PatientGeneSets

April 20, 2011

`cma.scores` *Cancer Mutation Prevalence Analysis Scores*

Description

Computes Gene-specific Scores for Cancer Mutation Prevalence Analysis.

Usage

```
cma.scores(cma.data,
   passenger.rates = t(data.frame(0.55*rep(1.0e-6,25))),
           number.genes,
           compute.poisson.BF=FALSE,
           compute.binomial.posterior=FALSE,
           allow.separate.rates = TRUE,
           filter.above=0,
           filter.below=0,
           filter.threshold=0,
   filter.mutations=0,
           aa=1e-10,
           bb=1e-10,
           priorH0=1-300/13020,
           prior.a0=100,
           prior.a1=5,
           prior.fold=10)
```

Arguments

`cma.data`	Data frame with mutation information broken down by gene, phase and mutation type. See `WoodMutationsBreast` for an example.
`passenger.rates`	Data frame of passenger mutation rates per nucleotide, by type, or "context". Columns denote types and must be in the same order as the first 25 columns in cma.data objects. If two rows are present, they must have row names "Discovery" and "Validation"
`number.genes`	The total number of genes analyzed, including those for whom no mutation were found.

`compute.poisson.BF`
: If TRUE, computes Bayes Factors (BF) using a Poisson model for mutation counts and a gamma priors on rates.

`compute.binomial.posterior`
: If TRUE, computes the posterior probability that a gene's mutation rates above the specified passenger rates using a binomial model.

`allow.separate.rates`
: If TRUE, allows for use separate rates for discovery and validation screens.

`filter.threshold`
: This and the following three input control filtering of genes, allowing to exclude genes from analysis, by size and number of mutations. Different criteria can be set above and below this threshold. The threshold is a gene size in base pairs.

`filter.above` Minimum number of mutations per Mb, applied to genes of size greater than `threshold.size`.

`filter.below` Minimum number of mutations per Mb, applied to genes of size lower than `threshold.size`.

`filter.mutations`
: Only consider genes whose total number of mutations is greater than or equal to `filter.mutations`.

`aa` Hyperparameter of beta prior used in compute.binomial.posterior.

`bb` Hyperparameter of beta prior used in compute.binomial.posterior

`priorH0` Prior probability of the null hypothesis, used to convert the BF in compute.poisson.BF to a posterior probability

`prior.a0` Shape hyperparameter of gamma prior on passenger rates used in compute.poisson.BF

`prior.a1` Shape hyperparameter of gamma prior on non-passenger rates used in compute.poisson.BF

`prior.fold` Hyperparameter of gamma prior on non-passenger rates used compute.poisson.BF. The mean of the gamma is set so that the ratio of the mean to the passenger rate is the specified `prior.fold` in each type.

Details

The scores computed by this function are relevant for two stage experiments like the one in the Sjoeblom article. In this design genes are sequenced in a first "discovery" sample. Genes in which mutations are found are also sequenced in a subsequent "validation" screen. The goal of this tool is to facilitate reanalysis of the Sjoeblom dataset. Application to other projects requires a detailed understanding of the Sjoeblom project.

Value

A data frame giving gene-by-gene values for each score. The columns in this data frame are:

`CaMP` The CaMP score of Sjoeblom and colleagues.

`neglogPg` The negative log10 of Pg, where Pg represents the probability that a gene has its exact observed mutation profile under the null, i.e. assuming the given passenger rates.

`logLRT` The log10 of the likelihood ratio test (LRT).

`logitBinomialPosteriorDriver`
: logit of the posterior probability that a gene's mutation rates above the specified passenger rates using a binomial model

`PoissonlogBF` The log10 of the Bayes Factor (BF) using a Poisson-Gamma model.

`PoissonPosterior`
: The posterior probability that a given gene is a driver, using a Poisson-Gamma model.

