

Package ‘wHC’

January 29, 2021

Title Goodness of fit Statistics Calculation based on weighted p-values

Version 0.0.0.9000

Description This package is used for the application of the series of goodness-of-fit tests on the given gene sets. It provides codes that is able to calculate the pvalues based on single fisher exact tests or the stratified CMH tests (Epstein, M. P., Allen, A. S., & Satten, G. A. ,2007).

Depends R (>= 3.2.2)

Imports Rcpp (>= 0.11.0), biomaRt, igraph

LinkingTo Rcpp

License GPL(>=3)

Encoding UTF-8

RoxygenNote 7.1.1

R topics documented:

cal_cdf	2
cmh_cal	2
find_rank	4
get_centrality	4
get_gene_expression	5
get_gene_length	6
get_genic_intolerance	7
gof_cal	7
hc_cal	9
match_prior_info	9
match_prior_info_centrality	10
pval_ss_cal	12
rawp_cal	13
sdminp	14
sdminp_fdr	14
strat_score_cal_glm	15
support_gtex_tissue	16
trans_w	16

Index	18
--------------	-----------

cal_cdf

This function calculates the weighted-pvalues.

Description

It calculates the CDF(Cumulative Distribution Function) of weighted pvalues from the original pvalues and their weights and assign new pvalues based on this CDF.

Usage

```
cal_cdf(pm, w = 1)
```

Arguments

pm	is a statistics matrix of P-values from nrow=n genes(independent tests),ncol=d. Pm are not encouraged to have only 1 rows, if that happend, warning message will produced and this function returns $\sqrt{1/p_m-1}$, rather than the double side statistics from linear regression. Thus, we have $S(t) \sim \text{Binomial}(n, p)$.
w	is the weight, if not specify, w=1, if specify w must have the same length as nrow Pm

Value

numeric vector with each elements is a Higher criticism values calculated from each colum of the Pm

References

Genovese, C. R., Roeder, K., & Wasserman, L. (2006). False Discovery Control with p-Value Weighting. *Biometrika*, 93(3), 509–524.

Examples

```
pval=matrix(runif(200,0,1),ncol=4,nrow=50)
w0=seq(0.5,1.5,length=50)
pwval=cal_cdf(pval,w=w0)
```

cmh_cal

cmh_cal is the function of the cmh test for single genes with stratas

Description

cmh_cal belongs to pval_ss_cal when there is only 1 strata, cmh_cal reduces to fisher exact test and ignore the exact_option

Usage

```
cmh_cal(
  dis,
  cur_g,
  strata_vector,
  k = 1,
  alter_option = "greater",
  exact_option = TRUE
)
```

Arguments

<code>dis</code>	is a n-length numeric vector of indicators of phenotypes from n samples.
<code>cur_g</code>	is a n-length numeric vector of indicators of mutations from n samples in a certain gene.
<code>strata_vector</code>	is a n-length numeric vector of categories from the k kinds of strata, it is converted from the result of <code>strat_score_cal_glm</code> , for example, when we have 5 categories, and when <code>strata = [[1,0,0,0,0];[0,1,0,0,0];[0,0,1,0,0];[0,0,0,0,0]]</code> , we have: <code>strata_vector = (2,3,4,0)</code> .
<code>k</code>	is the kinds of strata, in accordance with content of <code>strata_vector</code>
<code>alter_option</code>	decides the test for p-values calculation. <code>options=c("greater","less","two.sided")</code> : "greater"(default):apply one-side cmh test if mutation are enriched in cases. "less":apply one-side cmh test if mutation are enriched in controls. "two.sided": apply two-side cmh test
<code>exact_option</code>	A logical indicating whether the Mantel-Haenszel test(FALSE) or the exact conditional test (TRUE, default) is applied.If <code>k==1</code> , <code>exact_option</code> will be ignored.

Value

a p-value from cmh test.

References

William G. Cochran (December 1954). "Some Methods for Strengthening the Common chi-squared Tests". *Biometrics*. 10 (4): 417–451. doi:10.2307/3001616. JSTOR 3001616.

Nathan Mantel and William Haenszel (April 1959). "Statistical aspects of the analysis of data from retrospective studies of disease". *Journal of the National Cancer Institute*. 22 (4): 719–748. doi:10.1093/jnci/22.4.719. PMID 13655060.

Fisher, R. A. (1922). "On the interpretation of chi-squared from contingency tables, and the calculation of P". *Journal of the Royal Statistical Society*. 85 (1): 87–94. doi:10.2307/2340521. JSTOR 2340521.

