

# Package ‘countTransformers’

March 20, 2019

**Type** Package

**Title** Transform Counts in RNA-Seq Data Analysis

**Version** 0.0.6

**Date** 2019-03-20

**Maintainer** Zeyu Zhang <zhzyvv@gmail.com>

**Depends** R (>= 3.4.0), Biobase, limma

**Imports** MASS, graphics, stats

**biocViews** Bioinformatics, DifferentialExpression

**Description** Provide data transformation functions to transform counts in RNA-seq data analysis. Please see the reference: Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. (2019) <[doi.org/10.1038/s41598-019-41315-w](https://doi.org/10.1038/s41598-019-41315-w)>.

**License** GPL (>= 2)

**NeedsCompilation** no

**Author** Zeyu Zhang [aut, cre],  
Danyang Yu [aut, ctb],  
Minseok Seo [aut, ctb],  
Craig P. Hersh [aut, ctb],  
Scott T. Weiss [aut, ctb],  
Weiliang Qiu [aut, ctb]

**Repository** CRAN

**Date/Publication** 2019-03-20 12:56:43 UTC

## R topics documented:

es . . . . .	2
getJaccard . . . . .	3
l2Transformer . . . . .	4
lTransformer . . . . .	6
lv2Transformer . . . . .	7
lvTransformer . . . . .	9

r2Transformer . . . . .	11
rTransformer . . . . .	12
rv2Transformer . . . . .	14
rvTransformer . . . . .	16
wilcoxWrapper . . . . .	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

es	<i>A Simulated Data Set</i>
----	-----------------------------

---

## Description

A simulated data set based on the R code provided by Law et al.'s (2014) paper.

## Usage

```
data("es")
```

## Format

The format is: Formal class 'ExpressionSet' [package "Biobase"]

## Details

The simulated data set contains RNA-seq counts of 1000 genes for 6 samples (3 cases and 3 controls). The library sizes of the 6 samples are not equal.

## Source

The dataset was generated based on the R code Simulation\_Full.R from the website <http://bioinf.wehi.edu.au/voom/>.

## References

Law CW, Chen Y, Shi W, Smyth GK. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*. 2014; 15:R29

## Examples

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])
```

```
# phenotype data
pDat = pData(es)
print(dim(pDat))
print(pDat[1:2,])

# feature data
fDat = fData(es)
print(dim(fDat))
print(fDat[1:2,])
```

---

`getJaccard`*Calculate Jaccard Index for Two Binary Vectors*

---

**Description**

Calculate Jaccard index for two binary vectors.

**Usage**

```
getJaccard(c11, c12)
```

**Arguments**

<code>c11</code>	n by 1 binary vector of classification 1 for the n subjects
<code>c12</code>	n by 1 binary vector of classification 2 for the n subjects

**Details**

Jaccard Index is defined as the ratio

$$d/(b + c + d)$$

, where  $d$  is the number of subjects who were classified to group 1 by both classification rules,  $b$  is the number of subjects who were classified to group 1 by classification rule 1 and were classified to group 0 by classification rule 2,  $c$  is the number of subjects who were classified to group 0 by classification rule 1 and were classified to group 1 by classification rule 2.

**Value**

The Jaccard Index

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

**Examples**

```

n = 10
set.seed(1234567)

# generate two random binary vector of size n
c11 = sample(c(1,0), size = n, prob = c(0.5, 0.5), replace = TRUE)
c12 = sample(c(1,0), size = n, prob = c(0.5, 0.5), replace = TRUE)
cat("\n2x2 contingency table >>\n")
print(table(c11, c12))

JI = getJaccard(c11, c12)
cat("Jaccard index = ", JI, "\n")

```

---

l2Transformer

*Log Based Count Transformation Minimizing Sum of Sample-Specific Squared Difference*


---

**Description**

Log based count transformation minimizing sum of sample-specific squared difference.

**Usage**

