Package 'httrpathway'

October 15, 2019

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

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`acgnlsobj` *AC GNLS Objective Function*

Description

GNLS objective function set to y for gnls solver.

Usage

```
acgnlsobj(x, y, tp, ga, p, la, q)
```

Arguments

`x`	Concentration.
`y`	Desired activity level.
`tp`	Top.
`ga`	Gain AC50.
`p`	Gain power.
`la`	Loss AC50.
`q`	Loss power.

Value

Difference between GNLS model repsone at x and y.

acy *Activity Concentration y*

Description

Returns concentration at which model equals y.

Usage

```
acy(y, modpars, type = "hill", returntop = F, returntoploc = F,
  getloss = F, verbose = F)
```

Arguments

y	Activity value at which the concentration is desired. y should be less than the model's top, if there is one, and greater than zero.
modpars	List of named model parameters. Model parameters can include: "a", "b", "ga", "la", "p", "q", "tp". ga and la should NOT be in log units.
type	Model type; must be one of: "exp1", "exp2", "exp3", "exp4", "gnls", "hill", "poly1", "poly2", "pow".
returntop	When TRUE, returns actual top value for gnls. Has no effect for other models.
returntoploc	When TRUE, returns concentration of top for gnls. Has no effect for other models. If top location can't be found, NA is returned.
getloss	When TRUE, returns value on loss side of curve for gnls. Has no effect for other models.
verbose	When TRUE, shows warnings.

Details

Mathematically inverts model functions of the given type, except for gnls, which is numerically inverted. gnls returns NA when y > tp. Other options return the actual top (as opposed to theoretical tp) and top location for gnls model. gnls model defaults to giving concentration on gain side. Only one of getloss, returntop, and returntoploc should be TRUE at a time. If top location solution fails for gnls, top is set to tp. Returns NA if gnls numerical solver fails.

Value

Ouputs concentration at activity y, or gnls top or top concentration, when applicable.

Examples

```
acy(1, list(ga = 10, tp = 2, p = 3), type = "hill")
acy(1, list(ga = .1, tp = 2, p = 3, q = 3,la = 10), type = "gnls")
acy(1, list(ga = .1, tp = 2, p = 3, q = 3,la = 10), type = "gnls", getloss = TRUE)
acy(1, list(ga = .1, tp = 2, p = 3, q = 3,la = 10), type = "gnls", returntop = TRUE)
acy(1, list(ga = .1, tp = 2, p = 3, q = 3,la = 10), type = "gnls", returntoploc = TRUE)
```

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.

`bmdbounds` *BMD Bounds*

Description

Computes BMDU or BMDL.

Usage

```
bmdbounds(fit_method, bmr, pars, conc, resp, onesidedp = 0.05,
  bmd = NULL, which.bound = "lower")
```

Arguments

`fit_method`	Fit method: "exp2", "exp3", "exp4", "exp5", "hill", "gnls", "poly1", "poly2", or "pow".
`bmr`	Benchmark response.
`pars`	Named vector of model parameters: a,b,tp,ga,p,la,q,er output by httrfit, and in that order.
`conc`	Vector of concentrations (NOT in log units).
`resp`	Vector of responses corresponding to given concentrations.
`onesidedp`	The one-sided p-value. Default of .05 corresponds to 5 percentile BMDL, 95 percentile BMDU, and 90 percent CI.
`bmd`	Can optionally input the bmd when already known to avoid unnecessary calculation.
`which.bound`	Returns BMDU if which.bound = "upper"; returns BMDL if which.bound = "lower".

Details

Takes in concentration response fit details and outputs a bmdu or bmdl, as desired. If bmd is not finite, returns NA. If the objective function doesn't change sign or the root finding otherwise fails, it returns NA. These failures are not uncommon since some curves just don't reach the desired confidence level.

Value

Returns either the BMDU or BMDL.

Examples

```
conc = c(.03, .1, .3, 1, 3, 10, 30, 100)
resp = c(.1,-.1,0,1.1,1.9,2,2.1,1.9)
pars = c(tp = 1.973356, ga = 0.9401224, p = 3.589397, er =  -2.698579)
bmdbounds(fit_method = "hill", bmr = .5, pars, conc, resp)
bmdbounds(fit_method = "hill", bmr = .5, pars, conc, resp, which.bound = "upper")
```

---

bmdobj *BMD Objective Function*

Description

Utility function for bmdbounds

Usage

```
bmdobj(bmd, fname, bmr, conc, resp, ps, mll, onesp, partype = 2)
```

Arguments

bmd	Benchmark dose.
fname	Function name: "exp2", "exp3", "exp4", "exp5", "hillfn", "gnls", "poly1", "poly2", or "pow".
bmr	Benchmark response.
conc	Vector of concentrations NOT in log units.
resp	Vector of corresponding responses.
ps	Named list of paramters.
mll	Maximum log-likelihood of winning model.
onesp	One-sided p-value.
partype	Number for parameter type. Type 1 is y-scaling: a or tp. Type 2 is x-scaling: b or ga, when available, a otherwise. Type 3 is power scaling: p when available, then b or ga, then a if no others. Since bmd is linked to the x-scale, type 2 should always be used. Other types can also be vulnerable to underflow/overflow.