`Poissonlmlik0`
: Marginal likelihood under the null hypothesis in the Poisson-Gamma model

`Poissonlmlik1`
: Marginal likelihood under the alternative hypothesis in the Poisson-Gamma model

Author(s)

Giovanni Parmigiani, Simina M. Boca

References

Parmigiani G, Lin J, Boca S, Sjoeblom T, Kinzler WK, Velculescu VE, Vogelstein B. Statistical methods for the analysis of cancer genome sequencing data. http://www.bepress.com/jhubiostat/paper126/

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science.* DOI: 10.1126/science.1164382

Sjoeblom T, Jones S, Wood LD, Parsons DW, Lin J, Barber T, Mandelker D, Leary R, Ptak J, Silliman N, et al. The consensus coding sequences of breast and colorectal cancers. *Science.* DOI: 10.1126/science.1133427

Wood LD, Parsons DW, Jones S, Lin J, Sjoeblom, Leary RJ, Shen D, Boca SM, Barber T, Ptak J, et al. The Genomic Landscapes of Human Breast and Colorectal Cancer. *Science.* DOI: 10.1126/science.1145720

See Also

`MutationsBrain`, `GeneSizes08`, `do.gene.set.analysis`

Examples

```
## Not run:  data(Parsons)
ScoresBrain <- cma.scores(cma.data=MutationsBrain,
                          number.genes=nrow(GeneSizes08))

## End(Not run)
```

`combine.sims` *Combines two SetMethodSims objects.*

Description

This function is used to combine two `SetMethodSims` objects, which have the results from simulated datasets, provided that the values for `pass.null, perc.samples,` and `spiked.set.sizes` match up when the objects are generated with the `sim.data.p.values` function.

Usage

```
combine.sims(obj1, obj2)
```

Arguments

obj1	Object of the class SetMethodsSims.
obj2	Object of the class SetMethodsSims.

Value

An object of the class SetMethodsSims. See SetMethodsSims for more details.

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

SetMethodsSims-class, sim.data.p.values

Examples

```
## Not run:
##Note that this takes a few minutes to run:
library(KEGG.db)
data(Parsons)
data(ID2name)

set.seed(831984)

resultsSim <-
    sim.data.p.values(EventsBySample = EventsBySampleBrain,
                      Mutations = MutationsBrain,
                      GeneSizes = GeneSizes08,
                      Coverage = CoverageBrain,
      GeneSets = KEGGPATHID2EXTID[c("hsa05213",
                      "hsa05223", "hsa00250")],
      ID2name = ID2name,
                      nr.iter = 2,
                      pass.null = TRUE,
                      perc.samples = c(75, 95),
                      spiked.set.sizes = c(50),
                      show.iter = TRUE,
                      gene.method = FALSE,
                      perm.null.method = TRUE,
                      perm.null.het.method = FALSE,
                      pass.null.method = TRUE,
                      pass.null.het.method = FALSE)

resultsSim
```

```
extract.sims.method(resultsSim, resultsSim)

## End(Not run)
```

`CoverageBrain` *Data from Parsons et al. study: Total number of nucleotides "at risk"*

Description

Total numbers of nucleotides "at risk" that were successfully sequenced in RefSeq genes in the Parsons et al. glioblastoma study.

Usage

```
data(Parsons)
```

Format

Total number of nucleotides available for mutations in the glioblastoma study from Parsons et al., broken down by gene, study phase (Discovery or Validation), and mutation type. For this study, there was only a Discovery stage. The nucleotides availables for indels are all the successfully sequenced nucletides in a gene. The nucleotides availables for other mutations are excluding nucleotides who can only give rise to synonymous mutations. It also includes the total number of samples analyzed in each phase for each gene.

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science.* DOI: 10.1126/science.1164382

Parmigiani G, Lin J, Boca S, Sjoeblom T, Kinzler WK, Velculescu VE, Vogelstein B. Statistical methods for the analysis of cancer genome sequencing data. http://www.bepress.com/jhubiostat/paper126/

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

`do.gene.set.analysis`, `sim.data.p.values`, `SimMethodsSims-class`, `EventsBySampleBrain`, `GeneSizes08`, `MutationsBrain`

`do.gene.set.analysis` *Implements gene-set analysis methods.*

Description

This function implements the gene-set analysis methods. It returns a data-frame with p-values and q-values for all the methods selected.

Usage

```
do.gene.set.analysis(EventsBySample,
     Scores,
     GeneSizes,
     GeneSets,
     passenger.rates = t(data.frame(0.55*rep(1.0e-6,25))),
     Coverage,
     ID2name,
     BH = TRUE,
     gene.method = TRUE,
     perm.null.method = TRUE,
     perm.null.het.method = TRUE,
     pass.null.method = TRUE,
     pass.null.het.method = TRUE)
```

