0849>), Anna Surace [ctb] (<https://orcid.org/0000-0001-9589-3005>)
fcPlot

Plotly or ggplot fold change plots

Description
Plotly or ggplot fold change plots

Usage
fcPlot(
  object,
  x1var,
  x2var,
  x1Values = NULL,
  x2Values = NULL,
  pCutoff = 0.01,
  labels = c(),
  useAdjusted = FALSE,
  plotCutoff = 1,
  graphics = "ggplot",
  fontSize = 12,
  labelFontSize = 4,
  colours = c("grey", "goldenrod1", "red", "blue"),
  verbose = FALSE,
  ...
)

Maintainer  Katriona Goldmann <k.goldmann@qmul.ac.uk>
Repository  CRAN
Date/Publication  2022-08-12 13:10:07 UTC

R topics documented:

fcPlot ......................................................... 2
ggmmodelPlot ............................................... 4
glmmQvals .................................................... 6
eglmmRefit .................................................. 6
glmmSeq ...................................................... 7
GlmmSeq-class ............................................... 9
ImmSeq ........................................................ 10
ImmSeq-class ............................................... 12
maPlot ....................................................... 12
metadata ..................................................... 14
modelPlot .................................................... 14
tpm ............................................................ 16

Index 17

fcPlot  Plotly or ggplot fold change plots
Arguments

- **object**: A `glmmSeq` object created by `glmmSeq::glmmSeq()`.
- **x1var**: The name of the first (inner) x parameter.
- **x2var**: The name of the second (outer) x parameter.
- **x1Values**: Timepoints or categories in x1var used to calculate fold change. If NULL the first two levels in x1var are used.
- **x2Values**: Categories in x2var to be compared on x and y axis.
- **pCutoff**: The significance cut-off for colour-coding (default = 0.01).
- **labels**: Row names or indices to label on plot.
- **useAdjusted**: Whether to use adjusted p-values (must have q-values in object). Default = FALSE.
- **plotCutoff**: Which probes to include on plot by significance cut-off (default = 1, for all markers).
- **graphics**: Graphics system to use: "ggplot" or "plotly".
- **fontSize**: Font size.
- **labelFontSize**: Font size for labels.
- **colours**: Vector of colours to use for significance groups.
- **verbose**: Whether to print statistics.
- **...**: Other parameters to pass to plotly or ggplot.

Value

Returns a plot for fold change between x1Values in one x2Value subset on x axis and fold change in the other x2Value on the y axis.

Examples

```r
data(PEAC_minimal_load)

disp <- apply(tpm, 1, function(x) {
  (var(x, na.rm = TRUE)-mean(x, na.rm = TRUE))/(mean(x, na.rm = TRUE)**2)
})

g1mmFit <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  countdata = tpm[1:5, ],
  metadata = metadata,
  dispersion = disp,
  verbose = FALSE)

fcPlot(object = g1mmFit,
       x1var = "Timepoint",
       x2var = "EULAR_6m",
       x2Values = c("Good", "Non-response"),
       pCutoff = 0.05,
       useAdjusted = FALSE,
       plotCutoff = 1,
       graphics = "plotly")
```
ggmodelPlot  

Mixed model effects plot using ggplot2

Description

Plot to show differences between groups and over time using ggplot2.

Usage

```r
ggmodelPlot(
  object,
  geneName = NULL,
  x1var = NULL,
  x2var = NULL,
  x2shift = NULL,
  xlab = NULL,
  ylab = geneName,
  title = geneName,
  logTransform = is(object, "GlmmSeq"),
  shapes = 19,
  colours = "grey60",
  lineColours = "grey60",
  markerSize = 1,
  fontSize = 12,
  alpha = 0.7,
  x2Offset = 5,
  addPoints = TRUE,
  addModel = TRUE,
  modelSize = 4,
  modelColours = "blue",
  modelLineSize = 1,
  modelLineColours = modelColours,
  addBox = FALSE,
  ...
)
```