Value

Objective function value to find the zero of.

CHEM_DICT *Chemical Dictionary Example*

Description

Chemical dictionary for 10 randomly chosen chemicals from the phase 1 screen.

Usage

```
CHEM_DICT
```

Format

A data frame with 80 rows and 7 variables: sample_key (sample id + conc), sample_id (unique identifier for each sample), conc (concentration), time (length of the experiment in hours), casrn, name (chemical name), dsstox_substance_id.

cnst *Constant Model*

Description

Constant Model

Usage

```
cnst(ps, x)
```

Arguments

ps	Vector of parameters (ignored)
x	Vector of concentrations (regular units)

Value

Vector of model responses

ENDOCRINE_PATHWAYS *Endocrine Pathways Example*

Description

Pathway dictionary for 44 ER/AR related pathways

Usage

```
ENDOCRINE_PATHWAYS
```

Format

A data frame with 44 rows and 8 variables: pathset, pathway, ngene, ngene_in_httr, pathway_class, pathway_super_class, url, gene_list.

exp2 *Exponential 2 Model*

Description

Exponential 2 Model

Usage

```
exp2(ps, x)
```

Arguments

ps	Vector of parameters: a,b,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

exp3 *Exponential 3 Model*

Description

Exponential 3 Model

Usage

```
exp3(ps, x)
```

Arguments

ps	Vector of parameters: a,b,p,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

exp4 *Exponential 4 Model*

Description

Exponential 4 Model

Usage

```
exp4(ps, x)
```

Arguments

ps	Vector of parameters: tp,ga,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

exp5 *Exponential 5 Model*

Description

Exponential 5 Model

Usage

```
exp5(ps, x)
```

Arguments

ps	Vector of parameters: tp,ga,p,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

FCMAT2 *Fold Change Matrix Example*

Description

Fold change matrix for 10 randomly selected chemicals from the phase 1 screen.

Usage

```
FCMAT2
```

Format

A sample by gene matrix with 80 rows and 10,341 columns.

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 pathway 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.

fitcnst *Constant Model Fit*

Description

Function that fits a constant line and returns generic model outputs.

Usage

```
fitcnst(conc, resp, nofit = F)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
nofit	If nofit = T, returns formatted output filled with missing values.

Details

success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. aic, rme, and er are set to NA in case of nofit or failure. pars always equals "er".

Value

List of five elements: success, aic (Aikaike Information Criteria), rme (root mean square error), er (error parameter), pars (parameter names).

Examples

```
fitcnst(c(.1,1,10,100), c(1,2,0,-1))
fitcnst(c(.1,1,10,100), c(1,2,0,-1), nofit = TRUE)
```

`fitexp2` *Exponential 2 Model Fit*

Description

Function that fits to f(x) = a*(e^(x/b)- 1) and returns generic model outputs.

Usage

```
fitexp2(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F)
```

Arguments

`conc`	Vector of concentration values NOT in log units.
`resp`	Vector of corresponding responses.
`bidirectional`	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
`verbose`	If TRUE, gives optimization and hessian inversion details.
`nofit`	If nofit = T, returns formatted output filled with missing values.

Details

Zero background and increasing absolute response are assumed. Parameters are "a" (y scale), "b" (x scale), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitexp2(c(.1,1,10,100), c(0,.1,1,10))
```

`fitexp3` *Exponential 3 Model Fit*

Description

Function that fits to f(x) = a*(e^((x/b)^p)- 1) and returns generic model outputs.

Usage

```
fitexp3(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F,
  dmin = 0.3)
```

Arguments

`conc`	Vector of concentration values NOT in log units.
`resp`	Vector of corresponding responses.
`bidirectional`	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
`verbose`	If TRUE, gives optimization and hessian inversion details.
`nofit`	If nofit = T, returns formatted output filled with missing values.
`dmin`	Minimum allowed value of p.

Details

Zero background and increasing absolute response are assumed. Parameters are "a" (y scale), "b" (x scale), "p" (power), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitexp3(c(.03,.1,.3,1,3,10,30,100), c(0,0,.1, .2, .4, 1, 4, 50))
```

fitexp4 *Exponential 4 Model Fit*

Description

Function that fits to f(x) = tp*(1-2^(-x/ga)) and returns generic model outputs.

Usage

```
fitexp4(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
bidirectional	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
verbose	If TRUE, gives optimization and hessian inversion details.
nofit	If nofit = T, returns formatted output filled with missing values.

