

# Package ‘OmicsLonDA’

October 16, 2019

**Type** Package

**Title** Omics Longitudinal Differential Analysis

**Version** 1.0.0

**Date** 2019-03-29

**Author** Ahmed A. Metwally, Tom Zhang, Michael Snyder

**Maintainer** Ahmed A. Metwally <ametwall@stanford.edu>

**URL** <https://github.com/aametwally/OmicsLonDA>

**BugReports** <https://github.com/aametwally/OmicsLonDA/issues>

**Description** Statistical method that provides robust identification of time intervals where omics features (such as proteomics, lipidomics, metabolomics, transcriptomics, microbiome, as well as physiological parameters captured by wearable sensors such as heart rhythm, body temperature, and activity level) are significantly different between groups.

**License** MIT + file LICENSE

**Depends** R(>= 3.6)

**Imports** SummarizedExperiment, gss, plyr, zoo, pracma, ggplot2, BiocParallel, parallel, grDevices, graphics, stats, utils, methods, BiocGenerics

**Suggests** knitr, rmarkdown, testthat, devtools, BiocManager

**biocViews** TimeCourse, Survival, Microbiome, Metabolomics, Proteomics, Lipidomics, Transcriptomics, Regression

**NeedsCompilation** no

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/OmicsLonDA>

**git\_branch** RELEASE\_3\_9

**git\_last\_commit** 04ba038

**git\_last\_commit\_date** 2019-05-02

**Date/Publication** 2019-10-15

**R topics documented:**

adjustBaseline . . . . .	2
curveFitting . . . . .	3
findSigInterval . . . . .	4
omicslonda . . . . .	4
omicslonda_data_example . . . . .	6
permutationMC . . . . .	6
testStat . . . . .	7
testStatPermutation . . . . .	8
visualizeArea . . . . .	9
visualizeFeature . . . . .	10
visualizeFeatureSpline . . . . .	11
visualizeTestStatHistogram . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

adjustBaseline	<i>Adjust for baseline differences using Centered-Log Ratio</i>
----------------	---

---

**Description**

Adjust for baseline differences using Centered-Log Ratio

**Usage**

```
adjustBaseline(se_object = NULL)
```

**Arguments**

se\_object      SummarizedExperiment object contains omics count/level matrix and metadata

**Value**

a SummarizedExperiment object with the adjusted baseline assay

**References**

Ahmed Metwally (ametwall@stanford.edu)

**Examples**

```
data(omicslonda_data_example)
adjusted_se = adjustBaseline(omicslonda_data_example$omicslonda_se_object)
```

---

curveFitting

*Fit longitudinal data*


---

## Description

Fits longitudinal samples from the same group using negative binomial smoothing splines or LOWESS

## Usage

```
curveFitting(formula = Count ~ Time, df = "NULL",
             fit.method = "ssgaussian", points = NULL)
```

## Arguments

formula	formula to be passed to the regression model
df	dataframe has the Count, Group, Subject, Time
fit.method	fitting method (ssgaussian)
points	points at which the prediction should happen

## Value

a list that contains fitted smoothing spline for each group along with 95

## References

Ahmed Metwally (ametwall@stanford.edu)

## Examples

```
library(SummarizedExperiment)
data("omicslonda_data_example")
omicslonda_se_object_adjusted = adjustBaseline(
  se_object = omicslonda_data_example$omicslonda_se_object)
se_object = omicslonda_se_object_adjusted[1,]
dt = data.frame(colData(se_object))
dt$Count = as.vector(assay(se_object))
Group = as.character(dt$Group)
group.levels = sort(unique(Group))
gr.1 = as.character(group.levels[1])
gr.2 = as.character(group.levels[2])
df = dt
levels(df$Group) = c(levels(df$Group), "0", "1")
df$Group[which(df$Group == gr.1)] = 0
df$Group[which(df$Group == gr.2)] = 1
group.0 = df[df$Group == 0, ]
group.1 = df[df$Group == 1, ]
points = seq(1, 500)
model = curveFitting(formula = Count ~ Time, df = df, fit.method = "ssgaussian", points = points)
```

---

findSigInterval	<i>Find Significant Interval based on testStat</i>
-----------------	--

