Package 'ELMER'

October 27, 2015

Title Inferring Regulatory Element Landscapes and Transcription Factor Networks Using Cancer Methylomes

Version 1.2.0

Description
ELMER is designed to use DNA methylation and gene expression from a large number of samples to infere regulatory element landscape and transcription factor network in primary tissue.

Depends R (>= 3.2.0), IlluminaHumanMethylation450kanno.ilmn12.hg19, Homo.sapiens, ELMER.data

License GPL-3

LazyData true

VignetteBuilder knitr

Imports
GenomicRanges,ggplot2,reshape,grid,gridExtra,IRanges,GenomeInfoDb,S4Vectors,minfi,GenomicFeatures

Suggests parallel, snow, BiocStyle, knitr

biocViews DNAMethylation, GeneExpression, MotifAnnotation, Software, GeneRegulation

NeedsCompilation no

Author Lijing Yao [cre, aut],
Ben Berman [aut],
Peggy Farnham [aut],
Hui Shen [ctb],
Peter Laird [ctb],
Simon Coetzee [ctb]

Maintainer Lijing Yao <lijingya@usc.edu>

R topics documented:

`fetch.mee` *fetch.mee to generate MEE.data class object.*

Description

fetch.mee is a funtion to take in DNA methylation, RNA expression, sample information probe information and gene annotation generating a MEE.data class object, which is the input for main functions. Options (meth, exp, sample, probeInfo, geneInfo) can take in R object or read files by specifying file paths. When TCGA is specified, sample information will be automatically generated such as Control/Experiment labels.

Usage

```
fetch.mee(meth, exp, sample, probeInfo, geneInfo, probes = NULL, genes = NULL, TCGA = FALSE)
```

Arguments

`meth`	A matrix (R object) or path of XX.rda file which only stores a matrix of DNA methylation data.
`exp`	A matrix (R object) or path of XX.rda file which only stores a matrix of expression data.
`sample`	A data frame (R object) or path of XX.rda file which only contains sample information in data frame format.
`probeInfo`	A GRnage object or path of XX.rda file which only contains a GRange of probe information.
`geneInfo`	A GRnage object or path of XX.rda file which only contains a GRange of gene information such as Coordinates, GENEID and SYMBOL.

probes	A vector lists name of probes. If probes are specified, the DNA methylation matrix and probeInfo in MEE.data object will be restrained to this list of probes.
genes	A vector lists gene ids. If gene are specified, the methylation and probeInfo in output MEE.data object will be restrained this list of genes.
TCGA	A logical. FALSE indicates that data is not from TCGA (FALSE is default). TRUE indicates data is from TCGA and sample section will automatically filled in.

Details

Use path to load in data will help to reduce memory usage.

Value

A MEE.data object containing 5 slots. Detail see MEE.data-class

Note

Options (meth, exp, sample, probeInfo, geneInfo) don't need to be all specified. User can input one or more according the needs of the function. Such as get.meth.diff only need meth, sample and probeInfo.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
meth <- matrix(data=c(1:20),ncol=5,dimnames=list(paste0("probe",1:4),paste0("sample",1:5)))
exp <- matrix(data=c(101:110),ncol=5,dimnames=list(c("gene1","gene2"),paste0("sample",1:5)))
mee <- fetch.mee(meth=meth, exp=exp)
## only fetch probe 1 and 3
mee <- fetch.mee(meth=meth, exp=exp, probes=c("probe1","probe3"))
## only fetch gene 1
mee <- fetch.mee(meth=meth, exp=exp, genes="gene1")
```

---

fetch.pair *fetch.pair to generate Pair class object.*

Description

fetch.pair is a funtion to take in enhancer-gene linkage prediction information, probe information and gene annotation generating a Pair class object, which is the input for plotting functions. Options (pair, probeInfo, geneInfo) can take in R object or read files by specifying file paths.

Usage

```
fetch.pair(pair, probeInfo, geneInfo)
```

Arguments

pair	A data.frame (R object) or a path of XX.csv file containing pair information such as output of function get.pair.
probeInfo	A GRnage object or a path of XX.rda file which only contains a GRange of probe information.
geneInfo	A GRnage object or path of XX.rda file which only contains a GRange of gene information such as Coordinates, GENEID and SYMBOL.

Value

A Pair class object containing 3 slots. Detail see Pair-class

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

See Also

schematic.plot

Examples

```
df <- data.frame(Probe=c("cg19403323","cg12213388","cg26607897"),
GeneID =c("ID255928","ID84451","ID55811"),
Symbol =c("SYT14","KIAA1804","ADCY10"),
Pe=c(0.003322259,0.003322259,0.003322259))
geneInfo <- txs()
## input can be a path
pair <- fetch.pair(pair = df, geneInfo=geneInfo)
```

---

get.diff.meth *get.diff.meth to identify hypo/hyper-methylated CpG sites on HM450K between control and experimental groups such as normal verus tumor samples.*

Description

get.diff.meth applys one-way t-test to identify the CpG sites that are significantly hypo/hyper-methyalated using proportional samples (defined by percentage option) from control and experimental groups. The P values will be adjusted by Benjamini-Hochberg method. Option pvalue and sig.dif will be the criteria (cutoff) for selecting significant differentially methylated CpG sites. If save is TURE, two getMethdiff.XX.csv files will be generated (see detail).

Usage