See Also

[mantelhaen.test](#) which this function wraps

Examples

```
pheno=rbinom(100,1,0.5)
cur_pc=rbind(matrix(rnorm(500,0,1),ncol=10,nrow=50),matrix(rnorm(500,0.5,1),ncol=10,nrow=50))
strata=strat_score_cal_glm(pheno,cur_pc)
```

```
cur_geno=rbinom(100,1,0.1)
cur_pval=cmh_cal(pheno,cur_geno,strata, alter="greater",exact_option=TRUE)
```

find_rank	<i>This function calculates the ranks of a vector based on another vector</i>
-----------	---

Description

find_rank is used by sdminp_fdr to speed up the calculation

Usage

```
find_rank(target, ruler)
```

Arguments

target	a decreacing-sorted vector for measure by the ruler
ruler	has the same length as target, also a decreasing-sorted vector

Value

a numerica vector, with each element i shows the numbers of values in the target that is bigger than the ruler[i]

Examples

```
a=c(8,6,4,2)
b=c(7,4,3,1)
find_rank(a,b)
```

get_centrality	<i>get_centrality match the centrality prior information to the given gene set from collection of MSigDB</i>
----------------	--

Description

Different from match_prior_info, the match_prior_info_centrality calculates each interactions based on the true gene sets.

Usage

```
get_centrality(
  interact_m,
  w_option = "deg",
  direct_option = FALSE,
  mode_option = "all"
)
```

Arguments

interact_m	represents the genetic network. It is the adjacency matrix of a graph with each node represents a gene.
w_option	the kind of centralities.can be "deg" for degree,"closn" for closeness,"betn" for betweenness, "eigen" for eigenvector centrality, and "pagerank"
direct_option	if it is true, the network will be calculated as directed pathways, parameter especially for pagerank
mode_option	parameters for centrality calculation, "out" for out-degree, "in" for in-degree or "all" or "total" for the sum of the two. see igraph for more details.

References

- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Stanford InfoLab. Retrieved from <http://ilpubs.stanford.edu:8090/422>
- White, S., & Smyth, P. (2003). Algorithms for Estimating Relative Importance in Networks. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 266–275). New York, NY, USA: ACM. <https://doi.org/10.1145/956750.956782>

get_gene_expression *This function get gene expression data of specific tissues from GTEx*

Description

The data resources comes from tpm of genes counts of RNAseq data of GTEx (gtexportal.org/home/datasets). It based on GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_tpm.gct.gz (Retrive Nov (2013))

Usage

```
get_gene_expression(
  gene_label = "ensembl_gene_id",
  tissue = support_gtex_tissue(),
  comb = "none"
)
```

Arguments

gene_label	is the gene names of the returning vector or matrix.It can be "ensembl_gene_id" or "symbols"
tissue	a vector of filters the expression level of specific tissues. Use support_gtex_tissue() to see supported tissues.
comb	is the operation on combining selected categories of tissues. It can be "median","mean","max","min",and "none", which calculate the median, mean of genes or do nothing on them.

Value

the numeric vector or matrix(@param mode is "none") representing prior information for each single gene

References

Lonsdale, John, et al. "The genotype-tissue expression (GTEx) project." Nature genetics 45.6 (2013): 580.

Examples

```
prior_expression=get_gene_expression(gene_label="symbols",tissue=c("Liver","Lung"),comb="mean")
prior_expression=get_gene_expression()
```

get_gene_length	<i>This function estimates the transcripts length.</i>
-----------------	--

Description

The data resources comes from the ensembl genome browser (useast.ensembl.org/index.html).

Usage

```
get_gene_length(gene_label = "symbols", comb = "mean")
```

Arguments

gene_label	is the returning gene labels, it can be "ensembl_gene_id" or "symbols"
comb	is the estimation method based on multiple records of transcription lengths. It can be "median","mean","max","min", which calculate the median, mean,max and min of genes or do nothing on them.

Value

the numeric vector representing prior information for each single gene's length

References

Durinck, Steffen, et al. "Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt." Nature protocols 4.8 (2009): 1184.

See Also

[biomaRt](#)

Examples

```
prior_length=get_gene_length(gene_label="symbols",comb="median")
```

get_genic_intolerance *This function pull down the genic intolerance information*

Description

The data resources comes from the databased of genic intolerance. (genic-intolerance.org/data/RVIS_Unpublished_ExAC)
 The default gene symbol is "CCDS_r15" and " The genes are labeled with gene symbols.

