

Package: Protein2Tissue (via r-universe)

October 7, 2026

Title An R package for tissue expression analysis and gene set enrichment

Version 0.99.5

Description Protein2Tissue provides a comprehensive framework for analysing tissue-specific gene and protein expression patterns. The package allows users to map genes or proteins to their predominant tissue types, perform tissue-based enrichment analysis on gene sets, and explore expression profiles across a wide range of human tissues. Reference expression data is derived from publicly available databases the Human Protein Atlas (<https://www.proteinatlas.org>).

biocViews Proteomics, Transcriptomics, GeneSetEnrichment, Network

License file LICENSE

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.4.0)

Imports dplyr, tidyr, igraph, ggraph, ggplot2, ggpubr, clusterProfiler, AnnotationDbi, org.Hs.eg.db, methods, Exact

Suggests testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

URL <https://github.com/chryang/Protein2Tissue>

BugReports <https://github.com/chryang/Protein2Tissue/issues>

Collate 'AllClasses.R' 'AllGenerics.R' 'AllMethods.R' 'data.R' 'geneClassification.R' 'goEnriched.R' 'heatmapEnrich.R' 'internal_functions.R' 'networkPlot.R' 'plotProteinType.R' 'testByTissue.R' 'testPlot.R' 'tissueAnalysis.R' 'tissueAnalysis2.R' 'writeOutput.R'

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev zlib1g-dev

Repository <https://biocstaging.r-universe.dev>
Date/Publication 2026-10-06 16:25:24 UTC
RemoteUrl <https://github.com/BiocStaging/Protein2Tissue>
RemoteRef HEAD
RemoteSha 447083eb2537be05842e2732a2b9b4335d83d99b

Contents

ee_tb	2
Enrichment-class	3
geneClassification	4
getBackground	6
getInput	6
goEnriched	7
heatmapEnrich	8
networkPlot	9
plasma_data	10
plotProteinType	11
testByTissue	12
testPlot	14
tissue_map	15
tissueAnalysis	15
tissueAnalysis2	17
writeOutput	18
Index	20

ee_tb	<i>Human Protein Atlas tissue expression database</i>
-------	---

Description

A data frame containing tissue-specific gene expression data derived from the Human Protein Atlas (HPA).

Usage

ee_tb

Format

A data frame with n rows and the following columns:

- Gene** HGNC gene symbol
- Ensembl** Ensembl gene identifier
- EntrezID** NCBI Entrez ID

Uniprot UNIPROT ID

Is_secreted Whether the protein is a predicted secretory protein or not

Tissue.specificity expression category, one of "Tissue enhanced", "Group enriched", or "Tissue enriched"

tissue tissue names

Source

Human Protein Atlas <https://www.proteinatlas.org>. Uhlen M et al. (2017). A pathology atlas of the human cancer transcriptome. *Science*, 357(6352). doi:10.1126/science.aan2507

Examples

```
data(ee_tb)
head(ee_tb)
```

Enrichment-class	<i>Enrichment-class</i>
------------------	-------------------------

Description

An S4 class to store enrichment analysis results, including the enrichment results of input and background protein lists.

Slots

input An object containing enrichment results for the input set

background An object containing enrichment results for the background set

Examples

```
# Create an Enrichment object
obj <- methods::new("Enrichment",
  input      = data.frame(Gene = c("TP53", "BRCA1")),
  background = data.frame(Gene = c("TP53", "BRCA1", "EGFR", "MYC"))
)
show(obj)
```

geneClassification *Classify genes by tissue specificity*

Description

This function classifies genes into tissue specificity categories ("Tissue enriched", "Group enriched", "Tissue enhanced", "Low tissue specificity", or "Not detected") based on relative expression levels across tissues, following Human Protein Atlas-style rules.

