Package 'httrpathway'

May 21, 2021

Type Package

Title Pathway Scoring and Concentration Response for HTTr data

Version 1.1.0

Author Thomas Sheffield

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

Description

This package generates pathway (signature) scores with associated concentration response modeling; it also contains some important plotting functions. This package contains functions required to create input files (log2-fold change, or (l2fc) matrices) and run the signature/pathway based concentration-
response calculations. Another R project (httranalysis) contains a series of post-calculation analyses that are problem-specific. To run all of the calculations, use the function driver(). This version has also included gene-level concentration-response modeling
This package required a set of directories to be at the same level as the httrpathway folder
../input - various input files
../input/chemicals - collections of chemical information, not used in the standard calculations
../input/signatures -
the signature data, inluding the catalog (an Excel file) and the lists of genes per signature
../input/fcdata - where the l2fc data goes. See the functions buildFCMAT1 and buildFC-MAT2 for more information. These functions may need to be customized-
for the source of your data
../output - where all of the output goes [not clear if the subfolders are created on demand]
There are a series of data sets / objects that are names and carried around:
* dataset - this is the name of the data set being used. It corresponds to an experiment and the name ususally contains the cell type, the type of normalization, the time, media, etc. All input and output files will contain this dataset name
* sigcatalog - This is the name of the signature catalog. This is an excel file that lives in ../input/signatures.This file contains one row per signature and contains matching annotations such as the super_target
* sigset - One always uses a subset of the total set of signtures, indicated by having a value of 1 in the sigset column at the right hand of the signature catalog

Imports stats,
stringr,
grDevices,
graphics,
utils,
methods,
data.table,

future.apply,
future,
GSVA,
moments,
numDeriv,
openxlsx,
parallel,
RColorBrewer,
reshape2,
data.table,
openxlsx,
e1071,
tidyverse

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`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))
```

`baseline_gene_counts` *Gene the baseline gene counts for the cell atlas project*

Description

Gene the baseline gene counts for the cell atlas project

Usage

```
baseline_gene_counts(
  db = "httr_cell_atlas",
  dir = "../input/rawdata/cellatlas/"
)
```

Arguments

`db`	The name of the Mongo database
`dir`	The directory where the data will be stored This functions takes files created by export_mongo_httr_well() * httr_cell_atlas * httr_tox21_cpp2

`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.

---

buildFCMAT1.fromDB *Build the FCMAT1 data set*

Description

version to start with Logan's database export The difference between this version and the original is that there are extra columns The function just changes one column name and writes the file to a standard name and place

Usage

```
buildFCMAT1.fromDB(
  dataset = "mcf7_ph1_pe1_normal_block_123_excludePG",
  dir = "../input/fcdata/new_versions/",

    infile = "httr_mcf7_ph1_bl123_FCmat1_meanncnt0_5-plateteffect_1-shrinkage_nc
  pg.filter.file = "httr_mcf7_ph1_flagged_pg_block_123_exclude.xlsx",
  do.load = T
)
```

Arguments

dataset	The name to give to the data set
dir	The directory from which to read all of the raw files
infile	The nae of the input file
pg.filter.file	An optional file to use in filtering out bad plate groups
do.load	If TRUE, read the large input data file into memory

Value

A file with the FCMAT1 data is written to "../input/fcdata/FCMAT1_",dataset,".RData"

`buildFCMAT2.fromDB` *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.fromDB(
  dataset = "mcf7_ph1_pe1_normal_block_123_excludePG",
  time = 24,
  media = "DMEM",
  dir = "../input/fcdata/",
  method = "gene",
  do.read = T
)
```

Arguments

`dataset`	The name to give to the data set
`time`	The time in hours that the chemical dosing was run
`media`	THe name of the media used
`dir`	The directory from which to read all of the raw files
`method`	Either "gene" or "probe"
`do.read`	If TRUE, read in the FCMAT1 file and place in a global.

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

`buildFCMAT2.fromDB.refchems`
Transpose and filter the fold change matrix FCMAT1 in long format into a gene x sample format. This is the method to use when there are conc-response profiles of refchems

Description

Transpose and filter the fold change matrix FCMAT1 in long format into a gene x sample format. This is the method to use when there are conc-response profiles of refchems

Usage

```
buildFCMAT2.fromDB.refchems(
  dataset = "heparg2d_toxcast_pfas_pe1_normal",
  time = 24,
  media = "DMEM",
  dir = "../input/fcdata/",
  method = "gene",
  do.read = F,
  do.prep = F
)
```

Arguments

dataset	The name to give to the data set
time	The time in hours that the chemical dosing was run
media	THe name of the media used
dir	The directory from which to read all of the raw files
method	Either "gene" or "probe"
do.read	If TRUE, read in the FCMAT1 file and place in a global.