Arguments

- **object**: A glmmSeq/lmmSeq object created by `glmmSeq::glmmSeq()` or `lmmSeq::lmmSeq()`.
- **geneName**: The gene/row name to be plotted.
- **x1var**: The name of the first (inner) x parameter, typically 'time'. This is anticipated to have different values when matched by ID.
- **x2var**: The name of an optional second (outer) x parameter, which should be a factor.
- **x2shift**: Amount to shift along x axis for each level of x2var. By default the function will arrange each level of x2var side by side.
ggmodelPlot

xlab Title for the x axis
ylab Title for the y axis
title Plot title. If NULL gene name is used
logTransform Whether to perform a log10 transform on the y axis
shapes The marker shapes (default=19)
colours The marker colours as vector
lineColours The line colours (default=’grey60’) as vector
markerSize Size of markers (default=1)
fontSize Plot font size
alpha Line and marker opacity (default=0.7)
x2Offset Vertical adjustment to secondary x-axis labels (default=5)
addPoints Whether to add underlying data points (default=TRUE)
addModel Whether to add the fit model with markers (default=TRUE)
modelSize Size of model points (default=4)
modelColours Colour of model fit markers (default=’blue’) as vector
modelLineSize Size of model points (default=1) as vector
modelLineColours Colour of model fit lines
addBox Logical whether to add boxplots for mean and IQR
...
 Other parameters to pass to ggplot2::theme().

Value

Returns a paired plot for matched samples.

Examples

data(PEAC_minimal_load)

disp <- apply(tpm, 1, function(x){
  (var(x, na.rm=TRUE)-mean(x, na.rm=TRUE))/(mean(x, na.rm=TRUE)**2)
})

MS4A1glmm <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  countdata = tpm['MS4A1', , drop = FALSE],
  metadata = metadata,
  dispersion = disp,
  verbose = FALSE)

ggmodelPlot(object = MS4A1glmm,
  geneName = 'MS4A1',
  x1var = 'Timepoint',
  x2var = 'EULAR_6m',
  colours = c('skyblue', 'goldenrod1', 'mediumvioletred'))
glmmQvals

Glmm Sequencing qvalues

Description
Add qvalue columns to the glmmSeq dataframe

Usage
`glmmQvals(object, cutoff = 0.05, verbose = TRUE)`

Arguments

object
A glmmSeq/lmmSeq object created by `glmmSeq::glmmSeq()`.

cutoff
Prints a table showing the number of probes considered significant by the pvalue cut-off (default=0.05)

verbose
Logical whether to print the number of significant probes (default=TRUE)

Value
Returns a GlmmSeq object with results for gene-wise general linear mixed models with adjusted p-values using the qvalue function

Examples
```
data(PEAC_minimal_load)
disp <- apply(tpm, 1, function(x) {
  (var(x, na.rm=TRUE)-mean(x, na.rm = TRUE))/(mean(x, na.rm = TRUE)**2)
})
MS4A1glmm <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  countdata = tpm[1:5, ],
  metadata = metadata,
  dispersion = disp[1:5],
  verbose=FALSE)
MS4A1glmm <- glmmQvals(MS4A1glmm)
```

---

glmmRefit

Refit mixed effects model

Description
Based on a 'GlmmSeq' or 'lmmSeq' class result object, this function attempts to refit an identical model for a specific gene based on the data and fitting parameters stored in the results object and refitting using either `lme4::glmer()` for glmmSeq objects or `lmer()` for lmmSeq objects. The fitted model can then be passed on to other packages such as emmeans to look at estimated marginal means for the model.
**glmmSeq**

Usage

```r
glmmRefit(object, gene, ...)
```

Arguments

- `object`: A fitted results object of class `GlmmSeq` or `lmmSeq`
- `gene`: A character value specifying a single gene to extract a fitted model for
- `...`: Optional arguments passed to either `lme4::glmer` or `lme4::lmer`

Value

A fitted model of class `lmerMod` in the case of LMM or `glmerMod` for a GLMM

---

**glmmSeq**

*GLMM with negative binomial distribution for sequencing count data*

Description

Fits many generalised linear mixed effects models (GLMM) with negative binomial distribution for analysis of overdispersed count data with random effects. Designed for longitudinal analysis of RNA-Sequencing count data. Wald type 2 Chi-squared test is used to calculate p-values.

