Package ‘glmmSeq’

March 30, 2021

**Title**  General Linear Mixed Models for Gene-Level Differential Expression

**Version**  0.1.0

**Description**  Using random and fixed effects to model expression at an individual gene level can highlight differences between sample groups over time. The most widely used differential gene expression tools are unable to fit linear mixed effect models, therefore do not capture interaction terms. This package uses negative binomial mixed effects models to fit gene expression with matched samples. This is particularly useful for investigating changes in gene expression between groups of individuals over time, as seen in: Rivellese F., Surace A.E.A., Goldmann K., Sciaccia E., Giorli G., Cubuk C., John C.R., Nerviani A., Fossati-Jimack L., Thorbom G., Humby F., Bombardieri M., Lewis M.J., Pitzalis C. (2021) “Molecular Pathology Profiling of Synovial Tissue Predicts Response to Biologic Treatment in Rheumatoid Arthritis” [Manuscript in preparation].

**License**  MIT + file LICENSE

**Encoding**  UTF-8

**LazyData**  true

**RoxygenNote**  7.1.1

**Language**  en-gb

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

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

**Suggests**  knitr, rmarkdown, kableExtra, edgeR

**VignetteBuilder**  knitr

**Depends**  R (>= 3.6.0)

**Imports**  MASS, car, stats, gghalves, ggplot2, ggpubr, graphics, lme4, 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 Sciaccia [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(
  glmmResult,
  x1Label,
  x2Label,
  x1Values = NULL,
  x2Values = NULL,
  pCutoff = 0.01,
  labels = c(),
  useAdjusted = FALSE,
  plotCutoff = 1,
  graphics = "ggplot",
  fontSize = 12,
  labelFontSize = 5,
  colours = c("grey", "goldenrod1", "red", "blue"),
  verbose = FALSE
)

Arguments

- `glmmResult` A glmmSeq object created by `glmmSeq::glmmSeq()`.
- `x1Label` The name of the first (inner) x parameter
- `x2Label` The name of the second (outer) x parameter
- `x1Values` Subpopulations in `x1Label` to be used to calculate fold change. If NULL the first two levels in `x1Label` are used.
- `x2Values` Subpopulations in `x2Label` 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 pvalues (must have q_ columns in `glmmResult`). 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

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)
})

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

fcPlot(glmmResult = glmmFit,
    x1Label = "Timepoint",
    x2Label = "EULAR_6m",
    x2Values = c("Good responder", "Non responder"),
    pCutoff = 0.05,
    useAdjusted = FALSE,
    plotCutoff = 1,
    graphics = "plotly")
```
glmmGene

Glmm for sequencing results of a single gene

Description

Glmm for sequencing results of a single gene

Usage

```
glmmGene(
  modelFormula,
  countdata,
  gene,
  metadata,
  id,
  dispersion,
  sizeFactors = NULL,
  reducedFormula = "",
  modelData = NULL,
  control = glmerControl(optimizer = "bobyqa"),
  zeroCount = 0.125,
  removeDuplicatedMeasures = FALSE,
  removeSingles = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

- **modelFormula**: the model formula. For more information of formula structure see `lme4::glmer()`.
- **countdata**: the sequencing data
- **gene**: the row name in countdata to be used
- **metadata**: a data frame of sample information
- **id**: Column name in metadata which contains the sample IDs to be used in pairing
- **dispersion**: a numeric for the gene dispersion
- **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**: something something
- **control**: the glmer control (default=glmerControl(optimizer="bobyqa"). For more information see `lme4::glmerControl()`.
- **zeroCount**: numerical value to offset zeroes for the purpose of log (default=0.125)
**glmmQvals**

removeDuplicatedMeasures
   whether to remove duplicated conditions/repeated measurements for a given
time point (default=FALSE).
removeSingles
   whether to remove individuals with only one measurement (default=FALSE)
verbose
   Logical whether to display messaging (default=FALSE)
...
   Other parameters to pass to lme4::glmer().

**Value**

Returns the fit for the general linear mixed model of a single gene

**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)
})
MS4A1fit <- glmmGene(~ Timepoint * EULAR_6m + (1 | PATID),
gene = "MS4A1",
id = "PATID",
countdata = tpm,
metadata = metadata,
dispersion = disp["MS4A1"],
verbose=FALSE)
MS4A1fit

---

**glmmQvals**

*Glmm Sequencing qvalues*

**Description**

Add qvalue columns to the glmmSeq dataframe

**Usage**

glmmQvals(glmmResult, cutoff = 0.05, pi0 = NULL, verbose = TRUE)