Details

Zero background and increasing absolute response are assumed. Parameters are "tp" (top), "ga" (AC50), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitexp4(c(.03,.1,.3,1,3,10,30,100), c(0,0,.1, .2, .5, 1, 1.5, 2))
```

fitexp5 *Exponential 5 Model Fit*

Description

Function that fits to f(x) = tp*(1-2^(-(x/ga)^p)) and returns generic model outputs.

Usage

```
fitexp5(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F,
  dmin = 0.3)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
bidirectional	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
verbose	If TRUE, gives optimization and hessian inversion details.
nofit	If nofit = T, returns formatted output filled with missing values.
dmin	Minimum allowed value of p.

Details

Zero background and increasing absolute response are assumed. Parameters are "tp" (top), "ga" (AC50), "p" (power), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitexp5(c(.03,.1,.3,1,3,10,30,100), c(0,0,.1, .2, .5, 1, 1.5, 2))
```

---

fitgnls *Gain-Loss Model Fit*

Description

Function that fits to f(x) = tp/[(1 + (ga/x)^p)(1 + (x/la)^q)] and returns generic model outputs.

Usage

```
fitgnls(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F,
  minwidth = 1.5)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
bidirectional	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
verbose	If TRUE, gives optimization and hessian inversion details.
nofit	If nofit = T, returns formatted output filled with missing values.
minwidth	Minimum allowed distance between gain ac50 and loss ac50 (in log10 units).

Details

Concentrations are converted internally to log10 units and optimized with f(x) = tp/[(1 + 10^(p*(ga-x)))(1 + 10^(q*(x-la)))], then ga, la, ga_sd, and la_sd are converted back to regular units before returning. Zero background and increasing initial absolute response are assumed. Parameters are "tp" (top), "ga" (gain AC50), "p" (gain power), "la" (loss AC50),"q" (loss power) and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitgnls(c(.03,.1,.3,1,3,10,30,100), c(0,.3,1, 2, 2.1, 1.5, .8, .2))
```

---

fithill *Hill Model Fit*

Description

Function that fits to f(x) = tp/[(1 + (ga/x)^p)] and returns generic model outputs.

Usage

```
fithill(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
bidirectional	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
verbose	If TRUE, gives optimization and hessian inversion details.
nofit	If nofit = T, returns formatted output filled with missing values.

Details

Concentrations are converted internally to log10 units and optimized with f(x) = tp/(1 + 10^(p*(ga-x))), then ga and ga_sd are converted back to regular units before returning. Zero background and increasing initial absolute response are assumed. Parameters are "tp" (top), "ga" (gain AC50), "p" (gain power), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fithill(c(.03,.1,.3,1,3,10,30,100), c(0,0,.1, .2, .5, 1, 1.5, 2))
```

`fitpoly1` *Polynomial 1 (Linear) Model Fit*

Description

Function that fits to f(x) = a*x and returns generic model outputs.

Usage

```
fitpoly1(conc, resp, bidirectional = TRUE, verbose = FALSE,
  nofit = F)
```

Arguments

`conc`	Vector of concentration values NOT in log units.
`resp`	Vector of corresponding responses.
`bidirectional`	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
`verbose`	If TRUE, gives optimization and hessian inversion details.
`nofit`	If nofit = T, returns formatted output filled with missing values.

Details

Zero background and increasing absolute response are assumed. Parameters are "a" (y scale) and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitpoly1(c(.03,.1,.3,1,3,10,30,100), c(0,.01,.1, .1, .2, .5, 2, 5))
```

fitpoly2 *Polynomial 2 (Quadratic) Model Fit*

Description

Function that fits to f(x) = a*(x/b + x^2/b^2) and returns generic model outputs.

Usage

```
fitpoly2(conc, resp, bidirectional = TRUE, verbose = FALSE,
  nofit = F)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
bidirectional	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
verbose	If TRUE, gives optimization and hessian inversion details.
nofit	If nofit = T, returns formatted output filled with missing values.

Details

Zero background and monotonically increasing absolute response are assumed. Parameters are "a" (y scale), "b" (x scale), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitpoly2(c(.03,.1,.3,1,3,10,30,100), c(0,.01,.1, .1, .2, .5, 2, 8))
```

fitpow *Power Model Fit*

Description

Function that fits tof(x) = a*x^p and returns generic model outputs.

Usage

```
fitpow(conc, resp, bidirectional = TRUE, verbose = FALSE, nofit = F,
  nmin = 0.3)
```

Arguments

conc	Vector of concentration values NOT in log units.
resp	Vector of corresponding responses.
bidirectional	If TRUE, model can be positive or negative; if FALSE, it will be positive only.
verbose	If TRUE, gives optimization and hessian inversion details.
nofit	If nofit = T, returns formatted output filled with missing values.
nmin	Minimum allowed value of p.

