

# Package ‘PhenStat’

October 8, 2014

**Type** Package

**Title** Statistical analysis of phenotypic data

**Version** 1.2.0

**Date** 2014-04-02

**Author** Natalja Kurbatova, Natasha Karp, Jeremy Mason

**Maintainer** Natasha Karp <nk3@sanger.ac.uk>

**Depends** R (>= 2.3.0)

**Imports** methods, car, nlme, nortest, vcd, limma

**Suggests** RUnit, BiocGenerics

**Description** Package contains methods for statistical analysis of phenotypic data such as Mixed Models and Fisher Exact Test.

**License** file LICENSE

## R topics documented:

boxplotResidualBatch . . . . .	2
boxplotSexGenotype . . . . .	3
boxplotSexGenotypeBatch . . . . .	4
categoricalBarplot . . . . .	5
checkDataset . . . . .	6
classificationTag . . . . .	8
finalModel . . . . .	9
FisherExactTest . . . . .	11
generateGraphs . . . . .	12
modelFormula . . . . .	13
parserOutputSummary . . . . .	14
PhenList . . . . .	15
PhenList-class . . . . .	17
PhenTestResult . . . . .	18

PhenTestResult-class . . . . .	20
plotResidualPredicted . . . . .	23
qqplotGenotype . . . . .	24
qqplotRandomEffects . . . . .	25
qqplotRotatedResiduals . . . . .	26
scatterplotGenotypeWeight . . . . .	27
startModel . . . . .	28
summaryOutput . . . . .	30
testDataset . . . . .	31
testFinalModel . . . . .	33
vectorOutput . . . . .	34
vectorOutputMatrices . . . . .	35

<b>Index</b>	<b>37</b>
--------------	-----------

---

boxplotResidualBatch *Method "boxplotResidualBatch"*

---

## Description

Graph function for the Mixed Model framework's results. Creates a box plot with residue versus batch split by genotype.

## Usage

```
boxplotResidualBatch(phenTestResult, outputMessages=TRUE)
```

## Arguments

phenTestResult instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

outputMessages flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

## Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

## References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

## See Also

[PhenList](#) and [PhenTestResult](#)

### Examples

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
boxplotResidualBatch(result)
```

---

boxplotSexGenotype      *Method "boxplotSexGenotype"*

---

### Description

Graph function for the phenotypic dataset. Creates a box plot split by sex and genotype.

### Usage

```
boxplotSexGenotype(phenList, depVariable=NULL, graphingName=NULL, outputMessages=TRUE)
```

### Arguments

phenList	instance of the <a href="#">PhenList</a> class; mandatory argument
depVariable	a character string defining the dependent variable of interest; mandatory argument
graphingName	a character string defining the name to be used on the graph for the dependent variable; mandatory argument
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

### See Also

[PhenList](#)

## Examples

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
# box plot for dataset with two sexes: males and females
boxplotSexGenotype(test,
  depVariable="Bone.Mineral.Content",
  graphingName="BMC")

file <- system.file("extdata", "test4.csv", package="PhenStat")
test_1sex <- PhenList(dataset=read.csv(file),
  testGenotype="Mysm1/+")
# box plot for females only dataset
boxplotSexGenotype(test_1sex,
  depVariable="Lean.Mass",
  graphingName="Lean Mass (g)")
```

---

boxplotSexGenotypeBatch

*Method "boxplotSexGenotypeBatch"*

---

## Description

Graph function for the phenotypic dataset. Creates a box plot split by sex, genotype and batch.

Note: the batches are not ordered with time but allow assessment of how the treatment groups lie relative to the normal control variation.

## Usage

```
boxplotSexGenotypeBatch(phenList, depVariable=NULL, graphingName=NULL, outputMessages=TRUE)
```

## Arguments

phenList	instance of the <a href="#">PhenList</a> class; mandatory argument
depVariable	a character string defining the dependent variable of interest; mandatory argument
graphingName	a character string defining the name to be used on the graph for the dependent variable; mandatory argument
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

## Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

## References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

## See Also

[PhenList](#)

## Examples

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
# box plot for dataset with two sexes: males and females
boxplotSexGenotypeBatch(test,
  depVariable="Bone.Mineral.Content",
  graphingName="BMC" )

file <- system.file("extdata", "test4.csv", package="PhenStat")
test_1sex <- PhenList(dataset=read.csv(file),
  testGenotype="Mysm1/+")
# box plot for females only dataset
boxplotSexGenotypeBatch(test_1sex,depVariable="Lean.Mass",
  graphingName="Lean Mass")
```

---

categoricalBarplot      *Method "categoricalBarplot"*

---

## Description

Graph function for the Fisher Exact Test framework's results. Creates stacked bar plot(s) to compare proportions seen in a categorical variable between different genotypes. Graphs are created for all data and also for each sex individually.

## Usage

```
categoricalBarplot(phenTestResult,outputMessages=TRUE)
```

## Arguments

**phenTestResult** instance of the [PhenTestResult](#) class that comes from the function `testDataset`; mandatory argument

**outputMessages** flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#)

**Examples**