Usage

```
get_genic_intolerance()
```

Value

the numeric vector representing prior information for each single gene, with genes name corresponding to that in the net, which can be used as input of function match_prior_info

References

Petrovski, S., Wang, Q., Heinzen, E. L., Allen, A. S., & Goldstein, D. B. (2013). Genic Intolerance to Functional Variation and the Interpretation of Personal Genomes. *PLOS Genetics*, 9(8), e1003709. <https://doi.org/10.1371/journal.pgen.1003709>

Examples

```
genic_intolerance=get_genic_intolerance()
```

gof_cal

This function calculates the goodness-of-fit tests

Description

For ordered p-values $p(1) < p(2) < \dots < p(n)$. Define $S_n(t) = \sum_{i=1}^n 1_{p(i) \leq t}$. Then define the statistic T as $T = (S_n(p(i)) - n \times p(i)) / \sqrt{n \times p(i) \times (1 - p(i))}$ for score statistic. or $T = (S_n(p(i)) - n \times p(i)) / \sqrt{S_n(p(i)) \times (n - S_n(p(i))) / n}$ for wald statistic. or $T = n \times p(i) \times \log(p(i) \times n / S_n(p(i))) + (1 - p(i)) \times \log((1 - p(i)) \times n / (n - S_n(p(i))))$ for log likelihood ratio

Usage

```
gof_cal(
  pm,
  t0ratio = 0.1,
  gof_method = NA,
  single_statistic = "score",
  accumulate_option = "max",
  filter = 0,
  weight_option = "none",
  weight = 1
)
```

Arguments

<code>pm</code>	is a statistics matrix of P-values or weighted pvalues, each row represents a gene (independent tests) and each column represents a dataset (e.g. a permutation or an observation). Pm are not encouraged to have only 1 rows, if that happend, warning message will produced.
<code>t0ratio</code>	is the ratio for the region $c(0, t0ratio)$ of pvalues for statistic calculation.
<code>gof_method</code>	is the option for the set based analysis methods, available option includes: "higher_criticism", "berk_jo" see our publication for more details.
<code>single_statistic</code>	is the option for single statistic, it can be $c("score", "likelihood_ratio", "wald")$, it would be ignored is <code>gof_method</code> is assigned
<code>accumulate_option</code>	is the option for combining method, it can be $c("max", "sum", "rsum", "topsum", "max2", "sum2", "rsum2", "topsum2")$, it would be ignored is <code>gof_method</code> is assigned, the "max2", "sum2", "rsum2", "topsum2" will be replaced into "max", "sum", "rsum", "topsum" if <code>gof_method</code> is assigned.
<code>filter</code>	is the threshold to exclude extremely small pvalues to avoid them driving all signals.default 0.
<code>weight_option</code>	defines the external prior information as weight. It can be "none", "in" and "out". "none" assigns no weight, "in" assigns weight towards each single pvalues and "out" assigns weight towards each statistic T.
<code>weight</code>	is a vector which provides weight towards each genes.

Details

The max method calculates the $T_{max} = \max T_{(i)}$ where $0 < i < t0ratio \times n$ The sum method calculates the $T_{sum} = \sum T_{(i)}$ where $0 < i < t0ratio \times n$ The rsum is calculated with $T_{rsum} = 1/n \times \sum T_{(i)}$. where $0 < i \leq (i_{max})$, where i_{max} is the index of the maximum of Ts. The topsum is calculated with $T_{topsum} = 1/n \max T_{(r)}$. where r belongs to the subset that Ts in the subsets are the top `t0ratio` proportion among all Ts.

Value

a numeric vector with each elements is a Higher criticism values calculated from each column of the Pm

References

Donoho, D., & Jin, J. (2004). Higher Criticism for Detecting Sparse Heterogeneous Mixtures. The Annals of Statistics, 32(3), 962–994.