Arguments

`EventsBySample`	Data frame giving the specific mutations for each gene and each tumor sample. See `EventsBySampleBrain` for an example.
`Scores`	Data frame of gene scores. The logLRT scores are used for the gene.method option. It can be the output of `cma.scores`. If the gene.method option is set to FALSE, this parameter is not needed.
`GeneSizes`	Data frame of gene sizes. See `GeneSizes08` object for an example.
`GeneSets`	An object which annotates genes to gene-sets; it can either be a list with each component representing a set, or an object of the class AnnDbBimap.
`passenger.rates`	Data frame with 1 row and 25 columns, of passenger mutation rates per nucleotide, by type, or "context". Columns denote types and must be in the same order as the first 25 columns in the `MutationsBrain` objects.
`Coverage`	Data frame with coverage information, by gene, phase, and type. See `CoverageBrain` for an example.
`ID2name`	Vector mapping the gene identifiers used in the GeneSets object to the gene names used in the other objects; if they are the same, this parameter is not needed. See `ID2name` for an example.
`BH`	If set to `TRUE`, uses the Benjamini-Hochberg method to get q-values; if set to `FALSE`, uses the Storey method from the `qvalue` package.
`gene.method`	If set to `TRUE`, implements gene-oriented method.

`perm.null.method`
: If set to `TRUE`, implements patient-oriented method with permutation null and no heterogeneity.

`perm.null.het.method`
: If set to `TRUE`, implements patient-oriented method with permutation null and heterogeneity.

`pass.null.method`
: If set to `TRUE`, implements patient-oriented method with passenger null and no heterogeneity.

`pass.null.het.method`
: If set to `TRUE`, implements patient-oriented method with passenger null and heterogeneity.

Value

A data frame, with the rows representing set names and the columns representing the p-values and q-values corresponding to the different methods.

Author(s)

Simina M. Boca, Giovanni Parmigiani, Luigi Marchionni, Michael A. Newton.

References

Boca SM, Kinzler K, Velculescu VE, Vogelstein B, Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

Parmigiani G, Lin J, Boca S, Sjoeblom T, Kinzler WK, Velculescu VE, Vogelstein B. Statistical methods for the analysis of cancer genome sequencing data. http://www.bepress.com/jhubiostat/paper126/

Benjamini Y and Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57:289-300, 995.

Storey JD and Tibshirani R. Statistical significance for genome-wide experimens. *Proceedings of the National Academy of Sciences.* DOI: 10.1073/pnas.1530509100

Schaeffer EM, Marchionni L, Huang Z, Simons B, Blackman A, Yu W, Parmigiani G, Berman DM. Androgen-induced programs for prostate epithelial growth and invasion arise in embryogenesis and are reactivated in cancer. *Oncogene.* DOI: 10.1038/onc.2008.327

Thomas MA, Taub AE. Calculating binomial probabilities when the trial probabilities are unequal. *Journal of Statistical Computation and Simulation.* DOI: 10.1080/00949658208810534

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science.* DOI: 10.1126/science.1164382

Wood LD, Parsons DW, Jones S, Lin J, Sjoeblom, Leary RJ, Shen D, Boca SM, Barber T, Ptak J, et al. The Genomic Landscapes of Human Breast and Colorectal Cancer. *Science.* DOI: 10.1126/science.1145720

See Also

`CoverageBrain`, `EventsBySampleBrain`, `GeneSizes08`, `MutationsBrain`, `ID2name`

Examples

```
library(KEGG.db)
data(Parsons)
data(ID2name)

resultsBrain <- do.gene.set.analysis(EventsBySample = EventsBySampleBrain,
 GeneSizes = GeneSizes08, GeneSets = KEGGPATHID2EXTID[c("hsa05213",
 "hsa05223", "hsa00250")], Coverage = CoverageBrain, ID2name = ID2name,
 gene.method = FALSE, perm.null.method = TRUE, perm.null.het.method = FALSE,
 pass.null.method = TRUE, pass.null.het.method = FALSE)

resultsBrain
```

EventsBySampleBrain

Data from Parsons et al. study: Mutation types for every gene and sample

---

Description

All mutation types for each gene and tumor sample from the Parsons et al. glioblastoma study.

Usage

```
data(Parsons)
```

Format

Data frame giving the specific mutations for each gene and each tumor sample. It has 4 columns: "Event" (which should have the values "Mut" for "mutation"), "Sample" (which gives the name of the sample, for example "BR11P"), "Symbol" (which gives the gene symbol, for example "MRPL55"), and "MutationClass" (which gives the type of mutation, for example "C.in.CpG.to.T" means that a cytosine within a CpG island was mutated to a thymine).