```
file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Aff3/Aff3")
result <- testDataset(test,
  depVariable="Thoracic.Processes",
  method="FE")
categoricalBarplot(result)
```

---

checkDataset

*Method "checkDataset"*

---

**Description**

Checks dataset for the minimum required info:

1. Column names should present
2. Genotype column should present
3. Sex column should present
4. Two data points for each Genotype/Sex combination
5. Number of Genotype levels should be 2
6. Number of Sex levels should be 1 or 2
7. Sex levels have to be "Female" and/or "Male"
8. Records with reference genotype should be in the dataset
9. Records with test genotype should be in the dataset

Perform the following additional checks:

- presence of Weight column,
- presence of Batch column (Assay.Date).

Warning given in case of missed data indicating that you can only fit a glm or to use MM equation "withoutWeight".

Function [checkDataset](#) is called from [PhenList](#) function.

### Usage

```
checkDataset(dataset, testGenotype, refGenotype="+/+",
             outputMessages=TRUE, dataset.clean=TRUE)
```

### Arguments

dataset	data frame created from file or from another source; mandatory argument
testGenotype	defines the test genotype to be compared to the reference genotype; mandatory argument
refGenotype	defines the reference genotype; assigned default value is "+/+"
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; assigned default value is TRUE
dataset.clean	flag: "FALSE" value for no modification of the dataset; "TRUE" value to clean dataset if needed; assigned default value is TRUE

### Value

Returns an instance of the [PhenList](#) class.

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

### See Also

[PhenTestResult](#)

### Examples

```
# "checkDataset" is called internally from "PhenList" function
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
                 testGenotype="Sparc/Sparc")
```

---

classificationTag	<i>Method "classificationTag"</i>
-------------------	-----------------------------------

---

### Description

Returns a classification tag to assign a sexual dimorphism assessment of the phenotypic change.

### Usage

```
classificationTag(phenTestResult, userMode = "summaryOutput",
  phenotypeThreshold = 0.01, outputMessages=TRUE)
```

### Arguments

phenTestResult	instance of the <a href="#">PhenTestResult</a> class that comes from the function <a href="#">testDataset</a> ; mandatory argument
userMode	flag: "vectorOutput" a sexual dimorphic classification is assigned with a proviso of later assessing the overall statistical significance; "summaryOutput" the phenotypeThreshold is used to assess the overall statistical significance and then if significant the sexual dimorphic classification determined; defaults to summaryOutput
phenotypeThreshold	a numerical value defining the significance threshold to use in assigning the classification tag; default value 0.01
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

### Value

Returns a classification tag to assign a sexual dimorphism assessment of the phenotypic change.

If you are working interactively with the data, the argument "userMode" set to the value "summaryOutput" will use the "phenotypeThreshold" argument's value to assess statistical significance of the genotype effect and if significant then assign a sexual dimorphic classification. Alternatively, if the "userMode" set to the value "vectorMode", a sexual dimorphic classification will be returned with the MM framework where later you can globally assess whether the variable had a significant genotype effect. With the FE framework and the vectorMode, a NA is returned as the type of the effect cannot be assessed without assessing the statistical significance of the genotype effect.

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason



## References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

## See Also

[PhenTestResult](#)

## Examples

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
                 testGenotype="Sparc/Sparc")
result <- testDataset(test,
                     depVariable="Lean.Mass")
classificationTag(result,
                  userMode="summaryOutput",
                  phenotypeThreshold=0.001)
classificationTag(result,
                  userMode="vectorOutput")
```

---

finalModel	<i>Method "finalModel"</i>
------------	----------------------------

---

## Description

This is an internal function run within MM framework. It completes the final stage of the MM framework, which builds the final model and estimates effects. As an internal function, it doesn't include extensive error testing of inputs. Please use cautiously if calling directly.

Works with [PhenTestResult](#) object created by [startModel](#) function. The creation of MM final model is based on the significance of different fixed effects, depVariable and equation values stored in [PhenTestResult](#) object.

## Usage

```
finalModel(phenTestResult, outputMessages = TRUE)
```

## Arguments

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

`outputMessages` flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

**Value**

Returns results stored in instance of the [PhenTestResult](#) class

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenTestResult](#) and [testDataset](#)

**Examples**

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
# when "testDataset" functions argument "callAll" is set to FALSE
# only "startModel" function is called - the first step of MM framework
result <- testDataset(test,
  depVariable="Lean.Mass",
  equation="withoutWeight",
  callAll=FALSE)
# print out formula that has been created
# result$model.formula.genotype
# print out batch effects significance
# result$model.effect.batch
# change the model
# result <- testDataset(test,
#   depVariable="Lean.Mass",
#   equation="withWeight",
#   callAll=FALSE)
# print out new formula
#result$model.formula.genotype
# run the final model fitting when satisfied with the model
result <- finalModel(result)
```

---

FisherExactTest	<i>Method "FisherExactTest"</i>
-----------------	---------------------------------

---

### Description

The main function of the Fisher Exact Test framework. Creates  $n$  times 2 matrices with record counts, where  $n$  rows represent dependent variable levels and two columns represent genotype levels. Performs Fisher Exact Tests on calculated count matrices.

Three matrices can be created and three tests can be potentially perform depending on the dataset:

- all records together regardless the sex values - combined dataset,
- records where sex value is "Male" (if such exists) - males only dataset,
- records where sex value is "Female" (if such exists) - females only dataset.

Together with count matrices creates percentage matrices, calculates effect sizes and statistics for count matrices. Performs Fisher Exact Tests.

The results (matrices, statistics and Fisher Exact Test outputs) are stored in [PhenTestResult](#) object.

The function is called from "testDataset" function when "method" argument is set to "FE" meaning "Fisher Exact Test".

### Usage