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

---

buildSampleMap *Generate the sample_key x sample x DSSTox file*

Description

Generate the sample_key x sample x DSSTox file

Usage

```
buildSampleMap(
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  dsstox.file = "../input/DSSTox/DSSTox_sample_map.xlsx",
  dir = "../input/fcdata/",
  outfile = "../input/chemicals/HTTr_pilot_sample_map.xlsx",
  do.read = F
)
```

Arguments

dataset	Name of hte HTTr dataset
dsstox.file	Name of the DSStox chemical file
dir	Directory where the FCMAT1 files lives
outfile	Name of the output file
do.read	If TRUE, read in the input FCMAT1 file

`buildStudyChemicalMap` *Build a catalog of the chemicals in a dataset*

Description

Build a catalog of the chemicals in a dataset

Usage

```
buildStudyChemicalMap(dataset = "DMEM_6hr_screen_normal_pe_1")
```

Arguments

`dataset`	The name of the HTTr dataset

Value

No output.

`calcDEG` *Calculate the relative variability of genes to get the DEGs*

Description

Calculate the relative variability of genes to get the DEGs

Usage

```
calcDEG(
  dataset = "mcf7_ph1_pe1_normal_good_pg",
  dir = "../input/fcdata/",
  do.read = T
)
```

Arguments

`dataset`	The name to give to the data set
`dir`	The directory from which to read all of the raw filesatalog file
`do.read`	If TRUE, read in the HTTr data file

concatDESeq2Files *Concatenate the input DESeq2 files*

Description

Concatenate the input DESeq2 files

Usage

```
concatDESeq2Files(
  dataset = "DMEM_6hr_screen_normal_pe_1",

    indir = "../input/httr_mcf7_screen/meanncnt0_5-plateteffect_0-shrinkage_norm
  outdir = "../input/httr_mcf7_screen/"
)
```

Arguments

dataset	The name of the HTTr dataset
indir	The director to read from
outdir	The directory to write to

driver *Code to run all signature concentration-response calculations*

Description

Code to run all signature concentration-response calculations

Usage

```
driver(
  dataset = "mcf7_ph1_pe1_normal_block_123_allPG",
  sigcatalog = "signatureDB_master_catalog 2021-05-10",
  sigset = "screen_large",
  nullset = NULL,
  nrandom.chems = 1000,
  normfactor = 7500,
  mc.cores = 25,
  bmr_scale = 1.349,
  plotrange = c(1e-04, 100),
  method = "mygsea",
  celltype = "MCF7",
  do.build.random = T,
  do.run.random = T,
  do.run.all = T,
  do.scr.plots = T,
  do.signature.pod = T,
  do.supertarget.boxplot = T,
  do.all = F
)
```

Arguments

dataset	Name of the data set, produced by buildFCMAT2
sigcatalog	Name of the signature catalog
sigset	Name if the signature set. THis corresponds to a column in the signature catalog file
nullset	The name of the NULL set if it is custom, default is NULL
nrandom.chems	Number of random chemicals for the NULL distribution calculation, default is 1000
normfactor	Normalization factor for the conc-reap plots, default is 7500
mc.cores	Number of cores for paralell processing. Only works under Linux
bmr_scale	Scaling factor from the NULL SD to BMD, default is 1.349,
plotrange	The concentration range for the conc-resp plots in uM, default is c(0.0001,100),
method	signature scoring method in c("fc", "gsva", "mygsea"), default is fc
celltype	Name of the cull type, e.g. MCF7
do.build.random	If TRUE, build the random dataset
do.run.random	If True, run the calculations on the random data set. This is used to generate the NULL distribution
do.run.all	If true, run the calculations on the real data set
do.scr.plots	If TRUE, generate the signature concentration response plots
do.signature.pod	If TRUE, generate the signature PODs
do.supertarget.boxplot	If TRUE, generate the super target box plots
do.all	If TRUE, do all steps from do.build.random to the end
	Available data sets * heparg2d_toxcast_pfas_pe1_normal * mcf7_ph1_pe1_normal_block_123_allPC * mcf7_ph1_pe1_normal_block_123_excludePG * u2os_toxcast_pfas_pe1_normal * PFAS_HepaRG * PFAS_U2OS * u2os_pilot_pe1_normal_null_pilot_lowconc * u2os_toxcast_pfas_pe1_normal_refchems * heparg2d_toxcast_pfas_pe1_normal_refchems * DMEM_6hr_pilot_normal_pe_1 - MCF7 pilot
do.signature.summary.plot	if TRUE, generate the summary plots
do.signature.pod.laneplot	If TRUE, generate the signature lane plots (only useful for small sets of chemicals)

