Package ‘GCAI.bias’

February 19, 2015

Type Package

Title Guided Correction Approach for Inherited bias (GCAI.bias)

Version 1.0

Date 2014-07-14

Author Guoshuai Cai

Maintainer Guoshuai Cai <`GCAI.bioinfo@gmail.com`>

Description Many inherited biases and effects exists in RNA-seq due to both biological and technical effects. We observed the biological variance of testing target transcripts can influence the yield of sequencing reads, which might indicate a resource competition existing in RNA-seq. We developed this package to capture the bias depending on local sequence and perform the correction of this type of bias by borrowing information from spike-in measurement.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-07-16 01:07:26

R topics documented:

GCAI.bias-package 2
coeplot 3
corplot 5
correct.guided 6
counts.preprocess 7
index.mat.generation 8
index.preprocess 9
lm.estimate 10
obj.index 11
panel.cor 12
panel.smooth.r 12
posplot 13

test.dat.counts 15
test.dat.seq 16
train.dat.counts 19
train.dat.seq 20

Index **23**

`GCAI.bias-package` *Guided Correction Approach for Inherited bias (GCAI.bias)*

Description

Many inherited biases and effects exists in RNA-seq due to both biological and technical effects. We observed the biological variance of testing target transcripts can influence the yield of sequencing reads, which might indicate a resource competition existing in RNA-seq. We developed this package to capture the bias depending on local sequence and perform the correction of this type of bias by borrowing information from spike-in measurement.

Details

Package:	GCAI-bias
Type:	Package
Version:	1.0
Date:	2014-07-14
License:	GPL (>=2)
LazyLoad:	yes

This package is used for correcting bias introduced by the biological variance of sample transcripts sources. Batch effect in measurement of the same biological sample can be corrected by this package directly. However, spike-in are required to correct bias between different biological samples. For strand specific RNA-seq, antisense and sense reads should be corrected separately. Sequencing reads on each base pair are required to be formatted as train.dat.seq, train.dat.counts, test.dat.seq and test.dat.counts objects. Coefficients of local sequence will be estimated in lm.estimate and they will be used to correct bias by correct.guided function. Visualization of coefficients and correcting performance can be achieved by coeplot, corplot and posplot.

Author(s)

Guoshuai Cai

Maintainer: Guoshuai Cai <GCAI.bioinfo@gmail.com>

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
#initialize index matrix
word<-81
word.vec<-c("A","T","C","G")
pos.vec<-c((-(word-1)/2):((word-1)/2))

obj.index<-index.mat.generation(word.vec,pos.vec)

#train

data(train.dat.seq)
data(train.dat.counts)

train.index<-index.preprocess(train.dat.seq,word)
obj.train<-counts.preprocess(train.dat.counts)
obj.train[["index"]]<-train.index

coe.lm<-lm.estimate(obj.train,fit.cut.train=5)

coeplot(coe.lm,obj.index)

#test,correct

data(test.dat.seq)
data(test.dat.counts)

test.index<-index.preprocess(test.dat.seq,word)
obj.test<-counts.preprocess(test.dat.counts)
obj.test[["index"]]<-test.index

test.corrected<-correct.guided(coe.lm,obj.test)

corplot(test.corrected)
posplot(test.corrected,obj.test$pos)
```

`coeplot` *Estimated Coefficients Visualization*

Description

To plot estimated coefficients against their corresponding positions

Usage

```
coeplot(coe.lm, obj.index, ylim = c(-1, 1))
```

Arguments

`coe.lm`	a 1-column matrix of coefficients estimated from `lm.estimate`
`obj.index`	an object of index generated from `index.mat.generation`
`ylim`	the extremes of the range of y axis

Details

5 end of the mapped reads will be labeled 0 on x axis. Nucleotide A will be colored in red, T will be in yellowgreen, C will be in green and G will be in blue as the baseline.

Value

a plot produced on the current graphic device

Note

index object should match the length of coe.lm

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`lm.estimate`,`index.mat.generation`

Examples