```
FisherExactTest(phenList, depVariable, outputMessages=TRUE)
```

### Arguments

- |                |   |
|----------------|---|
| phenList       | instance of the <a href="#">PhenList</a> class; mandatory argument  |
| depVariable    | a character string defining the dependent variable of interest; mandatory argument                        |
| outputMessages | flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE |

### Value

Returns results stored in instance of the [PhenTestResult](#) class

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

- Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410
- West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenTestResult](#)**Examples**

```

file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Aff3/Aff3")
# "FisherExactTest" function is called from "testDataset" function.
result <- testDataset(test,depVariable="Thoracic.Processes",method="FE")
# Fisher Exact Test results can be printed out using function "summaryOutput"
summaryOutput(result)

```

---

generateGraphs	<i>Method "generateGraphs"</i>
----------------	--------------------------------

---

**Description**

Function generates graphs for the Mixed Model framework's results and stores generated graphs in the defined directory.

**Usage**

```

generateGraphs(phenTestResult, dir,
  graphingName=NULL, type="Xlib")

```

**Arguments**

phenTestResult	instance of the <a href="#">PhenTestResult</a> class that comes from the function <a href="#">testDataset</a> ; mandatory argument
dir	directory where to store generated graphs; mandatory argument
graphingName	a character string defining the name to be used on the graph for the dependent variable; defaults to NULL then the depVariable name will be plotted
type	character string, one of "Xlib" or "quartz" (some OS X builds) or "cairo", or "window", or cairo-png"; mandatory argument

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenTestResult](#)

---

modelFormula	<i>Method "modelFormula"</i>
--------------	------------------------------

---

**Description**

Creates formula for the start Mixed Model based on equation and number of sexes in the data.

**Usage**

```
modelFormula(equation, numberofsexes, depVariable)
```

**Arguments**

equation	a character string defining the equation to use (withWeight or withoutWeight); mandatory argument
numberofsexes	a numerical value for levels of sex in the dataset (1 or 2); mandatory argument
depVariable	a character string defining the dependent variable of interest; mandatory argument

**Value**

Returns formula to use.

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenTestResult](#)**Examples**

```
formula <- modelFormula(equation="withWeight",  
                        numberofsexes=2,  
                        depVariable="Lean.Mass")
```

parserOutputSummary    *Method "parserOutputSummary"*

---

### Description

Parses model output summary and returns it in readable named vector format

### Usage

```
parserOutputSummary(phenTestResult)
```

### Arguments

phenTestResult    instance of the [PhenTestResult](#) class that comes from the function [testDataset](#);  
mandatory argument

### Value

Returns a named vector with model output summary results

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

### See Also

[PhenTestResult](#)

### Examples

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
parserOutputSummary(result)
```

PhenList

*Method "PhenList"***Description**

Function to create data object from the data frame.

In addition to dependent variable column (the variable of interest) mandatory columns that should present in the data frame are "Genotype" and "Sex". The "Assay.Date" column is used to model "Batch" effect if not specified differently. "Weight" column is used to model body weight effect.

Function creates [PhenList](#) object, checks data integrity, renames columns when requested, provides simple statistics about dataset.

Returns an instance of the [PhenList](#) object created from the data file.

Dataset is cleaned to ensure there is a maximum two sex levels and there are exactly two levels for genotype. Data cleaning can be switched off by defining the argument "dataset.clean" as FALSE.

**Usage**

```
PhenList(dataset, testGenotype, refGenotype="+/+", hemiGenotype=NULL,
          outputMessages=TRUE, dataset.clean=TRUE,
          dataset.colname.batch=NULL, dataset.colname.genotype=NULL,
          dataset.colname.sex=NULL, dataset.colname.weight=NULL,
          dataset.values.missingValue=NULL, dataset.values.male=NULL,
          dataset.values.female=NULL)
```

**Arguments**

dataset	data frame created from file or from another source; mandatory argument
testGenotype	defines the test genotype to be compared to the reference genotype; mandatory argument
refGenotype	defines the reference genotype; assigned default value is "+/+"
hemiGenotype	defines the genotype value for hemizygous that will be changed to test genotype value
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; assigned default value is TRUE
dataset.clean	flag: "FALSE" value for no modification of the dataset; "TRUE" value to clean dataset if needed; assigned default value is TRUE
dataset.colname.batch	column name within dataset for the batch effect
dataset.colname.genotype	column name within dataset for the genotype info
dataset.colname.sex	column name within dataset for the sex info

dataset.colname.weight  
column name within dataset for the weight info

dataset.values.missingValue  
value used as missing value in the dataset

dataset.values.male  
value used to label "males" in the dataset

dataset.values.female  
value used to label "females" in the dataset

**Value**

Returns an instance of the [PhenList](#) class.

