httrpathway

February 6, 2020

Type Package

Title Pathway Scoring and Concentration Response for HTTr data

Version 1.0.0

Author Thomas Sheffield

Maintainer Richard Judson <judson.richard@epa.gov>

Description This package generates pathway scores with associated concentration response modeling; it also contains some important plotting functions. ``pathwayScore'' uses chemical/concentration by gene matrices of log2(fold change) values and pathway definitions to generate chemical/concentration by pathway matrices of pathway score values. Three pathway methods are included: ``fc'' (fold change in pathway - fold change outside pathway), ``mygsea'' (modified ssGSEA), and ``gsva''. ``pathwayConcResp_pval'' generates concentration response fits, related statistics, and plots for the pathway scores, given pathway scores run on null data (which itself can be generated by ``randomdata''). ``runAllPathwayCR'' wraps the main functions. ``pathwayAccumNullPlot'' generates BMD accumulation plots. ``referenceAC50'' checks the accuracy of a given pathway given ER reference data. ``runAllRepChemPidCR'', ``runAllRepChemCR'', ``repChemPidPlot'' and ``repChemPathwayPlot'' generate results for the replication study.

Imports stats,
stringr,
grDevices,
graphics,
utils,
methods,
data.table,
future.apply,
future,
GSVA,
moments,
numDeriv,
openxlsx,
parallel,
RColorBrewer,
reshape2,
data.table,

openxlsx,
e1071

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`allGeneBMD` *Calculate the all-gene cocnentraiton-response and BMDL*

Description

Calculate the all-gene cocnentraiton-response and BMDL

Usage

```
allGeneBMD(
  to.file = F,
  basedir = "../input/fcdata/",
  dataset = "DMEM_6hr_pilot_normal_00",
  metric = "z",
  l2fc.limit = 1.2
)
```

Arguments

to.file	If TRUE, write the plots to a pdf file
basedir	The base directory to find the raw fold change and chemcial data
dataset	The name of the datset to be used

Value

nothing is returned

---

auc *Area Under the Curve*

Description

Compute AUC for an ROC curve.

Usage

```
auc(tpr, fpr)
```

Arguments

tpr	Vector of true positive rates.
fpr	Vector of false positive rates.

Details

Uses trapezoid rule numerical integration to approximate AUC. Will be more accurate with more fine-grained inputs.

Value

AUC

Examples

```
auc(c(0,.5,1), c(0,.5,1))
auc(c(0,1,1), c(0,.5,1))
```

bioplanet_builder *BioPlanet Builder*

Description

Converts BioPlanet data into usable pathway data.

Usage

```
bioplanet_builder(
  pathfile = "../input/processed_pathway_data/bioplanet_pathway.csv",
  catfile = "../input/processed_pathway_data/bioplanet_pathway_category.csv",
  pwayout = "../input/processed_pathway_data/bioplanet_PATHWAYS.RData",
  pdataout = "../input/processed_pathway_data/PATHWAY_LIST_bioplanet.RData"
)
```

Arguments

pathfile	File name of bioplanet_pathway.csv.
catfile	File name of bioplanet_pathway_category.csv.
pwayout	File name of bioplanet_PATHWAYS.RData
pdataout	File name of

Details

This function shows how BioPlanet data was converted to usable pathway files. As BioPlanet is updated, this function will have to be updated. It requires two downloaded .csv files with location specified by pathfile and catfile. It saves usable pathway files with location specified by pwayout and pdataout to disk.

Value

No output.

bmde.potency *Export the most potent BMDExpress signatures*

Description

Export the most potent BMDExpress signatures

Usage

```
bmde.potency(to.file = F)
```

buildFCMAT1 *Build the FCMAT1 data set*

Description

Build the FCMAT1 data set

Usage

```
buildFCMAT1(
  dataset = "DMEM_6hr_pilot_normal_00",
 dir = "../input/httr_mcf7_pilot/meanncnt0_5-plateteffect_0-shrinkage_normal/DMEM_6/",
  filetype = "tsv"
)
```

Arguments

dataset	The name to give to the data set
dir	The directory from which to read all of the raw filesatalog file
filetype	Either tsv or RData

Value

A file with the FCMAT1 data is written to "../input/fcdata/FCMAT0_",dataset,".RData"

buildFCMAT2 *Transpose and filter the fold change matrix FCMAT1 in long format into a gene x sample format.*

Description

Transpose and filter the fold change matrix FCMAT1 in long format into a gene x sample format.

Usage

```
buildFCMAT2(
  dataset = "DMEM_6hr_pilot_normal_pe_0",
  dir = "../input/fcdata/",
  method = "gene",
  do.read = T,
  chemical.file = "../input/chemicals/HTTr.Sample.Matrix.2017.04.24.xlsx"
)
```

Arguments

method	=gene or probe_id
chems	The CHEMS data frame with chemical information

Value

Global variables are created for the FC matrix (FCMAT2), the SE matrix (SEMAT2) and the chemical dictionary (CHEM_DICT) which translates form the sample key (sample_id_conc_time) to the individual components

---

CMAP_RefchemDB *Extract annotations for CMAP chemicals*

Description

Extract annotations for CMAP chemicals

Usage

```
CMAP_RefchemDB()
```

Value

No output.

