Package 'pSI'

November 26, 2026

Type Package

Title Specificity Index Statistic

Version 1.2.0

Description Calculates the Specificity Index (pSI) statistic, which identifies genes enriched in specific cell types or tissues from a matrix of expression profiles, and provides post-processing tools: lists and counts of specific genes at standard pSI thresholds, overlap with candidate gene lists, and Fisher's exact tests for enrichment of a candidate gene list across all cell types and thresholds (Cell type-Specific Expression Analysis, CSEA). The methods are described in Dougherty et al. (2010) <doi:10.1093/nar/gkq130>, Xu et al. (2014) <doi:10.1523/JNEUROSCI.4488-13.2014> and Wells et al. (2015) <doi:10.1093/nar/gkv1262>. The precomputed pSI tables for mouse cell types and human brain regions used in those papers, formerly the 'pSI.data' package, are downloaded on first use with psi_data().

License GPL (>= 2)

URL https://github.com/DunderChief/pSI, https://dunderchief.github.io/pSI/

BugReports https://github.com/DunderChief/pSI/issues

Depends R (>= 4.0.0)

Imports graphics, grDevices, stats, tools, utils

Suggests curl, testthat (>= 3.0.0), withr

Encoding UTF-8

Language en-US

LazyData true

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Author Xiaoxiao Xu [aut],
Alan B. Wells [aut],
David R. O'Brien [aut, cre] (ORCID:

<https://orcid.org/0009-0007-0339-0517>),
Arye Nehorai [aut],
Joseph D. Dougherty [aut]

Maintainer David R. O'Brien <dunder.chief@gmail.com>

Repository CRAN

Date/Publication 2026-11-26 11:00:07 UTC

Contents

candidate.genes 2
candidate.overlap 3
dataset.s1 4
fisher.iteration 5
pSI.count 6
pSI.list 7
psi_data 8
sample.data 10
specificity.index 10

Index **13**

`candidate.genes` *Candidate Gene Lists*

Description

Five candidate gene lists, provided as examples for use with `fisher.iteration` and `candidate.overlap`.

Usage

```
candidate.genes
```

Format

A list of five character vectors of official gene symbols:

`AutDB` Hand-curated list of Autism Spectrum Disorder (ASD) candidate genes derived from human genetics studies, downloaded from AutDB (N = 328).

`protein.disrupting.rdnv` Genes affected by protein-disrupting rare de novo variants in ASD probands, collected from four exome studies published in 2012 (N = 122).

`silent.rdnv` Negative control: genes affected by silent rare de novo variants in unaffected siblings, from the same four studies (N = 122).

`hcrt.genes` Genes differentially regulated in narcoleptic mice with hypocretin (Hcrt) neuron ablation versus controls (N = 63).

`retinopathy.genes` Genes implicated in human congenital retinopathies, from the curated RetNet database (N = 120).

Source

AutDB: Basu SN, Kollu R, Banerjee-Basu S (2009). AutDB: a gene reference resource for autism research. *Nucleic Acids Research* 37:D832-D836. doi:10.1093/nar/gkn835

protein.disrupting.rdnv and silent.rdnv: Iossifov I, Ronemus M, Levy D, et al. (2012). De novo gene disruptions in children on the autistic spectrum. *Neuron* 74:285-299. doi:10.1016/j.neuron.2012.04.009. Neale BM, Kou Y, Liu L, et al. (2012). Patterns and rates of exonic de novo mutations in autism spectrum disorders. *Nature* 485:242-245. doi:10.1038/nature11011. Sanders SJ, Murtha MT, Gupta AR, et al. (2012). De novo mutations revealed by whole-exome sequencing are strongly associated with autism. *Nature* 485:237-241. doi:10.1038/nature10945. O'Roak BJ, Vives L, Girirajan S, et al. (2012). Sporadic autism exomes reveal a highly interconnected protein network of de novo mutations. *Nature* 485:246-250. doi:10.1038/nature10989.

hcrt.genes: Honda M, Eriksson KS, Zhang S, et al. (2009). IGFBP3 colocalizes with and regulates hypocretin (orexin). *PLoS ONE* 4:e4254. doi:10.1371/journal.pone.0004254

retinopathy.genes: Daiger SP. RetNet, the Retinal Information Network. https://web.sph.uth.edu/RetNet/

Examples

```
data(candidate.genes)
names(candidate.genes)
lengths(candidate.genes)
head(candidate.genes$AutDB)
```

---

candidate.overlap *Candidate Gene Overlap*

Description

candidate.overlap extracts the genes specific to each cell type/sample (as listed by pSI.list) that are also present in a candidate gene list, at each of the four standard pSI thresholds (0.05, 0.01, 0.001 and 0.0001).