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#)

**Examples**

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
```

```
file <- system.file("extdata", "test2.csv", package="PhenStat")
test2 <- PhenList(dataset=read.csv(file),
  testGenotype="Arid4a/Arid4a",
  refGenotype="+/+",
  hemiGenotype="Arid4a/+",
  dataset.colname.weight="Weight.Value")
```

```
file <- system.file("extdata", "test3.csv", package="PhenStat")
test3 <- PhenList(dataset=read.csv(file),
  dataset.clean=TRUE,
  dataset.values.female=1,
  dataset.values.male=2,
  testGenotype="Mysm1/+")
```



---

PhenList-class	<i>Class "PhenList"</i>
----------------	-------------------------

---

### Description

A list-based S4 class for storing phenotypic data. Helps to support data integrity checks and statistics calculation. The PhenList object can be created by using function [PhenList](#).

### Explore PhenList object

PhenList object instance contains the following sections:

1. "dataset" where cleaned and checked dataset is stored: `phenList$dataset`
2. "testGenotype" where the genotype level to test is stored: `phenList$testGenotype`
3. "refGenotype" where reference genotype value is stored with default value set to "+/+":  
`phenList$refGenotype`
4. "dataset.stat" where simple dataset statistics is stored: `phenList$dataset.stat`
5. "hemiGenotype" if defined contains the genotype value for hemizygous: `phenList$hemiGenotype`
6. "dataset.clean" flag value is stored, see [PhenList](#) for more details:  
`phenList$dataset.clean`
7. "dataset.colname" if defined contains dataset column names that have been renamed:
  - `phenList$dataset.colname.batch` column name for batch values
  - `phenList$dataset.colname.genotype` column name for genotype values
  - `phenList$dataset.colname.sex` column name for sex values
  - `phenList$dataset.colname.weight` column name for weight values
8. "dataset.values" if defined contains dataset values that have been changed during dataset cleaning process:
  - `phenList$dataset.values.missingValue` value used as missing value in the original dataset
  - `phenList$dataset.values.male` value used to label "males" in the original dataset
  - `phenList$dataset.values.female` value used to label "females" in the original dataset

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

- Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410
- West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenList](#)**Examples**

```

file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
class(test)

file <- system.file("extdata", "test2.csv", package="PhenStat")
test2 <- PhenList(dataset=read.csv(file),
  testGenotype="Arid4a/Arid4a",
  refGenotype="+/+",
  hemiGenotype="Arid4a/+",
  dataset.colname.weight="Weight.Value")
test2$dataset.stat
test2$testGenotype
test2$refGenotype
test2$hemiGenotype
test2$dataset.colname.weight

```

PhenTestResult

*Method "PhenTestResult"***Description**

Creates [PhenTestResult](#) object based on test results or model building results. Internal function for the package. Not build for users to directly call.

**Usage**

```

PhenTestResult(model.output=NULL, model.dataset=NULL, depVariable=NULL, equation="withWeight",
  method="MM", model.effect.batch=NULL,
  model.effect.variance=NULL, model.effect.sex=NULL,
  model.effect.interaction=NULL, model.output.interaction=NULL,
  model.effect.weight=NULL, numberSexes=NULL, pThreshold=0.05,
  model.formula.null=NULL, model.formula.genotype=NULL,
  model.output.genotype.nulltest.pVal=NULL,
  model.output.quality=NULL, model.output.summary=NULL)

```

**Arguments**

model.output	representing the model fit gls object or lme object
model.dataset	dataset used for modeling
depVariable	depending variable in the model
equation	possible values: "withWeight" to include weight effect into model, "withoutWeight" to exclude weight effect from the model

method	possible values: "MM" to work with Mixed Model framework, "FE" to work with Fisher Exact Test framework
model.effect.batch	Result of the test for batch effect significance
model.effect.variance	Result of the test for residual variance effect
model.effect.sex	Result of the test for sex effect significance
model.effect.weight	Result of the test for weight effect significance
model.effect.interaction	Result of the test for genotype by sex interaction significance
model.output.interaction	Interaction test result: p-value
numberSexes	number of possible sex values in the dataset
pThreshold	p-value threshold used to evaluate the effect significance, default value 0.05
model.formula.null	formula for the model without genotype effect
model.formula.genotype	formula for the model with genotype effect
model.output.genotype.nulltest.pVal	Genotype test results: p-value
model.output.quality	vector that contains diagnostic test output for mixed model quality of fit
model.output.summary	named vector that contains summary of the model output

**Value**

Returns an instance of the [PhenTestResult](#) class.

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

- Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410
- West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenTestResult](#)

**Examples**