---

compare.fc.gsva.outlers *Compare the outliers from the FC and GSVA calculations*

Description

Compare the outliers from the FC and GSVA calculations

Usage

```
compare.fc.gsva.outlers(
  to.file = F,
  method2 = "gsva",
  dataset = "DMEM_6hr_pilot_none_pe_1",
  pathset = "PathwaySet_20191107"
)
```

Arguments

to.file	If TRUE, write plots to a file

driver *Code to run all calculations*

Description

Code to run all calculations

Usage

```
driver(
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  sigcatalog = "signatureDB_master_catalog 2020-01-31",
  sigset = "pilot_tiny",
  nrandom.chems = 1000,
  mc.cores = 1,
  method = "mygsea",
  do.build.fcmat1.all = F,
  do.build.fcmat2.all = F,
  do.build.random = F,
  do.run.random = F,
  do.run.all = F,
  do.signature.summary.plot = F,
  do.signature.pod = F,
  do.signature.pod.laneplot = F,
  do.all = F
)
```

Arguments

method	signature scoring method in c("fc", "gsva", "mygsea")

driver.driver *Code to run all calculations*

Description

Code to run all calculations

Usage

```
driver.driver(
  pathset = "PathwaySet_20200108",
  method = "mygsea",
  nrandom.chems = 1000,
  mc.cores = 30,
  step1 = F,
```

```
  step2 = F,
  step3 = F,
  do.log = F
)
```

Arguments

method	Pathway scoring method in c("fc", "gsva", "mygsea")

`FC.vs.GSVA.plot` *Plot the difference between the FC and GSVA analysis at hte signature level*

Description

Plot the difference between the FC and GSVA analysis at hte signature level

Usage

```
FC.vs.GSVA.plot(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_0",
  pathset = "PathwaySet_20191107",
  cutoff = 0.5
)
```

Arguments

to.file	If TRUE, write plots to a file

`FCMATrepchems` *FCMAT for Replicate Chemicals*

Description

Generates fold change matrix for the pilot/phase 1 replicate experiment.

Usage

```
FCMATrepchems(study = "ph1", floor = 10, bygene = T)
```

Arguments

study	Which replicate? "ph1" or "pilot"
floor	Which flooring for input? 5 or 10
bygene	If bygene is TRUE, output will be chemical/concentration by gene, to be used for signature analysis; otherwise, it will output chemical/concentration by probe id for probe concentration response modeling.

Details

Converts deseq2 output to usable FCMAT2 matrices. Also builds CHEM_DICT files as a subset of pre-existing CHEM_DICT files. Generated files are saved directly to disk. Not intended to be run again, but rather to document how it was done originally.

Value

No output.

`geneConcResp` *Gene Concentration Response*

Description

Wrapper that performs concentration response modeling for gene or probe l2fc's

Usage

```
geneConcResp(
  dataset = "ph1_100normal_pid",
  mc.cores = 1,
  to.file = F,
  pval = 0.05,
  nametag = NULL,
  conthits = F,
  aicc = F,
 fitmodels = c("cnst", "hill", "gnls", "poly1", "poly2", "pow", "exp2", "exp3", "exp4",
    "exp5")
)
```

Arguments

`dataset`	String that identifies data set.
`mc.cores`	Number of parallel cores to use.
`to.file`	If TRUE, results are written to an RData file, otherwise they are returned.
`pval`	P-value cutoff between 0 and 1.
`nametag`	Optional identifier attached to the output name that usually is used to signify that an unusual option was used.
`conthits`	If conthits = T, continuous hitcalls are calculated; otherwise discrete hitcalls are used.
`aicc`	If aicc = T, corrected AIC is used insstead of first order (regular) AIC.
`fitmodels`	Vector of models names to be used. Default is all of them.

Details

If conthits = T and nametag is NULL, nametag will be set to "conthits". Loads an FCMAT2 and CHEM_DICT corresponding to given dataset. FCMAT should be chem/conc by gene or chem/conc by probe. Uses two lowest concentration of each column to estimate noise cutoff (as opposed to signature CR). Also, doesn't currently contain a plotting option.

Value

If to.file = F, data frame containing results; otherwise, nothing.

---

getpvalcutoff *Get P-Value Cutoff*

Description

Retrieves signature cutoffs for a given null dataset.

Usage

```
getpvalcutoff(pathset, nullset, method, pvals = NULL, numsds = NULL)
```

Arguments

pathset	Name of signature set used to score null data.
nullset	Name of null data set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
pvals	Vector of p-values to get cutoff for.
numsds	Vector of number of standard deviations to get cutoff for. For instance, numsds = 1 will return cutoffs at 1 standard deviation.

Details

Calculates median of all scores for a given signature as well as a cutoff based on the specified null dataset. P-values represent the percentage of scores that are greater in distance from the median than the cutoff. Numsd gives a cutoff that is the given number of standard deviations from the median. Each row of the output corresponds to one signature and one pvalue or numsd. If both pvals and numsds are specified, the output contains a column for each, and the unused identifier(pvalue or numsd) in each row will contain NA.

Value

Dataframe with 4 or 5 columns: signature, cutoff, bmed (median of all samples for that signature), pvalue (pvalue corresponding to each cutoff), numsd (number of sds corresponding to each cutoff).

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

largeFCgenes *Find genes that are often seen with large fold changes*

Description

Find genes that are often seen with large fold changes

Usage

```
largeFCgenes(
  to.file = F,
  basedir = "../input/fcdata/",
  dataset = "DMEM_6hr_pilot_normal_00",
  l2fc.limit = 1.2
)
```

Arguments

to.file	If TRUE, write the plots to a pdf file
basedir	The base directory to find the raw fold change and chemcial data
dataset	The name of the datset to be used

Value

nothing is returned

MYGSEA *My Gene Set Enrichment Analysis*

Description

Performs tweaked version of single sample GSEA.

Usage

```
MYGSEA(
  X,
  geneSets,
  min.sz = 1,
  max.sz = Inf,
  alpha = 0.25,
  verbose = T,
  useranks = T
)
```

Arguments

X	Transposed FCMAT2; i.e a gene by sample matrix of l2fc's including rownames and colnames. Equivalent to expr in gsva.
geneSets	Named list of signature definitions. Each element is a vector of gene names. Each element name is a signature name.E quivalent to gset.idx.list in gsva.
min.sz	Minimum signature size (deprecated).
max.sz	Maximum signature size (deprecated)
alpha	Power of R to use. Higher alpha will upweight more extreme ranks relative to middle ranks.
verbose	verbose = T prints gene set length message.
useranks	useranks = T uses ranks as in ssGSEA, while useranks = F uses the bare fold changes instead.

