Package 'EventPointer'

April 15, 2019

Type Package

Title An effective identification of alternative splicing events using junction arrays and RNA-Seq data

Version 2.0.1

Author Juan Pablo Romero, Juan Ferrer-Bonsoms, Ander Muniategui, Fernando Carazo, Ander Aramburu, Angel Rubio

Maintainer Juan Pablo Romero <jpromero@ceit.es>

Description EventPointer is an R package to identify alternative splicing events that involve either simple (case-control experiment) or complex experimental designs such as time course experiments and studies including paired-samples. The algorithm can be used to analyze data from either junction arrays (Affymetrix Arrays) or sequencing data (RNA-Seq).
The software returns a data.frame with the detected alternative splicing events: gene name, type of event (cassette, alternative 3',...,etc), genomic position, statistical significance and increment of the percent spliced in (Delta PSI) for all the events.
The algorithm can generate a series of files to visualize the detected alternative splicing events in IGV. This eases the interpretation of results and the design of primers for standard PCR validation.

Depends R (>= 3.4), SGSeq, Matrix, SummarizedExperiment

Imports GenomicFeatures, stringr, GenomeInfoDb, igraph, MASS, nnls, limma, matrixStats, RBGL, prodlim, graph, methods, utils, stats, doParallel, foreach, affxparser, GenomicRanges, S4Vectors, IRanges, qvalue, cobs, rhdf5

Suggests knitr, rmarkdown, BiocStyle, RUnit, BiocGenerics

License Artistic-2.0

LazyData true

RoxygenNote 6.1.0

biocViews AlternativeSplicing, DifferentialSplicing, mRNAMicroarray, RNASeq, Transcription, Sequencing, TimeCourse, ImmunoOncology

VignetteBuilder knitr

Url https://github.com/jpromeror/EventPointer

BugReports https://github.com/jpromeror/EventPointer/issues

git_url https://git.bioconductor.org/packages/EventPointer

git_branch RELEASE_3_8
git_last_commit e700729
git_last_commit_date 2018-12-05
Date/Publication 2019-04-15

R topics documented:

`AllEvents_RNASeq` *Alternative splicing events detected by EventPointer*

Description

Alternative splicing events detected by EventPointer

Usage

```
data(AllEvents_RNASeq)
```

Format

A `list` object `AllEvents_RNASeq[[i]][[j]]` displays the jth splicing event for the ith gene.

Value

AllEvents_RNASeq object contains all the detected alternativesplicing events using EventPointer-methodology. The splicing events where detected using the BAM files from the dataset published in Seshagiri et al. 2012 andused in the `SGSeq` R package vignette.

AllEvents_RNASeq_MP *Alternative splicing multi-path events detected by EventPointer*

Description

Alternative splicing multi-path events detected by EventPointer

Usage

```
data(AllEvents_RNASeq_MP)
```

Format

A `list` object `AllEvents_RNASeq[[i]][[j]]` displays the jth splicing event for the ith gene.

Value

AllEvents_RNASeq_MP object contains all the detected alternative splicing events using EventPointer methodology for multi-path events. The splicing events where detected using the BAM files from the dataset published in Seshagiri et al. 2012 and used in the `SGSeq` R package vignette.

ArrayDatamultipath *Preprocessed arrays data with multi-path events*

Description

Preprocessed arrays data with multi-path events

Usage

```
data(ArrayDatamultipath)
```

Format

A `data.frame` with preprocessed arrays data. The preprocessing was done using `aroma.affymetrix`. See the package vignette for the preprocessing pipeline

Value

ArrayDatamultipath object contains preprocessed junction arrays data. The preprocessing was done using `aroma.affymetrix` R package, refer to EventPointer vignette for the pipeline used for the preprocessing. The data corresponds to 4 samples from the SUM149 Cell line hybridized to the HTA 2.0 Affymetrix array. The first two samples are control and the second ones are treated.

ArraysData *Preprocessed arrays data*

Description

Preprocessed arrays data

Usage

```
data(ArraysData)
```

Format

A `data.frame` with preprocessed arrays data. The preprocessing was done using `aroma.affymetrix`. See the package vignette for the preprocessing pipeline

Value

ArraysData object contains preprocessed junction arrays data. The preprocessing was done using `aroma.affymetrix` R package, refer to EventPointer vignette for the pipeline used for the preprocessing. The data corresponds to 4 samples from the SUM149 Cell line hybridized to the HTA 2.0 Affymetrix array. The first two samples are control and the second ones are treated.