```

file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
# "testDataset" function calls "PhenTestResult" function internally
result <- testDataset(test,
  equation="withoutWeight",
  depVariable="Lean.Mass")
class(result)

```

---

PhenTestResult-class    *Class "PhenTestResult"*

---

**Description**

A list-based S4 class for storing of model fitting results and other information like dependent variable, method used to build a model, etc. Helps to support operations needed for model fitting process and modelling/testing results storage. In the package PhenTestResult object is created by function `testDataset` which in turn calls internal function `PhenTestResult`.

**Explore PhenTestResult object**

PhenTestResults object instance contains the following sections:

1. "model.dataset" contains cleaned and checked dataset that was used for the modelling/testing:

`phenTestResult$model.dataset`

2. "depVariable" contains dependent variable that was tested: `phenTestResult$depVariable`

3. "method" contains method name that was used for modelling/testing: `phenTestResult$method`

4. "equation" is specific for the Mixed Model framework and contains equation that was used for modelling ("withWeight" or "withoutWeight"):

`phenTestResult$equation`

5. "pThreshold" contains p-value threshold with default value set to 0.05:

`phenTestResult$pThreshold`

6. "numberSexes" contains the number of sexes in the dataset: `phenTestResult$numberSexes`

Modelling/testing results are stored in the sections "model.output", "model.formula" and "model.effect". The last two are specific for Mixed Model framework.

The contents of the result sections depend on framework that was used.

The results of Mixed Models framework:

1. Batch effect significance: `phenTestResult$model.effect.batch`

2. Variance equality: `phenTestResult$model.effect.variance`

3. Weight effect significance: `phenTestResult$model.effect.weight`

4. Sex effect significance: `phenTestResult$model.effect.interaction`

5. Evidence of sex effect (p-value): `phenTestResult$model.output.interaction`

6. Evidence of genotype effect (p-value): `phenTestResult$model.output.genotype.nulltest.pVal`
7. Formula for the final genotype model: `phenTestResult$model.formula.genotype`
8. Formula for the final null model: `phenTestResult$model.formula.null`
9. Model fitting output: `phenTestResult$model.output`
10. Model fitting summary: `summary(phenTestResult$model.output)$tTable`

The results of Fisher Exact Test framework.

Fisher Exact Test framework output contains matrices, statistics about matrices and Fisher Exact Tests results.

1. Count matrices.

1.1 Count matrix for combined dataset: `phenTestResult$model.output$count_matrix_all`

1.2 Count matrix for males only dataset: `phenTestResult$model.output$count_matrix_male`

1.3 Count matrix for female only dataset: `phenTestResult$model.output$count_matrix_female`

2. Percentage matrices.

2.1 Percentage matrix for combined dataset:

`phenTestResult$model.output$percentage_matrix_all`

2.2 Percentage matrix for males only dataset:

`phenTestResult$model.output$percentage_matrix_male`

2.3 Percentage matrix for females only dataset:

`phenTestResult$model.output$percentage_matrix_female`

3. Statistics.

3.1 Statistics for matrix for combined dataset: `phenTestResult$model.output$stat_all`

3.2 Statistics for matrix for males only dataset: `phenTestResult$model.output$stat_male`

3.3 Statistics for matrix for females only dataset: `phenTestResult$model.output$stat_female`

4. Effect sizes.

4.1 Effect size for combined dataset: `phenTestResult$model.output$ES_all`

4.2 Effect size for males only dataset: `phenTestResult$model.output$ES_male`

4.3 Effect size for females only dataset: `phenTestResult$model.output$ES_female`

5. Fisher Exact Test results.

5.1 Results for combined dataset: `phenTestResult$model.output$all`

5.2 Results for males only dataset: `phenTestResult$model.output$male`

5.3 Results for females only dataset: `phenTestResult$model.output$female`

5.4 p-value for combined dataset: `phenTestResult$model.output$all$p.value`

5.5 p-value for males only dataset: `phenTestResult$model.output$male$p.value`

5.6 p-value for females only dataset: `phenTestResult$model.output$female$p.value`

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

## References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

## See Also

[FisherExactTest](#), [startModel](#), [finalModel](#) and [PhenTestResult](#)

## Examples

```
# Mixed Model framework
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
                 testGenotype="Sparc/Sparc")
result <- testDataset(test,
                      equation="withoutWeight",
                      depVariable="Bone.Area")
result$depVariable
result$method
result$equation
result$numberSexes
# Batch effect is significant
result$model.effect.batch
# Variance homogeneity
result$model.effect.variance
# Weight effect is significant
result$model.effect.weight
# Sex effect is significant
result$model.effect.interaction
# Sex effect p-value - the result of the test
result$model.output.interaction
# Genotype effect p-value
result$model.output.genotype.nulltest.pVal
# Final model formula with genotype
result$model.formula.genotype
# Final model formula without genotype
result$model.formula.null
# Final model fitting output
# result$model.output
# Final model fitting summary
# summary(result$model.output)$tTable

# Fisher Exact Test framework
file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
                 testGenotype="Aff3/Aff3")
result <- testDataset(test,
                      depVariable="Thoracic.Processes",
                      method="FE")
```