Details

Zero background and monotonically increasing absolute response are assumed. Parameters are "a" (y scale), "p" (power), and error term "er". success = 1 for a successful fit, 0 if optimization failed, and NA if nofit = T. cov = 1 for a successful hessian inversion, 0 if it fails, and NA if nofit = T. aic, rme, modl, parameters, and parameter sds are set to NA in case of nofit or failure.

Value

Named list containing: success, aic (Aikaike Information Criteria), cov (success of covariance calculation), rme (root mean square error), modl (vector of model values at given concentrations), parameters values, parameter sd (standard deviation) estimates, pars (vector of parameter names), sds (vector of parameter sd names).

Examples

```
fitpow(c(.03,.1,.3,1,3,10,30,100), c(0,.01,.1, .1, .2, .5, 2, 8))
```

---

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 pathway 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 pathway cutoffs for a given null dataset.

Usage

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

Arguments

pathset	Name of pathway set used to score null data.
nullset	Name of null data set.
method	Name of pathway scoring method used on nullset.
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 pathway 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 pathway 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: pathway, cutoff, bmed (median of all samples for that pathway), pvalue (pvalue corresponding to each cutoff), numsd (number of sds corresponding to each cutoff).

gnls *Gain-Loss Model*

Description

Gain-Loss Model

Usage

```
gnls(ps, x)
```

Arguments

ps	Vector of parameters: tp,ga,p,la,q,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

gnlsderivobj *GNLS Derivative Objective Function*

Description

Derivative of the gnls function set to zero for top location solver.

Usage

```
gnlsderivobj(x, tp, ga, p, la, q)
```

Arguments

x	Concentration.
tp	Top.
ga	Gain AC50.
p	Gain power.
la	Loss AC50.
q	Loss power.

Value

Value of gnls derivative at x.

hillfn *Hill Model*

Description

Hill Model

Usage

```
hillfn(ps, x)
```

Arguments

ps	Vector of parameters: tp,ga,p,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

hitcont *Continuous Hitcalls*

Description

Wrapper that computes continuous hitcalls for a provided PATHWAY_CR dataframe.

Usage

```
hitcont(indf, xs = NULL, ys = NULL, newcutoff, mc.cores = 1)
```

Arguments

indf	Dataframe similar to PATHWAY_CR. Must contain "conc" and "resp" columns if xs and ys are not provided. Must contain "top", "ac50", "er", "fit_method", "caikwt", and "mll" columns as well as columns for each model parameter.
xs	List of concentration vectors that can be provided for speed.
ys	List of response vectors that can be provided for speed.
newcutoff	Vector of new cutoff values to use. Length should be equal to rows in indf.
mc.cores	Number of cores to use for large dataframes.

Details

indf parameter columns should be NA when not required by fit method. "conc" and "resp" entries should be a single string with values separated by |. Details on indf columns can be found in pathwayConcRespCore_pval.

Value

Vector of hitcalls between 0 and 1 with length equal to indf row number.

`hitcontinner` *Continuous Hitcalls Inner*

Description

Calculates continuous hitcall using 3 statistical metrics.

Usage

```
hitcontinner(conc, resp, top, cutoff, er, ps, fit_method, caikwt, mll)
```

Arguments

`conc`	Vector of concentrations.
`resp`	Vector of responses.
`top`	Model top.
`cutoff`	Desired cutoff.
`er`	Model error parameter.
`ps`	Vector of used model parameters in order: a, tp, b, ga, p, la, q, er.
`fit_method`	Name of winning fit method (should never be constant).
`caikwt`	Aikaike weight of constant model relative to winning model.
`mll`	Maximum log-likelihood of winning model.

Details

This function is called either directly from pathwayConcRespCore_pval or via hitcont using PATHWAY_CR. Details of how to compute function input are in pathwayConcRespCore_pval.

Value

Continuous hitcall between 0 and 1.

Examples

```
conc = c(.03,.1,.3,1,3,10,30,100)
resp = c(0,.1,0,.2,.6,.9,1.1,1)
top = 1.023239
er = -3.295307
ps = c(1.033239, 2.453014, 1.592714, er = -3.295307) #tp,ga,p,er
fit_method = "hill"
caikwt = 1.446966e-08
mll = 12.71495
hitcontinner(conc,resp,top,cutoff = 0.8, er,ps,fit_method, caikwt, mll)
hitcontinner(conc,resp,top,cutoff = 1, er,ps,fit_method, caikwt, mll)
hitcontinner(conc,resp,top,cutoff = 1.2, er,ps,fit_method, caikwt, mll)
```

hitlogic *Hit Logic (Discrete)*

Description

Wrapper that computes discrete hitcalls for a provided PATHWAY_CR dataframe.