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science.* DOI: 10.1126/science.1164382

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

`do.gene.set.analysis`, `sim.data.p.values`, `SimMethodsSims-class`, `CoverageBrain`, `GeneSizes08`, `MutationsBrain`

`extract.sims.method` *Extracts the p-values or q-values from a SetMethodsSims object for a specific method.*

Description

This function is used to obtain a single data frame with the p-values or q-values from one of the specific gene-set analysis methods, from a `SetMethodsSims` object which has the results from simulated datasets.

Usage

```
extract.sims.method(object, method)
```

Arguments

`object`	Object of the class `SetMethodsSims`.
`method`	Character string giving the method used for extraction, and whether p-values or q-values are extracted. The string should be one of the column names of the data frame resulting from the `do.gene.set.analysis` function.

Value

An object of the class `SetMethodsSims`. See `SetMethodsSims` for more details.

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

`SetMethodsSims-class`, `sim.data.p.values`, `do.gene.set.analysis`

Examples

```
## Not run:
##Note that this takes a few minutes to run:
library(KEGG.db)
data(Parsons)
data(ID2name)

set.seed(831984)

resultsSim <-
    sim.data.p.values(EventsBySample = EventsBySampleBrain,
                      Mutations = MutationsBrain,
                      GeneSizes = GeneSizes08,
```

```
                    Coverage = CoverageBrain,
        GeneSets = KEGGPATHID2EXTID[c("hsa05213",
                    "hsa05223", "hsa00250")],
        ID2name = ID2name,
                    nr.iter = 2,
                    pass.null = TRUE,
                    perc.samples = c(75, 95),
                    spiked.set.sizes = c(50),
                    show.iter = TRUE,
                    gene.method = FALSE,
                    perm.null.method = TRUE,
                    perm.null.het.method = FALSE,
                    pass.null.method = TRUE,
                    pass.null.het.method = FALSE)

resultsSim

extract.sims.method(resultsSim, "p.values.perm.null")

## End(Not run)
```

`GeneSizes08` *Data from Parsons et al. study: Sizes and composition of RefSeq genes*

Description

Sizes and composition of RefSeq genes from the Parsons et al. glioblastoma study.

Usage

```
data(Parsons)
```

Format

Data frame of RefSeq gene sizes by gene and type. Entries are total nucleotides in each type, per gene. Here and throughout transcript names are `rownames`. Each column represents one of 9 contexts (for example, the "C.in.CpG" column gives the number of nucleotides in each transcript which are cytosines and are located within CpG islands).

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science.* DOI: 10.1126/science.1164382

Parmigiani G, Lin J, Boca S, Sjoeblom T, Kinzler WK, Velculescu VE, Vogelstein B. Statistical methods for the analysis of cancer genome sequencing data. http://www.bepress.com/jhubiostat/paper126/

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

do.gene.set.analysis, sim.data.p.values, SimMethodsSims-class, CoverageBrain, EventsBySampleBrain, MutationsBrain

---

ID2name *Map of gene IDs to gene names*

Description

Entrez gene identifiers used in the KEGG.db package are mapped to the gene names used in the data from the Parsons et al. study.

Usage

```
data(ID2name)
```

Format

Vector having as names the Entrez gene identifiers used in the KEGG.db package and as entries the gene names used in the data objects available through data(Parsons).

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

do.gene.set.analysis, sim.data.p.values

MutationsBrain *Data from Parsons et al. study: Mutation counts*

Description

Mutation counts for RefSeq genes which are mutated in the Parsons et al. glioblastoma study.

Usage

```
data(Parsons)
```

Format

Data frame of number of mutations found in the glioblastoma dataset from the Parsons et al. study, broken down by gene, study phase (Validation or Discovery), and mutation type. For this study, there was only a Discovery stage. Entries are totals over all the samples sequenced in each phase. For convenience the data frame also replicates the Coverage and Gene Size information for the genes where mutations were found, and includes the total number of samples analyzed in each phase for each gene.

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science*. DOI: 10.1126/science.1164382

Parmigiani G, Lin J, Boca S, Sjoeblom T, Kinzler WK, Velculescu VE, Vogelstein B. Statistical methods for the analysis of cancer genome sequencing data. http://www.bepress.com/jhubiostat/paper126/

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

`do.gene.set.analysis`, `sim.data.p.values`, `SimMethodsSims-class`, `CoverageBrain`, `EventsBySampleBrain`, `GeneSizes08`

`SetMethodsSims-class` *Class representation for depositing output from simulations.*

Description

Stores results from the `sim.data.p.values` function.