Usage

```r
glmmSeq(
    modelFormula,
    countdata,
    metadata,
    id = NULL,
    dispersion,
    sizeFactors = NULL,
    reducedFormula = "",
    modelData = NULL,
    designMatrix = NULL,
    control = glmerControl(optimizer = "bobyqa"),
    cores = 1,
    removeSingles = FALSE,
    zeroCount = 0.125,
    verbose = TRUE,
    returnList = FALSE,
    progress = FALSE,
    ...
)
```
Arguments

- `modelFormula`: the model formula. This must be of the form "~..." where the structure is assumed to be "counts ~ ...". The formula must include a random effects term. For more information on formula structure for random effects see `lme4::glmer()`.
- `countdata`: the sequencing count data matrix with genes in rows and samples in columns.
- `metadata`: a dataframe of sample information with variables in columns and samples in rows.
- `id`: Optional. Used to specify the column in metadata which contains the sample IDs to be used in repeated samples for random effects. If not specified, the function defaults to using the variable after the "i" in the random effects term in the formula.
- `sizeFactors`: size factors (default = NULL). If provided the glmer offset is set to log(sizeFactors). For more information see "lme4::glmer()
- `reducedFormula`: Reduced design formula (default = "")
- `modelData`: Optional dataframe. Default is generated by call to expand.grid using levels of variables in the formula. Used to calculate model predictions (estimated means & 95% CI) for plotting via `modelPlot`. It can therefore be used to add/remove points in `modelPlot`.
- `designMatrix`: custom design matrix.
- `control`: the glmer optimizer control (default = `glmerControl(optimizer = "bobyqa")`). See `lme4::glmerControl()`.
- `cores`: number of cores to use. Default = 1.
- `removeSingles`: whether to remove individuals without repeated measures (default = FALSE)
- `zeroCount`: numerical value to offset zeroes for the purpose of log (default = 0.125)
- `verbose`: Logical whether to display messaging (default = TRUE)
- `returnList`: Logical whether to return results as a list or `glmmSeq` object (default = FALSE). Useful for debugging.
- `progress`: Logical whether to display a progress bar
- `...`: Other parameters to pass to `lme4::glmer()`

Details

This function is a wrapper for `lme4::glmer()`. Wald type 2 Chi-squared test is calculated as per `car::Anova()` optimised for speed. Parallelisation is provided using `parallel::mclapply` on Unix/Mac or `parallel::parLapply` on PC.

Value

Returns an S4 class `GlmmSeq` object with results for gene-wise general linear mixed models. A list of results is returned if `returnList` is TRUE which is useful for debugging.
Examples

```r
data(PEAC_minimal_load)
disp <- apply(tpm, 1, function(x) {
  (var(x, na.rm = TRUE)-mean(x, na.rm = TRUE))/(mean(x, na.rm = TRUE)**2)
})
MS4A1glmm <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  countdata = tpm[1:2, ],
  metadata = metadata,
  dispersion = disp,
  verbose = FALSE)
names(attributes(MS4A1glmm))
```

---

**GlmmSeq-class**

An S4 class to define the glmmSeq output

**Description**

An S4 class to define the glmmSeq output

**Slots**

- `info` List including the matched call, dispersions, offset, designMatrix
- `formula` The model formula
- `stats` Statistics from fitted models
- `predict` Predicted values
- `reducedFormula` The reduced formula with removed random effects
- `countdata` The input expression data with count data in rows
- `metadata` The input metadata
- `modelData` Model data for predictions
- `optInfo` Information on whether the model was singular or converged
- `errors` Any errors
- `vars` List of variables stored from the original call
ImmSeq

Linear mixed models for data matrix

Description

Fits many linear mixed effects models for analysis of gaussian data with random effects, with parallelisation and optimisation for speed. It is suitable for longitudinal analysis of high dimensional data. Wald type 2 Chi-squared test is used to calculate p-values.