Examples

```
a=matrix(runif(200,0,1),ncol=4,nrow=50)
gof_cal(a)
```

hc_cal	<i>This function calculates the higher criticism</i>
--------	--

Description

For ordered p-values $p(1) < p(2) < \dots < p(n)$. Define $S_n(t) = \sum_{i=1}^n 1_{p(i) \leq t}$. Then define the statistic T as $T = (S_n(p(i)) - n \times p(i)) / \sqrt{n \times p(i) \times (1 - p(i))}$. The higher criticism is calculated with $HC_{\max} = \max_i T_{(i)}$, where $0 < i \leq (t0ratio \times n)$

Usage

```
hc_cal(pm, t0ratio = 1, filter = 0)
```

Arguments

pm	is a statistics matrix of P-values or weighted pvalues, each row represents a gene (independent tests) and each column represents a dataset (e.g. a permutation or an observation). Pm are not encouraged to have only 1 rows, if that happend, warning message will produced.
t0ratio	is the ratio for the region $c(0, t0ratio)$ of pvalues for statistic calculation.
filter	is the threshold to exclude extremely small pvalues to avoid them driving all signals.default 0.

Value

a numeric vector with each elements is a Higher criticism values calculated from each colum of the Pm

References

Donoho, D., & Jin, J. (2004). Higher Criticism for Detecting Sparse Heterogeneous Mixtures. The Annals of Statistics, 32(3), 962–994.

Examples

```
pval=matrix(runif(200,0,1),ncol=4,nrow=50)
w0=seq(0.5,1.5,length=50)
pwval=cal_cdf(pval,w=w0)
hc_cal(pwval,t0ratio=0.4)
```

match_prior_info	<i>match_prior_info match the prior information to the given gene set from collection of MSigDB</i>
------------------	---

Description

match_prior_info match the prior information to the given gene set from collection of MSigDB

Usage

```
match_prior_info(net, prior_info, add_option = "none", report_option = TRUE)
```

Arguments

net	dataframe the gmt file from collections of Molecular signatures database (MSigDB), broad institute.
prior_info	the numeric vector representing prior information for each single gene, with genes name corresponding to that in the net.
add_option	defines the method of adding up missing values. It can be "none", "mean" or "median". No actions for adding up missing values if "none".
report_option	if TRUE, report current procedures of the path, which is the proportion of sets completed the matching steps.

Value

a dataframe with the same format as net, which is the gmt files

References

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., & Mesirov, J. P. (2011). Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, 27(12), 1739–1740. <https://doi.org/10.1093/bioinformatics/btr260>

Examples

```
net=net.h.all.v6.1.symbols
#net from MSigDB (software.broadinstitute.org/gsea/msigdb): "h.all.v6.1.symbols.gmt"
prior_gi=get_genic_intolerance()
prior_net=match_prior_info(net,prior_gi)
```

```
match_prior_info_centrality
```

match_prior_info_centrality matches the centrality prior information to the given gene set from collection of MSigDB

Description

Different from match_prior_info, the match_prior_info_centrality calculates each interactions based on the true gene sets.

Usage

```
match_prior_info_centrality(
  net,
  human_whole,
  add_option = "none",
  report_option = TRUE,
  w_option = "deg",
  direct_option = FALSE,
  mode_option = "all"
)
```

Arguments

net	dataframe the gmt file from collections of MSigDB, broad institute. each line represents a pathway. please read in with read.csv with header=FALSE and stringAsFactors = FALSE.
human_whole	the 2-column matrix with each line representing the connection from gene in column 1 to gene in column 2
add_option	defines the method of adding up missing values. It can be "none", "mean" or "median". No actions for adding up missing values if "none".
report_option	if TRUE, report current procedures of the path, which is the proportion of sets completed the matching steps.
w_option	the kind of centralities.
direct_option	if it is true, the network will be calculated as directed pathways, parameter especially for pagerank
mode_option	parameters for centrality calculation, "out" for out-degree, "in" for in-degree or "all" or "total" for the sum of the two.

Value

a dataframe with the same format as net, which is the gmt files

References

- Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., & Mesirov, J. P. (2011). Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, 27(12), 1739–1740. <https://doi.org/10.1093/bioinformatics/btr260>
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*. Retrieved from <http://ilpubs.stanford.edu:8090/422>
- White, S., & Smyth, P. (2003). Algorithms for Estimating Relative Importance in Networks. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 266–275). New York, NY, USA: ACM. <https://doi.org/10.1145/956750.956782>
- Chatr-aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., ... Tyers, M. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Research*, 45(Database issue), D369–D379. <https://doi.org/10.1093/nar/gkw1102>