exportDSSToxSample *Generate the sample x DSSTox file*

Description

Generate the sample x DSSTox file

Usage

```
exportDSSToxSample(outfile = "../input/DSSTox/DSSTox_sample_map.xlsx")
```

Arguments

outfile	Name of the file to be written

exportSignatureCutoffs *Export the signature-wise cutoffs*

Description

Export the signature-wise cutoffs

Usage

```
exportSignatureCutoffs(
  do.load = F,
  dataset = "mcf7_ph1_pe1_normal_block_123_excludePG",
  sigset = "screen_large",
  method = "fc"
)
```

Arguments

do.load	If TRUE, load hte large data file
dataset	The name of the HTTr data set to use
sigset	The name of the signature set to use
method	The scoring method to use

`export_mongo_httr_well` *Get the raw counts from the Mongo database*

Description

Get the raw counts from the Mongo database

Usage

```
export_mongo_httr_well(
  db = "httr_cell_atlas",
  collection = "httr_well_trt",
  dir = "../input/rawdata/cellatlas/"
)
```

Arguments

`db`	The name of the Mongo database
`collection`	THe name of the collection to export
`dir`	The directory where the data will be stored Collections * httr_cell_atlas * httr_tox21_cpp2

`fixSuperTarget` *Replace the super_target values in the signature output file with ones from a new catalog*

Description

Replace the super_target values in the signature output file with ones from a new catalog

Usage

```
fixSuperTarget(
  do.read = T,
  dataset = "PFAS_U2OS",
  sigcatalog = "signatureDB_master_catalog 2021-05-10",
  sigset = "screen_large",
  method = "fc"
)
```

Arguments

`do.read`	If TRUE, read in FCMAT2 to a global
`dataset`	The L2fc matrix data set
`sigcatalog`	THe name of the signature catalog file
`sigset`	The name of the signature set to use
`method`	The scoring method

geneBaseMeanDist *get the base mean distribution for each gene*

Description

get the base mean distribution for each gene

Usage

```
geneBaseMeanDist(
  to.file = F,
  do.read = F,
  dataset = "DMEM_6hr_screen_normal_pe_1"
)
```

Arguments

to.file	If TRUE, plot to a file
do.read	If TRUE, read the input file into memory
dataset	The name of the dataset

Value

No output.

geneConcResp *Gene Concentration Response*

Description

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

Usage

```
geneConcResp(
  dataset = "heparg2d_toxcast_pfas_pe1_normal_refchems",
  mc.cores = 20,
  to.file = T,
  pval = 0.05,
  nametag = NULL,
  conthits = T,
  aicc = F,
  fitmodels = c("cnst", "hill", "poly1", "poly2", "pow", "exp2", "exp3", "exp4",
    "exp5"),
  genefile = NULL
)
```

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.

---

geneConcRespPlot *Pathway Concentration Response Plot*

Description

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

Usage

```
geneConcRespPlot(row, plotrange = c(0.001, 100))
```

Arguments

row	Named list containing: • conc - conc string separated by l's • resp - response string separated by l'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.

`plotrange` The x-range of the plot as a vector of 2 elements, this can be changed for special cases, but defaults to 0.001 to 100

Details

row is one row of PATHWAY_CR, the signatureConcResp output.

Value

No output.