Usage

```
candidate.overlap(pSIs, candidate.genes, write.csv = FALSE)
```

Arguments

pSIs	data frame output from specificity.index, with one column per cell type/sample and genes as rows.
candidate.genes	character vector of official gene symbols to test for overlap with the cell type-specific genes.
write.csv	logical. If TRUE, one CSV file per threshold (pSi_0.0001.csv, pSi_0.001.csv, pSi_0.01.csv, pSi_0.05.csv) is written to the current working directory. Default FALSE.

Details

Gene symbols are compared in upper case, so mouse and human symbols can be matched against each other. (Before pSI 1.2.0 only the `pSIs` row names were upper-cased, so lower-case candidate symbols silently never matched.)

Value

A named list of four data frames (`pSi_0.0001`, `pSi_0.001`, `pSi_0.01`, `pSi_0.05`), one per pSI threshold, with one column per cell type/sample listing the candidate genes that are specific to it at that threshold, padded with `NA`.

Author(s)

Xiaoxiao Xu, Alan B. Wells, David R. O'Brien, Arye Nehorai, Joseph D. Dougherty

See Also

`fisher.iteration` to test whether each overlap is larger than expected by chance.

Examples

```
##load sample pSI output
data(sample.data)
##load sample candidate gene lists
data(candidate.genes)
##Generates lists of overlapping genes
candidate.gene.overlap.AutDB <- candidate.overlap(pSIs=sample.data$pSI.output,
                                                  candidate.genes=candidate.genes$AutDB)
candidate.gene.overlap.AutDB$pSi_0.01
```

---

dataset.s1 — *Supplementary Data Set 1: Genes Differentially Expressed in Autistic Cortex*

Description

Genes differentially expressed (up- and down-regulated) in the cortex of autistic versus control subjects, from human transcriptomic data (Voineagu et al. 2011).

Usage

```
dataset.s1
```

Format

A list of two character vectors of gene symbols:

`up.regulated` up-regulated genes (N = 234)

`down.regulated` down-regulated genes (N = 229)

Details

Differential expression was assessed with SAM (Significance Analysis of Microarrays) at FDR < 0.05 and fold change > 1.3.

Source

Voineagu I, Wang X, Johnston P, Lowe JK, Tian Y, Horvath S, et al. (2011). Transcriptomic analysis of autistic brain reveals convergent molecular pathology. *Nature* 474:380-384. doi:10.1038/nature10110

Examples

```
data(dataset.s1)
lengths(dataset.s1)
```

`fisher.iteration` *Fisher's Exact Test Across All Cell Types and pSI Thresholds*

Description

`fisher.iteration` tests a candidate gene list for over-representation among the genes specific to each cell type/sample, at each of the four standard pSI thresholds (0.05, 0.01, 0.001 and 0.0001), using a one-sided Fisher's exact test.

Usage

```
fisher.iteration(
  pSIs,
  candidate.genes,
  background = "data.set",
  p.adjust = TRUE
)
```

Arguments

`pSIs`	data frame output from `specificity.index`, with one column per cell type/sample and genes as rows.
`candidate.genes`	character vector of official gene symbols to test for over-representation.
`background`	character string naming the background gene list for the test. The default `"data.set"` uses all genes in `pSIs`; use it when the candidate genes and the pSI data set come from the same species. `"human.mouse"` restricts the background to the genes in `pSIs` (human) that have a clear mouse homolog, for testing a mouse candidate list against a human pSI data set; `"mouse.human"` is the converse. These two options download the UCSC kgXref table for mm10 or hg19 (a file of several megabytes) from `https://hgdownload.soe.ucsc.edu/` and stop with an informative error if the download fails.

p.adjust logical. If TRUE (the default) p-values are adjusted for multiple testing across cell types with the Benjamini-Hochberg method (p.adjust, method "BH"), separately for each threshold; if FALSE, nominal p-values are returned.