---

**Description**

Find Significant Interval based on testStat

**Usage**

```
findSigInterval(adjusted.pvalue, threshold = 0.05, sign)
```

**Arguments**

adjusted.pvalue	vector of the adjusted p-value
threshold	p-value cut off
sign	vector hold area sign of each time interval

**Value**

returns a list of the start and end points of all significant time intervals

**References**

Ahmed Metwally (ametwall@stanford.edu)

**Examples**

```
library(SummarizedExperiment)
padjusted = abs(rnorm(10, mean = 0.05, sd = 0.04))
sign = sample(x = c(1,-1), 10, replace = TRUE)
intervals = findSigInterval(adjusted.pvalue = padjusted,
                           threshold = 0.05, sign = sign)
```

---

omicslonda	<i>Omics Longitudinal Differential Analysis for one feature</i>
------------	---

---

**Description**

Find significant time intervals of omic feature

**Usage**

```
omicslonda(se_object = NULL, n.perm = 500, fit.method = "ssgaussian",
           points = NULL, text = "FeatureName", parall = FALSE,
           pvalue.threshold = 0.05, adjust.method = "BH", time.unit = "days",
           ylabel = "Normalized Count", col = c("blue", "firebrick"),
           prefix = "Test")
```

**Arguments**

<code>se_object</code>	SummarizedExperiment object contains omics count/level matrix and metadata contains (subject, time, group, and any any other covariates)
<code>n.perm</code>	number of permutations.
<code>fit.method</code>	fitting method (ssgaussian).
<code>points</code>	points at which the prediction should happen.
<code>text</code>	Feature's name.
<code>parall</code>	boolean to indicate whether to use multicore.
<code>pvalue.threshold</code>	p-value threshold cutoff for identifying significant time intervals.
<code>adjust.method</code>	multiple testing correction method.
<code>time.unit</code>	time unit used in the Time vector (hours, days, weeks, months, etc.)
<code>ylabel</code>	text to be shown on the y-axis of all generated figures (default: "Normalized Count")
<code>col</code>	two color to be used for the two groups (eg., c("red", "blue")).
<code>prefix</code>	prefix to be used to create directory for the analysis results

**Value**

a list of the significant time intervals for the tested feature, fitted model for each group, null distribution of the test statistic of the tested feature, and the original input data.

**References**

Ahmed Metwally (ametwall@stanford.edu)

**Examples**

```
library(SummarizedExperiment)
data(omicslonda_data_example)
omicslonda_se_object_adjusted = adjustBaseline(
  se_object = omicslonda_data_example$omicslonda_se_object)
omicslonda_test_object = omicslonda_se_object_adjusted[1,]
points = seq(1, 500, length.out = 500)
res = omicslonda(se_object = omicslonda_test_object, n.perm = 10,
  fit.method = "ssgaussian", points = points, text = "Feature_1",
  parall = FALSE, pvalue.threshold = 0.05,
  adjust.method = "BH", time.unit = "days",
  ylabel = "Normalized Count",
  col = c("blue", "firebrick"), prefix = tempfile())
```

---

omicslonda\_data\_example

*Simulated dataset*

---

### Description

Datasets used for examples/testing OmicsLonDA

### Usage

omicslonda\_data\_example

### Format

A data frame with 10 simulated features

---

permutationMC

*Permute group labels*

---

### Description

Permutes the group label of the samples in order to construct the testStatistics empirical distribution

### Usage

```
permutationMC(formula = Count ~ Time, perm.dat = NULL, n.perm = 500,
  fit.method = "ssgaussian", points, parall = FALSE, prefix = "Test")
```

### Arguments

formula	formula to be passed to the regression model
perm.dat	dataframe has the Count, Group, Subject, Time
n.perm	number of permutations
fit.method	The fitting method (ssgaussian)
points	The points at which the prediction should happen
parall	boolean to indicate whether to use multicore.
prefix	prefix to be used to create directory for the analysis results

### Value

a list of the fitted model for each group for all the permutations

### References

Ahmed Metwally (ametwall@stanford.edu)