Usage

```
geneClassification(
  expression,
  tissue_groups = NULL,
  min_expression = 1,
  secretory_analysis = FALSE,
  typeKey
)
```

Arguments

expression	A numeric matrix or data frame with genes as rows and tissues as columns. Row names of it must correspond to gene identifiers defined by typeKey.
tissue_groups	Optional data frame with two columns named Tissue and Group, defining pre-defined tissue groupings for group-enriched analysis. If not provided, the tissue groups would be automatically calculated among the top-expressing tissues.
min_expression	Numeric value specifying the minimum expression level required for a gene to be considered detected. Default is 1.
secretory_analysis	Logical indicating whether to include predicted secretory protein annotations from the human protein atlas. Default is FALSE.
typeKey	Character string specifying the gene identifier type used in expression row names. Must be one of "Gene", "Ensembl", or "EntrezID".

Details

Optionally, the function can integrate predicted secretory protein annotations from the **Protein2Tissue** database or derive gene, Ensembl, Entrez ID, and UniProt annotations using **org.Hs.eg.db**.

The classification is performed gene by gene using the following rules:

- **Not detected:** Maximum expression across all tissues is below min_expression.
- **Tissue enriched:** Expression in one tissue is at least 4-fold higher than expression in all other tissues.
- **Group enriched:** Mean expression of a group of tissues (either user-defined or automatically detected among the top-expressing tissues) is at least 4-fold higher than the maximum expression in all remaining tissues.

- **Tissue enhanced:** Expression in one tissue is at least 4-fold higher than the mean expression across all tissues.
- **Low tissue specificity:** Not meeting any of the criteria above.

When `secretory_analysis = TRUE`, gene annotations and predicted secretory status are retrieved from the internal `ee_tb` dataset provided by the **Protein2Tissue** package. Genes not present in this database are removed.

Value

data.frame with tissue specificity classification

A data frame containing:

- Gene annotation columns: Gene, Ensembl, EntrezID, and Uniprot
- Is_secreted (if available)
- Tissue.specificity classification
- One column per tissue indicating classification labels

Examples

```
url <- "https://raw.githubusercontent.com/ChrYang/DataSet/main/tissue_tpm_gtex.csv"
tissue_tpm_gtex <- read.csv(url, row.names=1)
res <- geneClassification(
  expression      = tissue_tpm_gtex,
  typeKey         = "Ensembl",
  min_expression  = 1,
  secretory_analysis = FALSE
)

head(res)

# Example with predefined tissue groups
tissue_groups <- data.frame(
  Tissue = c("Colon_Transverse", "Colon_Sigmoid", "Liver"),
  Group  = c("Colon", "Colon", "Liver")
)

res_grouped <- geneClassification(
  expression = tissue_tpm_gtex,
  tissue_groups = tissue_groups,
  typeKey = "Ensembl",
  min_expression = 1,
  secretory_analysis = FALSE
)
```

getBackground	<i>Get background</i>
---------------	-----------------------

Description

Accessor for the background slot of an Enrichment object

Usage

```
getBackground(object)

## S4 method for signature 'Enrichment'
getBackground(object)
```

Arguments

object An Enrichment object

Value

The background data frame

Examples

```
obj <- methods::new("Enrichment",
  input      = data.frame(Gene = c("TP53", "BRCA1")),
  background = data.frame(Gene = c("TP53", "BRCA1", "EGFR"))
)
getBackground(obj)
```

getInput	<i>Get input</i>
----------	------------------

Description

Accessor for the input slot of an Enrichment object

Usage

```
getInput(object)

## S4 method for signature 'Enrichment'
getInput(object)
```

Arguments

object An Enrichment object

Value

The input data frame

Examples

```
obj <- methods::new("Enrichment",
  input      = data.frame(Gene = c("TP53", "BRCA1")),
  background = data.frame(Gene = c("TP53", "BRCA1", "EGFR"))
)
getInput(obj)
```

goEnriched

GO enrichment analysis on tissue-specific genes

Description

Performs Gene Ontology (GO) enrichment analysis on genes showing tissue-specific or group-enriched expression patterns, using [enrichGO](#).

Usage

```
goEnriched(out, type = "BP", tissueTest = "", inclusion = "All")
```

Arguments

<code>out</code>	An object compatible with <code>.wideToLong()</code> , typically a <code>SummarizedExperiment</code> or similar Bioconductor object containing gene expression data across tissues.
<code>type</code>	A character string specifying the GO ontology category. One of "BP" (Biological Process), "MF" (Molecular Function), or "CC" (Cellular Component). Default is "BP".
<code>tissueTest</code>	A character string specifying the tissue to test. Must match a value in the <code>tissue</code> column of the long-format data. Default is "".
<code>inclusion</code>	A character vector specifying which signal categories to include. One of "Tissue enhanced", "Group enriched", "Tissue enriched", or "All" (default) which includes all three categories.