```
get.diff.meth(mee, diff.dir = "hypo", cores = NULL, percentage = 0.2, pvalue = 0.01, sig.dif = 0.3,
```

Arguments

mee	A MEE.data class object contains at least methy and probeInfo slots.
diff.dir	A character can be "hypo" or "hyper", showing dirction DNA methylation changes. If it is "hypo", get.diff.meth function will identify all significantly hypomethylated CpG sites; If "hyper", get.diff.meth function will identify all significantly hypoermethylated CpG sites
cores	A interger which defines the number of cores to be used in parallel process. Default is NULL: no parallel process.
percentage	A number ranges from 0 to 1 specifying the percentage of samples from control and experimental groups that are used to identify the differential methylation. Default is 0.2.
pvalue	A number specifies the significant P value (adjusted P value by BH) cutoff for selecting significant hypo/hyper-methylated probes. Default is 0.01.
sig.dif	A number specifies the smallest DNA methylation difference as a cutoff for selecting significant hypo/hyper-methylated probes. Default is 0.3.
dir.out	A path specifies the directory for outputs. Default is is current directory.
save	A logic. When TRUE, two getMethdiff.XX.csv files will be generated (see detail)

Details

save: When save is TRUE, function will generate two XX.csv files.The first one is named getMethdiff.hypo.probes.csv (or getMethdiff.hyper.probes.csv depends on diff.dir). The first file contains all statistic results for each probe. Based on this file, user can change different P value or sig.dir cutoff to select the significant results without redo the analysis. The second file is named getMethdiff.hypo.probes.significant.csv (or getMethdiff.hyper.probes.significant.csv depends on diff.dir). This file contains statistic results for the probes that pass the significant criteria (P value and sig.dir). When save is FALSE, a data frame R object will be generate which contains the same information with the second file.

Value

A data frame contains statistics from differential analysis for each probes.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
load(system.file("extdata","mee.example.rda",package = "ELMER"))
Hypo.probe <- get.diff.meth(mee, diff.dir="hypo") # get hypomethylated probes
```

`get.enriched.motif` *get.enriched.motif to identify the overrepresented motifs in a set of probes (HM450K) regions.*

Description

get.enriched.motif is a function make use of Probes.motif data from **ELMER.data** package to calculate the motif enrichment Odds Ratio and 95% confidence interval for a given set of probes. If save is TURE, two output files will be saved: getMotif.XX.enriched.motifs.rda and getMotif.XX.motif.enrichment.csv (see detail).

Usage

```
get.enriched.motif(probes.motif, probes, background.probes, lower.OR = 1.1, min.incidence = 10,
dir.out = "./", label = NULL, save=TRUE)
```

Arguments

`probes.motif`	A matrix contains motifs occurrence within probes regions. Probes.motif in **ELMER.data** will be used if probes.motif is missing (detail see `Probes.motif`).
`probes`	A vector lists the name of probes to define the set of probes in which motif enrichment OR and confidence interval will be calculated.
`background.probes`	A vector lists name of probes which are considered as background for motif.enrichment calculation (see detail).
`lower.OR`	A number specifies the smallest lower boundary of 95% confidence interval for Odds Ratio. The motif with higher lower boudnary of 95% confidence interval for Odds Ratio than the number are the significantly enriched motifs (detail see reference).
`min.incidence`	A non-negative integer specifies the minimum incidence of motif in the given probes set. 10 is default.
`dir.out`	A path specifies the directory for outputs. Default is current directory
`label`	A character labels the outputs such as "hypo", "hyper"
`save`	If save is TURE, two files will be saved: getMotif.XX.enriched.motifs.rda and getMotif.XX.motif.enrichment.csv (see detail).

Details

background.probes: For enhancer study, it is better to use probes within distal enhancer probes as background.probes. For promoter study, it is better to use probes within promoter regions as background.probes. Because enhancer and promoter have different CG content and harbors different clusters of TFs motif.

save: if save is TRUE, two files will be save on the disk. The first file is getMotif.XX.motif.enrichment.csv (XX depends on option label). This file reports the Odds Ratio and 95% confidence interval for these Odds Ratios which pass the signficant cutoff (lower.OR and min.incidence). The second file is getMotif.XX.enriched.motifs.rda (XX depends on option lable). This file contains a list R object with enriched motifs as name and probes containing the enriched motif as contents. This object will be used in `get.TFs` function. if save is FALSE, the function will return a R object which is the same with second file.

Value

A list (R object) with enriched motifs as name and probes containing the enriched motif as contents. And hypo.motif.enrichment.pdf plot will be generated.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
probes <- c("cg00329272","cg10097755","cg08928189", "cg17153775","cg21156590",
"cg19749688","cg12590404","cg24517858","cg00329272","cg09010107",
"cg15386853", "cg10097755", "cg09247779","cg09181054","cg19371916")
load(system.file("extdata","mee.example.rda",package = "ELMER"))
bg <- rownames(getMeth(mee))
enriched.motif <- get.enriched.motif(probes=probes,background.probes = bg,
min.incidence=2, label="hypo")
```

---

get.feature.probe *get.feature.probe to select probes within promoter regions or distal regions.*

Description

get.feature.probe is a function to select the probes falling into distal feature regions or promoter regions.

Usage