```
word<-81

data(obj.index)
data(train.dat.seq)
data(train.dat.counts)

train.index<-index.preprocess(train.dat.seq,word)
obj.train<-counts.preprocess(train.dat.counts)
obj.train[["index"]]<-train.index

coe.lm<-lm.estimate(obj.train,fit.cut.train=5)

coeplot(coe.lm,obj.index)
```

`corplot` *Pairwise Correlation Calculation and Visualization*

Description

Pairwise comparison matrix of the original and corrected measurements

Usage

```
corplot(mat)
```

Arguments

`mat`	a data matrix from `correct.guided` function

Details

Pairwise plots of sequencing reads mapped to each base pair on the log10 scale will be shown in the bottom panel; Pairwise Pearson correlation coefficients calculated on data points larger than 1 will be shown in the top panel.

Value

a plot produced on the current graphic device

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`correct.guided`

Examples

```
word<-81

data(obj.index)
data(train.dat.seq)
data(train.dat.counts)
data(test.dat.seq)
data(test.dat.counts)

#train
```

```
train.index<-index.preprocess(train.dat.seq,word)
obj.train<-counts.preprocess(train.dat.counts)
obj.train[["index"]]<-train.index

coe.lm<-lm.estimate(obj.train,fit.cut.train=5)

#test

test.index<-index.preprocess(test.dat.seq,word)
obj.test<-counts.preprocess(test.dat.counts)
obj.test[["index"]]<-test.index

test.corrected<-correct.guided(coe.lm,obj.test)

corplot(test.corrected)
```

`correct.guided` *Guided Correction*

Description

Guided by the coefficients estimated from `lm.estimate`, the bias correction could be performed.

Usage

```
correct.guided(coe.train, obj.test)
```

Arguments

`coe.train`	a 1-column matrix of coefficients estimated from training data by `lm.estimate`
`obj.test`	an object of testing dataset generated from `counts.preprocess` and `index.preprocess`

Details

Besides of corrected reads, βx will be also calculated.

Value

a matrix will be returned containing reads counts before and after correction and βx as well.

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`lm.estimate`, `counts.preprocess`, `index.preprocess`

Examples

```
word<-81

data(obj.index)
data(train.dat.seq)
data(train.dat.counts)
data(test.dat.seq)
data(test.dat.counts)

#train

train.index<-index.preprocess(train.dat.seq,word)
obj.train<-counts.preprocess(train.dat.counts)
obj.train[["index"]]<-train.index

coe.lm<-lm.estimate(obj.train,fit.cut.train=5)

#test

test.index<-index.preprocess(test.dat.seq,word)
obj.test<-counts.preprocess(test.dat.counts)
obj.test[["index"]]<-test.index

test.corrected<-correct.guided(coe.lm,obj.test)
```

`counts.preprocess` *Data Object Generation*

Description

To extract position and count information from `train.dat.counts` or `test.dat.counts`

Usage

```
counts.preprocess(mat.counts)
```

Arguments

`mat.counts`	a dataframe of train.dat.counts or test.dat.counts

Value

counts.preprocess return a list containing components:

`counts`	a matrix of counts
`pos`	a matrix of positions

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`train.dat.counts`,`test.dat.counts`

Examples

```
data(train.dat.counts)
obj.train<-counts.preprocess(train.dat.counts)

data(test.dat.counts)
obj.test<-counts.preprocess(test.dat.counts)
```

`index.mat.generation` *Index Matrix Generation*

Description

To generate S3 class of index

Usage

```
index.mat.generation(word.vec, pos.vec)
```

Arguments

`word.vec`	a vector of nucleotides, the default is c("A","T","C","G").
`pos.vec`	a vector of positions of local sequence, the default is -40 ~ 40.

Value

index.mat.generation return a list containing components:

`mat`	a matrix with nucleotides as row names and positions as column names will be returned, a 4×81 matrix will be returned with default arguments.

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
word<-81
word.vec<-c("A","T","C","G")
pos.vec<-c((-(word-1)/2):((word-1)/2))

obj.index<-index.mat.generation(word.vec,pos.vec)
```

`index.preprocess` *Binary Index Matrix Generation*

Description

To initiate the binary variable matrix for linear model

Usage

```
index.preprocess(mat.seq, word)
```

Arguments

`mat.seq`	a dataframe of train.dat.seq or test.dat.seq
`word`	the number of local sequence positions

Value

a binary index matrix will be returned.