Value

An `enrichResult` object from [enrichGO](#) containing GO enrichment results for the specified tissue and signal categories.

Examples

```
data("plasma_data", package = "Protein2Tissue")
tmp <- dplyr::filter(plasma_data, Coefficient.Age > 0 & q.Age < 0.05)
tmp <- dplyr::pull(tmp, UniqueSymbol)
myList <- toupper(tmp)

myBackground <- toupper(plasma_data$UniqueSymbol)
result <- tissueAnalysis(
  input      = myList,
  background = myBackground,
  typeKey    = "Gene",      # identifier type for input
  typeKeyBg  = "Gene",      # identifier type for background
)
goEnriched(result, type = "BP", tissueTest = "Liver", inclusion = "All")
```

heatmapEnrich

Heatmap of Tissue Expression Fingerprint

Description

Generates a heatmap showing tissue expression signals per gene from an Enrichment object, with optional filtering by signal category.

Usage

```
heatmapEnrich(out, inclusion = "All")
```

Arguments

<code>out</code>	An Enrichment S4 object containing the results of tissue enrichment analysis.
<code>inclusion</code>	A character vector specifying which signal categories to include. Valid options are "Tissue enhanced", "Group enriched", and "Tissue enriched". Use "All" to include all three categories. Default is "All".

Details

The function first converts the Enrichment object to long format using the internal `.wideToLong` function, retaining NA values to show absent signals in the heatmap. Signal categories are colour-coded for easy interpretation.

Value

A `ggplot2` object showing a heatmap of tissue expression signals per gene, with tissues on the x-axis and genes on the y-axis.

See Also

[plotProteinType](#), [tissueAnalysis](#)

Examples

```

result <- tissueAnalysis(
  input      = c("TP53", "BRCA1", "EGFR"),
  background = c("347", "948", "1050", "2167", "2819", "3625"),
  typeKey    = "Gene",
  typeKeyBg  = "EntrezID"
)

# plot all signal categories
heatmapEnrich(result)

# plot specific categories only
heatmapEnrich(result, inclusion = c("Tissue enriched", "Tissue enhanced"))

```

networkPlot

*Network plot of gene-tissue relationships***Description**

Creates a network visualisation showing gene-tissue relationships from a wide-format expression object.

Usage

```
networkPlot(out)
```

Arguments

out typically a SummarizedExperiment or similar Bioconductor object containing gene expression data across tissues.

Value

A ggplot object representing the gene-tissue network. Tissue nodes are displayed as large labelled circles; gene nodes as small points. Edges are coloured by signal category.

Examples

```

result <- tissueAnalysis(
  input      = c("TP53", "BRCA1", "EGFR"),
  background = c("347", "948", "1050", "2167", "2819", "3625"),
  typeKey    = "Gene",
  typeKeyBg  = "EntrezID",
  database   = NULL
)

p <- networkPlot(result)
print(p)

```

plasma_data	<i>Example plasma proteomics dataset of age-associated protein expression</i>
-------------	---

Description

A data frame containing differential expression results from a plasma proteomics study. The dataset captures associations between protein expression levels and age and is intended as example input for `tissueAnalysis()`.

Usage

```
plasma_data
```

Format

A data frame with n rows and 4 columns:

UniqueSymbol Character. Protein or gene identifier.

Coefficient.Age Numeric. Estimated effect size for association with age.

p.Age Numeric. Raw p-value for the age association.

q.Age Numeric. Adjusted p-value (q-value) controlling for multiple testing.

Details

This dataset is adapted from Lehallier et al. (2019), which profiled age-associated changes in the human plasma proteome.

Source

Lehallier et al. (2019). *Undulating changes in human plasma proteome profiles across the lifespan*. Nature Medicine. doi:[10.1038/s4159101906732](https://doi.org/10.1038/s4159101906732)

Examples

```
data(plasma_data)
head(plasma_data)
```

plotProteinType	<i>Plot Gene-level RNA Tissue Specificity</i>
-----------------	---

Description

Generates a bar chart showing the distribution of genes across RNA tissue specificity classes from an Enrichment object.