CDFfromGTF *CDF file creation for EventPointer*

Description

Generates the CDF file to be used under the aroma.affymetrix framework

Usage

```
CDFfromGTF(input = "Ensembl", inputFile = NULL, PSR, Junc, PathCDF,
  microarray = NULL)
```

Arguments

`input`	Reference transcriptome used to build the CDF file. Must be one of: 'Ensembl', 'UCSC' , 'AffyGTF' or 'CustomGTF'.
`inputFile`	If input is 'AffyGTF' or 'CustomGTF', inputFile should point to the GTF file to be used.
`PSR`	Path to the Exon probes txt file
`Junc`	Path to the Junction probes txt file
`PathCDF`	Directory where the output will be saved
`microarray`	Microarray used to create the CDF file. Must be one of: HTA-2_0, ClariomD, RTA or MTA

Value

The function displays a progress bar to show the user the progress of the function. However, there is no value returned in R as the function creates three files that are used later by other EventPointer functions.1) EventsFound.txt : Tab separated file with all the information of all the alternative splcing events found. 2) .flat file : Used to build the corresponding CDF file. 3) .CDF file: Output required for the aroma.affymetrix preprocessing pipeline. Both the .flat and .CDF file take large ammounts of memory in the hard drive, it is recommended to have at least 1.5 GB of free space.

Examples

```
PathFiles<-system.file('extdata',package='EventPointer')
DONSON_GTF<-paste(PathFiles,'/DONSON.gtf',sep='')
PSRProbes<-paste(PathFiles,'/PSR_Probes.txt',sep='')
JunctionProbes<-paste(PathFiles,'/Junction_Probes.txt',sep='')
Directory<-tempdir()
microarray<-'HTA-2_0'

# Run the function

CDFfromGTF(input='AffyGTF',inputFile=DONSON_GTF,PSR=PSRProbes,Junc=JunctionProbes,
           PathCDF=Directory,microarray=microarray)
```

CDFfromGTF_Multipath *CDF file creation for EventPointer*

Description

Generates the CDF file to be used under the aroma.affymetrix framework

Usage

```
CDFfromGTF_Multipath(input = "Ensembl", inputFile = NULL, PSR, Junc,
  PathCDF, microarray = NULL, paths = 2)
```

Arguments

input	Reference transcriptome used to build the CDF file. Must be one of Ensembl, UCSC or GTF.
inputFile	If input is GTF, inputFile should point to the GTF file to be used.
PSR	Path to the Exon probes txt file
Junc	Path to the Junction probes txt file
PathCDF	Directory where the output will be saved
microarray	Microarray used to create the CDF file. Must be one of: HTA-2_0, ClariomD, RTA or MTA
paths	Maximum number of paths of the events to find.

Value

The function displays a progress bar to show the user the progress of the function. However, there is no value returned in R as the function creates three files that are used later by other EventPointer functions. 1) EventsFound.txt : Tab separated file with all the information of all the alternative splcing events found. 2) .flat file : Used to build the corresponding CDF file. 3) .CDF file: Output required for the aroma.affymetrix preprocessing pipeline. Both the .flat and .CDF file take large ammounts of memory in the hard drive, it is recommended to have at least 1.5 GB of free space.

Examples

```
PathFiles<-system.file('extdata',package='EventPointer')
DONSON_GTF<-paste(PathFiles,'/DONSON.gtf',sep='')
PSRProbes<-paste(PathFiles,'/PSR_Probes.txt',sep='')
JunctionProbes<-paste(PathFiles,'/Junction_Probes.txt',sep='')
Directory<-tempdir()
microarray<-'HTA-2_0'

# Run the function

CDFfromGTF_Multipath(input='AffyGTF',inputFile=DONSON_GTF,PSR=PSRProbes,Junc=JunctionProbes,
                     PathCDF=Directory,microarray=microarray,paths=3)
```

EventDetection *Detect splicing events using EventPointer methodology*

Description

Identification of all the alternative splicing events in the splicing graphs

Usage

```
EventDetection(Input, cores, Path)
```

Arguments

Input	Output of the PrepareBam_EP function
cores	Number of cores used for parallel processing
Path	Directory where to write the EventsFound_RNASeq.txt file

Value

list with all the events found for all the genes present in the experiment. It also generates a file called EventsFound_RNASeq.txt with the information of each event.