Details

Based on the GSVA ssGSEA code. Main changes are: NAs are now handled correctly and rank is now centered on zero instead of beginning at one. Since signature sizes are undercounted here due to missing values, they are assessed more accurately in signatureScoreCoreMYGSEA and limits are enforced after scoring.

Value

Outputs signature by sample matrix of signature scores.

Examples

```
geneSets = list(signature1 = c("ABC", "DEF"), signature2 = c("ABC", "GHI"))
X = matrix(c(1:3,3:1), nrow = 3)
colnames(X) = c("Sample1", "Sample2")
rownames(X) = c("ABC", "DEF", "GHI")
MYGSEA(X,geneSets)
```

---

pidbar *PID Bar Plot*

Description

Specially formatted bar plot.

Usage

```
pidbar(x, ...)
```

Arguments

x	Named matrix or vector to pass to barplot.
...	Other options to pass to barplot.

Details

This function is a helper for repChemPidPlot. It fiddles with the margins and renames the labels so that they fit on the plot.

Value

No output.

---

plotouter *Plot Outer*

Description

Calls signatureConcResp plotting function.

Usage

```
plotouter(proper_name, SIGNATURE_CR, foldname, CYTOTOX)
```

Arguments

proper_name	Chemical name to be used in file name.
SIGNATURE_CR	Dataframe output of signatureConcResp_pval.
foldname	Folder name for output file.
CYTOTOX	The cytotoxicity data for all chemicals

Details

Calls signatureConcResp plotting function for one chemical and every signature. Saves a single pdf to disk for the given chemical containing every signature CR plot.

Value

No output.

---

podLaneplot *Build lane plots by chemical list and signature class, across the datasets*

Description

Build lane plots by chemical list and signature class, across the datasets

Usage

```
podLaneplot(to.file = F, dataset, sigset, method, plot.signature_min = F)
```

Arguments

to.file	If TRUE, write plots to a file

---

printCurrentFunction *Print the name of the current function*

Description

Print the name of the current function

Usage

```
printCurrentFunction(comment.string = NA)
```

Arguments

comment.string	An optinal string to be printed

pwaybar *Pathway Bar Plot*

Description

Specially formatted bar plot.

Usage

```
pwaybar(x, ...)
```

Arguments

x	Named matrix or vector to pass to barplot.
...	Other options to pass to barplot.

Details

This function is a helper for repChemPathwayPlot. It fiddles with the margins and renames the labels so that they fit on the plot.

Value

No output.

R2 *R Squared*

Description

Calculate coefficient of determination.

Usage

```
R2(y, pred)
```

Arguments

y	Vector of actual values.
pred	Vector of corresponding predicted values.

Details

Note that order matters: R2(x,y) != R2(y,x) in general.

Value

Coefficient of determination.

Examples

```
R2(c(1:10), c(1:10*.8))
R2(c(1:10*.8), c(1:10))
```

---

randomdata *Randomized Null Data*

Description

Generate randomized null data based on actual data.

Usage

```
randomdata(
  basedir = "../input/fcdata/",
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  nchem = 10,
  seed = 12345
)
```

Arguments

basedir	Directory that holds FCMAT2 and CHEM_DICT files.
dataset	Name of actual dataset to base null data on.
nchem	Number of null chemicals. Number of null samples is approximately eight times this value.
seed	Random seed.

Details

New FCMAT2 and CHEM_DICT files corresponding to the null dataset are written to disk in the basedir folder. The nullset name is paste0(dataset, "_", nchem). Randomization is performed by sampling the quantile function for each gene in the actual data. The nullset will have roughly the same distribution of values for each gene in the actual data,

Value

No output.

`referenceAC50` *Reference AC50 Plot*

Description

Scatter plot and accuracy statistics of signatures vs. reference values.

Usage

```
referenceAC50(
  method = "fc",
  dataset = "user_wneg",
  pathset = "bhrr",
  nullset = "user_wneg_RAND125",
  newpvals = c(0.2, 0.1, 0.05, 0.01, 0.005, 0.001),
  oldpval = 0.2,
  nametag = NULL,
  conthits = F,
  pathclass = "DUT",
  aucclass = "erac50"
)
```

Arguments

`method`	Pathway scoring method name.
`dataset`	Data set name.
`pathset`	Pathway set name.
`nullset`	Null data set name.
`newpvals`	Vector of p-values to make plots for.
`oldpval`	P-value used when running signatureConcResp.
`nametag`	Additional file identifier added during signatureConcResp.
`conthits`	Set conthits = T when using continuous hits.
`pathclass`	Some pre-defined sets of signatures to plot and run statistics on. "ER" is a group of ER signatures, "AR" is a group of AR signatures, and "DUT" is just the DUTERTRE_ESTRADIOL_RESPONSE_6HR_UP signature.
`aucclass`	Which type of reference value to compare against. "erac50" uses the pseudo.AC50.median, "bmd" uses the pseudo.ACB.median, "AR" uses the maximum AR AUC, and "ER" uses the maximum ER AUC. AR, ER, and bmd might no longer function correctly.

Details

Saves a plot to disk. Plot is a scatter plot of actual values (based on ER model) vs. predicted values (using some given signatures). For discrete hitcalls, only true positive are plotted and colors indicate model used. Continuous hitcalls plots all positives with colors indicating the hitcall. Other statistics assume that all chemicals that are not positives (defined by AUC >= .1) are negatives, so care must be taken not to include chemicals with borderline activity in the dataset. RMSE is only shown for true positives. Continuous hitcalls weights all statistics by the hitcall. oldpval should be >= newpvals when using discrete hitcalls.

