Package 'RNASeqBias'

March 7, 2011

Type Package

Title Bias Detection and Correction in RNA-Sequencing Data

Version 1.0

Date 2011-03-01

Author Wei Zheng, Lisa Chung, Hongyu Zhao

Maintainer Wei Zheng <wei.zheng@yale.edu>

Description The package contains functions and sample data for detecting and correcting for biases in gene expression levels from RNA-Seq experiments.

Depends Genominator, ShortRead, gam, gtools

License Artistic-2.0

LazyLoad yes

R topics documented:

RNASeqBias-package *Bias Detection and Correction in RNA-Sequencing Data*

Description

The package contains functions and sample data for detecting and correcting for biases in gene expression levels from RNA-Seq experiments. Considered bias factors are gene length, GC content and dinucleotide frequencies. Principal component analysis on GC content and dinucleotide frequencies are first performed, and the resulting principal components and gene length were used as covariates to fit a generalized additive model with smoothing spline on gene expression levels as response. The package also have codes to generate bias plots before and after bias correction, and to compare with other quantitative platforms.

Details

Package:	RNASeqBias
Type:	Package
Version:	1.0
Date:	2011-03-01
License:	Artistic-2.0
LazyLoad:	yes

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

Maintainer: Wei Zheng <wei.zheng@yale.edu>

check_agreement *Checking agreement between RTPCR/QuantiGene data and sequencing data*

Description

This function checks the correlation between gene expression measures obtained from different quantitative platforms, and compare the correlations before and after bias correction, to see whether thereis any improvement in cross-platform consistency.

Usage

```
check_agreement(checkdat, grpn = 3, cor.method = "pearson")
```

Arguments

checkdat	A dataframe with 3 columns. For example, column 1 is gene expression levels from RTPCR, column 2 is uncorrected gene expression levels from sequencing, column 3 is corrected gene expression levels from sequencing.
grpn	Number of groups to divide the data (default is 3). Groups are divided according to the fold change before and after bias correction.
cor.method	Type of correlation coeffcient (covariance) to calculate, default is "pearson". Could also be "kendall" or "spearman".

Value

A numeric matrix with 3 columns: correlation with uncorrected data, correlation with corrected data, and percentage improvement. Each row represents calculations for a group of data with corresponding fold change, except for the last row, which is calcualetd from all the data.

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

Examples

```
## data is not provided
rnaseq <- read.table("MAQC_pcg_genes_expr.txt", header=T)
rtpcr <- read.table("RTPCR_result_cleaned.txt", header=T)
guce <- read.table("ENS59_Gene_length_GC_dinuc.txt", header=T)

low.b <- which(apply(log(rnaseq[,1:7]),1,min) < -10)
low.rtb <- which(apply(log(rtpcr[,5:8]),1,mean) == -Inf)
common.b <- intersect(rownames(rtpcr)[-low.rtb], rownames(rnaseq)[-low.b])

biasg <- guce[match(rownames(rnaseq), guce[,1]),]
biasg[,2] <- log(biasg[,2])
colnames(biasg)[2] <- "log_L"

crt.brain <- gampc(apply(log(rnaseq[-low.b, 1:7]),1,mean), biasg[-low.b,2], biasg[-low.b,

rtid.b <- match(common.b, rownames(rtpcr))
sqid.b <- match(common.b, rownames(rnaseq))
rt.brain <- apply(log(rtpcr[rtid.b, 5:8]), 1, mean) #delta_CT.

seq.b <- apply(log(rnaseq[sqid.b,1:7]),1,mean)
crt.b <- crt.brain[match(common.b, rownames(crt.brain)),1]
br.dat <-cbind(rt.brain, seq.b, crt.b)
check.agreement(br.dat)
```

---

gampc — *Using generalized additive model combined with principal component analysis to correct for various biases in the gene expression levels from RNA-Seq data.*

Description

This function regress the gene expression levels from RNA-Seq data (usually in logRPKM scale) on various bias facors such as log(gene length), GC content, and dinucleotide frequencies in the gene using generalized additive model. Since some bias factors may be highly correlated, such as GC content and GC frequency, PCA was first performed on a subset of bias factors before GAM regression.

Usage

```
gampc(ymat, xmat1, xmat2, num.pc = NULL)
```

Arguments

ymat	a dataframe of gene expression levels from RNA-Seq data in log(RPKM) units. Each column represents data from one experiment.
xmat1	a dataframe of bias factors that does not need PCA adjustment. For example, log(gene length) is directly used in GAM without PCA.
xmat2	a dataframe of bias factors that needs PCA adjustment. For example, GC content and dinucleotide frequencies.
num.pc	Number of principal components to be used in GAM regreesion. Default value is to calculate the mimimal number of PC that explains 95

Details

xmat2 was first converted to a number of principal components that are independent from each other, and combined with xmat1 to fit a generalized additive model using smoothing splines.

Value

This function returns a dataframe with the same dimension as ymat, representing corrected gene expression levels.

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

Examples

```
##see vignette for usage on MAQCtestdata
```

---

group_plot *Grouping bias factors into bins and making bias plots*

Description

This function calls a subfunction plotk to group bias factors into bins and plot the gene expresison levels aginast each bias factor. Currently it handles 18 bias factors simulataneously.