Examples

```
# Run EventDetection function
data(SG_RNASeq)
TxtPath<-tempdir()
AllEvents_RNASeq<-EventDetection(SG_RNASeq,cores=1,Path=TxtPath)
```

`EventDetectionMultipath` *Detect splicing events using EventPointer methodology*

Description

Identification of all the alternative splicing events in the splicing graphs

Usage

```
EventDetectionMultipath(Input, cores, Path, paths = 2)
```

Arguments

`Input`	Output of the PrepareBam_EP function
`cores`	Number of cores used for parallel processing
`Path`	Directory where to write the EventsFound_RNASeq.txt file
`paths`	Maximum number of paths of the events to find.

Value

list with all the events found for all the genes present in the experiment. It also generates a file called EventsFound_RNASeq.txt with the information each event.

Examples

```
# Run EventDetection function
 data(SG_RNASeq)
 TxtPath<-tempdir()
 AllEvents_RNASeq_MP<-EventDetectionMultipath(SG_RNASeq,cores=1,Path=TxtPath,paths=3)
```

`EventPointer` *EventPointer*

Description

Statistical analysis of alternative splcing events

Usage

```
EventPointer(Design, Contrast, ExFit, Eventstxt, Filter = TRUE,
  Qn = 0.25, Statistic = "LogFC", PSI = FALSE)
```

Arguments

Design	The design matrix for the experiment.
Contrast	The contrast matrix for the experiment.
ExFit	aroma.affymetrix pre-processed variable after using `extractDataFrame(affy, addNames=TRUE)`
Eventstxt	Path to the EventsFound.txt file generated by CDFfromGTF function.
Filter	Boolean variable to indicate if an expression filter is applied
Qn	Quantile used to filter the events (Bounded between 0-1, Q1 would be 0.25).
Statistic	Statistical test to identify differential splicing events, must be one of : LogFC, Dif_LogFC or DRS.
PSI	Boolean variable to indicate if Delta PSI should be calculated for every splicing event.

Value

Data.frame ordered by the splicing p.value . The object contains the different information for each splicing event such as Gene name, event type, genomic position, p.value, z.value and delta PSI.

Examples

```
  data(ArraysData)

  Dmatrix<-matrix(c(1,1,1,1,0,0,1,1),nrow=4,ncol=2,byrow=FALSE)
  Cmatrix<-t(t(c(0,1)))
 EventsFound<-paste(system.file('extdata',package='EventPointer'),'/EventsFound.txt',sep='')

Events<-EventPointer(Design=Dmatrix,
                     Contrast=Cmatrix,
                     ExFit=ArraysData,
                     Eventstxt=EventsFound,
                     Filter=TRUE,
                     Qn=0.25,
                     Statistic='LogFC',
                     PSI=TRUE)
```

EventPointer_IGV *EventPointer IGV Visualization*

Description

Generates of files to be loaded in IGV for visualization and interpretation of events

Usage

```
EventPointer_IGV(Events, input, inputFile = NULL, PSR, Junc, PathGTF,
  EventsFile, microarray = NULL)
```

Arguments

Events	Data.frame generated by EventPointer with the events to be included in the GTF file.
input	Reference transcriprome. Must be one of: "Ensembl", "UCSC" , "AffyGTF" or "CustomGTF".
inputFile	If input is "AffyGTF" or "CustomGTF", inputFile should point to the GTF file to be used.
PSR	Path to the Exon probes txt file.
Junc	Path to the Junction probes txt file.
PathGTF	Directory where to write the GTF files.
EventsFile	Path to EventsFound.txt file generated with CDFfromGTF function.
microarray	Microarray used to create the CDF file. Must be one of: HTA-2_0, ClariomD, RTA or MTA

Value

The function displays a progress bar to show the user the progress of the function. Once the progress bar reaches 100 in PathGTF. The created files are: 1) paths.gtf : GTF file representing the alternative splicing events and 2) probes.gtf : GTF file representing the probes that measure each event and each path.