Objects from the class

New objects can be created using calls of the form `new("SetMethodsSims", null.dist, perc.samples, spiked.set.sizes, GeneSizes, GeneSets, Coverage, EventsBySample, Mutations, Scores, results)`

Slots

`null.dist`: Object of class `"character"`. Can be either "Passenger null" or "Permutation null," depending on what method is used to get the null data.

`perc.samples`: Object of class `"numeric"`. Vector representing the probabilities of the spiked-in gene-sets being altered in any given sample, as percentages; for example `perc.samples = c(75, 90)` means that these probabilities are `0.75` and `0.90`.

`spiked.set.sizes`: Object of class `"numeric"`. Vector representing the sizes, in genes, of the spiked-in gene-sets; for example, if `perc.samples = c(75, 90)` and `spiked.set.sizes = c(50, 100)`, there would be 4 spiked-in sets, one with 50 genes and probability of being altered of 0.75 in each sample, one with 50 genes and probability of being altered of 0.90 in each sample, one with 100 genes and probability of being altered of 0.75 in each sample, and one with 100 genes and probability of being altered of 0.90 in each sample.

`GeneSizes`: Object of class `"list"`. The entries of the list are objects similar to `GeneSizes08` and correspond to the simulation iterations.

`GeneSets`: Object of class `"list"`. The entries of the list correspond to gene-sets and give the genes annotated to them.

`Coverage`: Object of class `"list"`. The entries of the list are objects similar to `CoverageBrain` and correspond to the simulation iterations.

`EventsBySample`: Object of class `"list"`. The entries of the list are objects similar to `EventsBySampleBrain` and correspond to the simulation iterations.

`Mutations`: Object of class `"list"`. The entries of the list are objects similar to `MutationsBrain` and correspond to the simulation iterations.

`Scores`: Object of class `"list"`. The entries of this list are the output of `cma.scores` and correspond to the simulation iterations.

`results`: Object of class `"list"`. The entries of this list are the output of `do.gene.set.analysis` and correspond to the simulation iterations.

Methods

show `signature(object = "SetMethodsSims")`

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

See Also

`CoverageBrain`, `EventsBySampleBrain`, `GeneSizes08`, `MutationsBrain`, `sim.data.p.values`, `do.gene.set.analysis`, `combine.sims`, `extract.sims.method`

`sim.data.p.values` *Simulates data and performs gene-set analysis methods on the simulated datasets.*

Description

This function simulates data under the passenger or permutation null, either under the null or including spiked-in gene-sets. It then calculates the p-values and q-values for all the selected gene-set analysis methods.

Usage

```
sim.data.p.values(EventsBySample,
  Mutations,
  GeneSizes,
  Coverage,
  GeneSets,
  passenger.rates = t(data.frame(0.55*rep(1.0e-6,25))),
  ID2name,
  BH = TRUE,
  nr.iter,
  pass.null = FALSE,
  perc.samples = NULL,
  spiked.set.sizes = NULL,
  gene.method = TRUE,
  perm.null.method = TRUE,
  perm.null.het.method = TRUE,
  pass.null.method = TRUE,
  pass.null.het.method = TRUE,
  show.iter,
  KnownMountains = c("EGFR","SMAD4","KRAS",
  "TP53","CDKN2A","MYC","MYCN","PTEN","RB1"),
  exclude.mountains=TRUE)
```

Arguments

`EventsBySample`
: Data frame giving the specific mutations for each gene and each tumor sample. See `EventsBySampleBrain` for an example.

`Mutations`
: Data frame with mutation information broken down by gene, phase and mutation type. See `MutationsBrain` for an example.

`GeneSizes`
: Data frame of gene sizes. See `GeneSizes08` object for an example.

`Coverage`
: Data frame with coverage information, by gene, phase, and type. See `WoodCoverageBrain` for an example.

`GeneSets`
: An object which annotates genes to gene-sets; it can either be a list with each component representing a set, or an object of the class AnnDbBimap.

`passenger.rates`
: Data frame with 1 row and 25 columns, of passenger mutation rates per nucleotide, by type, or "context". Columns denote types and must be in the same order as the first 25 columns in the `MutationsBrain` objects.

`ID2name`
: Vector mapping the gene identifiers used in the GeneSets object to the gene names used in the other objects; if they are the same, this parameter is not needed. See `ID2name` for an example.