Value

No output.

---

repChemPathwayPlot *Replicate Chemical Pathway Plot*

Description

Generates plots and statistics for replicate chemicals' signatures.

Usage

```
repChemPathwayPlot(
  oldpval = 0.05,
  nametag = "conthits",
  method = "fc",
  pathset = "bhrr",
  mc.cores = 3
)
```

Arguments

oldpval	P-value used to generate PATHWAY_CR's.
nametag	Optional descriptor in filename.
method	Name of signature scoring method used.
pathset	Name of signature set.
mc.cores	Number of cores to use.

Details

This function is desigend to work with runAllRepChemCR, so the dataset names are hard-coded. This function may take some time to run. Concentration response should have been run using continuous hitcalls.

Value

No output.

repChemPidPlot *Replicate Chemical PID Plot*

Description

Generates plots and statistics for replicate chemicals' probe IDs.

Usage

```
repChemPidPlot(oldpval = 0.05, nametag = "conthits", mc.cores = 3)
```

Arguments

oldpval	P-value used to generate GENE_CR's.
nametag	Optional descriptor in filename.
mc.cores	Number of cores to use.

Details

This function is desigend to work with runAllRepChemPidCR, so the dataset names are hard-coded. This function may take some time to run. Concentration response should have been run using continuous hitcalls.

Value

No output.

RMSE *Root-mean-square-error*

Description

Computes root-mean-square-error between two vectors.

Usage

```
RMSE(x, y)
```

Arguments

x	First vector.
y	Second vector.

Value

RMSE

Examples

```
RMSE(1:3, c(1,3,5))
```

`runAllRepChemCR` *Run All Replicate Chemical Concentration Response*

Description

Runs signature scoring and concentration response for replicate chemicals.

Usage

```
runAllRepChemCR(
  basedir = "../input/fcdata/",
  pathset = "bhrr",
  method = "fc",
  minpathsize = 10,
  do.plot = F,
  pval = 0.05,
  mc.cores = c(39, 39),
  conthits = T,
  nchem = 125
)
```

Arguments

`basedir`	Folder that the FCMAT2's are stored in.
`pathset`	Name of signature set.
`method`	Name of signature scoring method.
`minpathsize`	Minimum signature size.
`do.plot`	do.plot = T generates plots for every chemical/signature/replicate combination. Adds a significant amount to the runtime.
`pval`	P-value to use for noise estimation.
`mc.cores`	Vector with two values: number of cores to use for signature scoring and number of cores to use for CR. CR can usually handle the maximum number, but gsva scoring might require a smaller number to avoid memory overflow.
`conthits`	conthits = T uses continuous hitcalls. Continuous hitcalls are a prerequisitie for using repChemPathwayPlot().
`nchem`	Number of null chemicals to use. The number of null samples is approximately eight times this value, so nchem = 125 generates ~1000 null samples.

Details

This function has hard-coded dataset names for the replicates. For each replicate, it computes signature scores, generates a null dataset, runs signature scores for the null dataset, and then runs concentration-response on the actual data. Pathway scores and CR are written to disk.

Value

No output.

`runAllRepChemPidCR` *Run All Replicate Chemical PID Concentration Response*

Description

Runs probe ID concentration response for replicate chemicals.

Usage

```
runAllRepChemPidCR(pval = 0.05, mc.cores = 39, conthits = T)
```

Arguments

`pval`	P-value to use for noise estimation. Noise is estimated using two lowest concentrations.
`mc.cores`	Number of cores to use for CR.
`conthits`	conthits = T uses continuous hitcalls. Continuous hitcalls are a prerequisitie for using repChemPidPlot().

Details

This function has hard-coded dataset names for the replicates. For each replicate, it runs concentration-response directly on the probe ID's. The result is written to disk.

Value

No output.

`runAllSignatureCR` *Run All Pathway Concentration Response (P-Value)*

Description

Driver for signature scoring and concentration response (CR).

Usage

```
runAllSignatureCR(
  basedir = "../input/fcdata/",
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  sigset,
  sigcatalog,
  method = "fc",
  minsigsize = 10,
  conthits = T,
  nullset,
  do.plot = T,
  pval = 0.05,
  mc.cores = c(1, 1),
 fitmodels = c("cnst", "hill", "poly1", "poly2", "pow", "exp2", "exp3", "exp4", "exp5")
)
```

Arguments

basedir	Folder that stores FCMAT2 and CHEM_DICT files.
dataset	Name of data set.
sigset	Name of signature set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
minsigsize	Minimum signature size.
conthits	conthits = T uses continous hitcall; conthits = F uses discrete hitcalls.
nullset	Name of null dataset. Set nullset = NULL to skip CR.
do.plot	do.plot = T generates a CR plot for every sample/signature combination.
pval	P-value to use for noise estimation.
mc.cores	Vector with two values: number of cores to use for signature scoring and number of cores to use for CR. CR can usually handle the maximum number, but gsva scoring might require a smaller number to avoid memory overflow.
fitmodels	Vector of model names to run conc/resp with. "cnst" should always be chosen.

Details

CR requires signature scores to have already been computed for a nullset. randomdata() can generate a nullset, and this function can compute signature scores for it by setting dataset = nullset and nullset = NULL. Pathway scores are written to disk in output/signature_score_summary/. CR results are written to disk in output/signature/conc_resp_summary/.