`geneConcRespPlotWrapper` *Wrapper for all of the conc-response plotting o genes*

Description

Wrapper for all of the conc-response plotting o genes

Usage

```
geneConcRespPlotWrapper(
  dataset = "heparg2d_toxcast_pfas_pe1_normal",
  mc.cores = 20,
  do.load = T,
  to.file = F,
  pval = 0.05,
  nametag = "_conthits",
  plotrange = c(1e-04, 100),
  onefile = T,
  chemfile = "../input/PFAS/Immuntox chemical evidence.xlsx"
)
```

Arguments

`dataset`	Name of the data set.
`mc.cores`	Number of cores to parallelize with.
`do.load`	If TRUE, load the SIGNATURE_CR file, otherwiseassume that it is in memory
`to.file`	to.file = T saves the output to a file; otherwise it's returned.
`pval`	Desired cutoff p-value.
`nametag`	Optional descriptor tag to attach to file outputs for experimental/non-default runs.
`plotrange`	The x-range of the plot as a vector of 2 elements, this can be changed for special cases, but defaults to 0.001 to 100
`onefile`	If TRUE, put all plots into one file, instead of one filer per chemical
`chemfile`	A file of chemicals to use. If NULL, plot all chemicals

`getpvalcutoff` *Get P-Value Cutoff*

Description

Retrieves signature cutoffs for a given null dataset.

Usage

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

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.
`verbose`	If TRUE, write extra output

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).

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)
```

pg_id.to.sample_id *get the mapping between the plate groups and the samples*

Description

version to start with Logan's database export The difference between this version and the original is that there are extra columns The function just changes one column name and writes the file to a standard name and place

Usage

```
pg_id.to.sample_id(
  do.load = F,
  dataset = "mcf7_ph1_pe1_normal_block_123_allPG",
  dir = "../input/fcdata/"
)
```

Arguments

do.load	if T, load the initial file
dataset	The name to give to the data set
dir	The directory from which to read all of the raw files

Value

A mapping the sampels to the plate groups f

plotouter *Plot Outer*

Description

Calls signatureConcResp plotting function.

Usage

```
plotouter(proper_name, SIGNATURE_CR, foldname, plotrange = c(0.001, 100))
```

Arguments

proper_name	Chemical name to be used in file name.
SIGNATURE_CR	Dataframe output of signatureConcResp_pval.
foldname	Folder name for output file.
plotrange	The x-range of the plot as a vector of 2 elements, this can be changed for special cases, but defaults to 0.001 to 100

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.

`plotouterGene` *Plot Outer*

Description

Calls signatureConcResp plotting function.

Usage

```
plotouterGene(proper_name, GENE_CR, foldname, plotrange = c(0.001, 100))
```

Arguments

`proper_name`	Chemical name to be used in file name.
`GENE_CR`	Dataframe output of geneConcResp_pval.
`foldname`	Folder name for output file.
`plotrange`	The x-range of the plot as a vector of 2 elements, this can be changed for special cases, but defaults to 0.001 to 100

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 = "DMEM_6hr_pilot_normal_pe_1",
  sigset = "pilot_large_all_100CMAP",
  method = "mygsea",
  hccut = 0.9,
  plot.signature_min = F,
  bmd.mode = "percent"
)
```

Arguments

to.file	If TRUE, write plots to a file
dataset	The data set to use
sigset	THe signature set to use
method	Scoring method
hccut	Exclude rows with hitcall less than this value
bmd.mode	percent or abs
plot.signature.min	If TRUE, plot the minimum signature

---

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

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 = "u2os_pilot_pe1_normal_null_pilot_lowconc",
  nchem = 1000,
  seed = 12345,
  maxconc = 1e+06,
  nlowconc = 2,
  dtxsid.exclude = NULL
)
```

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.
maxconc	Only use concentrations less than maxconc, default 1000000
nlowconc	If not NULL, only include the loest nlowconc concentrations for eahc chemical
dtxsid.exclude	dtxsids to exclude, default NULL for U2OS pilot dtxsid.exclude=c('DTXSID9020031','DTXSID004

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.

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 = "mygsea",
  bmr_scale = 1.349,
  normfactor = 7500,
  minsigsize = 10,
  conthits = T,
  nullset,
  do.plot = T,
```

```
  do.cr = 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.
sigcatalog	Name of the signature catalog
method	Pathway scoring method in c("fc", "gsva", "mygsea")
bmr_scale	bmr scaling factor. Default = 1.349
normfactor	Factor to scale the native units up by to get onto a reasonable plotting value (~ -1 to 1)
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.
do.cr	Run the concentration-response step (set to FALSE for the null set)
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 = "ccte-mysql-res.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

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 = "wgcna",
  sigcatalog = "signatureDB_wgcna_mcf7_ph1_pe1_normal_good_pg_MCF7_12_10_catalog
)
```

Arguments

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

Value

the trimmed signature table

signatureConcResp *Pathway Concentration Response (P-value)*

Description

Performs signature concentration response using p-value based cutoffs.