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`train.dat.seq`,`test.dat.seq`

Examples

```
word<-81

data(train.dat.seq)
train.index<-index.preprocess(train.dat.seq,word)

data(test.dat.seq)
test.index<-index.preprocess(test.dat.seq,word)
```

`lm.estimate` *Linear Model Fitting*

Description

To estimating the coefficients by fitting the linear model

Usage

```
lm.estimate(obj.train, fit.cut.train = 5)
```

Arguments

`obj.train`	an object of training data generated from `index.preprocess` and `counts.preprocess`
`fit.cut.train`	the minimum counts of the data points used for model fitting, the default for spike-in training dataset is 5.

Details

It models the influence on the local sequence from the dissimilarity of spike-in transcripts measurement. The region of positions around hexamer primers can be defined by the variable "word".

Value

a 1-column matrix of coefficients will be returned.

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`index.preprocess`,`counts.preprocess`

Examples

```
word<-81

data(train.dat.seq)
data(train.dat.counts)

train.index<-index.preprocess(train.dat.seq,word)
obj.train<-counts.preprocess(train.dat.counts)
obj.train[["index"]]<-train.index

coe.lm<-lm.estimate(obj.train,fit.cut.train=5)
```

`obj.index` *Index S3 Class*

Description

Index S3 class contains two objects: a 2-dimension matrix with nucleotides as row names and positions as column names and a flatted one with 1-dimension combinations of nucleotides and positions.

Usage

```
data(obj.index)
```

Format

The format is:

List of 2

$ mat : num [1:4, 1:81] 0 0 0 0 0 0 0 0 0 0 ...

..- attr(*, "dimnames")=List of 2

.. ..$: chr [1:4] "A" "T" "C" "G"

.. ..$: chr [1:81] "-40" "-39" "-38" "-37" ...

$ flat: chr [1:324, 1:3] "1" "2" "3" "4" ...

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
data(obj.index)
```

`panel.cor` *Pearson Correlation Calculation*

Description

To calculate the Pearson correlation coefficient, a part of `corplot`

Usage

```
panel.cor(x, y, digits = 2, prefix = "", cex.cor, cor.cut = 1)
```

Arguments

`x`

`y`

`digits`

`prefix`

`cex.cor`

`cor.cut`

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`corplot`

`panel.smooth.r` *Scatter Plot Generation*

Description

To generate a base pair x-y plot, a part of `corplot`

Usage

```
panel.smooth.r(x, y, col = par("col"), bg = NA, pch = par("pch"),
cex = par("cex"), col.smooth = "red", span = 2/3, iter = 3, ...)
```

Arguments

`x`

`y`

`col`

`bg`

`pch`

`cex`

`col.smooth`

`span`

`iter`

`...`

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`corplot`

`posplot` *Correction Performance Visualization*

Description

Pairwise comparison matrix of the original and corrected measurements

Usage

```
posplot(test.corrected, pos.test, n.lim = 1000, fit.cut.lr = 50)
```

Arguments

`test.corrected`	a data matrix from `correct.guided` function
`pos.test`	a data matrix of test transcripts' position information
`n.lim`	the maximum number of log ratios points will be plotted
`fit.cut.lr`	the minimum number of sequencing reads points will be used for log ratio plotting

Details

Distributions of sequencing reads of samples measured before and after correction will be plotted with the positions on the x-axis and the number of sequencing reads as bars. Also patterns of the fluctuation factor, log ratio before the correction (black line) and log ratio after the correction (blue line) will be plotted.

Value

a plot produced on the current graphic device

Author(s)

Guoshuai Cai

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

See Also

`correct.guided`, `counts.preprocess`

Examples