```
get.feature.probe(feature, TSS, TSS.range = list(upstream = 2000, downstream = 2000), promoter = F
```

Arguments

feature	A GRange object containing biofeature coordinates such as enhancer coordinates. Default is comprehensive genomic enhancer regions from REMC and FANTOM5 which is Union.enhancer data in **ELMER.data**. feature option is only usable when promoter option is FALSE.
TSS	A GRange object contains the transcription start sites. When promoter is FALSE, Union.TSS in **ELMER.data** will be used for default. When promoter is TRUE, UCSC gene TSS will be used as default (see detail). User can specify their own preference TSS annotation.
TSS.range	A list specify how to define promoter regions. Default is upstream =2000bp and downstream=2000bp.
promoter	A logical. If it is TRUE, function will ouput the promoter probes. If FALSE, function will ouput the distal probes overlaping with features. The default is FALSE.
rm.chr	A vector of chromosome. Once specified, the probes on these chromosome will be removed such as chrX chrY or chrM

Details

TSS: In order to get real distal probes, we use more comprehensive annotated TSS by both GENCODE and UCSC. However, to get probes within promoter regions need more accurate annotated TSS such as UCSC. Therefore, there are different settings for promoter and distal probe selection. But user can specify their own favorable TSS annotation. Then there won't be any difference between promoter and distal probe selection.

Value

A GRanges object contains the coordinate of probes which locate within promoter regions or distal feature regions such as union enhancer from REMC and FANTOM5.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
  # get distal enhancer probe
## Not run:
Probe <- get.feature.probe()

## End(Not run)
# get promoter probes
## Not run:
Probe <- get.feature.probe(promoter=TRUE)

## End(Not run)
# get distal enhancer probe remove chrX chrY
Probe2 <- get.feature.probe(rm.chr=c("chrX", "chrY"))
```

---

get.pair *get.pair to predict enhancer-gene linkages.*

Description

get.pair is a function to predict enhancer-gene linkages using associations between DNA methylation at enhancer CpG sites and expression of 20 nearby genes of the CpG sites (see reference). Two files will be saved if save is true: getPair.XX.all.pairs.statistic.csv and getPair.XX.pairs.significant.csv (see detail).

Usage

```
get.pair(mee, probes, nearGenes, percentage = 0.2, permu.size = 10000, permu.dir = NULL,
Pe = 0.001, dir.out = "./", diffExp = FALSE, cores = NULL, label = NULL, save=TRUE)
```

Arguments

mee	A MEE.data object contains at least meth, exp, probeInfo, geneInfo slots.
probes	A vector lists name of probes that need to be linked to genes.
nearGenes	A list (R object) containing output of GetNearGenes function or a path of XX.rda file containing output of GetNearGenes function.
percentage	A number ranges from 0 to 1 specifying the percentage of samples of control and experimental groups used to link probes to genes. Default is 0.2.
permu.size	A number specifies the number of permuation. Default is 1000.
permu.dir	A path shows the directory of permutation outputs.
Pe	A number specifies the empircal pvalue cutoff for defining signficant pairs. Default is 0.01
dir.out	A path specifies the directory for outputs of get.pair function. Default is current directory
diffExp	A logic. Default is FALSE. If TRUE, t test will be applied to test whether putative target gene are differentially expressed between two control and experimental groups.
cores	A interger which defines the number of cores to be used in parallel process. Default is NULL: no parallel process.
label	A character labels the outputs.
save	A logic. If save is true, two files will be saved for publication or analysis reusage purpose: getPair.XX.all.pairs.statistic.csv and getPair.XX.pairs.significant.csv (see detail)

Details

save: When save is TRUE, function will generate two XX.csv files. The first one is named getPair.XX.all.pairs.statistic.csv (XX depends on option label). This file contains all statistic results for each probe-gene pair. Based on this file, user can change different P value or sig.dir cutoff to select the significant results without redo the analysis. The second file is named getPair.XX.pairs.significant.csv (XX depends on option label). This file contains statistic results for the probes that pass the significant criteria (Pe). When save is FALSE, a data frame R object will be generate which contains the same information with the second file.

Value

A data frame contains statistic result for significant pairs

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
load(system.file("extdata","mee.example.rda",package = "ELMER"))
nearGenes <-GetNearGenes(TRange=getProbeInfo(mee,probe=c("cg00329272","cg10097755")),
                         geneAnnot=getGeneInfo(mee))
                       Hypo.pair <-get.pair(mee=mee,probes=c("cg00329272","cg10097755"),
                                            nearGenes=nearGenes,permu.size=5,Pe = 0.2,
                                            dir.out="./",
                                            label= "hypo")
```

`get.permu` *get.permu to generate permutation results for calculation of empirical P values for each enhancer-gene linkage.*

Description

get.permu is a function to use the same statistic model to calculate random enhancer-gene pairs. Based on the permutation value, empirical P value can be calculated for the real enhancer-gene pair (see reference).

Usage

```
get.permu(mee, geneID, percentage = 0.2, rm.probes = NULL, permu.size = 10000,
permu.dir = NULL, cores = NULL)
```

Arguments

`mee`	A MEE.data object contains at least meth, exp, probeInfo, geneInfo.
`geneID`	A vector lists gene id which need to be have permutation.
`percentage`	A number ranges from 0 to 1 specifying the percentage of samples of control and experimental groups used to link probes to genes. Default is 0.2.
`rm.probes`	A vector lists name of probes which belongs to the set of probes fed into `get.pair` function.
`permu.size`	A number specifies the number of permuation. Default is 1000.
`permu.dir`	A path shows the directory of permutation outputs
`cores`	A interger which defines the number of cores to be used in parallel process. Default is NULL: no parallel process.

Value

Certain number of permutation for each gene of interets.

Note

Permutation is the most time consuming step. It is recommended to use multiple cores for this step. Default permutation time is 1000 which may need 12 hrs by 4 cores. However 10,000 permutations is recommended to get high confidence results. But it may cost 2 days.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
load(system.file("extdata","mee.example.rda",package = "ELMER"))
permu <-get.permu(mee=mee,geneID=rownames(getExp(mee)),
                  rm.probes=c("cg00329272","cg10097755"),
                  permu.size=5)
```

`get.TFs` *get.TFs to identify regulatory TFs.*

Description

get.TFs is a function to identify regulatory TFs based on motif analysis and association analysis between the probes containing a particular motif and expression of all known TFs. If save is ture, two files will be saved: getTF.XX.significant.TFs.with.motif.summary.csv and getTF.hypo.TFs.with.motif.pvalue.rda (see detail).