Usage

```
signatureConcResp(
  sigset = "pilot_tiny",
  sigcatalog = "signatureDB_master_catalog 2020-03-12",
  dataset = "DMEM_6hr_screen_normal_pe_1_pgnorm",
  method = "mygsea",
  bmr_scale = 1.349,
  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")
)
```

Arguments

sigset	Name of the signature set.
dataset	Name of the data set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
bmr_scale	bmr scaling factor. Default = 1.349
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.

---

signatureConcRespFilter

Filter the conc-response data for just the most potent results and plot the conc-response curves if desired

---

Description

Filter the conc-response data for just the most potent results and plot the conc-response curves if desired

Usage

```
signatureConcRespFilter(
  to.file = F,
  do.plot = F,
  do.load = T,
  hccut = 0.9,
  tccut = 1.5,
  dataset = "heparg2d_toxcast_pfas_pe1_normal",
  sigset = "screen_large",
  method = "fc",
  do.pfas = F
)
```

Arguments

to.file	If TRUE, send plots to a file
do.plot	If TRUE do the plotting
do.load	If TRUE, load the data file
hccut	Exclude rows with hitcall below this value
tccut	Exclude rows with top_over_cutoff below this value
dataset	Dataset to use
sigset	Signature set to use
method	signature scoring method in c("fc", "gsva", "mygsea")
do.pfas=F	Error bars are exp(er)*qt(.025,4) = exp(er)*2.7765

---

signatureConcRespNullDistThreshold

Export the Null set thresholds

Description

Null dataset and dataset should have already been scored using signatureScore and the given sigset and method.

Usage

```
signatureConcRespNullDistThreshold(
  sigset = "screen_large",
  dataset = "mcf7_ph1_pe1_normal_all_pg",
  method = "fc",
  pval = 0.05
)
```

Arguments

sigset	Name of the signature set.
dataset	Name of the data set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
pval	Desired cutoff p-value.

`signatureConcRespPlot` *Pathway Concentration Response Plot*

Description

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

Usage

```
signatureConcRespPlot(row, plotrange = c(0.001, 100))
```

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.
`plotrange`	The x-range of the plot as a vector of 2 elements, this can be changed for special cases, but defaults to 0.001 to 100

Details

row is one row of PATHWAY_CR, the signatureConcResp output.

Value

No output.

`signatureConcRespPlotWrapper` *Wrapper for all of the conc-response plotting*

Description

Wrapper for all of the conc-response plotting

Usage

```
signatureConcRespPlotWrapper(
  sigset = "screen_large",
  dataset = "u2os_pilot_pe1_normal_null_pilot_lowconc",
  sigcatalog = "signatureDB_master_catalog 2021-03-05",
  method = "fc",
  bmr_scale = 1.349,
  mc.cores = 20,
  do.load = T,
  pval = 0.05,
  nametag = "_conthits",
  plotrange = c(1e-04, 100)
)
```

Arguments

`sigset`	Name of the signature set.
`dataset`	Name of the data set.
`sigcatalog`	Name of the signature catalog
`method`	Pathway scoring method in c("fc", "gsva", "mygsea")
`bmr_scale`	bmr scaling factor. Default = 1.349
`mc.cores`	Number of cores to parallelize with.
`do.load`	If TRUE, load the SIGNATURE_CR file, otherwiseassume that it is in memory to.file to.file = T saves the output to a file; otherwise it's returned.
`pval`	Desired cutoff p-value.
`nametag`	Optional descriptor tag to attach to file outputs for experimental/non-default runs.
`plotrange`	The x-range of the plot as a vector of 2 elements, this can be changed for special cases, but defaults to 0.001 to 100

`signatureConcRespToZ` *Convert the conc-response data to a z score*

Description

Convert the conc-response data to a z score

Usage

```
signatureConcRespToZ(
  do.load = T,
  mc.cores = 2,
  dataset = "heparg2d_toxcast_pfas_pe1_normal",
  sigset = "screen_large",
  method = "fc",
  celltype = "HepaRG",
  hccut = 0.95,
  tccut = 1.5
)
```