Value

No output.

remove gnls from default set

`runInsert` *Insert a record into a database. if auto.increment=TRUE, return the auto incremented primary key of the record. otherwise, return -1*

Description

Insert a record into a database. if auto.increment=TRUE, return the auto incremented primary key of the record. otherwise, return -1

Usage

```
runInsert(query, db, do.halt = F, verbose = F, auto.increment.id = F)
```

Arguments

`query`	a properly formatted SQL query as a string
`db`	the name of the database
`do.halt`	if TRUE, halt on errors or warnings
`verbose`	if TRUE, print diagnostic information
`auto.increment`	if TRUE, add the auto increment primary key even if not part of the query

Value

Returns the database table auto incremented primary key ID

`runInsertTable` *Inserts multiple rows into a database table*

Description

Inserts multiple rows into a database table

Usage

```
runInsertTable(mat, table, db, do.halt = T, verbose = F, get.id = T)
```

Arguments

`mat`	data frame containing the data, with the column names corresponding
`table`	name of the database table to which data will be inserted
`db`	the name of the database
`do.halt`	if TRUE, halt on errors or warnings
`verbose`	if TRUE, print diagnostic information

`runQuery` *Runs a database query and returns a result set*

Description

Runs a database query and returns a result set

Usage

```
runQuery(query, db, do.halt = T, verbose = F)
```

Arguments

`query`	a properly formatted SQL query as a string
`db`	the name of the database
`do.halt`	if TRUE, halt on errors or warnings
`verbose`	if TRUE, print diagnostic information

`setDBConn` *set SQL connection to the database*

Description

set SQL connection to the database

Usage

```
setDBConn(server = "mysql-res1.epa.gov", user = "rjudson", password = NA)
```

Arguments

`server`	SQL server on which relevant database lives
`user`	SQL username to access database
`password`	SQL password corresponding to username

`signatureAccumNullPlot` *BMD Accumulation Plot With Nulls*

Description

Creates signature BMD accumulation plot vs. null and computes accumulation BMD.

Usage

```
signatureAccumNullPlot(
  pathset = "PathwaySet_20191107",
  dataset = "DMEM_6hr_pilot_normal_pe_0",
  method = "fc",
  nullset = "DMEM_6hr_pilot_normal_pe_0_RAND1000",
  newpval = 0.05,
  oldpval = 0.05,
  to.file = F,
  usecont = T,
  nametag = "conthits",
  mc.cores = 1,
  hit.threshold = 0.5,
  verbose = F
)
```

Arguments

`pathset`	Name of pathset.
`dataset`	Name of dataset.
`method`	Name of signature scoring method.
`nullset`	Name of null dataset.
`newpval`	P-value for cutoff to be used in plot.
`oldpval`	P-value that nullset and dataset were originally run with.
`to.file`	If to.file = T, plots to file.
`usecont`	Set usecont = T for continuous hitcalls, usecont = F for discrete. Should probably match the original hitcall type use in CR modeling.
`nametag`	Set additional name descriptor that was attached to CR modeling, if applicable.
`mc.cores`	Number of cores to use when altering continuous hitcalls; has no effect if usecont = F or newpval = oldpval.

Details

Nullset and dataset should already have been run through signatureConcResp using the given pathset, method, and oldpval. Only generates proof of concept plots for accumulation BMDs. There is not currently a method to extract the accumulation BMDs directly.

Value

No output.

`signatureBmdeDeseq` *Compare the signature PODs between BMDExpress and DESeq2 / TCPL*

Description

Compare the signature PODs between BMDExpress and DESeq2 / TCPL

Usage

```
signatureBmdeDeseq(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_0",
  pathset = "PathwaySet_20191107",
  method = "fc"
)
```

Arguments

`to.file`	If TRUE, write plots to a file

`signatureBuildBioplanet` *Build the standard input file for the Bioplanet signatures*

Description

Build the standard input file for the Bioplanet signatures

Usage

```
signatureBuildBioplanet()
```

Value

No output.

signatureBuildCMAP *Build the standard input file for the CMAP signatures*

Description

Build the standard input file for the CMAP signatures

Usage

```
signatureBuildCMAP()
```

Value

No output.

signatureBuildDisGeNET *Build the standard input file for the Ryan signatures*

Description

Build the standard input file for the Ryan signatures

Usage

```
signatureBuildDisGeNET()
```

Value

No output.

signatureBuilder *Create the files needed for the signature calculations before adding random genes*

Description

Create the files needed for the signature calculations before adding random genes

Usage

```
signatureBuilder(min.ngene = 10, max.ngene = 1e+05)
```

Arguments

min.ngene	Signatures will only be saved if the number of genes is >= this value
max.ngene	Signatures will only be saved if the number of genes is <= this value

Value

No output.

---

signatureBuilderRandom *Add the random gene sets to the signature files*

Description

Add the random gene sets to the signature files

Usage

```
signatureBuilderRandom(nrandom = 1000, mc.cores = 1)
```

Arguments

nrandom	Number of random gene sets
mc.cores	The number of cores to use in parallel

Value

No output.

---

signatureBuilder_bhrr *bhrr Pathway Builder*

Description

Builds bhrr pathset based on msigdb, bioplanet, and ryan pathsets.

Usage

```
signatureBuilder_bhrr()
```

Details

Shows how the bhrr pathset was built from pre-existing pathsets.