Details

For every cell type/threshold combination the function asks: what is the probability that this many candidate genes would be specific to this cell type by chance? Each gene in the background is classified by two binary variables, specific to the cell type at this threshold or not, and in the candidate list or not, and the resulting 2 x 2 table is tested with fisher.test (alternative "greater"). Gene symbols are compared in upper case, and duplicated candidate genes are counted once.

Value

A data frame with one row per cell type/sample and four columns, one per pSI threshold (0.05, 0.01, 0.001, 0.0001). Column names end in "adjusted" or "nominal" depending on p.adjust.

Author(s)

Xiaoxiao Xu, Alan B. Wells, David R. O'Brien, Arye Nehorai, Joseph D. Dougherty

See Also

candidate.overlap for the overlapping genes themselves.

Examples

```
##load sample pSI output
data(sample.data)
##load sample candidate gene lists
data(candidate.genes)
##run Fisher's exact test for overrepresentation on pSI.out for the AutDB
##candidate gene list across all cell types/sample types & pSI thresholds
fisher.out.AutDB <- fisher.iteration(pSIs=sample.data$pSI.output,
                                     candidate.genes=candidate.genes$AutDB)
fisher.out.AutDB
```

---

pSI.count *Convert pSI output to gene counts*

Description

pSI.count counts the number of genes specific to each cell type/sample at each of the four standard pSI thresholds (0.05, 0.01, 0.001 and 0.0001).

Usage

```
pSI.count(pSIs, write.csv = FALSE)
```

Arguments

pSIs	data frame output from specificity.index, with one column per cell type/sample and genes as rows.
write.csv	logical. If TRUE, the table is also written to gene_count_1.csv in the current working directory. Default FALSE.

Value

A data frame with four rows, one per pSI threshold ("0.0001", "0.001", "0.01", "0.05"), and one column per cell type/sample, giving the number of genes whose pSI value falls below each threshold.

Author(s)

Xiaoxiao Xu, Alan B. Wells, David R. O'Brien, Arye Nehorai, Joseph D. Dougherty

See Also

pSI.list for the gene lists themselves.

Examples

```
##load sample pSI output
data(sample.data)
##Count the number of genes specific to each cell type/sample type across all pSI thresholds
pSI.out.count <- pSI.count(pSIs=sample.data$pSI.output)
pSI.out.count
```

---

pSI.list *Convert pSI output to gene lists*

Description

pSI.list extracts, for each cell type/sample and each of the four standard pSI thresholds (0.05, 0.01, 0.001 and 0.0001), the genes whose pSI values fall below the threshold, i.e. the genes specific to that cell type.

Usage

```
pSI.list(pSIs, write.csv = FALSE)
```

Arguments

pSIs	data frame output from specificity.index, with one column per cell type/sample and genes as rows.
write.csv	logical. If TRUE, one CSV file per threshold (pSi_0.05.csv, pSi_0.01.csv, pSi_0.001.csv, pSi_1e-04.csv) is written to the current working directory. The default is FALSE: since pSI 1.2.0 nothing is written unless you ask for it.

Details

Gene symbols are converted to upper case for consistency in downstream analysis. Within each data frame the columns hold different numbers of genes and are padded with NA to equal length.

Value

A named list of four data frames (pSi_0.0001, pSi_0.001, pSi_0.01, pSi_0.05), one per pSI threshold. Each has one column per cell type/sample, named <sample>_<threshold>, containing the symbols of the genes with pSI below that threshold, padded with NA.

Author(s)

Xiaoxiao Xu, Alan B. Wells, David R. O'Brien, Arye Nehorai, Joseph D. Dougherty

See Also

pSI.count for the counts alone, candidate.overlap to intersect these lists with a candidate gene list.