Usage

```
get.TFs(mee, enriched.motif, TFs, motif.relavent.TFs, percentage = 0.2,
dir.out = "./", label = NULL, cores = NULL,save=TRUE)
```

Arguments

`mee`	A MEE.data object contains at least meth, exp, probeInfo, geneInfo.
`enriched.motif`	A list containing output of `get.enriched.motif` function or a path of XX.rda file containing output of get.enriched.motif function.
`TFs`	A data.frame containing TF GeneID and Symbol or a path of XX.csv file containing TF GeneID and Symbol. If missing, human.TF list will be used (human.TF data in **ELMER.data**). For detail information, refer the reference paper.
`motif.relavent.TFs`	A list containing motif as names and relavent TFs as contents for each list element or a path of XX.rda file containing a list as above. If missing, motif.relavent.TFs will be used (motif.relavent.TFs data in **ELMER.data**). For detail information, refer the reference paper.
`percentage`	A number ranges from 0 to 1 specifying the percentage of samples of control and experimental groups used to link probes to genes. Default is 0.2.
`dir.out`	A path specifies the directory for outputs of get.pair function. Default is current directory
`cores`	A interger which defines the number of cores to be used in parallel process. Default is NULL: no parallel process.
`label`	A character labels the outputs.
`save`	A logic. If save is ture, two files will be saved: getTF.XX.significant.TFs.with.motif.summary.csv and getTF.hypo.TFs.with.motif.pvalue.rda (see detail). If save is false, a data frame contains the same content with the first file.

Details

save: If save is ture, two files will be saved. The first file is getTF.XX.significant.TFs.with.motif.summary.csv (XX depends on option lable). This file contain the regulatory TF significantly associate with average DNA methylation at particular motif sites. The second file is getTF.hypo.TFs.with.motif.pvalue.rda (XX depends on option label). This file contains a matrix storing the statistic results for significant associations between TFs (row) and average DNA methylation at motifs (column). If save is false, a data frame which contains the same content with the first file will be reported.

Value

Potential responsible TFs will be reported.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
load(system.file("extdata","mee.example.rda",package = "ELMER"))
enriched.motif <- list("TP53"= c("cg00329272", "cg10097755", "cg08928189",
                                 "cg17153775", "cg21156590", "cg19749688", "cg12590404",
                                 "cg24517858", "cg00329272", "cg09010107", "cg15386853",
                                 "cg10097755", "cg09247779", "cg09181054"))
TF <- get.TFs(mee, enriched.motif,
              TFs=data.frame(GeneID=c("ID7157","ID8626","ID7161"),
              Symbol=c("TP53","TP63","TP73"),
              stringsAsFactors = FALSE),
              label="hypo")
```

---

get450K *get450K to download HM40K DNA methylation data for certain cancer types from TCGA website.*

Description

get450K is a function to download latest version of HM450K DNA methylation for all samples of certain cancer types from TCGA website.

Usage

```
get450K(disease, basedir = "./Data")
```

Arguments

disease	A character specifies the disease to download from TCGA such as BLCA
basedir	A path shows where the data will be stored.

Value

Download all DNA methylation from HM450K level 3 data for the specified disease.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

https://tcga-data.nci.nih.gov/tcga/tcgaHome2.jsp

`getClinic` *getClinic to download clinic data for certain cancer types from TCGA website.*

Description

getClinic is a function to download latest version of clinic data for all samples of certain cancer types from TCGA website.

Usage

```
getClinic(disease, basedir = "./Data")
```

Arguments

`disease`	A character specifies the disease to download from TCGA such as BLCA
`basedir`	A path shows where the data will be stored.

Value

Download all clinic information for the specified disease.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

https://tcga-data.nci.nih.gov/tcga/tcgaHome2.jsp

getExp *getExp to extract RNA expression slot from a MEE.data class object.*

Description

getExp is a function to easily extract RNA expression data out of a MEE.data object. By specifying geneID or ID for samples, a matrix of RNA expression data for defined genes and samples will be extract out of MEE.data object.

Usage

```
getExp(object, geneID, ID)

## S4 method for signature 'MEE.data'
getExp(object, geneID, ID)
```

Arguments

object	MEE.data object
geneID	A vector of genes' id. When specified, only these gene expressions will be output.
ID	A vector of sample ID. When specified, gene expression only for these samples will be output.

Value

A matrix of gene expression values.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
exp <- matrix(data=c(101:110),ncol=5,dimnames=list(c("gene1","gene2"),paste0("sample",1:5)))
mee <- fetch.mee(exp=exp)
Exp <- getExp(mee, geneID = "gene1") ## get gene expression for certain genes
Exp <- getExp(mee, ID = c("sample1","sample5")) ## get gene expression for certain samples
```

getGeneID *getGeneID to report gene id from symbol*

Description

getGeneID is a function to report the gene ids from gene symbols.