```
word<-81

data(obj.index)
data(train.dat.seq)
data(train.dat.counts)
data(test.dat.seq)
data(test.dat.counts)

#train

train.index<-index.preprocess(train.dat.seq,word)
obj.train<-counts.preprocess(train.dat.counts)
obj.train[["index"]]<-train.index

coe.lm<-lm.estimate(obj.train,fit.cut.train=5)

#test

test.index<-index.preprocess(test.dat.seq,word)
obj.test<-counts.preprocess(test.dat.counts)
obj.test[["index"]]<-test.index

test.corrected<-correct.guided(coe.lm,obj.test)

posplot(test.corrected,obj.test$pos)
```

`test.dat.counts` *Testing Counts Dataset*

Description

An example, a data frame of reads counts of testing samples

Usage

```
data(test.dat.counts)
```

Format

A data frame with 1527 observations on the following 6 variables.

`gene` a factor with levels `ERCC-00002` `ERCC-00003` `ERCC-00007` `ERCC-00009` `ERCC-00012` `ERCC-00013` `ERCC-00014` `ERCC-00016` `ERCC-00017` `ERCC-00018` `ERCC-00019` `ERCC-00022` `ERCC-00023` `ERCC-00024` `ERCC-00025` `ERCC-00028` `ERCC-00031` `ERCC-00033` `ERCC-00034` `ERCC-00035` `ERCC-00039` `ERCC-00040` `ERCC-00041` `ERCC-00042` `ERCC-00044` `ERCC-00048` `ERCC-00054` `ERCC-00057` `ERCC-00058` `ERCC-00059` `ERCC-00061` `ERCC-00079` `ERCC-00084` `ERCC-00170`

`pos` a numeric vector

`code_strand` a factor with levels `+`

`seq_strand` a factor with levels `S`

`GSM516589_run32_s_2_ERCC_map_Pos` a numeric vector

`GSM517059_run29_s_1_ERCC_map_Pos` a numeric vector

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
data(test.dat.counts)
```

`test.dat.seq` *Testing Sequence Index Dataset*

Description

An example, a data frame of sequence index of testing samples

Usage

```
data(test.dat.seq)
```

Format

A data frame with 1527 observations on the following 85 variables.

`gene` a factor with levels `ERCC-00002` `ERCC-00003` `ERCC-00007` `ERCC-00009` `ERCC-00012` `ERCC-00013` `ERCC-00014` `ERCC-00016` `ERCC-00017` `ERCC-00018` `ERCC-00019` `ERCC-00022` `ERCC-00023` `ERCC-00024` `ERCC-00025` `ERCC-00028` `ERCC-00031` `ERCC-00033` `ERCC-00034` `ERCC-00035` `ERCC-00039` `ERCC-00040` `ERCC-00041` `ERCC-00042` `ERCC-00044` `ERCC-00048` `ERCC-00054` `ERCC-00057` `ERCC-00058` `ERCC-00059` `ERCC-00061` `ERCC-00079` `ERCC-00084` `ERCC-00170`