Examples

```
##load sample pSI output
data(sample.data)
##List the genes specific to each cell type/sample type across all pSI thresholds
pSI.out.list <- pSI.list(pSIs=sample.data$pSI.output)
names(pSI.out.list)
head(pSI.out.list$pSi_0.01)
```

---

psi_data *Download the companion pSI data sets (the former pSI.data package)*

Description

psi_data fetches the precomputed Specificity Index (pSI) tables that used to be distributed as the companion package pSI.data: mouse cell types profiled by bacTRAP, and human brain regions from BrainSpan. They are too large for CRAN (about 10 MB together), so they are published as a release on the pSI GitHub repository and downloaded on first use.

Usage

```
psi_data(dataset = c("mouse", "human"), cache = NULL, quiet = FALSE)
```

Arguments

dataset	"mouse" or "human".
cache	logical or NULL; see Details.
quiet	logical. If TRUE, no download messages are shown.

Details

The two data sets, unchanged from pSI.data 1.0 (2014):

`"mouse"` a list of three data frames of 16907 genes by 35 mouse cell types or regions, profiled by translating ribosome affinity purification (bacTRAP) on one microarray platform, as of October 2013: `psi.in` (averaged expression values), `psi.in.filter` (`NA` marks genes to exclude for a cell type) and `psi.out` (pSI values).

`"human"` a list of two aggregations of BrainSpan RNA-seq data for 16866 genes: `developmental.periods` (six brain regions across ten developmental periods, 60 columns) and `young.adulthood` (the same six regions for young adulthood only). Each holds `psi.in` and `psi.out`.

The `psi.out` tables are ready to use as the `pSIs` argument of `fisher.iteration`, `candidate.overlap`, `pSI.list` and `pSI.count`.

Every download is verified against the file's MD5 checksum. Where the file is kept depends on `cache`. With `cache = TRUE` it goes under the package's user cache directory, `tools::R_user_dir("pSI", "cache")`, so later R sessions reuse it. With `cache = FALSE` it goes under `tempdir()` and disappears with the session. The default, `NULL`, asks you once per session when R is interactive and otherwise behaves like `FALSE`. To skip the question in every session, set `options(pSI.data.cache = TRUE)` (or `FALSE`) in your `.Rprofile`. To delete the cached files, remove that directory: `unlink(tools::R_user_dir("pSI", "cache"), recursive = TRUE)`.

Value

The requested data set, a named list of data frames as described above. If the file is not already available and cannot be downloaded (for example when offline), `NULL` is returned invisibly with a message rather than an error.

Source

https://github.com/DunderChief/pSI/releases/tag/pSI.data-1.0. Mouse expression data from GEO series GSE13379 and GSE30626; human data from the BrainSpan Atlas of the Developing Human Brain, https://www.brainspan.org.

See Also

`sample.data` for the small built-in example data.

Examples

```
## about 4 MB, kept only for this session
mouse <- psi_data("mouse", cache = FALSE)
if (!is.null(mouse)) {
  names(mouse)
  dim(mouse$psi.out)
  ## how many genes are specific to each mouse cell type at each threshold
  pSI.count(mouse$psi.out)
  ## CSEA of a candidate list against these cell types (a few seconds):
  ## fisher.iteration(mouse$psi.out, candidate.genes$AutDB)
}
```

sample.data *Sample Input and Output pSI Data Sets*

Description

A sample expression matrix and the corresponding specificity index (pSI) output, used to illustrate the pSI package functions.

Usage

```
sample.data
```

Format

A list of two data frames, each with 16866 rows (genes, as row names) and 6 columns (brain regions):

pSI.input expression values

pSI.output pSI values, NA where the gene is not specific to that region

Details

pSI.input is the BrainSpan RNA-seq data for young adulthood, condensed into six major regional divisions (one column per region). pSI.output is the pSI table calculated from it with specificity.index.

Source

BrainSpan: Atlas of the Developing Human Brain, https://www.brainspan.org

Examples

```
data(sample.data)
str(sample.data, max.level = 1)
head(sample.data$pSI.input)
```

specificity.index *Specificity Index Statistic*

Description

specificity.index calculates the specificity index statistic (pSI) for every gene in every cell type/sample of an expression matrix. The pSI identifies genes enriched in specific cell populations across a large number of profiles, and correctly predicts in situ hybridization patterns for many cell types (Dougherty et al. 2010).