Usage

```
plotProteinType(out)
```

Arguments

out	An Enrichment S4 object containing the results of tissue enrichment analysis.
-----	---

Details

The function summarises the `Tissue.specificity` column from the input slot of the Enrichment object and plots the count of genes in each specificity class as a horizontal bar chart.

Value

A ggplot2 object showing the number of genes in each RNA tissue specificity class.

See Also

[heatmapEnrich](#), [tissueAnalysis](#)

Examples

```
result <- tissueAnalysis(  
  input      = c("TP53", "BRCA1", "EGFR"),  
  background = c("347", "948", "1050", "2167", "2819", "3625"),  
  typeKey    = "Gene",  
  typeKeyBg  = "EntrezID"  
)  
plotProteinType(result)
```

testByTissue

*Statistical Test of significance for Tissue Enrichment***Description**

Performs Fisher's exact test for each tissue to identify tissues that are significantly enriched in the input protein list compared to the background.

Usage

```
testByTissue(
  out,
  inclusion = "All",
  testMethod = "fisher",
  padj = "none",
  secretoryOnly = FALSE
)
```

Arguments

out	An Enrichment S4 object containing the results of tissue enrichment analysis.
inclusion	A character vector specifying which signal categories to include in the analysis. Valid options are "Tissue enhanced", "Group enriched", and "Tissue enriched". Use "All" to include all three categories. Default is "All".
testMethod	A character string specifying the method for statistics tests for the contingency table. Passed to fisher.test or exact.test . Common options include "fisher", "barnard", and "boschloo". Default is "fisher".
padj	A character string specifying the method for multiple testing correction. Passed to p.adjust . Common options include "BH", "bonferroni", and "none". Default is "none".
secretoryOnly	A logical value indicating whether to restrict the analysis to secretory proteins only (i.e. rows where <code>Is_secreted</code> is TRUE). Default is FALSE.

Details

For each tissue, a 2x2 contingency table is constructed and exact test is applied to test whether the proportion of input proteins expressed in that tissue differs significantly from the background proportion.

The contingency table is structured as follows:

	Expressed	Not expressed
Input	a	b
Background	c	d

Multiple testing correction is applied across all tissues using the method specified in `padj`.

Value

A data frame with one row per tissue containing the following columns:

- `tissue` — tissue name
- `input_in` — number of input proteins present in the tissue
- `input_total` — total number of input proteins
- `bg_in` — number of background proteins present in the tissue
- `bg_total` — total number of background proteins
- `input_prop` — proportion of input proteins in the tissue
- `bg_prop` — proportion of background proteins in the tissue
- `OR` — odds ratio from Fisher's exact test
- `p_value` — p-value from Fisher's exact test
- `p_adj` — adjusted p-value

See Also

[tissueAnalysis](#), [heatmapEnrich](#), [fisher.test](#), [exact.test](#), [p.adjust](#)

Examples

```
result <- tissueAnalysis(
  input      = c("TP53", "BRCA1", "EGFR"),
  background = c("347", "948", "1050", "2167", "2819", "3625"),
  typeKey    = "Gene",
  typeKeyBg  = "EntrezID"
)

# run Fisher's exact test across all tissues
testResult <- testByTissue(result)

# with BH correction and secretory proteins only
testResult <- testByTissue(
  result,
  inclusion      = c("Tissue enriched", "Tissue enhanced"),
  testMethod     = "barnard",
  padj           = "BH",
  secretoryOnly = TRUE
)
```

testPlot	<i>Plot Exact Test Results by Tissue</i>
----------	--

Description

Generates a horizontal grouped bar chart comparing the proportion of input proteins versus background proteins across tissues, with p-values annotated from exact test.

Usage

```
testPlot(testTable)
```

Arguments

testTable	A data frame output from testByTissue , containing columns tissue, input_prop, bg_prop, p_value, and p_adj.
-----------	---

Details

The function reshapes the exact test result table from wide to long format and plots input and background proportions side by side per tissue. Tissues are ordered by proportion for easier visual comparison. P-values are rounded to 3 decimal places and displayed to the right of the bars.