Usage

```
hitlogic(indf, newbmad = NULL, xs = NULL, ys = NULL,
  newcutoff = NULL)
```

Arguments

indf	Dataframe similar to PATHWAY_CR. Must contain "conc" and "resp" columns if xs and ys are not provided. Must contain "cutoff" and "bmad_factor" columns if newbmad is not NULL. Must contain "top" and "ac50" columns. "conc" and "resp" entries should be a single string with values separated by |.
newbmad	(Deprecated) New number of bmads to use for the cutoff.
xs	List of concentration vectors that can be provided for speed.
ys	List of response vectors that can be provided for speed.
newcutoff	Vector of new cutoff values to use. Length should be equal to rows in indf.

Value

Vector of hitcalls with length equal to number of rows in indf.

Examples

```
conc = rep(".03|.1|.3|1|3|10|30|100",2)
resp = rep("0|0|.1|.1|.5|.5|1|1",2)
indf = data.frame(top = c(1,1), ac50 = c(3,4), conc = conc, resp = resp,
  stringsAsFactors = FALSE)
hitlogic(indf, newcutoff = c(.8,1.2))
```

hitloginner *Hit Logic Inner (Discrete)*

Description

Contains hit logic, called directly during CR fitting or later through "hitlogic".

Usage

```
hitloginner(conc = NULL, resp, top, cutoff, ac50 = NULL)
```

Arguments

conc	Vector of concentrations (No longer necessary).
resp	Vector of responses.
top	Model top.
cutoff	Desired cutoff.
ac50	Model AC50 (No longer necessary).

Details

The purpose of this function is to keep the actual hit rules in one location so it can be called during CR fitting, and then again after the fact for a variety of cutoffs. Curves fit with constant winning should have top = NA, generating a miss.

Value

Outputs 1 for hit, 0 for miss.

Examples

```
hitloginner(resp = 1:8, top = 7, cutoff = 5) #hit
hitloginner(resp = 1:8, top = 7, cutoff = 7.5) #miss: top too low
hitloginner(resp = 1:8, top = 9, cutoff = 8.5) #miss: no response> cutoff
hitloginner(resp = 1:8, top = NA, cutoff = 5) #miss: no top (constant)
```

---

httrFit *HTTr Fit*

Description

Concentration response curve fitting for HTTr.

Usage

```
httrFit(conc, resp, cutoff, force.fit = FALSE, bidirectional = TRUE,
  verbose = FALSE, do.plot = F, fitmodels = c("cnst", "hill", "gnls",
  "poly1", "poly2", "pow", "exp2", "exp3", "exp4", "exp5"), ...)
```

Arguments

conc	Vector of concentrations (NOT in log units).
resp	Vector of responses.
cutoff	Desired cutoff. If no absolute responses > cutoff and force.fit = F, will only fit constant model.
force.fit	If force.fit = T, will fit all models regardless of cutoff.
bidirectional	If bidirectional = F, will only give positive fits.
verbose	If verbose = T, will print optimization details and aics.
do.plot	If do.plot = T, will generate a plot comparing model curves.
fitmodels	Vector of model names to try fitting. Missing models still return a skeleton output filled with NAs.
...	Other fitting parameters (deprecated).

Details

All models are monotonic and equal to 0 at 0 concentration (zero background). To add more models in the future, write a fit____ function, and add the model name to the fitmodels and modelnames vectors.

Value

List of 11 elements. First 10 elements are the output generated by each fit function with their given model names. Last element is "modelnames": a vector of model names so other functions can easily cycle through the output.

Examples

```
conc = c(.03,.1,.3,1,3,10,30,100)
resp = c(0,.1,0,.2,.6,.9,1.1,1)
output = httrFit(conc,resp, .8, fitmodels = c("cnst", "hill"),verbose = TRUE,
  do.plot = TRUE)
```

---

loggnls *Log Gain-Loss Model*

Description

Log Gain-Loss Model

Usage

```
loggnls(ps, x)
```

Arguments

ps	Vector of parameters: tp,ga,p,la,q,er
x	Vector of concentrations (log10 units)

Value

Vector of model responses

---

loghill *Log Hill Model*

Description

Log Hill Model

Usage

```
loghill(ps, x)
```

Arguments

ps	Vector of parameters: tp,ga,p,er
x	Vector of concentrations (log10 units)

Value

Vector of model responses

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 pathway definitions. Each element is a vector of gene names. Each element name is a pathway name.E quivalent to gset.idx.list in gsva.
min.sz	Minimum pathway size (deprecated).
max.sz	Maximum pathway 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 pathway sizes are undercounted here due to missing values, they are assessed more accurately in pathwayScoreCoreMYGSEA and limits are enforced after scoring.

Value

Outputs pathway by sample matrix of pathway scores.

Examples

```
geneSets = list(pathway1 = c("ABC", "DEF"), pathway2 = 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)
```

nestselect *Nest Select*

Description

Chooses between nested models.