Examples

```
  PathFiles<-system.file('extdata',package='EventPointer')
  DONSON_GTF<-paste(PathFiles,'/DONSON.gtf',sep='')
  PSRProbes<-paste(PathFiles,'/PSR_Probes.txt',sep='')
  JunctionProbes<-paste(PathFiles,'/Junction_Probes.txt',sep='')
  Directory<-tempdir()

  data(ArraysData)

  Dmatrix<-matrix(c(1,1,1,1,0,0,1,1),nrow=4,ncol=2,byrow=FALSE)
  Cmatrix<-t(t(c(0,1)))
 EventsFound<-paste(system.file('extdata',package='EventPointer'),'/EventsFound.txt',sep='')

  Events<-EventPointer(Design=Dmatrix,
                       Contrast=Cmatrix,
                       ExFit=ArraysData,
                       Eventstxt=EventsFound,
                       Filter=TRUE,
                       Qn=0.25,
                       Statistic='LogFC',
                       PSI=TRUE)

EventPointer_IGV(Events=Events[1,,drop=FALSE],
                 input='AffyGTF',
                 inputFile=DONSON_GTF,
                 PSR=PSRProbes,
                 Junc=JunctionProbes,
                 PathGTF=Directory,
                EventsFile= EventsFound,
                microarray="HTA-2_0")
```

EventPointer_RNASeq *Statistical analysis of alternative splcing events for RNASeq data*

Description

Statistical analysis of all the alternative splicing events found in the given bam files.

Usage

```
EventPointer_RNASeq(Events, Design, Contrast, Statistic = "LogFC",
  PSI = FALSE)
```

Arguments

Events	Output from EventDetection function
Design	The design matrix for the experiment.
Contrast	The contrast matrix for the experiment.
Statistic	Statistical test to identify differential splicing events, must be one of : LogFC, Dif_LogFC and DRS.
PSI	Boolean variable to indicate if PSI should be calculated for every splicing event.

Value

Data.frame ordered by the splicing p.value . The object contains the different information for each splicing event such as Gene name, event type, genomic position, p.value, z.value and delta PSI.

Examples

```
data(AllEvents_RNASeq)
Dmatrix<-matrix(c(1,1,1,1,1,1,1,0,0,0,0,1,1,1,1),ncol=2,byrow=FALSE)
Cmatrix<-t(t(c(0,1)))
Events <- EventPointer_RNASeq(AllEvents_RNASeq,Dmatrix,Cmatrix,Statistic='LogFC',PSI=TRUE)
```

EventPointer_RNASeq_IGV *EventPointer RNASeq IGV Visualization*

Description

Generates of files to be loaded in IGV for visualization and interpretation of events

Usage

```
EventPointer_RNASeq_IGV(Events, SG_RNASeq, EventsTxt, PathGTF)
```

Arguments

Events	Data.frame generated by EventPointer_RNASeq with the events to be included in the GTF file.
SG_RNASeq	Output from PrepareBam_EP function. Contains splicing graphs components.
EventsTxt	Path to EventsFound.txt file generated with EventDetection function
PathGTF	Directory where to write the GTF files.

Value

The function displays a progress bar to show the user the progress of the function. Once the progress bar reaches 100 file is written to the specified directory in PathGTF. The created file: 1) paths_RNASeq.gtf : GTF file representing the alternative splicing events.

Examples

```
data(AllEvents_RNASeq)
data(SG_RNASeq)

 # Run EventPointer

 Dmatrix<-matrix(c(1,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1),ncol=2,byrow=FALSE)
 Cmatrix<-t(t(c(0,1)))
Events <- EventPointer_RNASeq(AllEvents_RNASeq,Dmatrix,Cmatrix,Statistic='LogFC',PSI=TRUE)

 # IGV Visualization

EventsTxt<-paste(system.file('extdata',package='EventPointer'),'/EventsFound_RNASeq.txt',sep='')
 PathGTF<-tempdir()
 EventPointer_RNASeq_IGV(Events,SG_RNASeq,EventsTxt,PathGTF)
```

EventPointer_RNASeq_TranRef

Statistical analysis with the output of GetPSI_FromTranRef

Description

Statistical analysis with the output of GetPSI_FromTranRef

Usage

```
EventPointer_RNASeq_TranRef(Count_Matrix, Statistic = "LogFC", Design,
  Contrast)
```

Arguments

Count_Matrix	The list containing the expression data taken from the ouput of GetPSI_FromTranRef
Statistic	The type of statistic to apply. Default = 'LogFC' (can be 'logFC, 'Dif_LogFC','DRS')
Design	The design matrix of the experiment.
Contrast	The Contrast matrix of the experiment.