```

result$depVariable
result$method
result$numberSexes
# Count matrix for all data
result$model.output$count_matrix_all
# Count matrix for males only records
result$model.output$count_matrix_male
# Count matrix for females only records
# result$model.output$count_matrix_female
# Percentage matrix for all data
result$model.output$percentage_matrix_all
# Percentage matrix for males only records
# result$model.output$percentage_matrix_male
# Percentage matrix for females only records
result$model.output$percentage_matrix_female
# Matrix statistics for all data
result$model.output$stat_all
# Matrix statistics for males only records
result$model.output$stat_male
# Matrix statistics for females only records
result$model.output$stat_female
# Effect size for all data
result$model.output$ES_all
# Effect size for males only records
result$model.output$ES_male
# Effect size for females only records
result$model.output$ES_female
# Fisher Exact Test results for all data
result$model.output$all
# p-value for all data
result$model.output$all$p.value

```

---

plotResidualPredicted *Method "plotResidualPredicted"*

---

### Description

Graph function for the Mixed Model framework's results. Creates predicted versus residual values plots split by genotype.

### Usage

```
plotResidualPredicted(phenTestResult, outputMessages=TRUE)
```

### Arguments

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

`outputMessages` flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#) and [PhenTestResult](#)

**Examples**

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
                 testGenotype="Sparc/Sparc")
result <- testDataset(test,
                      depVariable="Lean.Mass")
plotResidualPredicted(result)
```

---

qqplotGenotype	<i>Method "qqplotGenotype"</i>
----------------	--------------------------------

---

**Description**

Graph function for the Mixed Model framework's results. Creates a Q-Q plot of residuals for each genotype.

**Usage**

```
qqplotGenotype(phenTestResult, outputMessages=TRUE)
```

**Arguments**

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

`outputMessages` flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason



## References

- Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410
- West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

## See Also

[PhenList](#) and [PhenTestResult](#)

## Examples

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
qqplotGenotype(result)
```

---

qqplotRandomEffects *Method "qqplotRandomEffects"*

---

## Description

Graph function for the Mixed Model framework's results. Creates a Q-Q plot of blups (best linear unbiased predictions).

## Usage

```
qqplotRandomEffects(phenTestResult, outputMessages=TRUE)
```

## Arguments

- `phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument
- `outputMessages` flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

## Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

## References

- Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410
- West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#) and [PhenTestResult](#)

**Examples**

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
qqplotRandomEffects(result)
```

---

qqplotRotatedResiduals

*Method "qqplotRotatedResiduals"*

---

**Description**

Graph function for the Mixed Model framework's results. Creates a Q-Q plot of rotated residuals.

**Usage**

```
qqplotRotatedResiduals(phenTestResult, outputMessages=TRUE)
```

**Arguments**

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

`outputMessages` flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* **7**(12): e52410. doi:10.1371/journal.pone.0052410

Houseman, E. A., Ryan, L. M., Coull, B. A. (2004): Cholesky residuals for assessing normal errors in a linear model with correlated outcomes. *Journal of the American Statistical Association* **99**:466: pg 383-394. Doi 10.1198

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#) and [PhenTestResult](#)

## Examples

```
file <- system.file("extdata", "test2.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  dataset.colname.weight="Weight.Value",
  testGenotype="Arid4a/Arid4a",
  hemiGenotype="Arid4a/+")
result <- testDataset(test,
  depVariable="C1")
qqplotRotatedResiduals(result)
```

---

scatterplotGenotypeWeight

*Method "scatterplotGenotypeWeight"*

---

## Description

Graph function for the phenotypic dataset. Creates a scatter plot body weight versus dependent variable.

Both a regression line and a loess line (locally weighted line) is fitted for each genotype.

## Usage

```
scatterplotGenotypeWeight(phenList,depVariable=NULL,graphingName=NULL,outputMessages=TRUE)
```

## Arguments

phenList	instance of the <a href="#">PhenList</a> class; mandatory argument
depVariable	a character string defining the dependent variable of interest; mandatory argument
graphingName	a character string defining the name to be used on the graph for the dependent variable; mandatory argument
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

## Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

## References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenList](#)**Examples**

```

file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
scatterplotGenotypeWeight(test,
  depVariable="Bone.Mineral.Content",
  graphingName="BMC" )
file <- system.file("extdata", "test4.csv", package="PhenStat")
test_1sex <- PhenList(dataset=read.csv(file),
  testGenotype="Mysm1/+")
scatterplotGenotypeWeight(test_1sex,
  depVariable="Lean.Mass",
  graphingName="Lean Mass")

```

startModel

*Method "startModel"***Description**

This is an internal function run within MM framework. It completes the testing stage of which effects are significant. As an internal function, it doesn't include extensive error testing of inputs. Please use cautiously if calling directly.

It creates start model and modify it after testing of different hypothesis (the model fixed effects).

The model fixed effects are:

- batch effect (TRUE if batch variation is significant, FALSE if not),
- variance effect (TRUE if residual variances for genotype groups are homogeneous and FALSE if they are heterogeneous),
- interaction effect (TRUE if genotype by sex interaction is significant),
- sex effect (TRUE if sex is significant),
- weight effect (TRUE if weight is significant).