**Arguments**

- `glmmResult`: A glmmSeq object created by glmmSeq::glmmSeq().
- `cutoff`: Prints a table showing the number of probes considered significant by the pvalue cut-off (default=0.05)
- `pi0`: It is recommended not to input an estimate of pi0. Experienced users can use their own methodology to estimate the proportion of true nulls or set it equal to 1 for the BH procedure (default = NULL).
- `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),
  id = "PATID",
  countdata = tpm[1:5, ],
  metadata = metadata,
  dispersion = disp[1:5],
  verbose=FALSE)
MS4A1glmm <- glmmQvals(MS4A1glmm, pi0=1)
Arguments

modelFormula  the model formula. For more information of formula structure see `lme4::glmer()`
countdata  the sequencing count data
metadata  a data frame of sample information
id  Column name in metadata which contains the sample IDs to be used in pairing samples
dispersion  a numeric vector of gene dispersion
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  Expanded design matrix
control  the glmer control (default = glmerControl(optimizer = "bobyqa")). For more information see `lme4::glmerControl()`
cores  number of cores to use. Default = 1.
removeDuplicatedMeasures  whether to remove duplicated conditions/repeated measurements for a given time point (default = FALSE).
removeSingles  whether to remove individuals with only one measurement (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).
progress  Logical whether to display a progress bar
...
Other parameters to pass to `lme4::glmer()`

Value

Returns a GlmmSeq object with results for gene-wise general linear mixed models or a list of results if returnList is TRUE.

Examples

data(PEAC_minimal_load)
disp <- apply(tpm, 1, function(x) {
  (var(x, na.rm = TRUE)-mean(x, na.rm = TRUE))/sqrt(mean(x, na.rm = TRUE)**2)
})
MS4A1glmm <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
  id = "PATID",
  countdata = tpm["MS4A1", ],
  metadata = metadata,
  dispersion = disp["MS4A1"],
  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

- formula: The model formula
- stats: the statistics from the glmm fit
- predict: The predicted interception values
- reducedFormula: The reduced formula with removed random effects
- countdata: The input expression data
- metadata: The input metadata
- modelData: the model data for the glmm
- optInfo: Information on whether the model was singular or converged
- errors: Any errors
- variables: The variables used in the formula

maPlot

MA plots

Description

MA plots

Usage

maPlot(
  glmmResult,
  x1Label, 
  x2Label, 
  x1Values = NULL, 
  x2Values = NULL, 
  pCutoff = 0.01, 
  plotCutoff = 1, 
  zeroCountCutoff = 50, 
  colours = c("grey", "midnightblue", "mediumvioletred", "goldenrod"), 
  labels = c(), 
  fontSize = 12, 
  labelFontSize = 5, 
  useAdjusted = FALSE, 
  graphics = "ggplot", 
  verbose = FALSE 
)
Arguments

- `glmmResult`: A glmmSeq object created by `glmmSeq::glmmSeq()`.
- `x1Label`: The name of the first (inner) x parameter.
- `x2Label`: The name of the second (outer) x parameter.
- `x1Values`: Subpopulations in `x1Label` to be used to calculate fold change. If NULL, the first two levels in `x1Label` are used.
- `x2Values`: Subpopulations in `x2Label` 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 pvalues (must have q_ columns in `glmmResult`).
- `graphics`: Either "ggplot" or "plotly".
- `verbose`: Whether to print statistics.

Value

List of three plots. One plot for each `x2Value` and one combined figure.

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)
})

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

plots <- maPlot(resultTable,
  x1Label='Timepoint',
  x2Label='EULAR_6m',
  x2Values=c('Good responder', 'Non responder'),
  graphics="plotly")

plots$combined
```
**metadata**  
*Minimal metadata from PEAC*

### Description

Minimal metadata for paired longitudinal response analysis.

### Usage

```r
metadata
```

### Format

A data frame

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

---

**modelPlot**  
*Model plot*

### Description

Model plots to show the overall differences between groups and over time

### Usage

```r
modelPlot(
  glmmResult,
  geneName,
  x1Label = "Timepoint",
  x2Label,
  xTitle = NULL,
  yTitle = "Gene Expression",
  title = NULL,
  logTransform = FALSE,
  shapes = 21,
  colours = c("blue"),
  x2Offset = 6,
  lineWidth = 1,
  markerSize = 5,
  fontSize = NULL,
  overlap = TRUE,
)```
modelPlot

```
addErrorbars = TRUE,
graphics = "ggplot",
...
```

**Arguments**

- `glmmResult`: A glmmSeq object created by `glmmSeq::glmmSeq()`.
- `geneName`: Gene/row name to plot.
- `x1Label`: The name of the first (inner) x parameter.
- `x2Label`: The name of the second (outer) x parameter.
- `xTitle`: Title for the x axis.
- `yTitle`: 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=21.
- `colours`: The marker colours, default=c('blue').
- `x2Offset`: Vertical adjustment to secondary x-axis (default=6).
- `lineWidth`: Plot line size (default=1).
- `markerSize`: Size of markers (default=5).
- `fontSize`: Plot font size (default=10).
- `overlap`: Logical whether x2Label fits should be plotted overlapping one another (default=TRUE).
- `addErrorbars`: Logical whether to add error bars.
- `graphics`: Which graphic system to use (options = "base" or "ggplot")
- `...`: Other parameters to pass to `graphics::plot()` or `ggplot2::theme()`.