Value

a data.frame with the information of the names of the event, its p.values and the corresponding z.value. If there is more than one contrast, the function returns as many data.frames as number of contrast and all these data.frame are sotred in an unique list.

Examples

```
data(EventXtrans)
data(PSIss)
# Design and contrast matrix:

Design <- matrix(c(1,1,1,1,0,0,1,1),nrow=4)
Contrast <- matrix(c(0,1),nrow=2)

# Statistical analysis:


Fit <- EventPointer_RNASeq_TranRef(Count_Matrix =  PSIss$ExpEvs,
                                   Statistic = 'LogFC',Design = Design,
                                   Contrast = Contrast)
```

EventsGTFfromTrancriptomeGTF

Events .gtf from trancriptome .gtf

Description

Events .gtf from trancriptome .gtf

Usage

```
EventsGTFfromTrancriptomeGTF(inputFile = NULL, Transcriptome = NULL,
  Pathtxt = NULL, PathGTF = NULL)
```

Arguments

inputFile	If input is GTF, inputFile should point to the GTF file to be used.
Transcriptome	the name of the transcriptome
Pathtxt	Directory to save the .txt of the events founded
PathGTF	Directory where the output will be saved

Value

a list containing four elements: three sparce matrices that relate which isoforms build up the paths (path1,path2 and pathRef) of each event. The fourth element contains the name of the reference annotation: only appear the name of the transcript.

Examples

```
PathFiles<-system.file("extdata",package="EventPointer")
inputFile <- paste(PathFiles,"/gencode.v24.ann_2genes.gtf",sep="")
Transcriptome <- "Gencode24_2genes"
Pathtxt <- tempdir()
PathGTF <- tempdir()

# Run the function

EventXtrans <- EventsGTFfromTrancriptomeGTF(inputFile = inputFile,
                                            Transcriptome = Transcriptome,
                                            Pathtxt=Pathtxt,PathGTF=PathGTF)
```

`EventXtrans` *relationship between isoforms and events*

Description

relationship between isoforms and events

Usage

```
data(EventXtrans)
```

Format

A `list` object `EventXtrans[[1]]` displays the isoform that build up the path1 of each event.

Value

EventXtrans object contains the relationship between the isoforms and the events. It is a list of 4 elements. the first three stored sparse matrices relating the isoforms with the events. The fourth element stores de names of the reference annotation used (isoforms names)

`getbootstrapkallisto` *Author: JF*

Description

Function to load the values of the bootstrap

Inputs:

Usage

```
getbootstrapkallisto(pathValues = NA, nb)
```

Arguments

pathValues	A vector with the complete directory to the folder of the output of kallisto
nb	number of bootstrap

Value

A list containing the quantification data and with the bootrap information.

Examples

```
PathFiles <- system.file('extdata',package='EventPointer')
PathFiles <- dir(paste0(PathFiles,'/output'),full.names = TRUE)

#load the data

mydatab <- getbootstrapkallisto(pathValues = PathFiles,nb = 20)
```

GetPSI_FromTranRef *GetPSI_FromTranRef*

Description

Get the values of PSI.A filer expresion is applied if the user select the option of filter.

Usage

```
GetPSI_FromTranRef(PathsxTranscript, Samples, Filter = TRUE, Qn = 0.25)
```

Arguments

PathsxTranscript	the outpu of EventGTFfromTrancriptomeGTF
Samples	the samples (in the rowname of the samples must be written only the name of the transcript)
Filter	Boolean variable to indicate if an expression filter is applied. Defaul T
Qn	Quantile used to filter the events (Bounded between 0-1, Q1 would be 0.25).

Value

The output of the function is a list containing two elements: a matrix with the values of PSI and a list containing as many matrices as number of events. In each matrix is stored the expression of the different paths of an event along the samples.