See Also

[igraph](#) which this function wraps

Examples

```
net=net.h.all.v6.1.entrez
#from MSigDB (http://software.broadinstitute.org/gsea/msigdb): "h.all.v6.1.entrez.gmt"
human_whole=human_whole_biogird_3.4.147
#from BioGRID(https://thebiogrid.org/): "BIOGRID-ORGANISM-Homo_sapiens-3.4.147.tab2.txt"
human_whole=as.matrix(human_whole[,c(2,3,8,9,10,11)])
human_whole=unique(human_whole)
human_whole=as.matrix(human_whole[order(human_whole[,2]),])
human_whole=as.matrix(human_whole[order(human_whole[,1]),])
human_whole[,1]=as.numeric(human_whole[,1])
human_whole[,2]=as.numeric(human_whole[,2])
human_whole=human_whole[,1:2]
```

```
res=match_prior_info_centrality(net,human_whole,add_option="none",
report_option=TRUE,w_option="pagerank",direct_option=TRUE,mode_option="all")
```

pval_ss_cal	<i>pval_ss_cal calculates pvalues of each genes between disease and test genotype</i>
-------------	---

Description

It featuring for options of adjusting for the stratification-score indicators followed with cmh test, for observation and permutation. If no stratification, this code did simple fisher exact test instead.

Usage

```
pval_ss_cal(dis_ob, g, strata = NA, nperm = 0, alter = "greater", exact = TRUE)
```

Arguments

dis_ob	0-1 vector shows the subjects has disease (1) or not (0)
g	matrix with each row demonstrate the genes and each column demonstrates the subjects. ncol(g)=length(dis_ob)
strata	the output of function strat_score_cal, which are the n rows, k-1 columns matrix for samples stratification information. If ignored, only fisher exact test are applied.
nperm	the number of permutation we should perform. If nperm=0 (default), that means only calculate the observed/input situation. Otherwise, only calculated the permuted situation.
alter	only works if cmh_option is TRUE. The test for p-values calculation. options=c("greater","less","two.sided"): "greater"(default):apply one-side cmh test if mutation are enriched in cases."less":apply one-side cmh test if mutation are enriched in controls. "two.sided": apply two-side cmh test.
exact	only works if cmh_option is TRUE. A logical indicating whether the Mantel-Haenszel test(FALSE) or the exact conditional test(TRUE, default) is applied.

Value

a numeric vector of p-values

References

Epstein MP, Allen AS, Satten GA (2007) A simple and improved correction for population stratification in case-control studies. American Journal of Human Genetics 80: 921-930

See Also

[mantelhaen.test](#) which this function wraps

[cmh_cal](#) which this function wraps

Examples

```
pheno=rbinom(100,1,0.5)
cur_pc=rbind(matrix(rnorm(500,0,1),ncol=10,nrow=50),matrix(rnorm(500,0.5,1),ncol=10,nrow=50))
strata=strat_score_cal_glm(pheno,cur_pc)
geno=matrix(rbinom(2000,1,0.1),nrow=20,ncol=100)
pval=pval_ss_cal(pheno,geno,strata,nperm=0, alter="greater",exact_option=TRUE)
```

rawp_cal

This function calculates the empirical raw p-values.

Description

It implements the raw-pvalues based on permutation, defined by Ge et al, 2003 (box 1)

Usage

```
rawp_cal(res_ob, res_perm)
```

Arguments

res_ob	is a numeric vectors of statistics
res_perm	is a matrix with each colum is a permuated statistics with the same length as res_ob

Value

a numeric matrix with raw pvalues, defined by Ge et al, 2003 (box 1). NAs will be ignored.

References

Ge, Y., Dudoit, S., & Speed, T. P. (2003). Resampling-based multiple testing for microarray data analysis. *Test*, 12(1), 1–77. <https://doi.org/10.1007/BF02595811>

Examples

```
ob=rnorm(4,2,2)
perm=matrix(rnorm(20,2,2),ncol=5,nrow=4)
rawp_cal(ob,perm)
```

sdminp

This function calculates the Step Down minP method.

Description

It implement the improved step-down minP algorithm by Ge et al, 2003 (box 4)

Usage

```
sdminp(res_ob, res_perm)
```

Arguments

res_ob	is a numeric vectors of statistics
res_perm	is a matrix with each colum is a permuated statistics with the same length as res_ob

Value

a numeric vector adjusted pvalues

References

Ge, Y., Dudoit, S., & Speed, T. P. (2003). Resampling-based multiple testing for microarray data analysis. *Test*, 12(1), 1–77. <https://doi.org/10.1007/BF02595811>

Examples

```
ob=rnorm(10,2,2)
perm=matrix(rnorm(100,2,2),ncol=10,nrow=10)
sdminp(ob,perm)
```

sdminp_fdr

This function calculates the Step Down minP and report the results, This is downstream analysis for sdminp It implements the improved step-down minP algorithm based on FDR by Ge et al, 2003 (box 5) It requires another function find_rank

Description

This function calculates the Step Down minP and report the results, This is downstream analysis for sdminp It implements the improved step-down minP algorithm based on FDR by Ge et al, 2003 (box 5) It requires another function find_rank

Usage

```
sdminp_fdr(res_ob, res_perm, tao0 = 0.2)
```

Arguments

res_ob	is a numeric vectors of statistics
res_perm	is a matrix with each colum is a permuated statistics with the same length as res_ob
tao0	is a proportion threshold that more than tao0, there will expected to be no significant results.