Usage

```
nestselect(aics, mod1, mod2, dfdiff, pval = 0.05)
```

Arguments

aics	Named vector of model aics (can include extra models).
mod1	Name of model 1, the model with fewer degrees of freedom.
mod2	Name of model 2, the model with more degrees of freedom.
dfdiff	Absolute difference in number of degrees of freedom (i.e. the difference in parameters).
pval	P-value for nested model test.

Value

Named aic vector with losing model removed.

Examples

```
aics = c(-5,-6,-3)
names(aics) = c("poly1", "poly2", "hill")
nestselect(aics, "poly1", "poly2", 1)

aics = c(-5,-7,-3)
names(aics) = c("poly1", "poly2", "hill")
nestselect(aics, "poly1", "poly2", 1)
```

pathwayAccumNullPlot *BMD Accumulation Plot With Nulls*

Description

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

Usage

```
pathwayAccumNullPlot(pathset = "bhrr", dataset = "arer",
  method = "fc", nullset = "arer_RAND125", newpval = 0.05,
  oldpval = 0.05, to.file = T, usecont = T, nametag = NULL,
  mc.cores = 1)
```

Arguments

pathset	Name of pathset.
dataset	Name of dataset.
method	Name of pathway 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 pathwayConcResp 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.

---

pathwayBuilder_bhrr *bhrr Pathway Builder*

Description

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

Usage

```
pathwayBuilder_bhrr()
```

Details

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

Value

No output.

pathwayConcRespCore_pval
Pathway Concentration Response Core

Description

Core of concentration response curve fitting for pvalue based cutoff. This function calls httrFit to get curve fits, chooses the winning model, extracts the top and ac50, computes the hitcall, and calculates bmd/bmdl/bmdu among other statistics. Nested model selection is used to choose between poly1/poly2, then the model with the lowest AIC (or AICc) is declared the winner. Continuous hitcalls requires httrFit to be run with force.fit = T and "cnst" never to be chosen as the winner.

Usage

```
pathwayConcRespCore_pval(row, fitmodels = c("cnst", "hill", "gnls",
  "poly1", "poly2", "pow", "exp2", "exp3", "exp4", "exp5"), conthits = F,
  aicc = F)
```

Arguments

row	A named list that must include: • conc - list of concentrations (not in log units) • resp - list of corresponding responses • bmed - median of noise estimate. • cutoff - noise cutoff • onesd - 1 standard deviation of the noise (for bmd calculation) Other elements (usually identifiers, like casrn) of row will be attached to the final output.
fitmodels	Vector of model names to use.
conthits	conthits = T uses continuous hitcalls, otherwise they're discrete.
aicc	aicc = T uses corrected AIC to choose winning method; otherwise regular AIC.

Value

One row dataframe containing all CR output and statistics and any identifiers from row.

Examples

```
conc = list(.03,.1,.3,1,3,10,30,100)
resp = list(0,.2,.1,.4,.7,.9,.6, 1.2)
row = list(conc = conc, resp = resp, bmed = 0, cutoff = 1, onesd = .5)
pathwayConcRespCore_pval(row, conthits = TRUE)
pathwayConcRespCore_pval(row, aicc = TRUE)
```

`pathwayConcRespPlot` *Pathway Concentration Response Plot*

Description

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

Usage

```
pathwayConcRespPlot(row)
```

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 • bmd10, bmdl, bmdu - bmd, bmd lower bound, and bmd upper bound • ac50, acc - curve value at 50 • top - curve top • time, pathway, pathway_class, pathway_size - other identifiers Other elements are ignored.

Details

row is one row of PATHWAY_CR, the pathwayConcResp output.

Value

No output.

pathwayConcResp_pval *Pathway Concentration Response (P-value)*

Description

Performs pathway concentration response using p-value based cutoffs.

Usage

```
pathwayConcResp_pval(pathset = "bhrr", dataset = "arer",
  method = "fc", nullset = "arer_RAND1000", mc.cores = 1,
  to.file = T, do.plot = F, pval = 0.05, nametag = NULL,
  conthits = T, aicc = F, minpathsize = 10, fitmodels = c("cnst",
  "hill", "gnls", "poly1", "poly2", "pow", "exp2", "exp3", "exp4", "exp5"))
```

Arguments

pathset	Name of the pathway set.
dataset	Name of the data set.
method	Name of the pathway scoring method.
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/pathway 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.
minpathsize	Minimum allowed pathway size. Sample/pathway 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 pathwayScore and the given pathset and method. This function prepares pathwayScore output for CR processing, calls pathwayConcRespCore_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 pathway CR output.

pathwayDistributionPlot
Pathway Distribution Plot

Description

Plots null and actual pdfs for given pathway and cutoffs.