Usage

ImmSeq(
  modelFormula,  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  maindata,  # data matrix with genes in rows and samples in columns
  metadata,  # a dataframe of sample information with variables in columns and samples in rows
  id = NULL,  # Optional. Used to specify the column in metadata which contains the sample IDs to be used in repeated samples for random effects. If not specified, the function defaults to using the variable after the "|" in the random effects term in the formula.
  offset = NULL,  # Vector containing model offsets (default = NULL). If provided the \texttt{lmer()} offset is set to offset. See \texttt{lme4::lmer()}
  test.stat = c("Wald", "F"),  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  reducedFormula = "",  # data matrix with genes in rows and samples in columns
  modelData = NULL,  # a dataframe of sample information with variables in columns and samples in rows
  designMatrix = NULL,  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  control = lmerControl(),  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  cores = 1,  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  removeSingles = FALSE,  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  verbose = TRUE,  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  returnList = FALSE,  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
  progress = FALSE,
  ...  # the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
)

Arguments

- \texttt{modelFormula}: data matrix with genes in rows and samples in columns
- \texttt{maindata}: a dataframe of sample information with variables in columns and samples in rows
- \texttt{metadata}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{id}: Optional. Used to specify the column in metadata which contains the sample IDs to be used in repeated samples for random effects. If not specified, the function defaults to using the variable after the "|" in the random effects term in the formula.
- \texttt{offset}: Vector containing model offsets (default = NULL). If provided the \texttt{lmer()} offset is set to offset. See \texttt{lme4::lmer()}
- \texttt{test.stat}: vector containing model offsets (default = NULL). If provided the \texttt{lmer()} offset is set to offset. See \texttt{lme4::lmer()}
- \texttt{reducedFormula}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{modelData}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{designMatrix}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{control}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{cores}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{removeSingles}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{verbose}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{returnList}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}
- \texttt{progress}: the model formula. This must be of the form "~ ..." where the structure is assumed to be "gene ~ ...". The formula must include a random effects term. See formula structure for random effects in \texttt{lme4::lmer()}

...
test.stat  Character value specifying test statistic. Current options are "Wald" for type 2 Wald Chi square test using code derived and modified from \texttt{car::Anova} to improve speed for matrix tests. Or "F" for conditional F tests using Saiterthwaite's method of approximated Df. This uses \texttt{lmerTest::lmer} and is somewhat slower.

reducedFormula  Optional design formula without random effects. If not given, it is automatically generated by removing the random effects from the main formula. Used to calculate confidence intervals for final fitted models on each gene for plotting purposes.

modelData  Optional dataframe. Default is generated by call to \texttt{expand.grid} using levels of variables in the formula. Used to calculate model predictions (estimated means & 95\% CI) for plotting via \texttt{modelPlot}. It can therefore be used to add/remove points in \texttt{modelPlot}.

designMatrix  Optional custom design matrix generated by call to \texttt{model.matrix} using \texttt{modelData} and \texttt{reducedFormula}. Used to calculate model predictions for plotting.

control  the \texttt{lmer} optimizer control (default = \texttt{lmerControl()}). See \texttt{lme4::lmerControl()}.

cores  number of cores to use for parallelisation. Default = 1.

removeSingles  whether to remove individuals with no repeated measures (default = FALSE)

verbose  Logical whether to display messaging (default = TRUE)

returnList  Logical whether to return results as a list or \texttt{lmmSeq} object (default = FALSE). Helpful for debugging.

progress  Logical whether to display a progress bar

...  Other parameters passed to \texttt{lme4::lmer()}

Details
Two key methods are used to speed up computation above and beyond simple parallelisation. The first is to speed up \texttt{lme4::lmer()} by calling \texttt{lme4::lFormula} once at the start and then updating the \texttt{lFormula} output with new data. The 2nd speed up is through optimised code for repeated type 2 Wald Chi-squared tests (original code was derived from \texttt{car::Anova}). For example, elements such as the hypothesis matrices are generated only once to reduce unnecessarily repetitive computation, and the generation of p-values from Chi-squared values is vectorised and performed at the end. F-tests using the \texttt{lmerTest} package have not been optimised and are therefore slower.