```
l2Transformer(mat, low = 1e-04, upp = 1000)
```

**Arguments**

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
low	lower bound for the model parameter
upp	upper bound for the model parameter

**Details**

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the log transformation

$$y_{gi} = \log_2 \left( x_{gi} + \frac{1}{\delta} \right)$$

. The optimal value for the parameter  $\delta$  is to minimize the sum of the squared difference between the sample mean and the sample median across  $n$  subjects

$$\sum_{i=1}^n (\bar{y}_i - \tilde{y}_i)^2$$

,  $\bar{y}_i = \sum_{g=1}^G y_{gi}/G$  and  $\tilde{y}_i$  is the median of  $y_{1i}, \dots, y_{Gi}$ , and where  $G$  is the number of genes and  $n$  is the number of subjects.

**Value**

A list with 3 elements:

res.delta	An object returned by optimize function
delta	model parameter
mat2	transformed data matrix having the same dimension as mat

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

**Examples**

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")

res = l2Transformer(mat = ex)

# estimated model parameter
print(res$delta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")
```

---

lTransformer

*Log-based transformation*


---

**Description**

Log-based transformation.

**Usage**

```
lTransformer(mat, low = 1e-04, upp = 100)
```

**Arguments**

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
low	lower bound for the model parameter
upp	upper bound for the model parameter

**Details**

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the log transformation

$$y_{gi} = \log_2 \left( x_{gi} + \frac{1}{\delta} \right)$$

. The optimal value for the parameter  $\delta$  is to minimize the squared difference between the sample mean and the sample median of the pooled data  $y_{gi}$ ,  $g = 1, \dots, G$ ,  $i = 1, \dots, n$ , where  $G$  is the number of genes and  $n$  is the number of subjects.

**Value**

A list with 3 elements:

res.delta	An object returned by optimize function
delta	model parameter
mat2	transformed data matrix having the same dimension as mat

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

**Examples**

```

library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")

res = lv2Transformer(mat = ex)

# estimated model parameter
print(res$delta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")

```

---

lv2Transformer	<i>Log and VOOM Based Count Transformation Minimizing Sum of Sample-Specific Squared Difference</i>
----------------	---

---

**Description**

Log and VOOM based count transformation minimizing sum of sample-specific squared difference.

**Usage**

```
lv2Transformer(mat, lib.size = NULL, low = 0.001, upp = 1000)
```

**Arguments**

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
lib.size	By default, lib.size is a vector of column sums of mat
low	lower bound for the model parameter
upp	upper bound for the model parameter

**Details**

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the log transformation

$$y_{gi} = \log_2 \left( t_{gi} + \frac{1}{\delta} \right)$$

, where

$$t_{gi} = \frac{(x_{gi} + 0.5)}{X_i + 1} \times 10^6$$

and  $X_i = \sum_{g=1}^G x_{gi}$  is the column sum for the  $i$ -th column of the matrix `mat`. The optimal value for the parameter  $\delta$  is to minimize the sum of the squared difference between the sample mean and the sample median across  $n$  subjects

$$\sum_{i=1}^n (\bar{y}_i - \tilde{y}_i)^2$$

,  $\bar{y}_i = \sum_{g=1}^G y_{gi}/G$  and  $\tilde{y}_i$  is the median of  $y_{1i}, \dots, y_{Gi}$ , and where  $G$  is the number of genes and  $n$  is the number of subjects.

**Value**

A list with 3 elements:

<code>res.delta</code>	An object returned by optimize function
<code>delta</code>	model parameter
<code>mat2</code>	transformed data matrix having the same dimension as <code>mat</code>

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

**Examples**

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
```



```

md = median(vec)
diff = m - md
cat("m=", m, ", ", md=", md, ", ", diff=", diff, "\n")

res = lv2Transformer(mat = ex)

# estimated model parameter
print(res$delta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", ", md2=", md2, ", ", diff2=", diff2, "\n")

```

lvTransformer

*Log and VOOM Transformation***Description**

Log and VOOM Transformation.

**Usage**