Usage

```
getGeneID(mee, symbol)
```

Arguments

mee	A MEE.data or Pair object.
symbol	A vector of characters which are gene symbols

Value

The gene ID for these gene symbols

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
geneInfo <- txs()
## input can be a path
pair <- fetch.pair(geneInfo=geneInfo)
getGeneID(pair, symbol="KIAA1804")
```

---

getGeneInfo *getGeneInfo to extract geneInfo slot from MEE.data or Pair object.*

Description

getGeneInfo is a function to easily extract geneInfo out of a MEE.data or Pair object. By specifying geneID or symbol, geneInfo for the defined genes (geneID or symbol) will be extracted out of MEE.data or Pair object. When range is specified, the geneInfo falling into the range will be extracted.

Usage

```
getGeneInfo(object, geneID, symbol, range)

## S4 method for signature 'ANY'
getGeneInfo(object, geneID, symbol, range)
```

Arguments

object	MEE.data or Pair object
geneID	A vector of genes' id. When specified, only these gene coordinates will be output.
symbol	A vector of genes' symbols . When specified, only these gene coordinates will be output.
range	A GRanges object. When specified, only the geneInfo locating within these loci will be output.

Value

Gene annotation information such as gene id, symbol and coordinates.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
geneInfo <- txs()
mee <- fetch.mee(geneInfo=geneInfo)
Genes <- getGeneInfo(mee, geneID = "55811")
Genes <- getGeneInfo(mee, symbol ="ADCY10")
Genes <- getGeneInfo(mee, range = GRanges(seqnames="chr1", ranges=IRanges(1000000,1600000)))
```

---

getMeth *getMeth to extract DNA methylation slot from a MEE.data class object.*

Description

getMeth is a function to easily extract DNA methylation data out of a MEE.data object. By specifying probe or ID for samples, a matrix of DNA methylation values for defined probes and samples will be extracted out of MEE.data object.

Usage

```
getMeth(object, probe, ID)

## S4 method for signature 'MEE.data'
getMeth(object, probe, ID)
```

Arguments

object	MEE.data object
probe	A vector of probes' name. When specified, DNA methylation only for these probes will be output.
ID	A vector of sample ID. When specified, DNA methylation only for these samples will be output.

Value

A matrix of DNA methylation values.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
meth <- matrix(data=c(1:20),ncol=5,dimnames=list(paste0("probe",1:4),paste0("sample",1:5)))
mee <- fetch.mee(meth=meth)
Meth <- getMeth(mee,probe = "probe1")
Meth <- getMeth(mee, ID = c("sample1","sample2"))
```

GetNearGenes *GetNearGenes to collect nearby genes for one locus.*

Description

GetNearGenes is a function to collect equal number of gene on each side of one locus.

Usage

```
GetNearGenes(geneAnnot = NULL, TRange = NULL, geneNum = 20,
  cores = NULL)
```

Arguments

geneAnnot	A GRange object contains coordinates of promoter for human genome.
TRange	A GRange object contains coordinates of a list of target loci.
geneNum	A number determines how many gene will be collected totally. Then the number devided by 2 is the number of genes collected from each side of targets (number shoule be even) Default to 20.
cores	A interger which defines the number of cores to be used in parallel process. Default is NULL: no parallel process.

Value

A data frame of nearby genes' information: genes' IDs, genes' symbols, distance with target and side to which the gene locate to the target.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
geneAnnot <- txs(TSS=list(upstream=0, downstream=0))
probe <- GRanges(seqnames = c("chr1","chr2"),
range=IRanges(start = c(16058489,236417627), end= c(16058489,236417627)),
name= c("cg18108049","cg17125141"))
NearbyGenes <- GetNearGenes(geneNum=20,geneAnnot=geneAnnot,TRange=probe)
```

getPair *getPair to extract pairInfo slot from Pair object.*

Description

getPair is a function to easily extract pairInfo out of a Pair object. By specifying geneID or probe, geneInfo for the defined genes (geneID) and probes (probe) will be extracted out of Pair object.

Usage

```
getPair(object, probe, geneID)

## S4 method for signature 'Pair'
getPair(object, probe, geneID)
```

Arguments

object	Pair object
probe	A vector of probes' name. When specified, only the pair containing these probes will be output.
geneID	A vector of genes' id. When specified, only the pair containing these genes will be output.

Value

Pair information such as empirical P values, probe and gene ID.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
df <- data.frame(Probe=c("cg19403323","cg12213388","cg26607897"),
GeneID =c("ID255928","ID84451","ID55811"),
Symbol =c("SYT14","KIAA1804","ADCY10"),
Pe=c(0.003322259,0.003322259,0.003322259))
pair <- fetch.pair(pair = df)
Pairs <- getPair(pair, probe = "cg19403323") # get pair information for a probe
Pairs <- getPair(pair, geneID = "ID55811") # get pair information for a gene
```

getProbeInfo *getProbeInfo to extract probeInfo slot from MEE.data or Pair object.*

Description

getProbeInfo is a function to easily extract probeInfo out of a MEE.data or Pair object. By specifying probe, probeInfo for the defined set of probes will be extracted out of MEE.data or Pair object. Option chr will restrain the output probeInfo to certain chromosomes. When range is specified, the probeInfo falling into the range will be extracted.

Usage

```
getProbeInfo(object, chr, probe, range)