If user would like to assign other TRUE/FALSE values to the fixed effects of the model then he or she has to define keepList argument which is vector of TRUE/FALSE values.

If user has defined model fixed effects (keepList argument) then function prints out calculated and user defined effects (only when outputMessages argument is set to TRUE), checks user defined effects for consistency (for instance, if there are no "Weight" column in the dataset then weight effect can't be assigned to TRUE, etc.) and modifies start model according to user defined effects.

As the result PhenTestResult object that contains calculated or user defined model fixed effects and MM start model is created.

**Usage**

```
startModel(phenList, depVariable,
           equation="withWeight", outputMessages=TRUE,
           pThreshold=0.05, keepList=NULL)
```

**Arguments**

phenList	instance of the <a href="#">PhenList</a> class; mandatory argument
depVariable	a character string defining the dependent variable of interest; mandatory argument
equation	a character string defining the equation to use. Possible values "withWeight" (default),"withoutWeight"
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE
pThreshold	a numerical value for the p-value threshold to use in tests, default value 0.05
keepList	a logical vector defining the significance of different model effects: keep_batch, keep_equalvar, keep_weight, keep_sex, keep_interaction; default value NULL

**Value**

Returns results stored in instance of the [PhenTestResult](#) class

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#)

**Examples**

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
                testGenotype="Sparc/Sparc")
# when "testDataset" functions argument "callAll" is set to FALSE
# only "startModel" function is called - the first step of MM framework
result <- testDataset(test,
                     depVariable="Lean.Mass",
                     callAll=FALSE)
# print out formula that has been created
```

```
result$model.formula.genotype
# print out batch effects significance
result$model.effect.batch
# change the model
result <- testDataset(test,
  depVariable="Lean.Mass",
  equation="withWeight",
  callAll=FALSE)
# print out new formula
result$model.formula.genotype
```

---

summaryOutput	<i>Method "summaryOutput"</i>
---------------	-------------------------------

---

### Description

Wrapper for the model fitting results. Returns model fitting and testing results in a user friendly format.

### Usage

```
summaryOutput(phenTestResult, phenotypeThreshold = 0.01)
```

### Arguments

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

`phenotypeThreshold` a numerical value defining the threshold to use in `classificationTag`; ; default value 0.01

### Value

Returns model fitting results in a text format for the screen

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

### See Also

[PhenTestResult](#)

**Examples**

```

# Mixed Model framework
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
summaryOutput(result)

# Fishe Exact Test framework
file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test2 <- PhenList(dataset=read.csv(file),
  testGenotype="Aff3/Aff3")
result2 <- testDataset(test2,
  depVariable="Thoracic.Processes",
  method="FE")
summaryOutput(result2)

```

---

testDataset	<i>Method "testDataset"</i>
-------------	-----------------------------

---

**Description**

Performs the following checks for dependent variable:

1. "depVariable" column should present in the dataset
2. "depVariable" should be numeric for Mixed Model (MM) framework, otherwise performs Fisher Exact Test (FE)
3. Each one genotype level should have more than one "depVariable" level (variability) for MM framework, otherwise recommends FE framework.

Runs the stages of selected framework to analyse dependent variable.

For instance, runs startModel and finalModel for the MM framework if the argument "callAll" is set to TRUE.

If framework contains only one stage like in Fisher Exact Test case then runs that one stage regardless "callAll" value.

**Usage**

```

testDataset(phenList=NULL, depVariable=NULL,
  equation="withWeight", outputMessages=TRUE,
  pThreshold=0.05, method="MM", callAll=TRUE, keepList=NULL, dataPointsThreshold=4)

```

**Arguments**

phenList	instance of the <a href="#">PhenList</a> class; mandatory argument
depVariable	a character string defining the dependent variable of interest; mandatory argument

equation	a character string defining the equation to use. Possible values "withWeight" (default),"withoutWeight"
outputMessages	flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE
pThreshold	a numerical value for the p-value threshold to use in tests, default value 0.05
method	a character string ("MM" or "FE") defining the method to use for model building, default value "MM" mixed model
callAll	flag: "FALSE" value to run only the first stage of the selected framework; "TRUE" value (default) to run all stages of framework one after another
keepList	a logical vector defining the significance of different model effects: keep_batch, keep_equalvar, keep_weight, keep_sex, keep_interaction; default value NULL
dataPointsThreshold	threshold for the number of data points in the MM framework

**Value**

Returns results stored in instance of the [PhenTestResult](#) class

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

- Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410
- West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**

[PhenList](#)

**Examples**

```
# Mixed Model framework
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
# print out formula that has been created
result$model.formula.genotype
summaryOutput(result)

# Mixed Model framework with user defined effects
user_defined_effects <- c(keep_batch=TRUE,
  keep_equalvar=TRUE,
  keep_weight=TRUE,
```