Value

a numeric matrix with adjusted pvalues, the first column is the FDR adjusted pvalues, the second colum is the corresponding q-values

References

Ge, Y., Dudoit, S., & Speed, T. P. (2003). Resampling-based multiple testing for microarray data analysis. *Test*, 12(1), 1–77. <https://doi.org/10.1007/BF02595811>

Examples

```
ob=rnorm(10,2,2)
perm=matrix(rnorm(100,2,2),ncol=10,nrow=10)
sdminp_fdr(ob,perm)
```

strat_score_cal_glm	<i>strat_score_cal_glm calculates the population stratification</i>
---------------------	---

Description

strat_score_cal_glm calculates the population stratification

Usage

```
strat_score_cal_glm(dis, pc, nstrat_ss = 5)
```

Arguments

dis	vector of disease outcomes (1=case, 0=control) with Ntot subjects.
pc	first 10 principle components(default colum=10).
nstrat_ss	number of strata to be separated to, default 5

Value

a numeric matrix with nstrat_ss-1 columns, dis length rows. each elements are 0-1 indicators which shows if this subjects belongs to this strata. the last strata are subjects with all other nstrat_ss-1 columns be zero.

References

Epstein MP, Allen AS, Satten GA (2007) A simple and improved correction for population stratification in case-control studies. *American Journal of Human Genetics* 80: 921-930

See Also

[glm](#) which this function wraps

Examples

```
pheno=rbinom(100,1,0.5)
cur_pc=rbind(matrix(rnorm(500,0,1),ncol=10,nrow=50),matrix(rnorm(500,0.5,1),ncol=10,nrow=50))
strata=strat_score_cal_glm(pheno,cur_pc)
```

support_gtex_tissue	<i>support_gtex_tissue provides supported option for "tissue" in function get_gene_expression</i>
---------------------	---

Description

The data resources comes from tpm of genes counts of RNAseq data of "<https://gtexportal.org/home/datasets>".

Usage

```
support_gtex_tissue()
```

References

Lonsdale, John, et al. "The genotype-tissue expression (GTEx) project." Nature genetics 45.6 (2013): 580.

https://storage.googleapis.com/gtex_analysis_v7/rna_seq_data/GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gen (Retrive Nov (2013)).

See Also

[get_gene_expression](#)

Examples

```
support_gtex_tissue()
```

trans_w	<i>This function transfers prior information into the weights</i>
---------	---

Description

The weight is calculated with the equation: $w=1/(a \times \text{prior_info}+b \times \text{mean}(\text{prior_info}))$ and then scaled into $\text{mean}(w)=1$. Here we take $a=0.95$ and $b=0.05$.

Usage

```
trans_w(w)
```


Arguments

w a numeric vector which is the original statistic for weight, need to be non-negative

Value

a numeric vector of the transferred weight with around mean~1, min~0.05. Missing values in input are transferred into 1

Examples

```
a=c(NA,exp(rnorm(7,0,1)))  
trans_w(a)
```

Index

biomaRt, [6](#)

cal_cdf, [2](#)
cmh_cal, [2](#), [12](#)

find_rank, [4](#)

get_centrality, [4](#)
get_gene_expression, [5](#), [16](#)
get_gene_length, [6](#)
get_genic_intolerance, [7](#)
glm, [16](#)
gof_cal, [7](#)

hc_cal, [9](#)

igraph, [5](#), [11](#)

mantelhaen.test, [3](#), [12](#)
match_prior_info, [9](#)
match_prior_info_centrality, [10](#)

pval_ss_cal, [12](#)

rawp_cal, [13](#)

sdmnp, [14](#)
sdmnp_fdr, [14](#)
strat_score_cal_glm, [15](#)
support_gtex_tissue, [16](#)
support_gtex_tissue(), [5](#)

trans_w, [16](#)