Examples

```
data(EventXtrans)
PathFiles <- system.file('extdata',package='EventPointer')
filesnames <- dir(paste0(PathFiles,'/output'))
PathFiles <- dir(paste0(PathFiles,'/output'),full.names = TRUE)
dirtoload <- paste0(PathFiles,'/','abundance.tsv')
RNASeq <- read.delim(dirtoload[1],sep = '\t', colClasses = c(NA,'NULL','NULL','NULL',NA))
for (n in 2:length(dirtoload)){
  RNASeq[,n+1] <- read.delim(dirtoload[n],sep = '\t',
                             colClasses = c('NULL','NULL','NULL','NULL',NA))
}
rownames(RNASeq)<-RNASeq[,1]
RNASeq<-RNASeq[,-1]
colnames(RNASeq) <- filesnames
rownames(RNASeq) <- sapply(strsplit(rownames(RNASeq),'\\|'),function(X) return(X[1]))
RNASeq<-as.matrix(RNASeq) #must be a matrix variable

#Obtain values of PSI

PSIss <- GetPSI_FromTranRef(PathsxTranscript = EventXtrans,Samples = RNASeq,Filter = FALSE)

PSI <- PSIss$PSI
Expression <- PSIss$ExpEvs
```

`PrepareBam_EP` *Bam files preparation for EventPointer*

Description

Prepares the information contained in .bam files to be analyzed by EventPointer

Usage

```
PrepareBam_EP(Samples, SamplePath, Ref_Transc = "Ensembl",
  fileTransc = NULL, cores = 1, Alpha = 2)
```

Arguments

`Samples`	Name of the .bam files to be analyzed (Sample1.bam,Sample2.bam,...,etc).
`SamplePath`	Path where the bam files are stored.
`Ref_Transc`	Reference transcriptome used to name the genes found in bam files. Options are: Ensembl, UCSC or GTF.
`fileTransc`	Path to the GTF reference transcriptome ff Ref_Transc is GTF.
`cores`	Number of cores used for parallel processing.
`Alpha`	Internal SGSeq parameter to include or exclude regions

Value

SGFeaturesCounts object. It contains a GRanges object with the corresponding elements to build the different splicing graphs found and the counts related to each of the elements.

Examples

```
## Not run:
 # Obtain the samples and directory for .bam files

   BamInfo<-si
   Samples<-BamInfo[,2]
   PathToSamples <- system.file('extdata/bams', package = 'SGSeq')
   PathToGTF<-paste(system.file('extdata',package='EventPointer'),'/FBXO31.gtf',sep='')

  # Run PrepareBam function
   SG_RNASeq<-PrepareBam_EP(Samples=Samples,
                            SamplePath=PathToSamples,
                            Ref_Transc='GTF',
                            fileTransc=PathToGTF,
                            cores=1)

## End(Not run)
```

`PSIss` *relationship between isoforms and events*

Description

relationship between isoforms and events

Usage

```
data(PSIss)
```

Format

A object `PSIss[[1]]` displays the values of PSI and `PSIss[[2]]` the valeus of expression.

Value

PSIss object the values of PSI calculated by the funcion GetPSI_FromTranRef and also the values of expression.

`PSI_Statistic` *PSI_Statistic*

Description

Statistical analysis of the alternative splicing events. This function takes as input the values of PSI. Perform a statistical analysis using the bootstrap method

Usage

```
PSI_Statistic(PSI, Design, Contrast, nboot)
```

Arguments

PSI	A matrix with the values of the PSI.
Design	The design matrix for the experiment.
Contrast	The contrast matrix for the experiment.
nboot	The number of random analysis.

Value

The output of these functions is a list containing: two data.frame (deltaPSI and Pvalues) with the values of the deltaPSI and the p.values for each contrast, and a third element (LocalFDR) with the information of the local false discovery rate.

Examples

```
data(PSIss)
Design <- matrix(c(1,1,1,1,0,0,1,1),nrow=4)
Contrast <- matrix(c(0,1),nrow=1)

# Statistical analysis:

table <- PSI_Statistic(PSIss$PSI,Design = Design, Contrast = Contrast, nboot = 50)
```

SG_RNASeq *Splicing graph elements predicted from BAM files*

Description

Splicing graph elements predicted from BAM files

Usage

```
data(SG_RNASeq)
```

Format

A SGFeatureCounts objects with predicted splicing graph features and counts

Value

SG_RNASeq object displays the predicted features found in the BAM files from the dataset published in Seshagiri et al. 2012 and used in the SGSeq R package vignette.

Index