## S4 method for signature 'ANY'
getProbeInfo(object, chr, probe, range)
```

Arguments

object	MEE.data or Pair object
chr	A vector of chromosome such chr1, chr2. When specified, only the probeInfo locating on these chromosome will be output.
probe	A vector of probes' name. When specified, only these probes' coordinate will be output.
range	A GRanges object. When specified, only probeInfo locating within these loci will be output.

Value

Probe information such as names, coordinates.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
probeInfo <- GRanges(seqnames = c("chr1","chr1","chr3"),
ranges = IRanges(start = c(1,6,20),end = c(2,7,21)),
name=c("cg1","cg2","cg3"))
mee <- fetch.mee(probeInfo=probeInfo)
Probes <- getProbeInfo(mee,chr="chr1") # get probes which locate on the chr1
Probes <- getProbeInfo(mee, probe = "cg1") # get certain probes information
Probes <- getProbeInfo(mee, range = GRanges(seqnames="chr1", ranges=IRanges(5,20)))
```

`getRNAseq` *getRNAseq to download all RNAseq data for a certain cancer type from TCGA.*

Description

getRNAseq is a function to download RNAseq data for all samples of a certain cancer type from TCGA

Usage

```
getRNAseq(disease, basedir = "./Data")
```

Arguments

`disease`	A character specifies disease in TCGA such as BLCA
`basedir`	A path shows where the data will be stored.

Value

Download all RNA seq level 3 data for the specified disease.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

https://tcga-data.nci.nih.gov/tcga/tcgaHome2.jsp

`getSample` *getSample to extract sample information from MEE.data object.*

Description

getSample is a function to easily extract sample information from MEE.data object. By specifying ID for samples, only that set of samples' informtion will be extracted. When certain columns of data need to be extracted, just specify columns names in cols option and sample information for wanted columns will be reported.

Usage

```
getSample(object, ID, cols)

## S4 method for signature 'MEE.data'
getSample(object, ID, cols)
```

Arguments

object	MEE.data object
ID	A vector of sample ID. When specified, sample informtion only for these samples will be output.
cols	A vector of column names of sampleInfo slots of MEE.data object.

Value

Sample information.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
SampleInfo <- data.frame(ID=paste0("sample",1:5),
TN=c("Tumor","Tumor","Normal","Normal","Tumor"))
mee <- fetch.mee(sample = SampleInfo)
Samples <- getSample(mee,ID = "sample2") ## get sample2's information
Samples <- getSample(mee, cols = "TN")  ## get 'TN' information for each samples
```

---

getSymbol *getSymbol to report gene symbol from id*

Description

getSymbol is a function to report the gene symbols from gene IDs.

Usage

```
getSymbol(mee, geneID)
```

Arguments

mee	A MEE.data or Pair object.
geneID	A character which is the geneID

Value

The gene symbol for input genes.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
geneInfo <- txs()
## input can be a path
pair <- fetch.pair(geneInfo=geneInfo)
getSymbol(pair, geneID="84451")
```

getTCGA *getTCGA to download DNA methylation, RNA expression and clinic data for all samples of certain cancer type from TCGA.*

Description

getTCGA is a function to download DNA methylation, RNA expression and clinic data for all samples of certain cancer type from TCGA website. And downloaded data will be transform to matrixes or data frame for further analysis.

Usage

```
getTCGA(disease, Meth = TRUE, RNA = TRUE, Clinic = TRUE,
  basedir = "./Data", RNAtype = "gene", Methfilter = 0.2)
```

Arguments

disease	A character specifies the disease to download in TCGA such as BLCA
Meth	A logic if TRUE HM450K DNA methylation data will download.
RNA	A logic if TRUE RNA-seq Hiseq-V2 from TCGA level 3 will be download.
Clinic	A logic if TRUE clinic data will be download for that disease.
basedir	A path shows where the data will be stored.
RNAtype	A charactor to specify whether use isoform level or gene level. When RNAtype=gene, gene level gene expression will be used. When isoform, then isoform data will be used.
Methfilter	A number. For each probe, the percentage of NA among the all the samples should smaller than Methfilter.

Value

Download DNA methylation (HM450K)/RNAseq(HiseqV2)/Clinic data for the specified disease from TCGA.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

https://tcga-data.nci.nih.gov/tcga/tcgaHome2.jsp

Examples

```
getTCGA("BRCA",Meth=FALSE, RNA=FALSE, Clinic=TRUE, basedir="~")
```

MEE.data-class *MEE.data An S4 class contains 5 slots: methylation, expression, sample information, probe information and gene information. MEE.data class are the main input for main functions.*

Description

MEE.data An S4 class contains 5 slots: methylation, expression, sample information, probe information and gene information. MEE.data class are the main input for main functions.

Slots

`meth` A matrix of DNA methylation. Each row is one probe and each column is one sample

`exp` A matrix of expression. Each row is one gene and each column is one sample

`sample` A data.frame contains sample information

`probeInfo` A GRange object contains probe information

`geneInfo` A GRange object contains gene information

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

See Also

`get.diff.meth`, `get.pair`, `get.TFs`, `get.permu`, `getMeth`,`getExp`, `getGeneInfo`, `getProbeInfo`, `getSample`,`fetch.mee`,`scatter.plot`

---

motif.enrichment.plot *motif.enrichment.plot to plot bar plots showing motif enrichment ORs and 95% confidence interval for ORs*

Description

motif.enrichment.plot to plot bar plots showing motif enrichment ORs and 95% confidence interval for ORs. Option motif.enrichment can be a data frame generated by `get.enriched.motif` or a path of XX.csv saved by the same function.