Arguments

`do.load`	If TRUE, load hte large HTTr data set
`mc.cores`	NUmber of cores to use in multi-core mode=2,
`dataset`	Name of the HTTr data set being used
`sigset`	Name of the signature set used
`method`	Scoring method used
`celltype`	name of cell type ebing used
`hccut`	Exclude signature rows with hitcall less than this value
`tccut`	Exclude signature rows with top_over_cutoff less than this value

`signaturePOD` *Calculate PODs at the signature level*

Description

Calculate PODs at the signature level

Usage

```
signaturePOD(
  do.load = F,
  sigset = "screen_large",
  dataset = "PFAS_U2OS",
  method = "fc",
  bmr_scale = 1.349,
  hccut = 0.95
)
```

Arguments

sigset	Name of signature set.
dataset	Name of data set.
method	Pathway scoring method in c("fc", "gsva", "mygsea")
bmr_scale	bmr scaling factor. Default = 1.349
hccut	Remove rows with hitcall less than this value
do.laod	If TRUE, load the input data into memory

---

signaturePOD.BMRcompare

Compare the PODs with different BMR values

Description

Compare the PODs with different BMR values

Usage

```
signaturePOD.BMRcompare(
  to.file = F,
  dataset = "mcf7_ph1_pe1_normal_block_123",
  sigset = "screen_large",
  method = "fc",
  bmr_scale = 1,
  hccut = 0.9
)
```

Arguments

to.file	If TRUE, write plots to a file
dataset	Name of data set.
sigset	Name of signature set.
method	Pathway scoring method
bmr_scale	bmr scaling factor. Default = 1.349
hccut	Remove rows with hitcall less than this value

`signaturePODsummary` *Summarize the POD overlap with ToxCast*

Description

Summarize the POD overlap with ToxCast

Usage

```
signaturePODsummary(
  sigset = "pilot_large_all_100CMAP",
  dataset = "DMEM_6hr_pilot_normal_pe_1",
  method = "mygsea"
)
```

Arguments

`sigset`	THe name of the signature set
`dataset`	Name of the HTTr data set
`method`	THe signature scoring method

`signatureScore` *Signature Score*

Description

Computes and saves signature scores.

Usage

```
signatureScore(
  FCMAT2,
  CHEM_DICT,
  sigset,
  sigcatalog,
  dataset,
  method,
  normfactor = 7500,
  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.
sigcatalog	Name of the signature catalog file
dataset	Name of data set.
method	Signature scoring method in c("fc", "gsva", "mygsea")
normfactor	Value passed ot the plotting code to scale the y values
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,
  ngenemax = NULL,
  verbose = F
)
```

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.
ngenemax	If ngene is not NULL, then tonly the most extreme n genes of the signature will be used for the "in" set
verbose	If TRUE, weite extra diagnostic output

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",
  normfactor = 7500,
  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 = "screen_large",
  sigcatalog = "signatureDB_master_catalog 2020-04-04",
  dataset = "DMEM_6hr_screen_normal_pe_1_RAND1000",
  method = "mygsea",
  nullset = "DMEM_6hr_screen_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

---

superTargetBoxplot *Generate chemical-wise boxplot of the BMD distributions by super_target*

Description

Generate chemical-wise boxplot of the BMD distributions by super_target

Usage

```
superTargetBoxplot(
  to.file = T,
  do.load = T,
  dataset = "heparg2d_toxcast_pfas_pe1_normal",
  sigcatalog = "signatureDB_master_catalog 2021-04-24",
  sigset = "screen_large",
  method = "fc",
  celltype = "HepaRG",
  hccut = 0.95,
  tccut = 1.5,
  minconc = 0.001,
  maxconc = 100
)
```

Arguments

to.file	If TRUE, send the plots to a file
do.load	If TRUE, load hte large HTTr data set into memory
dataset	Name of the HTTr data set
sigcatalog	Name of the signature catalog to use
sigset	Name of the signature set
method	Scoring method
celltype	Name of the cell type
hccut	Exclude rows in the data set with hitcall less than this value
tccut	Exclude rows in the data set with top_over_cutoff less than this value
minconc	Minimum concentration used in the plots
maxconc	Maximum concentration used in the plots
	After running this, run the following ... superTargetPODplot superTargetStats

`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