**Value**

Returns a plot with the glmm fit for a given gene/row.

**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)
})
Fit <- glmmSeq(~ Timepoint * EULAR_6m + (1 | PATID),
               id = 'PATID',
               countdata = tpm['ADAM12', ],
               metadata = metadata,
               dispersion = disp,
               verbose=FALSE)
modelPlot(Fit,
          "ADAM12",
```
pairedPlot

**Paired plots**

**Description**

Paired plots to show differences between groups and over time

**Usage**

```r
pairedPlot(
  glmmResult,
  geneName = NULL,
  x1Label = NULL,
  x2Label = NULL,
  IDColumn = "ID",
  xTitle = NULL,
  yTitle = "Gene Expression",
  title = NULL,
  logTransform = FALSE,
  shapes = 21,
  colours = "red",
  lineColour = "grey60",
  markerSize = 2,
  fontSize = NULL,
  alpha = 0.7,
  x2Offset = 6,
  pairedOnly = TRUE,
  graphics = "base",
  addModel = TRUE,
  modelSize = 3,
  modelColour = "black",
  modelLineWidth = 1,
  modelLineColour = "black",
  addBox = FALSE,
  addViolins = TRUE,
  violinWidth = 0.5,
  ...
)
```

pairedPlot

Arguments

- `glmmResult`: A `glmmSeq` object created by `glmmSeq::glmmSeq()`.
- `geneName`: The gene/row name to be plotted.
- `x1Label`: The name of the first (inner) x parameter. This must be able to be paired using the ID.
- `x2Label`: The name of the second (outer) x parameter.
- `IDColumn`: Column name of sample IDs for pairing.
- `xTitle`: Title for the x axis.
- `yTitle`: 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=21).
- `colours`: The marker colours (default='red').
- `lineColour`: The line colours (default='grey60').
- `markerSize`: Size of markers (default=2).
- `fontSize`: Plot font size.
- `alpha`: Line and marker opacity (default=0.7).
- `x2Offset`: Vertical adjustment to secondary x-axis (default=6).
- `pairedOnly`: Logical whether to only plot paired samples (default=TRUE).
- `graphics`: Which graphic system to use (options = "base" or "ggplot").
- `addModel`: Whether to add the fit model with markers (default=TRUE).
- `modelSize`: Size of model points (default=3).
- `modelColour`: Colour of model fit markers (default="black").
- `modelLineSize`: Size of model points (default=1).
- `modelLineColour`: Colour of model fit lines (default="black").
- `addBox`: Logical whether to add boxplots for mean and IQR.
- `addViolins`: Logical whether to add half violin-plots (ggplot only), default=TRUE.
- `violinWidth`: Width of violin plots (default=0.5).
- `...`: Other parameters to pass to `graphics::plot()` or `ggplot2::theme()`.

Value

Returns a paired plot for matched samples.
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),
  id = 'PATID',
  countdata = tpm[,MS4A1],
  metadata = metadata,
  dispersion = disp[,MS4A1],
  removeDuplicatedMeasures=TRUE,
  verbose=FALSE)

pairedPlot(glmmResult=MS4A1glmm,
  geneName = 'MS4A1',
  x1Label = 'Timepoint',
  x2Label='EULAR_6m',
  IDColumn = 'PATID',
  colours = c('skyblue', 'goldenrod1', 'mediumvioletred'),
  graphics = 'base')
```

---

tpm  

**TPM count data from PEAC**

Description

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

Usage

tpm

Format

An object of class `data.frame` with 50 rows and 149 columns.
Index

* datasets
  metadata, 10
  tpm, 14
* hplot
  fcPlot, 2
  maPlot, 8

  fcPlot, 2

  ggplot2::theme(), 11, 13
  glmmGene, 4
  glmmQvals, 5
  glmmSeq, 6
  GlmmSeq-class, 8
  glmmSeq::glmmSeq(), 3, 5, 9, 11, 13
  graphics::plot(), 11, 13

  lme4::glmer(), 4, 5, 7
  lme4::glmerControl(), 4, 7

  maPlot, 8
  metadata, 10
  modelPlot, 10

  pairedPlot, 12

  tpm, 14