```

        keep_sex=TRUE,
        keep_interaction=TRUE)
result3 <- testDataset(test,
  depVariable="Lean.Mass",
  keepList=user_defined_effects)
# print out formula that has been created
result3$model.formula.genotype
summaryOutput(result3)

# Fisher Exact Test framework
file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test2 <- PhenList(dataset=read.csv(file),
  testGenotype="Aff3/Aff3")
result2 <- testDataset(test2,
  depVariable="Thoracic.Processes",
  method="FE")
summaryOutput(result2)

```

---

testFinalModel	<i>Method "testFinalModel"</i>
----------------	--------------------------------

---

### Description

Performs diagnostic tests for Mixed Model quality of fit. There are no arguments checks assuming that function is called internally from the "finalModel" function. Otherwise should be used with precaution.

### Usage

```
testFinalModel(phenTestResult)
```

### Arguments

phenTestResult instance of the [PhenTestResult](#) class that comes from the function testDataset ; mandatory argument

### Value

Returns a vector with model fitting results

### Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

### References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410  
 West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenTestResult](#)**Examples**

```
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
testFinalModel(result)
```

---

`vectorOutput`*Method "vectorOutput"*

---

**Description**

Wrapper for the model fitting results. Returns model fitting and testing results in a vector format. Assumes that modelling results are stored in the [PhenTestResult](#) object (output from function [testDataset](#)).

**Usage**

```
vectorOutput(phenTestResult)
```

**Arguments**

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

**Value**

Returns a vector with model fitting results: model output summary and other values

**Author(s)**

Natalja Kurbatova, Natasha Karp, Jeremy Mason

**References**

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

**See Also**[PhenTestResult](#)

## Examples

```
# Mixed Model framework
file <- system.file("extdata", "test1.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Sparc/Sparc")
result <- testDataset(test,
  depVariable="Lean.Mass")
vectorOutput(result)

# Fishe Exact Test framework
file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test2 <- PhenList(dataset=read.csv(file),
  testGenotype="Aff3/Aff3")
result2 <- testDataset(test2,
  depVariable="Thoracic.Processes",
  method="FE")
vectorOutput(result2)
```

---

vectorOutputMatrices *Method "vectorOutputMatrices"*

---

## Description

Returns count matrices from Fisher Exact Test framework in a vector form.

Note: Maximal supported number of dependent variable levels is 10.

## Usage

```
vectorOutputMatrices(phenTestResult, outputMessages=TRUE)
```

## Arguments

`phenTestResult` instance of the [PhenTestResult](#) class that comes from the function [testDataset](#); mandatory argument

`outputMessages` flag: "FALSE" value to suppress output messages; "TRUE" value to show output messages; default value TRUE

## Value

Returns a vector with count values.

## Author(s)

Natalja Kurbatova, Natasha Karp, Jeremy Mason

## References

Karp N, Melvin D, Sanger Mouse Genetics Project, Mott R (2012): Robust and Sensitive Analysis of Mouse Knockout Phenotypes. *PLoS ONE* 7(12): e52410. doi:10.1371/journal.pone.0052410

West B, Welch K, Galecki A (2007): Linear Mixed Models: A practical guide using statistical software *New York: Chapman & Hall/CRC* 353 p.

## See Also

[PhenTestResult](#)

## Examples

```
file <- system.file("extdata", "test_categorical.csv", package="PhenStat")
test <- PhenList(dataset=read.csv(file),
  testGenotype="Aff3/Aff3")
result <- testDataset(test,
  depVariable="Thoracic.Processes",
  method="FE")
vectorOutputMatrices(result)
```

# Index

## \*Topic **classes**

- PhenList-class, [17](#)
- PhenTestResult-class, [20](#)
  
- boxplotResidualBatch, [2](#)
- boxplotSexGenotype, [3](#)
- boxplotSexGenotypeBatch, [4](#)
  
- categoricalBarplot, [5](#)
- checkDataset, [6](#), [7](#)
- classificationTag, [8](#)
  
- finalModel, [9](#), [22](#)
- FisherExactTest, [11](#), [22](#)
  
- generateGraphs, [12](#)
  
- modelFormula, [13](#)
  
- parserOutputSummary, [14](#)
- PhenList, [2–7](#), [11](#), [15](#), [15–18](#), [24–29](#), [31](#), [32](#)
- PhenList-class, [17](#)
- PhenTestResult, [2](#), [5](#), [7–14](#), [18](#), [18–20](#),  
[22–26](#), [29](#), [30](#), [32–36](#)
- PhenTestResult-class, [20](#)
- plotResidualPredicted, [23](#)
  
- qqplotGenotype, [24](#)
- qqplotRandomEffects, [25](#)
- qqplotRotatedResiduals, [26](#)
  
- scatterplotGenotypeWeight, [27](#)
- startModel, [9](#), [22](#), [28](#)
- summaryOutput, [30](#)
  
- testDataset, [2](#), [8–10](#), [12](#), [14](#), [20](#), [23–26](#), [30](#),  
[31](#), [34](#), [35](#)
- testFinalModel, [33](#)
  
- vectorOutput, [34](#)
- vectorOutputMatrices, [35](#)