Parallelisation is performed using \texttt{parallel::mclapply} on unix/mac and \texttt{parallel::parLapply} on windows. Progress bars use \texttt{pbmcapply::pbmcapply} on unix/mac and \texttt{pbapply::pblapply} on windows.

Value
Returns an S4 class \texttt{lmmSeq} object with results for gene-wise linear mixed models; or a list of results if \texttt{returnList} is TRUE.

Examples
```r
data(PEAC_minimal_load)
logtpm <- log2(tpm + 1)
Immttest <- lmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
                   maindata = logtpm[1:2, ],
```
maPlot

```r
metadata = metadata,
verbose = FALSE)
```

names(attributes(lmmtest))

---

**lmmSeq-class**

An S4 class to define the lmmSeq output

---

**Description**

An S4 class to define the lmmSeq output

**Slots**

- `info` List including matched call, offset, designMatrix
- `formula` The model formula
- `stats` Statistics from fitted models
- `predict` Predicted values
- `reducedFormula` The reduced formula with removed random effects
- `maindata` The input expression data with variables in rows
- `metadata` The input metadata
- `modelData` Model data for predictions
- `optInfo` Information on whether the model was singular or converged
- `errors` Any errors
- `vars` List of variables stored from the original call

---

**maPlot**

MA plots

---

**Description**

MA plots

**Usage**

```r
maPlot(
    object,
    x1var, 
    x2var, 
    x1Values = NULL, 
    x2Values = NULL, 
    pCutoff = 0.01, 
    plotCutoff = 1,
```
maPlot

zeroCountCutoff = 50,
colours = c("grey", "midnightblue", "mediumvioletred", "goldenrod"),
labels = c(),
fontSize = 12,
labelFontSize = 4,
useAdjusted = FALSE,
graphics = "ggplot",
verbose = FALSE
)

Arguments

object A glmmSeq object created by \texttt{glmmSeq::glmmSeq()}
x1var The name of the first (inner) x parameter
x2var The name of the second (outer) x parameter
x1Values Timepoints or categories in \texttt{x1var} to be used to calculate fold change. If \texttt{NULL} the first two levels in \texttt{x1var} are used.
x2Values Categories in \texttt{x2var} to be compared on x and y axis.
pCutoff The significance cut-off for colour-coding (default=0.01)
plotCutoff Which probes to include by significance cut-off (default=1 for all markers)
zeroCountCutoff Which probes to include by minimum counts cut-off (default=50)
colours Vector of colours to use for significance groups
labels Row names or indices to label on plot
fontSize Font size
labelFontSize Font size for labels
useAdjusted whether to use adjusted p-values (must have q-values in \texttt{object})
graphics Either "ggplot" or "plotly"
verbose Whether to print statistics

Value

List of three plots. One plot for each \texttt{x2Value} and one combined figure

Examples

data(PEAC_minimal_load)

disp <- apply(tpm, 1, function(x){
  (var(x, na.rm=TRUE)-mean(x, na.rm=TRUE))/(mean(x, na.rm=TRUE)**2)
})

resultTable <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  countdata = tpm[1:5, ],
  metadata = metadata,
  dispersion = disp)
```r
plots <- maPlot(resultTable, 
    x1var='Timepoint',
    x2var='EULAR_6m',
    x2Values=c('Good', 'Non-response'),
    graphics="plotly")

plots$combined
```

---

### metadata

**Minimal metadata from PEAC**

**Description**

Minimal metadata for paired longitudinal response analysis.

**Usage**

```r
metadata
```

**Format**

A data frame

- **PATID** Id for matching patients
- **Timepoint** timepoints
- **EULAR_6m** response data

---

### modelPlot

**Mixed model effects plot**

**Description**

Plot to show differences between groups over time using base graphics.