Value

No output.

signatureBuildMsigDB *Build the standard input file for the MSigDB signatures*

Description

Build the standard input file for the MSigDB signatures

Usage

```
signatureBuildMsigDB()
```

Value

No output.

signatureBuildRyan *Build the standard input file for the Ryan signatures*

Description

Build the standard input file for the Ryan signatures

Usage

```
signatureBuildRyan()
```

Value

No output.

signatureCatalogLoader *Merge the up and down halves of the pathway data*

Description

Merge the up and down halves of the pathway data

Usage

```
signatureCatalogLoader(
  sigset = "pilot_small",
  sigcatalog = "signatureDB_master_catalog 2020-01-31"
)
```

Arguments

sigset	Name of the signature set.
sigcatlog	Nmae of the catalog file

Value

the trimmed signature table

---

signatureChemicalDiffAcrossDatasets

Build lane plots by chemical list and signature class, across the datasets

---

Description

Build lane plots by chemical list and signature class, across the datasets

Usage

```
signatureChemicalDiffAcrossDatasets(
  to.file = F,
  chemical.target = "ER",
  signature.super_class = "estrogen",
  pathset = "PathwaySet_20191107",
  method = "fc"
)
```

Arguments

to.file	If TRUE, write plots to a file
method	Pathway scoring method in c("fc", "gsva", "mygsea")

---

signatureChemicalLanePlotAcrossDatasets

Build lane plots by chemical list and signature class, across the datasets

---

Description

Build lane plots by chemical list and signature class, across the datasets

Usage

```
signatureChemicalLanePlotAcrossDatasets(
  to.file = F,
  chemical.target = "ER",
  signature.super_class = "estrogen",
  pathset = "PathwaySet_20191107",
  method = "fc"
)
```

Arguments

to.file	If TRUE, write plots to a file
method	Pathway scoring method in c("fc", "gsva", "mygsea")

---

signatureClassHM *Build lane plots by chemical list and signature class, across the datasets*

Description

Build lane plots by chemical list and signature class, across the datasets

Usage

```
signatureClassHM(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  pathset = "PathwaySet_20191107",
  method = "gsva",
  threshold = 0.5
)
```

Arguments

to.file	If TRUE, write plots to a file

signatureClassSummaryPlot *Build summary plots by signature class*

Description

Build summary plots by signature class

Usage

```
signatureClassSummaryPlot(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  sigset = "PathwaySet_20191107",
  method = "mygsea",
  hitcall.threshold = 0.5
)
```

Arguments

to.file	If TRUE, write plots to a file

signatureConcResp *Pathway Concentration Response (P-value)*

Description

Performs signature concentration response using p-value based cutoffs.

Usage

```
signatureConcResp(
  sigset,
  sigcatalog,
  dataset,
  method = "mygsea",
  nullset,
  mc.cores = 1,
  to.file = T,
  do.plot = F,
  pval = 0.05,
  nametag = NULL,
  conthits = T,
  aicc = F,
  minsigsize = 10,
```

```
  fitmodels = c("cnst", "hill", "gnls", "poly1", "poly2", "pow", "exp2", "exp3", "exp4",
    "exp5"),
  CYTOTOX
)
```

Arguments

sigset	Name of the signature set.
dataset	Name of the data set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
nullset	Name of the null data set.
mc.cores	Number of cores to parallelize with.
to.file	to.file = T saves the output to a file; otherwise it's returned.
do.plot	do.plot = T creates concentration-response plots for every sample/signature combination and saves to disk.
pval	Desired cutoff p-value.
nametag	Optional descriptor tag to attach to file outputs for experimental/non-default runs.
conthits	conthits = T uses continuous hitcalls, otherwise it's discrete.
aicc	aicc = T uses corrected AIC to choose winning method; otherwise regular AIC.
minsigsize	Minimum allowed signature size. Sample/signature combinations with less than this number of non-missing l2fc's will be discarded.
fitmodels	Vector of model names to use. Probably should include "cnst".

Details

Null dataset and dataset should have already been scored using signatureScore and the given sigset and method. This function prepares signatureScore output for CR processing, calls signatureConcRespCore_pval, formats the output, saves it to disk, then calls plotouter for CR plots, if desired. If conthits = T and nametag = NULL, the nametag "conthits" is automatically added to the output file.

Value

If to.file = T, nothing. If to.file = F, dataframe with signature CR output.

---

signatureConcRespPlot *Pathway Concentration Response Plot*

Description

Plots a concentration response curve for one sample/signature combination.

Usage

```
signatureConcRespPlot(row, CYTOTOX)
```

Arguments

row	Named list containing: • conc - conc string separated by |'s • resp - response string separated by |'s • method - scoring method determines plot bounds • proper_name - chemical name for plot title • cutoff - noise cutoff • bmr - baseline median response; level at which bmd is calculated • er - fitted error term for plotting error bars • a, tp, b, ga, p, la, q - other model parameters for fit curve • fit_method - curve fit method • bmd, bmdl, bmdu - bmd, bmd lower bound, and bmd upper bound • ac50, acc - curve value at 50 • top - curve top • time, signature, signature_class, signature_size - other identifiers Other elements are ignored.
CYTOTOX	The cytotoxicity data for all chemicals

Details

row is one row of PATHWAY_CR, the signatureConcResp output.

Value

No output.

---

signatureDistributionPlot

Pathway Distribution Plot

Description

Plots null and actual pdfs for given signature and cutoffs.

Usage

```
signatureDistributionPlot(
  sigset = "bhrr",
  dataset = "ph1_100normal_gene",
  method = "fc",
  nullset = "ph1_100normal_gene_RAND125",
  perc = 0.95,
  fdr = 0.25,
  comparetype = "Null",
```

```
  samplepaths = c("HALLMARK_ESTROGEN_RESPONSE_EARLY",
    "DUTERTRE_ESTRADIOL_RESPONSE_6HR_UP", "HALLMARK_CHOLESTEROL_HOMEOSTASIS",
    "Vitamin A and carotenoid metabolism", "Cytochrome P450 signature",
    "HALLMARK_ANDROGEN_RESPONSE"),
  to.file = T,
  seed = 12345
)
```

Arguments

sigset	Name of signature set.
dataset	Name of data set.
method	Name of signature scoring method.
nullset	Name of null data set.
perc	1-p-value for pvalue cutoff.
fdr	False discovery rate for FDR cutoff.
comparetype	Type of noise to use: "Null" for null data scores, "Low Conc" for lowest concentrations.
samplepaths	Vector of sample signature names to plot.
to.file	If to.file = T, write plot to disk.
seed	Randomization seed to use to choose additional sample signatures.