Usage

```
motif.enrichment.plot(motif.enrichment, significant = NULL, dir.out = "./",
  save = TRUE, label = NULL)
```

Arguments

motif.enrichment	A data frame or a file path of get.enriched.motif output motif.enrichment.csv file.
significant	A list to select subset of motif. Default is NULL.
dir.out	A path specifies the directory to which the figures will be saved. Current directory is default.
save	A logic. If true (default), figure will be saved to dir.out.
label	A character labels the output figures.

Value

A figure shows the enrichment level for selected motifs.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
motif.enrichment <- data.frame(motif=c("TP53","NR3C1","E2F1","EBF1","RFX5",
"ZNF143", "CTCF"),
OR=c(19.33,4.83,1, 4.18, 3.67,3.03,2.49),
lowerOR =c(10,3,1.09,1.9,1.5,1.5, 0.82),
upperOR =c(23,5,3,7,6,5,5),
stringsAsFactors=FALSE)
motif.enrichment.plot(motif.enrichment=motif.enrichment,
                      significant=list(OR=3),
                      label="hypo", save=FALSE)
```

Pair-class *An S4 class that pairs information, probe information and gene information.*

Description

An S4 class that pairs information, probe information and gene information.

Slots

pairInfo A data.frame

probeInfo A GRanges object.

geneInfo A GRanges object.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

See Also

schematic.plot, getGeneInfo, getProbeInfo, fetch.pair

---

promoterMeth *promoterMeth to calculate associations of gene expression with DNA methylation at promoter regions.*

Description

promoterMeth is a function to calculate associations of gene expression with DNA methylation at promoter regions.

Usage

```
promoterMeth(mee, sig.pvalue = 0.01, percentage = 0.2, save = TRUE)
```

Arguments

mee	A MEE.data object must contains four components: meth, exp, probeInfo, geneInfo.
sig.pvalue	A number specifies significant cutoff for gene silenced by promoter methylation. Default is 0.01. P value is raw P value without adjustment.
percentage	A number ranges from 0 to 1 specifying the percentage of samples of control and experimental groups used to link promoter DNA methylation to genes. Default is 0.2.
save	A logic. If it is true, the result will be saved.

Value

A data frame contains genes whose expression significantly anti-correlated with promoter methylation.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

References

Yao L, Shen H, Laird PW, Farnham PJ,Berman BP: Inferring Regulatory Element Landscapes and Transcription Factor Networks from Cancer Methylomes. in revision of Genome Biology

Examples

```
load(system.file("extdata","mee.example.rda",package = "ELMER"))
Gene.promoter <- promoterMeth(mee)
```

scatter.plot *scatter.plot to plot scatter plots between gene expression and DNA methylation.*

Description

scatter.plot is a function to plot various scatter plots between gene expression and DNA methylation. When byPair is specified, scatter plot for individual probe-gene pairs will be generated. When byProbe is specified, scatter plots for one probes with nearby 20 gene pairs will be generated. When byTF is specified, scatter plot for TF expression and average DNA methylation at certain motif sites will be generated.

Usage

```
scatter.plot(mee, byPair = list(probe = c(), gene = c()),
  byProbe = list(probe = c(), geneNum = 20), byTF = list(TF = c(), probe =
  c()), category = NULL, dir.out = "./", save = TRUE, ...)
```

Arguments

mee	A MEE.data object includes DNA methylation data, expression data, probeInfo and geneInfo.
byPair	A list: byPair =list(probe=c(),gene=c()); probe contains a vector of probes' name and gene contains a vector of gene IDs. The length of probe should be the same with length of gene.
byProbe	A list byProbe =list(probe=c(), geneNum=20); probe contains a vector of probes' name and geneNum specify the number of gene near the probes will ploted. 20 is default for geneNum.
byTF	A list byTF =list(TF=c(), probe=c()); TF contains a vector of TF's symbol and probe contains the a vector of probes' name.
category	A vector labels subtype of samples or a character which is the column name in the sampleInfo in the MEE.data object. Once specified, samples will label different color. The color can be customized by using color.value.
dir.out	A path specifies the directory to which the figures will be saved. Current directory is default.
save	A logic. If true, figure will be saved to dir.out.
...	color.value, lm_line in scatter function

Value

Scatter plots.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
load(system.file("extdata","mee.example.rda",package = "ELMER"))
scatter.plot(mee,byProbe=list(probe=c("cg19403323"),geneNum=20),
          category="TN", save=FALSE)
scatter.plot(mee,byProbe=list(probe=c("cg19403323"),geneNum=20),
          category="TN", save=TRUE) ## save to pdf
# b. generate one probe-gene pair
scatter.plot(mee,byPair=list(probe=c("cg19403323"),gene=c("ID255928")),
           category="TN", save=FALSE,lm_line=TRUE)
```

`schematic.plot` *schematic.plot to plot schematic plots showing the locations of genes and probes.*

Description

schematic.plot is a function to plot schematic plots showing the locations of genes and probes.

Usage

```
schematic.plot(pair, byProbe, byGene, byCoordinate = list(chr = c(), start =
  c(), end = c()), dir.out = "./", save = TRUE, ...)
```

Arguments

`pair`	A Pair object. All slots of Pair class should be included
`byProbe`	A vector of probe names.
`byGene`	A vector of gene ID
`byCoordinate`	A list contains chr, start and end. byCoordinate=list(chr=c(),start=c(),end=c()).
`dir.out`	A path specifies the directory for outputs. Default is current directory
`save`	A logic. If true, figures will be saved to dir.out.
`...`	Parameters for GetNearGenes function. See `GetNearGenes`

Details

byProbes: When a vector of probes' name are provided, function will produce schematic plots for each individual probes. The schematic plot contains probe, nearby 20 (or the number of gene user specified.) genes and the significantly linked gene to the probe.

byGene: When a vector of gene ID are provided, function will produce schematic plots for each individual genes. The schematic plot contains the gene and all the significantly linked probes.

byCoordinate: When a genomic coordinate is provided, function will produce a schematic plot for this coordinate. The schematic plot contains all genes and significantly linked probes in the range and the significant links.