`pos` a numeric vector

`code_strand` a factor with levels `+`

`seq_strand` a factor with levels `S`

`X1` a numeric vector

`X2` a numeric vector

`X3` a numeric vector

`X4` a numeric vector

`X5` a numeric vector

`X6` a numeric vector

`X7` a numeric vector

`X8` a numeric vector

`X9` a numeric vector

`X10` a numeric vector

`X11` a numeric vector

`X12` a numeric vector

`X13` a numeric vector

`X14` a numeric vector

`X15` a numeric vector

`X16` a numeric vector

`X17` a numeric vector

`X18` a numeric vector

`X19`	a numeric vector
`X20`	a numeric vector
`X21`	a numeric vector
`X22`	a numeric vector
`X23`	a numeric vector
`X24`	a numeric vector
`X25`	a numeric vector
`X26`	a numeric vector
`X27`	a numeric vector
`X28`	a numeric vector
`X29`	a numeric vector
`X30`	a numeric vector
`X31`	a numeric vector
`X32`	a numeric vector
`X33`	a numeric vector
`X34`	a numeric vector
`X35`	a numeric vector
`X36`	a numeric vector
`X37`	a numeric vector
`X38`	a numeric vector
`X39`	a numeric vector
`X40`	a numeric vector
`X41`	a numeric vector
`X42`	a numeric vector
`X43`	a numeric vector
`X44`	a numeric vector
`X45`	a numeric vector
`X46`	a numeric vector
`X47`	a numeric vector
`X48`	a numeric vector
`X49`	a numeric vector
`X50`	a numeric vector
`X51`	a numeric vector
`X52`	a numeric vector
`X53`	a numeric vector
`X54`	a numeric vector
`X55`	a numeric vector

`X56`	a numeric vector
`X57`	a numeric vector
`X58`	a numeric vector
`X59`	a numeric vector
`X60`	a numeric vector
`X61`	a numeric vector
`X62`	a numeric vector
`X63`	a numeric vector
`X64`	a numeric vector
`X65`	a numeric vector
`X66`	a numeric vector
`X67`	a numeric vector
`X68`	a numeric vector
`X69`	a numeric vector
`X70`	a numeric vector
`X71`	a numeric vector
`X72`	a numeric vector
`X73`	a numeric vector
`X74`	a numeric vector
`X75`	a numeric vector
`X76`	a numeric vector
`X77`	a numeric vector
`X78`	a numeric vector
`X79`	a numeric vector
`X80`	a numeric vector
`X81`	a numeric vector

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
data(test.dat.seq)
```

`train.dat.counts` *Training Counts Dataset*

Description

An example, a data frame of reads counts of training samples

Usage

```
data(train.dat.counts)
```

Format

A data frame with 5203 observations on the following 6 variables.

`gene` a factor with levels `ERCC-00002` `ERCC-00003` `ERCC-00007` `ERCC-00009` `ERCC-00012` `ERCC-00013` `ERCC-00014` `ERCC-00016` `ERCC-00017` `ERCC-00018` `ERCC-00019` `ERCC-00022` `ERCC-00023` `ERCC-00024` `ERCC-00025` `ERCC-00028` `ERCC-00031` `ERCC-00033` `ERCC-00034` `ERCC-00035` `ERCC-00039` `ERCC-00040` `ERCC-00041` `ERCC-00042` `ERCC-00044` `ERCC-00048` `ERCC-00054` `ERCC-00058` `ERCC-00061` `ERCC-00079` `ERCC-00171`

`pos` a numeric vector

`code_strand` a factor with levels `+`

`seq_strand` a factor with levels `S`

`GSM516589_run32_s_2_ERCC_map_Pos` a numeric vector

`GSM517059_run29_s_1_ERCC_map_Pos` a numeric vector

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
data(train.dat.counts)
```

`train.dat.seq` *Training Sequence Index Dataset*

Description

An example, a data frame of sequence index of training samples

Usage

```
data(train.dat.seq)
```

Format

A data frame with 5203 observations on the following 85 variables.

`gene` a factor with levels `ERCC-00002` `ERCC-00003` `ERCC-00007` `ERCC-00009` `ERCC-00012` `ERCC-00013` `ERCC-00014` `ERCC-00016` `ERCC-00017` `ERCC-00018` `ERCC-00019` `ERCC-00022` `ERCC-00023` `ERCC-00024` `ERCC-00025` `ERCC-00028` `ERCC-00031` `ERCC-00033` `ERCC-00034` `ERCC-00035` `ERCC-00039` `ERCC-00040` `ERCC-00041` `ERCC-00042` `ERCC-00044` `ERCC-00048` `ERCC-00054` `ERCC-00058` `ERCC-00061` `ERCC-00079` `ERCC-00171`