Details

This function requires that a signaturescoremat file has already been generated for the given sigset/dataset/method using signatureScore. There should also be signaturescoremat file for the nullset if comparetype = "Null". This function has also been used to get crossing-based cutoffs, but that feature has been deprecated.

Value

No output.

---

signatureGeneHM *Build lane plots by chemical list and signature class, across the datasets*

Description

Build lane plots by chemical list and signature class, across the datasets

Usage

```
signatureGeneHM(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  chemical.target = "ER",
  signature.super_class = "estrogen",
  pathset = "PathwaySet_20191107",
  method = "gsva",
  threshold = 0.5
)
```

Arguments

to.file	If TRUE, write plots to a file

---

signatureHeatmap *Create heatmaps of the htpp data frames*

Description

Create heatmaps of the htpp data frames

Usage

```
signatureHeatmap(
  to.file = F,
  pathset = "PathwaySet_20191025",
  dataset = "DMEM_6hr_pilot_normal_00",
  method = "fc",
  conthits = T,
  nametag = NULL,
  pval = 0.05
)
```

Arguments

to.file	If TRUE, write plots to a file
dataset	The set of data to be included

signaturePOD *Build lane plots by chemical list and signature class, across the datasets*

Description

Build lane plots by chemical list and signature class, across the datasets

Usage

```
signaturePOD(sigset, dataset, method, hit.threshold = 0.5)
```

Arguments

to.file	If TRUE, write plots to a file

signatureRank *Get the signature ranks for chemicals*

Description

Get the signature ranks for chemicals

Usage

```
signatureRank(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  pathset = "PathwaySet_20191107",
  method = "mygsea"
)
```

signatureRank.DESEQ2 *Get the signature ranks for chemicals*

Description

Get the signature ranks for chemicals

Usage

```
signatureRank.DESEQ2(
  to.file = F,
  dataset1 = "DMEM_6hr_pilot_none_pe_1",
  dataset2 = "DMEM_6hr_pilot_normal_pe_1",
  pathset = "PathwaySet_20191107",
  method = "gsva",
  cutoff = 0.5
)
```

---

signatureScore *Signature Score*

Description

Computes and saves signature scores.

Usage

```
signatureScore(
  FCMAT2,
  CHEM_DICT,
  sigset,
  sigcatalog,
  dataset,
  method,
  mc.cores = 1,
  minsigsize = 10
)
```

Arguments

FCMAT2	Sample by gene matrix of log2(fold change)'s. Rownames are sample keys and colnames are genes.
CHEM_DICT	Dataframe with one row per sample key and seven columns: sample_key, sample_id, conc, time, casrn, name, dtxsid.
sigset	Name of signature set.
dataset	Name of data set.
method	Signature scoring method in c("fc", "gsva", "mygsea")
mc.cores	Number of cores to use.
minsigsize	Minimum allowed signature size BEFORE accounting for missing values.

Details

signatureScore is a driver for various scoring methods. The three that are currently available are "gsva", "mygsea", "fc", and "mygsea_norank" (a version of mygsea that uses fold changes instead of ranks as weights). Deprecated methods include the Fisher method and gsvae (gsva with empirical cdfs). Beware running out of memory on large runs with gsva, Linux, and many cores. Signature size is counted according to number of genes in the signature that are also in the column names of FCMAT2. However, each method performs a more rigorous size count internally that accounts for missing values and adds this to the output. This minsigsize is enforced when running signatureConcResp_pval.

Value

No output.

`signatureScoreCoreFC` *Signature Score Core - FC*

Description

Computes fold change signature scores.

Usage

```
signatureScoreCoreFC(fcdata, sigset, dataset, chem_dict, signature_data)
```

Arguments

`fcdata`	Sample by gene matrix of log2(fold change)'s. Rownames are sample keys and colnames are genes.
`sigset`	Name of signature set.
`dataset`	Name of data set.
`chem_dict`	Dataframe with one row per sample key and seven columns: sample_key, sample_id, conc, time, casrn, name, dtxsid.
`signature_data`	Named ist of gene name vectors. Each element is one signature, defined by the genes it contains.

Details

This fast implementation of fold change signature scores uses matrix multiplication. The score is simply: mean(fold change of genes in signature) - mean(fold change of genes outside signature).

Value

Dataframe with one row per chemical/conc/signature combination. Columns are: sample_id, dtxsid, casrn, name, time, conc, sigset, signature, size (signature size accounting for missing values), mean_fc_scaled_in, mean_fc_scaled_out, signature_score.

`signatureScoreCoreGSVA` *Signature Score Core - GSVA*

Description

Computes GSVA signature scores.

Usage

```
signatureScoreCoreGSVA(
  sk.list,
  sigset = "FILTERED",
  dataset,
  fcmat,
  chem_dict,
  signature_data,
  mc.cores = 1
)
```

Arguments

`sk.list`	Sample keys to use; should correspond to fcmat rownames.
`sigset`	Name of signature set.
`dataset`	Name of data set.
`fcmat`	Sample by gene matrix of log2(fold change)'s. Rownames are sample keys and colnames are genes.
`chem_dict`	Dataframe with one row per sample key and seven columns: sample_key, sample_id, conc, time, casrn, name, dtxsid.
`signature_data`	Named ist of gene name vectors. Each element is one signature, defined by the genes it contains.
`mc.cores`	Number of cores to use. Parallelization is performed by gsva itself.