Usage

```
specificity.index(
  pSI.in,
  pSI.in.filter,
  bts = 50,
  p_max = 0.1,
  e_min = 0.3,
  hist = FALSE,
  SI = FALSE,
  hist_file = NULL
)
```

Arguments

pSI.in	data frame with expression values for genes in rows and samples or cell types in columns (replicate arrays already averaged, so one column per cell type).
pSI.in.filter	optional matched data frame (same genes and samples) with NA for any gene that should be excluded for a particular cell type.
bts	numeric. Number of resampled distributions to average for the permutation test.
p_max	numeric. Maximum p-value to calculate; genes with larger p-values are returned as NA.
e_min	numeric. Minimum expression value for a gene to be included. A value of 50 has been used for microarray studies and 0.3 for RNA-seq studies.
hist	logical. If TRUE, draw histograms of the sampled (null) and actual distributions of mean gene rank for each cell type on the current graphics device, one page per cell type. (Before pSI 1.2.0 these were written as PNG files into the working directory.)
SI	logical. If TRUE, return the specificity index (mean rank) itself instead of the pSI p-value.
hist_file	optional path to a PDF file. If supplied, the histograms are written there (one page per cell type) instead of to the current device, and hist is treated as TRUE.

Details

For cell type 1 and gene n, the specificity index is the mean, over the other $m-1$ cell types k, of the rank (among all genes) of the expression ratio of that gene between cell type 1 and cell type k:

$$SI_{n,1} = \frac{\sum_{k=2}^{m} rank(\frac{IP_{1,n}}{IP_{k,n}})}{m-1}$$

A low SI means the gene is highly enriched in cell type 1 relative to the others. The pSI is a p-value for the observed SI, obtained by comparing it with a null distribution of SI values computed from bts expression matrices resampled with replacement. To save time it is only calculated for genes whose p-value is below p_max; all other genes, and genes with expression below e_min or set to NA in pSI.in.filter, are returned as NA.

Because the null distribution is resampled, results vary slightly from run to run; call set.seed first for a reproducible result.

Value

A data frame with the same row and column names as `pSI.in`. Each cell holds the pSI value (a p-value) of that gene for that cell type/sample, or `NA` if the gene was excluded (expression below `e_min`, filtered by `pSI.in.filter`, or p-value above `p_max`). With `SI = TRUE` the specificity index is returned instead of the p-value.

Author(s)

Xiaoxiao Xu, Alan B. Wells, David R. O'Brien, Arye Nehorai, Joseph D. Dougherty

References

Dougherty JD, Schmidt EF, Nakajima M, Heintz N (2010). Analytical approaches to RNA profiling data for the identification of genes enriched in specific cells. *Nucleic Acids Research* 38(13):4218-4230. doi:10.1093/nar/gkq130

See Also

`pSI.list`, `pSI.count`, `candidate.overlap`, `fisher.iteration`

Examples

```
##load sample expression matrix
data(sample.data)
##calculate specificity index on expression matrix
##(Normally for RNAseq data, and e_min of 0.3, microarrays: e_min= 50)
set.seed(1)
pSI.output <- specificity.index(pSI.in=sample.data$pSI.input, e_min=20)
head(pSI.output[rowSums(!is.na(pSI.output)) > 0, ])

##histograms of the sampled and actual rank distributions, to a temporary PDF
pdf_file <- tempfile(fileext = ".pdf")
pSI.output <- specificity.index(pSI.in=sample.data$pSI.input, e_min=20,
                                hist_file=pdf_file)
unlink(pdf_file)
```

Index

∗ **datasets**
candidate.genes, 2
dataset.s1, 4
sample.data, 10

candidate.genes, 2
candidate.overlap, *2*, 3, *6*, *8, 9*, *12*

dataset.s1, 4

fisher.iteration, *2*, *4*, 5, *9*, *12*
fisher.test, *6*

p.adjust, *6*
pSI.count, 6, *8, 9*, *12*
pSI.list, *3*, *7*, 7, *9*, *12*
psi_data, 8

sample.data, *9*, 10
set.seed, *11*
specificity.index, *3*, *5*, *7*, *10*, 10