Value

Schematic plots will be produced.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
library(grid)
load(system.file("extdata","mee.example.rda",package = "ELMER"))
nearGenes <-GetNearGenes(TRange=getProbeInfo(mee,probe=c("cg00329272","cg19403323")),
                         geneAnnot=getGeneInfo(mee))
Hypo.pair <-get.pair(mee=mee,probes=c("cg00329272","cg19403323"),
                     nearGenes=nearGenes,permu.size=5,Pe = 0.2,dir.out="./",
                     label= "hypo")
pair <- fetch.pair(pair=Hypo.pair,
                   probeInfo = getProbeInfo(mee),
                   geneInfo = getGeneInfo(mee))
# a. generate schematic plot of one probe with nearby 20 genes and label
#the gene significantly linked with the probe.
grid.newpage()
schematic.plot(pair=pair, byProbe="cg19403323" ,save=FALSE)
#b. generate schematic plot of ont gene with the probe which the gene significanlty linked to.
grid.newpage()
schematic.plot(pair=pair, byGene="ID255928",save=FALSE)
```

`TCGA.pipe` *ELMER analysis pipeline for TCGA data.*

Description

ELMER analysis pipeline for TCGA data. This pipeline combine every steps of **ELMER** analyses: get.feature.probe, get.diff.meth, get.pair, get.permu, get.enriched.motif and get.TFs. Every steps' results are saved.

Usage

```
TCGA.pipe(disease, analysis = "all", wd = "./", cores = NULL,
  Data = NULL, diff.dir = "hypo", ...)
```

Arguments

`disease`	TCGA disease name in short form such as COAD
`analysis`	A vector of characters listing the analysis need to be done. Analysis can be "download","distal.probes","diffMeth","pair","motif","TF.search". Default is "all" meaning all the analysis will be processed.
`wd`	A path shows working dirctory. Default is "./"
`cores`	A interger defines number of core to be used in parallel process. Default is NULL: don't use parallel process.
`Data`	A path shows the folder containing DNA methylation, expression and clinic data
`diff.dir`	A character can be "hypo" or "hyper", showing dirction DNA methylation changes. If it is "hypo", get.diff.meth function will identify all significantly hypomethylated CpG sites; If "hyper", get.diff.meth function will identify all significantly hypermethylated CpG sites

...	A list of parameters for functions: GetNearGenes, get.feature.probe, get.diff.meth, get.pair,

Value

Different analysis results.

Examples

```
## Not run:
distal.probe <- TCGA.pipe(disease = "LUSC", analysis="Probe.selection", wd="~/")

## End(Not run)
```

`TF.rank.plot` *TF.rank.plot to plot the scores (-log10(P value)) which assess the correlation between TF expression and average DNA methylation at motif sites.*

Description

TF.rank.plot is a function to plot the scores (-log10(P value)) which assess the correlation between TF expression and average DNA methylation at motif sites. The the motif relevant TF and top3 TFs will be labeled in a different color.

Usage

```
TF.rank.plot(motif.pvalue, motif, TF.label, dir.out = "./", save = TRUE)
```

Arguments

`motif.pvalue`	A matrix or a path specifying location of "XXX.with.motif.pvalue.rda" which is output of get.TFs.
`motif`	A vector of charactors specify the motif to plot
`TF.label`	A list shows the labels for each motif. If TF.label is not specified, the motif relevant TF and top3 TF will be labeled.
`dir.out`	A path specify the directory to which the figures will be saved. Current directory is default.
`save`	A logic. If true (default), figure will be saved to dir.out.

Value

A plot shows the score (-log(P value)) of association between TF expression and DNA methylation at sites of a certain motif.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
load(system.file("extdata","getTF.hypo.TFs.with.motif.pvalue.rda",package="ELMER"))
TF.rank.plot(motif.pvalue=TF.meth.cor, motif="TP53", TF.label=list(TP53=c("TP53","TP63","TP73")),
            save=FALSE)
```

`txs` *txs to fetch USCS gene annotation (transcripts level) from Bioconductor package Homo.sapians. If upstream and downstream are specified in TSS list, promoter regions of USCS gene will be generated.*

Description

txs is a function to fetch USCS gene annotation (transcripts level) from Bioconductor package Homo.sapians. If upstream and downstream are specified in TSS list, promoter regions of USCS gene will be generated.

Usage

```
txs(TSS = list(upstream = NULL, downstream = NULL))
```

Arguments

TSS	A list contains upstream and downstream like TSS=list(upstream, downstream). When upstream and downstream is specified, coordinates of promoter regions with gene annotation will be generated.

Value

UCSC gene annotation if TSS is not specified. Coordinates of UCSC gene promoter regions if TSS is specified.

Author(s)

Lijing Yao (maintainer: lijingya@usc.edu)

Examples

```
# get UCSC gene annotation (transcripts level)
txs <- txs()
# get coordinate of all UCSC promoter regions +/-1000bp of TSSs
## Not run:
txs <- txs(TSS=list(upstream=1000, downstream=1000))

## End(Not run)
```

Index