**Examples**

```

library(SummarizedExperiment)
data("omicslonda_data_example")
omicslonda_se_object_adjusted = adjustBaseline(
  se_object = omicslonda_data_example$omicslonda_se_object)
omicslonda_test_object = omicslonda_se_object_adjusted[1,]
se_object = omicslonda_test_object
dt = data.frame(colData(se_object))
dt$Count = as.vector(assay(se_object))
Group = as.character(dt$Group)
group.levels = sort(unique(Group))
gr.1 = as.character(group.levels[1])
gr.2 = as.character(group.levels[2])
df = dt
levels(df$Group) = c(levels(df$Group), "0", "1")
df$Group[which(df$Group == gr.1)] = 0
df$Group[which(df$Group == gr.2)] = 1
group.0 = df[df$Group == 0, ]
group.1 = df[df$Group == 1, ]
points = seq(100, 130)
perm = permutationMC(formula = Count ~ Time, perm.dat = df, n.perm = 10,
  fit.method = "ssgaussian", points = points,
  parall = FALSE, prefix = tempfile())

```

testStat

*Calculate Test-Statistic of each feature's time interval***Description**

Calculate Test-Statistic of each feature's time interval

**Usage**

```
testStat(curve.fit.df)
```

**Arguments**

curve.fit.df    gss data object of the fitted spline

**Value**

a list that has the test statistic for each time interval

**References**

Ahmed Metwally (ametwall@stanford.edu)

**Examples**

```

library(SummarizedExperiment)
data("omicslonda_data_example")
model = omicslonda_data_example$omicslonda_results$model
stat = testStat(model)$testStat

```

---

testStatPermutation     *Calculate testStat of each feature's time interval for all permutations*

---

### Description

Calculate testStat of each feature's time interval for all permutations

### Usage

```
testStatPermutation(perm)
```

### Arguments

perm                    list has all the permuted models

### Value

a list of test statistic for each time interval for all all permutations

### References

Ahmed Metwally (ametwall@stanford.edu)

### Examples

```
library(SummarizedExperiment)
data("omicslonda_data_example")
omicslonda_se_object_adjusted = adjustBaseline(
  se_object = omicslonda_data_example$omicslonda_se_object)
omicslonda_test_object = omicslonda_se_object_adjusted[1,]
se_object = omicslonda_test_object
dt = data.frame(colData(se_object))
dt$Count = as.vector(assay(se_object))
Group = as.character(dt$Group)
group.levels = sort(unique(Group))
gr.1 = as.character(group.levels[1])
gr.2 = as.character(group.levels[2])
df = dt
levels(df$Group) = c(levels(df$Group), "0", "1")
df$Group[which(df$Group == gr.1)] = 0
df$Group[which(df$Group == gr.2)] = 1
group.0 = df[df$Group == 0, ]
group.1 = df[df$Group == 1, ]
points = seq(100, 130)
perm = permutationMC(formula = Count ~ Time, perm.dat = df, n.perm = 10,
  fit.method = "ssgaussian", points = points,
  parall = FALSE, prefix = tempfile())
test.stat.prem = testStatPermutation(perm)
```



---

visualizeArea	<i>Visualize significant time interval</i>
---------------	--

---

## Description

Visualize significant time interval

## Usage

```
visualizeArea(omicslonda_object = NULL, fit.method = "ssgaussian",
  text = "FeatureName", unit = "days", ylabel = "Normalized Count",
  col = c("blue", "firebrick"), prefix = "Test")
```

## Arguments

omicslonda_object	The returned object from omicslonda analysis
fit.method	Fitting method (ssgaussian)
text	Feature name
unit	time unit used in the Time vector (hours, days, weeks, months, etc.)
ylabel	text to be shown on the y-axis of all generated figures (default: "Normalized Count")
col	two color to be used for the two groups (eg., c("red", "blue")).
prefix	prefix to be used to create directory for the analysis results

## Value

null

## References

Ahmed Metwally (ametwall@stanford.edu)

## Examples

```
library(SummarizedExperiment)
data(omicslonda_data_example)
visualizeArea(omicslonda_object = omicslonda_data_example$omicslonda_results,
  fit.method = "ssgaussian",
  text = "Feature_1", unit = "days",
  ylabel = "Normalized Count", col =
  c("blue", "firebrick"), prefix = tempfile())
```

---

visualizeFeature	<i>Visualize Longitudinal Feature</i>
------------------	---------------------------------------

---

### Description

Visualize Longitudinal Feature

### Usage

```
visualizeFeature(se_object = NULL, text = "featureName",
  unit = "days", ylabel = "Normalized Count", col = c("blue",
  "firebrick"), prefix = "Test")
```