Usage

```
group_plot(mat, minval = NULL, maxval = NULL, type = "median", grpn = 500, spec
```

Arguments

mat	a matrix of 19 columns, the first column is log(RPKM), 2nd to 19th columns are bias factors
minval	lower limit of y coordinate range.
maxval	upper limit of y coordinate range.
type	a character string with values "bin" or "median", indicating the type of plots to make. If type = "median" (default), plots with median exp level against median bias level for each bin will be generated. If type = "bin", plots with median exp level against the bin index of bias factors will be generated.
grpn	Number of genes in each bin. Default is 500.
spec	a numeric flag. If spec= -1 (default), will make bias plots for all 18 bas factors; otherwise only the bias factor in column spec will be plotted.
titleon	a logical value to control whether to add a title named after column names of mat for each bias plot. Default is TRUE.

Value

This function will generate bias plots and return a dataframe with y and x values for each bias plot.

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

See Also

See Also plotk, group_plotm

Examples

```
##see vignette for usage on MAQCtestdata
```

---

group_plotm *Overlay bias plots to compare.*

Description

This function takes output from different group_plot runs, and make overlapping bias plots for comparison.

Usage

```
group_plotm(toCmp, minval = NULL, maxval = NULL, titl = rep("not specified", 18)
```

Arguments

toCmp	A list of group_plot objects.
minval	lower limit of y coordinate range.
maxval	upper limit of y coordinate range.
titl	A vector of 18 character strings to make titles for each bias plot. Usually named after the column names of bias factors.

Value

This function makes overlapping bias plots without return values.

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

See Also

See Also `group_plot`

Examples

```
##see vignette for usage on MAQCtestdata
```

`MAQCtestdata` *RNA-Seq counts data and annotation*

Description

This data set contains

Usage

```
MAQCtestdata
```

Format

a list with four components: "Counts" contains counts mapped to each ENSEMBL transcript in one MAQC2 experiment; "reweightedCounts" contains counts after prime reweighting adjustment; "tanno" contains annotation of bias factors for each transcript; "sinog" contains information of single-isoform non-overlapping genes.

Source

The brain data is from SRR037452, and the UHR data is from SRR037459. The counts data and corresponding RPKM values before and after prime reweighting can also be generated from alignment files using the function "prime.reweight.counts" in the package.

References

Hansen KD, Brenner SE, Dudoit S: Biases in Illumina transcriptome sequencing caused by random hexamer priming. Nucleic Acids Res 2010, 38(12):e131.

plotk *A subfunction called by group_plot*

Description

This function divides a bias factor into bins and make a plot against gene expresison levels.

Usage

```
plotk(x, y, minval, maxval, type, grpn = 500)
```

Arguments

x	a numeric vector to partition into bins, usually a bias factor such as GC content.
y	a numeric vector to plot on y-axis, usually log(RPKM)
minval	lower limit of y coordinate range.
maxval	upper limit of y coordinate range.
type	a character string with values "bin" or "median", indicating the type of plots to make. If type = "median" (default), plots with median exp level against median bias level for each bin will be generated. If type = "bin", plots with median exp level against the bin index of bias factors will be generated.
grpn	Number of genes in each bin. Default is 500.

Details

only for internal use. argument values passed through group_plot

Value

A dataframe with two columns: xvalues to plot and yvalues to plot. A scatter plot with a lowess fitted line is also generated.

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

See Also

See Also group_plot

`prime.reweight.counts` *Calculate gene expression levels from alignemnt files and implementing prime reweighting method*

Description

This function calculates counts per gene/transcript from alignment files in bam format, and implements prime reweighting method as described in Hansen et al. 2010.

Usage

```
prime.reweight.counts(infile, annotation, chrMap, group_fac)
```

Arguments

`infile`	a character data frame with two columns: id (the name for each experiment), and path (the path to corresponding alignment file)
`annotation`	a dataframe with "Ensembl.Gene.ID", "Ensembl.Transcript.ID", "Ensembl.Exon.ID", "Exon.Chr.Start..bp.", "Exon.Chr.End..bp.", "Chromosome.Name", "Strand", "start", "end", "strand", "chr". Usually obtained from BioConductor.
`chrMap`	a simple character vector with names of each chromosome, must match "Chromosome.Name" in `annotation`.
`group_fac`	a character string indicating the unit of counts, e.g. "Ensembl.Transcript.ID"

Value

`allCounts`	a data frame with counts per gene for each sample, followed by the prime reweighted counts for each sample.
`trExp`	a data frame with the same dimention as allCounts, but contains gene expression levels in RPKM unit.

Note

Reads in the alignment files are not probablistically assigned to each transcript. Instead one read may be counted for multiple transcripts.

Author(s)

Wei Zheng, Lisa Chung, Hongyu Zhao

References

Hansen KD, Brenner SE, Dudoit S: Biases in Illumina transcriptome sequencing caused by random hexamer priming. Nucleic Acids Res 2010, 38(12):e131.

Examples

```
# if you have a bunch of alignment files in bam format, e.g. brain samples in MAQC
# you can use the following codes

insample <- data.frame("id" = c( "BRN_SRR037452",
"BRN_SRR037453","BRN_SRR037454", "BRN_SRR037455",
"BRN_SRR037456", "BRN_SRR037457", "BRN_SRR037458"
 ),
                     "path" = c(
"SRR037452_map.bam", "SRR037453_map.bam", "SRR037454_map.bam",
"SRR037455_map.bam", "SRR037456_map.bam", "SRR037457_map.bam",
"SRR037458_map.bam") )

#### chrMap is a simple vector with the chromosome names
chrMap <- c(paste("chr", 1:22, sep = ""), "chrM", "chrX", "chrY")

#### annotation file for ENS59 protein coding genes
annotation <- read.table("ENS59_pcg_annotation.txt",  header=T)

allCounts <- prime.reweight.counts(insample, annotation,
chrMap, group_fac = "Ensembl.Transcript.ID")
```

Index