`BH`
: If set to `TRUE`, uses the Benjamini-Hochberg method to get q-values; if set to `FALSE`, uses the Storey method from the `qvalue` package.

`nr.iter`
: The number of iterations to be simulated.

`pass.null`
: If set to true `TRUE`, implements the passenger null hypothesis, using the rates from `passenger.rates`; otherwise, implements the permutation null, permuting mutational events.

`perc.samples` Vector representing the probabilities of the spiked-in gene-sets being altered in any given sample, as percentages; for example `perc.samples = c(75, 90)` means that these probabilities are `0.75` and `0.90`.

`spiked.set.sizes` Vector representing the sizes, in genes, of the spiked-in gene-sets; for example, if `perc.samples = c(75, 90)` and `spiked.set.sizes = c(50, 100)`, there would be 4 spiked-in sets, one with 50 genes and probability of being altered of 0.75 in each sample, one with 50 genes and probability of being altered of 0.90 in each sample, one with 100 genes and probability of being altered of 0.75 in each sample, and one with 100 genes and probability of being altered of 0.90 in each sample.

`gene.method` If set to `TRUE`, implements gene-oriented method.

`perm.null.method` If set to `TRUE`, implements patient-oriented method with permutation null and no heterogeneity.

`perm.null.het.method` If set to `TRUE`, implements patient-oriented method with permutation null and heterogeneity.

`pass.null.method` If set to `TRUE`, implements patient-oriented method with passenger null and no heterogeneity.

`pass.null.het.method` If set to `TRUE`, implements patient-oriented method with passenger null and heterogeneity.

`show.iter` If set to `TRUE`, shows what simulation is currently running.

`KnownMountains` Vector of genes to be excluded from the permutation null simulations if `exclude.mountains = TRUE`.

`exclude.mountains` If set to `TRUE`, excludes the genes in `KnownMountains`.

Value

An object of the class `SetMethodsSims`. See `SetMethodsSims` for more details.

Author(s)

Simina M. Boca, Giovanni Parmigiani.

References

Boca S.M., Kinzler K., Velculescu V.E., Vogelstein B., Parmigiani G. Patient-oriented gene-set analysis for cancer mutation data. *Submitted*, 2010.

Parmigiani G, Lin J, Boca S, Sjoeblom T, Kinzler WK, Velculescu VE, Vogelstein B. Statistical methods for the analysis of cancer genome sequencing data. http://www.bepress.com/jhubiostat/paper126/

Benjamini Y and Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57:289-300, 995.

Storey JD and Tibshirani R. Statistical significance for genome-wide experimens. *Proceedings of the National Academy of Sciences.* doi: 10.1073/pnas.1530509100

Parsons DW, Jones S, Zhang X, Lin JCH, Leary RJ, Angenendt P, Mankoo P, Carter H, Siu I, et al. An Integrated Genomic Analysis of Human Glioblastoma Multiforme. *Science.* DOI: 10.1126/science.1164382

Wood LD, Parsons DW, Jones S, Lin J, Sjoeblom, Leary RJ, Shen D, Boca SM, Barber T, Ptak J, et al. The Genomic Landscapes of Human Breast and Colorectal Cancer. *Science.* DOI: 10.1126/science.1145720

See Also

`SetMethodsSims-class`, `CoverageBrain`, `EventsBySampleBrain`, `GeneSizes08`, `MutationsBrain`, `ID2name` `extract.sims.method`, `combine.sims`

Examples

```
##Note that this takes a few minutes to run:
library(KEGG.db)
data(Parsons)
data(ID2name)

set.seed(831984)

resultsSim <-
    sim.data.p.values(EventsBySample = EventsBySampleBrain,
                      Mutations = MutationsBrain,
                      GeneSizes = GeneSizes08,
                      Coverage = CoverageBrain,
      GeneSets = KEGGPATHID2EXTID[c("hsa05213",
                      "hsa05223", "hsa00250")],
      ID2name = ID2name,
                      nr.iter = 2,
                      pass.null = TRUE,
                      perc.samples = c(75, 95),
                      spiked.set.sizes = c(50),
                      show.iter = TRUE,
                      gene.method = FALSE,
                      perm.null.method = TRUE,
                      perm.null.het.method = FALSE,
                      pass.null.method = TRUE,
                      pass.null.het.method = FALSE)

resultsSim
```

Index