### Arguments

se_object	SummarizedExperiment object contains omics count/level matrix and metadata
text	feature name
unit	time interval unit
ylabel	text to be shown on the y-axis of all generated figures (default: "Normalized Count")
col	two color to be used for the two groups (eg., c("red", "blue")).
prefix	prefix to be used to create directory for the analysis results

### Value

null

### References

Ahmed Metwally (ametwall@stanford.edu)

### Examples

```
library(SummarizedExperiment)
data("omicslonda_data_example")
omicslonda_se_object_adjusted = adjustBaseline(
  se_object = omicslonda_data_example$omicslonda_se_object)
omicslonda_test_object = omicslonda_se_object_adjusted[1,]
visualizeFeature(se_object = omicslonda_test_object, text = "Feature_1",
  unit = "days", ylabel = "Normalized Count",
  col = c("blue", "firebrick"), prefix = tempfile())
```

---

 visualizeFeatureSpline

*Visualize the feature trajectory with the fitted Splines*


---

## Description

Plot the longitudinal features along with the fitted splines

## Usage

```
visualizeFeatureSpline(se_object = NULL, omicslonda_object = NULL,
  fit.method = "ssgaussian", text = "FeatureName", unit = "days",
  ylabel = "Normalized Count", col = c("blue", "firebrick"),
  prefix = "Test")
```

## Arguments

se_object	SummarizedExperiment object contains omics count/level matrix and metadata
omicslonda_object	The returned object from omicslonda analysis
fit.method	The fitting method (ssgaussian)
text	feature name
unit	time unit used in Time vector (hours, days, weeks, months, etc.)
ylabel	text to be shown on the y-axis of all generated figures (default: "Normalized Count")
col	two color to be used for the two groups (eg., c("red", "blue")).
prefix	prefix to be used to create directory for the analysis results

## Value

null

## References

Ahmed Metwally (ametwall@stanford.edu)

## Examples

```
library(SummarizedExperiment)
data(omicslonda_data_example)
omicslonda_se_object_adjusted = adjustBaseline(
  se_object = omicslonda_data_example$omicslonda_se_object)
omicslonda_test_object = omicslonda_se_object_adjusted[1,]
points = seq(1, 500, length.out = 500)
res = omicslonda(se_object = omicslonda_test_object, n.perm = 10,
  fit.method = "ssgaussian", points = points, text = "Feature_1",
  parall = FALSE, pvalue.threshold = 0.05,
  adjust.method = "BH", time.unit = "days",
  ylabel = "Normalized Count",
  col = c("blue", "firebrick"), prefix = "OmicsLonDA_example")
```

```
visualizeFeatureSpline(se_object = omicslonda_test_object, omicslonda_object =
  omicslonda_data_example$omicslonda_results, fit.method = "ssgaussian",
  text = "Feature_1", unit = "days",
  ylabel = "Normalized Count",
  col = c("blue", "firebrick"),
  prefix = tempfile())
```

---

visualizeTestStatHistogram

*Visualize test statistics empirical distribution*

---

## Description

Visualize test statistics empirical distribution

## Usage

```
visualizeTestStatHistogram(omicslonda_object = NULL,
  text = "FeatureName", fit.method = "ssgaussian", prefix = "Test")
```

## Arguments

omicslonda_object	The returned object from omicslonda analysis
text	Feature name
fit.method	fitting method
prefix	prefix to be used to create directory for the analysis results

## Value

null

## References

Ahmed Metwally (ametwall@stanford.edu)

## Examples

```
library(SummarizedExperiment)
data(omicslonda_data_example)
visualizeTestStatHistogram(omicslonda_object = omicslonda_data_example$omicslonda_results,
  text = "Feature_1",
  fit.method = "ssgaussian", prefix = tempfile())
```

# Index

## \*Topic **datasets**

omicslonda\_data\_example, 6

adjustBaseline, 2

curveFitting, 3

findSigInterval, 4

omicslonda, 4

omicslonda\_data\_example, 6

permutationMC, 6

testStat, 7

testStatPermutation, 8

visualizeArea, 9

visualizeFeature, 10

visualizeFeatureSpline, 11

visualizeTestStatHistogram, 12