**Usage**

```r
modelPlot(
    object, 
    geneName = NULL, 
    x1var = NULL, 
    x2var = NULL, 
    x2shift = NULL, 
    xlab = NA, 
    ylab = geneName, 
    title = geneName,
```

logTransform = is(object, "GlmmSeq"),
shapes = 21,
colours = "grey60",
lineColours = "grey60",
markerSize = 0.5,
fontSize = NULL,
alpha = 0.7,
addModel = TRUE,
addPoints = TRUE,
modelSize = 2,
modelColours = "royalblue",
modellineSize = 1,
modellineColours = modelColours,
errorBarLwd = 2.5,
errorBarLength = 0.05,
...
)

Arguments

object A glmmSeq/lmmSeq object created by glmmSeq::glmmSeq() or glmmSeq::lmmSeq()
geneName The gene/row name to be plotted
x1var The name of the first (inner) x parameter, typically 'time'. This is anticipated to have different values when matched by ID.
x2var The name of an optional second (outer) x parameter, which should be a factor.
x2shift Amount to shift along x axis for each level of x2var. By default the function will arrange each level of x2var side by side. Lower values of x2shift or x2shift=0 can be used to overlap plots similar to 'dodge' or stagger them.
xlab Title for the x axis
ylab Title for the y axis
title Plot title. If NULL gene name is used
logTransform Whether to perform a log10 transform on the y axis
shapes The marker shapes (default=19)
colours The marker colours (default='red') as vector or named vector
lineColours The line colours (default='grey60') as vector or named vector
markerSize Size of markers (default=2)
fontSize Plot font size
alpha Line and marker opacity (default=0.7)
addModel Whether to add the fit model with markers (default=TRUE)
addPoints Whether to add underlying data points (default=TRUE)
modelSize Size of model points (default=2)
modelColours Colour of model fit markers (default="black") as vector or named vector
modellineSize Size of model points (default=1) as vector or named vector
modellineColours
  Colour of model fit lines.
errorBarLwd    Line width of error bars
errorBarLength Head width of error bars
...  Other parameters to pass to `graphics::plot()`

Value

Returns a paired plot for matched samples

Examples

data(PEAC_minimal_load)

disp <- apply(tpm, 1, function(x){
  (var(x, na.rm=TRUE)-mean(x, na.rm=TRUE))/(mean(x, na.rm=TRUE)**2)
})

MS4A1glmm <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  countdata = tpm[1:2, ],
  metadata = metadata,
  dispersion = disp)

modelPlot(object=MS4A1glmm,
  geneName = 'MS4A1',
  x1var = 'Timepoint',
  x2var='EULAR_6m')

---

tpm  

TPM count data from PEAC

Description

Transcripts Per Million (TPM) count data for PEAC synovial biopsies.

Usage

tpm

Format

An object of class `matrix` (inherits from `array`) with 50 rows and 123 columns.
Index

* datasets
  metadata, 14
  tpm, 16
* hplot
  fcPlot, 2
  maPlot, 12

car::Anova, 11
car::Anova(), 8
fcPlot, 2
ggmodelPlot, 4
ggplot2::theme(), 5
glmmQvals, 6
glmmRefit, 6
glmmSeq, 7
GlmmSeq-class, 9
glmmSeq::glmmSeq(), 3, 4, 6, 13, 15
glmmSeq::lmnSeq(), 4, 15
graphics::plot(), 16
lme4::glmer, 7
lme4::glmer(), 6, 8
lme4::glmerControl(), 8
lme4::lFormula, 11
lme4::lmer, 7
lme4::lmer(), 10, 11
lme4::lmerControl(), 11
lmerTest::lmer, 11
lmmSeq, 10
lmmSeq-class, 12

maPlot, 12
metadata, 14
modelPlot, 8, 11, 14

parallel::mclapply, 8, 11
parallel::parLapply, 8, 11
pbapply::pblapply, 11
pbmcapply::pbmclapply, 11