Value

A ggplot2 object showing a horizontal grouped bar chart with:

- Blue bars representing the proportion of input proteins per tissue
- Grey bars representing the proportion of background proteins
- P-values from exact test annotated per tissue

See Also

[testByTissue](#), [tissueAnalysis](#)

Examples

```
result <- tissueAnalysis(
  input      = c("TP53", "BRCA1", "EGFR"),
  background = c("347", "948", "1050", "2167", "2819", "3625"),
  typeKey    = "Gene",
  typeKeyBg  = "EntrezID",
  database   = NULL
)

# run test's exact test across all tissues
testResult <- testByTissue(result)
```

```
# plot results
testPlot(testResult)
```

tissue_map	<i>An example of the data frame the for the 'tissue_groups' parameter of geneClassification.</i>
------------	--

Description

An example of the data frame the for the 'tissue_groups' parameter of geneClassification.

Usage

```
tissue_map
```

Format

An example of the data frame the for the 'tissue_groups' parameter of geneClassification. This data frame has two columns named Tissue and Group, defining predefined tissue groupings for group-enriched analysis.

Examples

```
data(tissue_map)
head(tissue_map)
```

tissueAnalysis	<i>Tissue Enrichment Analysis</i>
----------------	-----------------------------------

Description

Performs tissue enrichment analysis by mapping input proteins and background proteins against a reference expression database. Proteins not found in the database are removed with a warning, and the matched results are returned as an Enrichment object for downstream analysis.

Usage

```
tissueAnalysis(
  input,
  background = "Default",
  typeKey = "Gene",
  typeKeyBg = "Ensembl",
  database = NULL
)
```

Arguments

input	A character vector of protein identifiers to be analysed (e.g. gene symbols or Ensembl IDs).
background	A character vector of background protein identifiers representing the universe of proteins. Defaults to "Default".
typeKey	A character string specifying the identifier type for the input proteins. Must match a column name in the reference database. Default is "Gene".
typeKeyBg	A character string indicating the identifier type used for the background proteins. This must be "Gene", "Ensembl", or "EntrezID". The default is "Ensembl".
database	Defaults to ee_tb (Human Protein Atlas data bundled with this package). Can be replaced with a custom data frame provided it contains columns matching typeKey and typeKeyBg.

Details

The function maps input and background protein lists against the internal reference expression database ee_tb. Proteins not found in the database are removed before analysis. If no proteins are found in the database, the function stops with an informative error message.

Note that typeKey and typeKeyBg may differ because input proteins and background proteins can use different identifier systems (e.g. gene symbols vs Ensembl IDs). Both are mapped to the same reference database ee_tb which contains multiple identifier columns.

Value

An Enrichment S4 object containing:

- input — a data frame of matched input proteins with reference database annotations
- background — a data frame of matched background proteins with reference database annotations

See Also

[enrichmentAnalysis](#), [Enrichment-class](#)

Examples

```
result <- tissueAnalysis(
  input      = c("TP53", "BRCA1", "EGFR"),
  background = c("347", "948", "1050", "2167", "2819", "3625"),
  typeKey    = "Gene",
  typeKeyBg  = "EntrezID",
  database   = NULL
)

# access results
getInput(result)
getBackground(result)
```

tissueAnalysis2

*Specific tissue Enrichment Analysis***Description**

Performs enrichment analysis in the tissue designated by the parameter `tissueToTest` by mapping input genes or proteins against a reference expression database. Genes or proteins not found in the database are removed with a warning, and the matched results are returned as a data frame showing genes or proteins enriched in the corresponding tissue.