Usage

```
pathwayDistributionPlot(pathset = "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 pathway",
  "HALLMARK_ANDROGEN_RESPONSE"), to.file = T, seed = 12345)
```

Arguments

pathset	Name of pathway set.
dataset	Name of data set.
method	Name of pathway 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 pathway names to plot.
to.file	If to.file = T, write plot to disk.
seed	Randomization seed to use to choose additional sample pathways.

Details

This function requires that a PATHSCOREMAT file has already been generated for the given pathset/dataset/method using pathwayScore. There should also be PATHSCOREMAT 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.

pathwayScore *Pathway Score*

Description

Computes and saves pathway scores.

Usage

```
pathwayScore(FCMAT2, CHEM_DICT, pathset = "bhrr",
  dataset = "PlateEffect", method = "10chems", mc.cores = 1,
  minpathsize = 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, dsstox_substance_id.
pathset	Name of pathway set.
dataset	Name of data set.
method	Name of desired scoring method.
mc.cores	Number of cores to use.
minpathsize	Minimum allowed pathway size BEFORE accounting for missing values.

Details

pathwayScore 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. Pathway size is counted according to number of genes in the pathway 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 minpathsize is enforced when running pathwayConcResp_pval.

Value

No output.

pathwayScoreCoreFC *Pathway Score Core - FC*

Description

Computes fold change pathway scores.

Usage

```
pathwayScoreCoreFC(fcdata, pathset, dataset, chem_dict, pathway_data)
```

Arguments

fcdata	Sample by gene matrix of log2(fold change)'s. Rownames are sample keys and colnames are genes.
pathset	Name of pathway 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, dsstox_substance_id.
pathway_data	Named ist of gene name vectors. Each element is one pathway, defined by the genes it contains.

Details

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

Value

Dataframe with one row per chemical/conc/pathway combination. Columns are: sample_id, dsstox_substance_id, casrn, name, time, conc, pathset, pathway, size (pathway size accounting for missing values), mean_fc_scaled_in, mean_fc_scaled_out, pathway_score.

pathwayScoreCoreGSVA *Pathway Score Core - GSVA*

Description

Computes GSVA pathway scores.

Usage

```
pathwayScoreCoreGSVA(sk.list, pathset = "FILTERED", dataset, fcmat,
  chem_dict, pathway_data, mc.cores = 1)
```

Arguments

sk.list	Sample keys to use; should correspond to fcmat rownames.
pathset	Name of pathway 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, dsstox_substance_id.
pathway_data	Named ist of gene name vectors. Each element is one pathway, 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. pathscoremat output is saved directly to disk.

Value

No output.

---

pathwayScoreCoreMYGSEA

Pathway Score Core - MYGSEA

Description

Computes pathway scores for mygsea.

Usage

```
pathwayScoreCoreMYGSEA(sk.list, method = "mygsea", pathset = "bhrr",
  dataset, fcmat, chem_dict, pathway_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"
pathset	Name of pathway 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, dsstox_substance_id.
pathway_data	Named ist of gene name vectors. Each element is one pathway, 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 pathscoremat is written to disk.

Value

No output.

---

pathway_data *Endocrine Pathway Data*

Description

Pathway data example for 44 ER/AR related pathways

Usage

```
pathway_data
```

Format

A named list containing 44 elements. Each elements corresponds to a pathway and consists of a vector of gene names.

---

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 pathwayConcResp plotting function.

Usage

```
plotouter(proper_name, PATHWAY_CR, foldname)
```

Arguments

`proper_name`	Chemical name to be used in file name.
`PATHWAY_CR`	Dataframe output of pathwayConcResp_pval.
`foldname`	Folder name for output file.

Details

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

Value

No output.

`poly1` *Polynomial 1 Model*

Description

Polynomial 1 Model

Usage

```
poly1(ps, x)
```

Arguments

`ps`	Vector of parameters: a,er
`x`	Vector of concentrations (regular units)

Value

Vector of model responses

---

poly2 *Polynomial 2 Model*

Description

Polynomial 2 Model

Usage

```
poly2(ps, x)
```

Arguments

ps	Vector of parameters: a,b,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

---

pow *Power Model*

Description

Power Model

Usage

```
pow(ps, x)
```

Arguments

ps	Vector of parameters: a,p,er
x	Vector of concentrations (regular units)

Value

Vector of model responses

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 = "Phase1_6fixed",
  nchem = 125, 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 pathways 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 pathwayConcResp.
nametag	Additional file identifier added during pathwayConcResp.
conthits	Set conthits = T when using continuous hits.
pathclass	Some pre-defined sets of pathways to plot and run statistics on. "ER" is a group of ER pathways, "AR" is a group of AR pathways, and "DUT" is just the DUTERTRE_ESTRADIOL_RESPONSE_6HR_UP pathway.
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 pathways). 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' pathways.