Details

This function is a wrapper for GSVA with Gaussian cdf kernels. signaturescoremat output is saved directly to disk.

Value

No output.

signatureScoreCoreMYGSEA *Signature Score Core - MYGSEA*

Description

Computes signature scores for mygsea.

Usage

```
signatureScoreCoreMYGSEA(
  sk.list,
  method = "mygsea",
  sigset,
  dataset,
  fcmat,
  chem_dict,
  signature_data,
  mc.cores = 1,
  normalization = T,
  useranks = T
)
```

Arguments

sk.list	Sample keys to use; should correspond to fcmat rownames.
method	Method name to use in file output. "mygsea" or "mygsea_norank"
sigset	Name of signature set.
dataset	Name of data set.
fcmat	Sample by gene matrix of log2(fold change)'s. Rownames are sample keys and colnames are genes.
chem_dict	Dataframe with one row per sample key and seven columns: sample_key, sample_id, conc, time, casrn, name, dtxsid.
signature_data	Named ist of gene name vectors. Each element is one signature, defined by the genes it contains.
mc.cores	Number of cores to use. Parallelization is performed by gsva itself.
normalization	normalization = T normalizes final scores.
useranks	useranks = T uses score ranks for weighting; otherwise, fold changes are used for weights.

Details

This function is a parallelized wrapper for MYGSEA, which does the actual scoring. mygsea method uses ranks and normalization, while mygsea_norank method does not use ranks or normalization. Normalization divides final scores by difference between max and min score. Without normalization, scores from individual samples have no impact on each other. Final signaturescoremat is written to disk.

Value

No output.

---

signatureScoreMerge *Merge the up and down halves of the pathway data*

Description

Merge the up and down halves of the pathway data

Usage

```
signatureScoreMerge(
  sigset = "pilot_small",
  sigcatalog = "signatureDB_master_catalog 2020-01-31",
  dataset = "DMEM_6hr_pilot_normal_pe_1_RAND1000",
  method = "mygsea",
  nullset = "DMEM_6hr_pilot_normal_pe_1_RAND1000"
)
```

Arguments

sigset	Name of the signature set.
dataset	Name of the data set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
nullset	Name of the null data set.
sigcatlog	Nmae of the catalog file

Value

nothing

signatureTestMerging *Examine the merging code effect*

Description

Examine the merging code effect

Usage

```
signatureTestMerging(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  sigset = "pilot_tiny",
  method = "mygsea",
  cname = "Fulvestrant"
)
```

Arguments

to.file	If TRUE, write plots to a file

smoothecdf *Smooth ECDF*

Description

Converts a data frame containing bmd, bmdl, bmu, to a smooth ecdf.

Usage

```
smoothecdf(x, mymat, verbose = F, bmdrange = c(0.001, 100))
```

Arguments

x	ECDF plotting location x-values.
mymat	Dataframe containing bmd, bmdu, bmdl, and hitcall columns.
verbose	verbose = F suppresses both bounds NA warning.
bmdrange	Maximum expected BMD range. The farthest value from the bmd is used to compute standard deviation of gaussian when both bounds are missing.

Details

Models each bmd as a gaussian with mean bmd uses bmdl (bmdu if bmdl is na) to compute sd. Each gaussian is scaled by the hitcall.

Value

Outputs a vector corresponding to the locations in x.

Examples

```
x = 10^(-50:50/30)
mymat =data.frame(list(bmd = c(.1,1,10), bmdl = c(.05,NA,NA),
  bmdu = c(.6,1.5,NA), hitcall = c(1,1,1)))
out = smoothecdf(x, mymat)
plot(log10(x),out, type = "l")
mymat$hitcall = c(1,.5,0)
out2 = smoothecdf(x, mymat)
plot(log10(x),out2, type = "l")
```

---

stressPathwayHM *Build a heatmap of the stress genes*

Description

Build a heatmap of the stress genes

Usage

```
stressPathwayHM(
  to.file = F,
  dataset = "DMEM_6hr_pilot_normal_pe_0",
  pathset = "PathwaySet_20191107",
  threshold = 0.5,
  method = "fc"
)
```

Arguments

to.file	If TRUE, write plots to a file

---

toxcastPOD *Get the TOxCast PODs using input from*

Description

Get the TOxCast PODs using input from

Usage

```
toxcastPOD(do.prep = F)
```

TxT *Calculate several statistics on a 2 x 2 matrix*

Description

Calculate several statistics on a 2 x 2 matrix

Usage

```
TxT(tp, fp, fn, tn, do.p = TRUE, rowname = NA)
```

Arguments

tp	number of true positives
fp	number of false positives
fn	number of false negatives
tn	number of true negatives
do.p	if TRUE, calcualte an exact p-value
rowname	if not NA, adda column to the output with this rowname
	Returns: a list of the results a: TP b: FP c: FN d: TN sens: sensitivity spec: specificity ba: Balanced Accuracy accuracy: Accuracy relative.risk: Relative Risk odds.ratio: Odds Ratio or.ci.lwr: lower confidence interval of the Odds Ratio or.ci.upr: upper confidence interval of the Odds Ratio ppv: Positive Predictive Value npv: Negative Predictive Value p.value: Chi-squared p-value F1: 2TP/(2TP+FP+FN)
	sval: All of the results as a tab-delimited string title: the title of the results as a tab-delimited string mat: The results as a 1-row data frame @export

WRMSE *Weighted Root-mean-square-error*

Description

Computes root-mean-square error with weighted average.

Usage

```
WRMSE(x, y, w)
```

Arguments

x	First vector of numbers.
y	Second vector of numbers.
w	Vector of weights.

Details

x,y,w should all be the same length. Order of x and y won't change output.

Value

Weighted RMSE.

Examples

```
WRMSE(1:3, c(1,3,5), 1:3)
```

Index