Usage

```
tissueAnalysis2(
  input,
  typeKey = "Gene",
  database = NULL,
  inclusion = "All",
  secretoryOnly = FALSE,
  tissueToTest
)
```

Arguments

<code>input</code>	A character vector of gene or proteins identifiers to be analysed (e.g. gene symbols or Ensembl IDs).
<code>typeKey</code>	A character string specifying the identifier type for the input genes or proteins. Must match a column name in the reference database. Default is "Gene" (gene symbol).
<code>database</code>	Defaults to <code>ee_tb</code> (Human Protein Atlas data bundled with this package). Can be replaced with a custom data frame provided it contains columns matching <code>typeKey</code> .
<code>inclusion</code>	A character vector specifying which signal categories to include in the analysis. Valid options are "Tissue enhanced", "Group enriched", and "Tissue enriched". Use "All" to include all three categories. Default is "All".
<code>secretoryOnly</code>	A logical value indicating whether to restrict the analysis to secretory proteins only (i.e. rows where <code>Is_secreted</code> is TRUE). Default is FALSE.
<code>tissueToTest</code>	The tissue in which the input proteins or genes are tested for enrichment. Must be one of the tissue listed in the database provided. Tissues in the default database are <code>c("Adipose tissue", "Breast", "Seminal vesicle", "Parathyroid gland", "Bone marrow", "Lung", "Lymphoid tissue", "Liver", "Testis", "Brain", "Esophagus", "Gall-bladder", "Intestine", "Pancreas", "Ovary", "Kidney", "Placenta", "Cervix", "Skin", "Endometrium", "Skeletal muscle", "Retina", "Tongue", "Heart muscle", "Choroid plexus", "Fallopian tube", "Epididymis", "Salivary gland", "Vagina", "Smooth muscle", "Adrenal gland", "Blood vessel", "Pituitary gland", "Stomach", "Urinary bladder", "Thyroid gland", "Prostate")</code>

Details

The function maps input gene or proteins lists against the internal reference expression database like the default `tee_tb` in the tissue designated by the parameter `tissueToTest`. Proteins or genes not found in the database are removed before analysis. If no proteins are found in the database, the function stops with an informative error message.

Value

A data frame of matched input genes with reference database annotations for the tissue designated by the `tissueToTest` parameter.

See Also

[enrichmentAnalysis](#), [Enrichment-class](#)

Examples

```
# using gene symbols for input, Ensembl IDs for background
result2 <- tissueAnalysis2(
  input      = c("BRCA1", "EGFR"),
  typeKey    = "Gene",
  database   = NULL,
  inclusion  = "All",
  secretoryOnly = FALSE,
  tissueToTest = "Testis"
)

# access results
result2
```

writeOutput

Write Enrichment Analysis Output to File

Description

Write Enrichment Analysis Output to File

Usage

```
writeOutput(out, type = "csv", filename)
```

Arguments

<code>out</code>	An Enrichment S4 object containing the results of tissue enrichment analysis.
<code>type</code>	A character string specifying the output file format. Must be either "csv" or "tsv". Default is "csv".
<code>filename</code>	A character string specifying the output file path.

Value

Invisibly returns NULL. Called for its side effect of writing a file to disk.

Examples

```
result <- tissueAnalysis(  
  input      = c("TP53", "BRCA1", "EGFR"),  
  background = c("347", "948", "1050", "2167", "2819", "3625"),  
  typeKey    = "Gene",  
  typeKeyBg  = "EntrezID",  
  database   = NULL  
)  
  
writeOutput(result, type = "csv", "~/Enrichment_Results.csv")
```

Index

* datasets

- ee_tb, [2](#)
- plasma_data, [10](#)
- tissue_map, [15](#)

* write

- writeOutput, [18](#)

ee_tb, [2](#)

enrichGO, [7](#)

Enrichment-class, [3](#)

enrichmentAnalysis, [16](#), [18](#)

exact.test, [12](#), [13](#)

fisher.test, [12](#), [13](#)

geneClassification, [4](#)

getBackground, [6](#)

getBackground, Enrichment-method
(getBackground), [6](#)

getInput, [6](#)

getInput, Enrichment-method (getInput), [6](#)

goEnriched, [7](#)

heatmapEnrich, [8](#), [11](#), [13](#)

networkPlot, [9](#)

p.adjust, [12](#), [13](#)

plasma_data, [10](#)

plotProteinType, [8](#), [11](#)

testByTissue, [12](#), [14](#)

testPlot, [14](#)

tissue_map, [15](#)

tissueAnalysis, [8](#), [11](#), [13](#), [14](#), [15](#)

tissueAnalysis2, [17](#)

writeOutput, [18](#)