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 pathway scoring method used.
pathset	Name of pathway 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))
```

---

runAllPathwayCR_pval *Run All Pathway Concentration Response (P-Value)*

Description

Driver for pathway scoring and concentration response (CR).

Usage

```
runAllPathwayCR_pval(basedir = "input/fcdata/", dataset = "arer",
  pathset = "bhrr", method = "fc", minpathsize = 10, conthits = T,
  nullset = "arer_RAND125", do.plot = T, pval = 0.05,
  mc.cores = c(39, 39), fitmodels = c("cnst", "hill", "gnls", "poly1",
  "poly2", "pow", "exp2", "exp3", "exp4", "exp5"))
```

Arguments

basedir	Folder that stores FCMAT2 and CHEM_DICT files.
dataset	Name of data set.
pathset	Name of pathway set.
method	Name of pathway scoring method.
minpathsize	Minimum pathway 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/pathway combination.
pval	P-value to use for noise estimation.
mc.cores	Vector with two values: number of cores to use for pathway 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 pathway scores to have already been computed for a nullset. randomdata() can generate a nullset, and this function can compute pathway scores for it by setting dataset = nullset and nullset = NULL. Pathway scores are written to disk in output/pathway_score_summary/. CR results are written to disk in output/pathway/conc_resp_summary/.

Value

No output.

`runAllRepChemCR` *Run All Replicate Chemical Concentration Response*

Description

Runs pathway 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 pathway set.
`method`	Name of pathway scoring method.
`minpathsize`	Minimum pathway size.
`do.plot`	do.plot = T generates plots for every chemical/pathway/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 pathway 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 pathway scores, generates a null dataset, runs pathway 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.

smoothecdf *Smooth ECDF*

Description

Converts a data frame containing bmd10, 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 bmd10, bmdu, bmdl, and hitcall columns.
verbose	verbose = F suppresses both bounds NA warning.
bmdrange	Maximum expected BMD range. The farthest value from the bmd10 is used to compute standard deviation of gaussian when both bounds are missing.

Details

Models each bmd as a gaussian with mean bmd10 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(bmd10 = 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")
```

---

tcplObj *Concentration Response Objective Function*

Description

Log-likelihood to be maximized during CR fitting.

Usage

```
tcplObj(p, conc, resp, fname, errfun = "dt4", err = NULL)
```

Arguments

p	Vector of parameters, must be in order: a, tp, b, ga, p, la, q, er. Does not require names.
conc	Vector of concentrations in log10 units for loghill/loggnls, in regular units otherwise.
resp	Vector of corresponding responses.
fname	Name of model function.
errfun	Which error distribution to assume for each point. "dt4" is the original 4 degrees of freedom t-distribution. "dnorm" is the normal distribution.
err	An optional estimation of error for the given fit.

Details

This function is a generalized version of the log-likelihood estimation functions used in the ToxCast Pipeline (TCPL). Hill model uses fname "loghill" and gnls uses fname "loggnls". Other model functions have the same fname as their model name; i.e. exp2 uses "exp2", etc. errfun = "dnorm" may be better suited to gsva pathway scores than "dt4". Setting err could be used to fix error based on the null data noise distribution instead of fitting the error when maximizing log-likelihood.

Value

Log-likelihood.

Examples

```
conc = c(.03,.1 , .3 , 1  , 3 , 10 , 30  , 100)
resp = c( 0 , 0 , .1 ,.2 , .5 , 1  , 1.5 , 2  )
p = c(tp = 2, ga = 3, p = 4, er = .5)
tcplObj(p,conc,resp,"exp5")

lconc = log10(conc)
tcplObj(p,lconc,resp,"loghill")
```

---

toplikelihood *Top Likelihood*

Description

Probability of top being above cutoff.

Usage

```
toplikelihood(fname, cutoff, conc, resp, ps, top, mll)
```

Arguments

fname	Model function name (equal to model name except hill which uses "hillfn")
cutoff	Desired cutoff.
conc	Vector of concentrations.
resp	Vector of responses.
ps	Vector of parameters, must be in order: a, tp, b, ga, p, la, q, er
top	Model top.
mll	Winning model maximum log-likelihood.

Details

Should only be called by hitcontinner. Uses profile likelihood, similar to bmdbounds. Here, the y-scale type parameter is substituted in such a way that the top equals the cutoff. Then the log-likelihood is compared to the maximum log-likelihood using chisq function to retrieve probability.

Value

Probability of top being above cutoff.

Examples

```
fname = "hillfn"
conc = c(.03,.1,.3,1,3,10,30,100)
resp = c(0,.1,0,.2,.6,.9,1.1,1)
ps = c(1.033239, 2.453014, 1.592714, er = -3.295307)
top = 1.023239
mll = 12.71495
toplikelihood(fname, cutoff = .8, conc, resp, ps, top, mll)
toplikelihood(fname, cutoff = 1, conc, resp, ps, top, mll)
toplikelihood(fname, cutoff = 1.2, conc, resp, ps, top, mll)
```

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