```
lvTransformer(mat, lib.size=NULL, low=0.001, upp=1000)
```

**Arguments**

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
lib.size	By default, lib.size is a vector of column sums of mat
low	lower bound for the model parameter
upp	upper bound for the model parameter

**Details**

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the log transformation

$$y_{gi} = \log_2 \left( t_{gi} + \frac{1}{\delta} \right)$$

, where

$$t_{gi} = \frac{(x_{gi} + 0.5)}{X_i + 1} \times 10^6$$

and  $X_i = \sum_{g=1}^G x_{gi}$  is the column sum for the  $i$ -th column of the matrix mat. The optimal value for the parameter  $\delta$  is to minimize the squared difference between the sample mean and the sample median of the pooled data  $y_{gi}$ ,  $g = 1, \dots, G$ ,  $i = 1, \dots, n$ , where  $G$  is the number of genes and  $n$  is the number of subjects.

**Value**

A list with 3 elements:

<code>res.delta</code>	An object returned by optimize function
<code>delta</code>	model parameter
<code>mat2</code>	transformed data matrix having the same dimension as <code>mat</code>

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

**Examples**

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")

res = lvTransformer(mat = ex)

# estimated model parameter
print(res$delta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")
```

---

r2Transformer	<i>Root Based Count Transformation Minimizing Sum of Sample-Specific Squared Difference</i>
---------------	---

---

### Description

Root based count transformation minimizing sum of sample-specific squared difference.

### Usage

```
r2Transformer(mat, low = 1e-04, upp = 1000)
```

### Arguments

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
low	lower bound for the model parameter
upp	upper bound for the model parameter

### Details

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the root and voom transformation

$$y_{gi} = \frac{x_{gi}^{(1/\eta)}}{(1/\eta)}$$

, The optimal value for the parameter  $\eta$  is to minimize the sum of the squared difference between the sample mean and the sample median across  $n$  subjects

$$\sum_{i=1}^n (\bar{y}_i - \tilde{y}_i)^2$$

,  $\bar{y}_i = \sum_{g=1}^G y_{gi}/G$  and  $\tilde{y}_i$  is the median of  $y_{1i}, \dots, y_{Gi}$ , and where  $G$  is the number of genes and  $n$  is the number of subjects.

### Value

A list with 3 elements:

res.delta	An object returned by optimize function
eta	model parameter
mat2	transformed data matrix having the same dimension as mat

### Author(s)

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

## References

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

## Examples

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")

res = r2Transformer(mat = ex)

# estimated model parameter
print(res$eta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")
```

---

rTransformer

*Root Based Transformation*

---

## Description

Root based transformation.

## Usage

```
rTransformer(mat, low = 1e-04, upp = 100)
```

**Arguments**

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
low	lower bound for the model parameter
upp	upper bound for the model parameter

**Details**

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the root transformation

$$y_{gi} = \frac{x_{gi}^{(1/\eta)}}{(1/\eta)}$$

. The optimal value for the parameter  $\eta$  is to minimize the squared difference between the sample mean and the sample median of the pooled data  $y_{gi}$ ,  $g = 1, \dots, G$ ,  $i = 1, \dots, n$ , where  $G$  is the number of genes and  $n$  is the number of subjects.

**Value**

res.eta	An object returned by optimize function
eta	model parameter
mat2	transformed data matrix having the same dimension as mat

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

**Examples**

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")
```

```

res = rTransformer(mat = ex)

# estimated model parameter
print(res$eta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")

```

---

rv2Transformer	<i>Root and VOOM Based Count Transformation Minimizing Sum of Sample-Specific Squared Difference</i>
----------------	--

---

### Description

Root and VOOM based count transformation minimizing sum of sample-specific squared difference.

### Usage

```
rv2Transformer(mat, low = 1e-04, upp = 1000, lib.size = NULL)
```

### Arguments

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
lib.size	By default, lib.size is a vector of column sums of mat
low	lower bound for the model parameter
upp	upper bound for the model parameter

### Details

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the root and voom transformation

$$y_{gi} = \frac{t_{gi}^{(1/\eta)}}{(1/\eta)}$$

, where

$$t_{gi} = \frac{(x_{gi} + 0.5)}{X_i + 1} \times 10^6$$

and  $X_i = \sum_{g=1}^G x_{gi}$  is the column sum for the  $i$ -th column of the matrix mat. The optimal value for the parameter  $\eta$  is to minimize the sum of the squared difference between the sample mean and the sample median across  $n$  subjects

$$\sum_{i=1}^n (\bar{y}_i - \tilde{y}_i)^2$$

,  $\bar{y}_i = \sum_{g=1}^G y_{gi}/G$  and  $\tilde{y}_i$  is the median of  $y_{1i}, \dots, y_{Gi}$ , and where  $G$  is the number of genes and  $n$  is the number of subjects.