`pos` a numeric vector

`code_strand` a factor with levels `+`

`seq_strand` a factor with levels `S`

`X1` a numeric vector

`X2` a numeric vector

`X3` a numeric vector

`X4` a numeric vector

`X5` a numeric vector

`X6` a numeric vector

`X7` a numeric vector

`X8` a numeric vector

`X9` a numeric vector

`X10` a numeric vector

`X11` a numeric vector

`X12` a numeric vector

`X13` a numeric vector

`X14` a numeric vector

`X15` a numeric vector

`X16` a numeric vector

`X17` a numeric vector

`X18` a numeric vector

`X19`	a numeric vector
`X20`	a numeric vector
`X21`	a numeric vector
`X22`	a numeric vector
`X23`	a numeric vector
`X24`	a numeric vector
`X25`	a numeric vector
`X26`	a numeric vector
`X27`	a numeric vector
`X28`	a numeric vector
`X29`	a numeric vector
`X30`	a numeric vector
`X31`	a numeric vector
`X32`	a numeric vector
`X33`	a numeric vector
`X34`	a numeric vector
`X35`	a numeric vector
`X36`	a numeric vector
`X37`	a numeric vector
`X38`	a numeric vector
`X39`	a numeric vector
`X40`	a numeric vector
`X41`	a numeric vector
`X42`	a numeric vector
`X43`	a numeric vector
`X44`	a numeric vector
`X45`	a numeric vector
`X46`	a numeric vector
`X47`	a numeric vector
`X48`	a numeric vector
`X49`	a numeric vector
`X50`	a numeric vector
`X51`	a numeric vector
`X52`	a numeric vector
`X53`	a numeric vector
`X54`	a numeric vector
`X55`	a numeric vector

`X56`	a numeric vector
`X57`	a numeric vector
`X58`	a numeric vector
`X59`	a numeric vector
`X60`	a numeric vector
`X61`	a numeric vector
`X62`	a numeric vector
`X63`	a numeric vector
`X64`	a numeric vector
`X65`	a numeric vector
`X66`	a numeric vector
`X67`	a numeric vector
`X68`	a numeric vector
`X69`	a numeric vector
`X70`	a numeric vector
`X71`	a numeric vector
`X72`	a numeric vector
`X73`	a numeric vector
`X74`	a numeric vector
`X75`	a numeric vector
`X76`	a numeric vector
`X77`	a numeric vector
`X78`	a numeric vector
`X79`	a numeric vector
`X80`	a numeric vector
`X81`	a numeric vector

References

Cai G, RNA-SEQUENCING APPLICATIONS: GENE EXPRESSION QUANTIFICATION AND METHYLATOR PHENOTYPE IDENTIFICATION, Ph.D. Thesis, 2013

Examples

```
data(train.dat.seq)
```

Index

∗Topic **RNA-seq**
    GCAI.bias-package, 2
∗Topic **Spike-in**
    GCAI.bias-package, 2
∗Topic **bias**
    GCAI.bias-package, 2
∗Topic **correction**
    correct.guided, 6
∗Topic **datasets**
    obj.index, 11
    test.dat.counts, 15
    test.dat.seq, 16
    train.dat.counts, 19
    train.dat.seq, 20
∗Topic **fitting**
    lm.estimate, 10
∗Topic **initialization**
    counts.preprocess, 7
    index.mat.generation, 8
    index.preprocess, 9
∗Topic **package**
    GCAI.bias-package, 2
∗Topic **visualization**
    coeplot, 3
    corplot, 5
    panel.cor, 12
    panel.smooth.r, 12
    posplot, 13
∗Topic
    GCAI.bias-package, 2

coeplot, 3
corplot, 5, *12*, *13*
correct.guided, *5*, 6, *13*, *14*
counts.preprocess, *6*, *7*, 7, *10*, *14*

GCAI.bias *(*GCAI.bias-package*)*, 2
GCAI.bias-package, 2

index.mat.generation, *4*, 8
index.preprocess, *6*, *7*, 9, *10*

lm.estimate, *4*, *6*, *7*, 10

obj.index, 11

panel.cor, 12
panel.smooth.r, 12
posplot, 13

test.dat.counts, *7*, *8*, 15
test.dat.seq, *9*, 16
train.dat.counts, *7*, *8*, 19
train.dat.seq, *9*, 20