### Value

A list with 3 elements:

res.delta	An object returned by optimize function
eta	model parameter
mat2	transformed data matrix having the same dimension as mat

### Author(s)

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

### References

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

### Examples

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")

res = rv2Transformer(mat = ex)

# estimated model parameter
print(res$eta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")
```

rvTransformer

*Root and VOOM Transformation***Description**

Root and vOOM transformation.

**Usage**

```
rvTransformer(mat, lib.size = NULL, low = 0.001, upp = 1000)
```

**Arguments**

mat	G x n data matrix, where G is the number of genes and n is the number of subjects
lib.size	By default, lib.size is a vector of column sums of mat
low	lower bound for the model parameter
upp	upper bound for the model parameter

**Details**

Denote  $x_{gi}$  as the expression level of the  $g$ -th gene for the  $i$ -th subject. We perform the root transformation

$$y_{gi} = \frac{t_{gi}^{(1/\eta)}}{(1/\eta)}$$

, where

$$t_{gi} = \frac{(x_{gi} + 0.5)}{X_i + 1} \times 10^6$$

and  $X_i = \sum_{g=1}^G x_{gi}$  is the column sum for the  $i$ -th column of the matrix mat. The optimal value for the parameter  $\delta$  is to minimize the squared difference between the sample mean and the sample median of the pooled data  $y_{gi}$ ,  $g = 1, \dots, G$ ,  $i = 1, \dots, n$ , where  $G$  is the number of genes and  $n$  is the number of subjects.

**Value**

A list with 3 elements:

res.eta	An object returned by optimize function
eta	model parameter
mat2	transformed data matrix having the same dimension as mat

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu



## References

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

## Examples

```
library(Biobase)

data(es)
print(es)

# expression set
ex = exprs(es)
print(dim(ex))
print(ex[1:3,1:2])

# mean-median before transformation
vec = c(ex)
m = mean(vec)
md = median(vec)
diff = m - md
cat("m=", m, ", md=", md, ", diff=", diff, "\n")

res = rvTransformer(mat = ex)

# estimated model parameter
print(res$eta)

# mean-median after transformation
vec2 = c(res$mat2)
m2 = mean(vec2)
md2 = median(vec2)
diff2 = m2 - md2
cat("m2=", m2, ", md2=", md2, ", diff2=", diff2, "\n")
```

---

wilcoxWrapper

*Wrapper Function for Wilcoxon Rank Sum Test*

---

## Description

Wrapper function for wilcoxon rank sum test.

## Usage

```
wilcoxWrapper(mat, grp)
```

**Arguments**

`mat`                     $G \times n$  data matrix, where  $G$  is the number of genes and  $n$  is the number of subjects

`grp`                     $n \times 1$  vector of subject group info

**Details**

For each row of `mat`, we perform Wilcoxon rank sum test.

**Value**

A  $G \times 1$  vector of p-values.

**Author(s)**

Zeyu Zhang, Danyang Yu, Minseok Seo, Craig P. Hersh, Scott T. Weiss, Weiliang Qiu

**References**

Zhang Z, Yu D, Seo M, Hersh CP, Weiss ST, Qiu W. Novel Data Transformations for RNA-seq Differential Expression Analysis. (2019) 9:4820 <https://rdcu.be/brDe5>

# Index

## \*Topic **datasets**

es, [2](#)

## \*Topic **method**

getJaccard, [3](#)

l2Transformer, [4](#)

l1Transformer, [6](#)

lv2Transformer, [7](#)

lvTransformer, [9](#)

r2Transformer, [11](#)

rTransformer, [12](#)

rv2Transformer, [14](#)

rvTransformer, [16](#)

wilcoxWrapper, [17](#)

es, [2](#)

getJaccard, [3](#)

l2Transformer, [4](#)

l1Transformer, [6](#)

lv2Transformer, [7](#)

lvTransformer, [9](#)

r2Transformer, [11](#)

rTransformer, [12](#)

rv2Transformer, [14](#)

rvTransformer, [16](#)

wilcoxWrapper, [17](#)