

# Package ‘ssPATHS’

January 21, 2025

**Type** Package

**Title** ssPATHS: Single Sample PATHway Score

**Version** 1.20.0

**Author** Natalie R. Davidson

**Maintainer** Natalie R. Davidson <natalie.davidson@inf.ethz.ch>

**Description** This package generates pathway scores from expression data for single samples after training on a reference cohort. The score is generated by taking the expression of a gene set (pathway) from a reference cohort and performing linear discriminant analysis to distinguish samples in the cohort that have the pathway augmented and not. The separating hyperplane is then used to score new samples.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** ROCR, dml, MESS

**Suggests** ggplot2, testthat (>= 2.1.0)

**Depends** SummarizedExperiment

**RoxygenNote** 6.1.1

**biocViews** Software, GeneExpression, BiomedicalInformatics, RNASeq, Pathways, Transcriptomics, DimensionReduction, Classification

**git\_url** <https://git.bioconductor.org/packages/ssPATHS>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** 37a0886

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2025-01-20

## Contents

expected_score_output . . . . .	2
gene_weights_reference . . . . .	2
get_classification_accuracy . . . . .	3
get_gene_weights . . . . .	4

get_hypoxia_genes . . . . .	6
get_new_samp_score . . . . .	6
new_samp_df . . . . .	8
tcga_expr_df . . . . .	8

## Index 10

---

expected\_score\_output *Gene Expression Values for PDAC Cancer Cell Lines exposed to Hypoxia*

---

### Description

Gene Expression Values for PDAC Cancer Cell Lines exposed to Hypoxia

### Usage

```
data(expected_score_output)
```

### Format

A data frame with columns:

**sample\_id** String. The name of the sample. Samples with "hyp" or "norm" in the sample id are cell lines that were exposed to hypoxic or normoxic conditions respectively. Samples with "ctrl" or "noHIF" were samples that were able to produce a HIF-mediated hypoxic response or not, respectively.

**pathway\_score** Float. The estimated hypoxia score for this sample.

### Source

Derived Data

### Examples

```
## Not run:
  expected_score_output

## End(Not run)
```

---

gene\_weights\_reference  
*Gene Expression Values for PDAC Cancer Cell Lines exposed to Hypoxia*

---

### Description

Gene Expression Values for PDAC Cancer Cell Lines exposed to Hypoxia

### Usage

```
data(gene_weights_reference)
```

**Format**

A data frame with columns:

**gene\_weight** Float. Gene weighting learned from reference data.

**gene\_id** String. The ensembl id of the gene.

**Source**

Derived data

**Examples**

```
## Not run:  
gene_weights_reference  
  
## End(Not run)
```

---

```
get_classification_accuracy  
      Get Classification Accuracy
```

---

**Description**

Get the AUC-ROC, AUC-PR, and ROC/PR curves for plotting.

**Usage**

```
get_classification_accuracy(sample_scores, positive_val)
```

**Arguments**

**sample\_scores** This is a data.frame containing the sample id, score, and true label Y. This object is returned by the method `get_gene_weights`.

**positive\_val** This is the value that will denote a true positive. It must be one of the two values in the Y column in `sample_scores`.

**Value**

This returns a list of performance metrics

<code>auc_pr</code>	Area under the PR-curve
<code>auc_roc</code>	Area under the ROC-curve
<code>perf_pr</code>	ROCR object for plotting the PR-curve
<code>perf_roc</code>	ROCR object for plotting the ROC-curve

**Author(s)**

Natalie R. Davidson

## Examples

```

data(tcga_expr_df)

# transform from data.frame to SummarizedExperiment
tcga_se <- SummarizedExperiment(t(tcga_expr_df[, -(1:4)]),
                               colData=tcga_expr_df[, 2:4])
colnames(tcga_se) <- tcga_expr_df$tcga_id
colData(tcga_se)$sample_id <- tcga_expr_df$tcga_id

hypoxia_gene_ids <- get_hypoxia_genes()
hypoxia_gene_ids <- intersect(hypoxia_gene_ids, rownames(tcga_se))

colData(tcga_se)$Y <- ifelse(colData(tcga_se)$is_normal, 0, 1)

# now we can get the gene weightings
res <- get_gene_weights(tcga_se, hypoxia_gene_ids, unidirectional=TRUE)
sample_scores <- res[[2]]

# check how well we did
training_res <- get_classification_accuracy(sample_scores, positive_val=1)
print(training_res[[2]])

plot(training_res[[3]], col="orange", ylim=c(0, 1))
legend(0.1,0.8,c(training_res$auc_pr,"\n"), border="white", cex=1.7,
       box.col = "white")

plot(training_res[[4]], col="blue", ylim=c(0, 1))
legend(0.1,0.8,c(training_res$auc_roc,"\n"),border="white",cex=1.7,
       box.col = "white")

```

---

get\_gene\_weights

*Get Gene Weights from Reference Data*

---

## Description

This method performs linear discriminant analysis on a reference dataset using a pre-defined set of genes related to a pathway of interest.

## Usage

```
get_gene_weights(expression_se, gene_ids, unidirectional)
```

## Arguments

**expression\_se** This is an SummarizedExperiment object of the reference samples. Rows are genes and columns are samples. The colData component must contain a sample\_id column. Within this method, there is a normalization step where each sample is scaled across all genes in the SummarizedExperiment assay. For this to be stable and consistent, we recommend that the assay contain at least 500 genes that are consistently expressed across all samples in addition to the genes in the pathway of interest.

- `gene_ids` This is a vector of strings, where each element is a `gene_id` in the pathway of interest. The `gene_ids` must be present in `rownames(expression_se)`.
- `unidirectional` This is a boolean, `default=TRUE`. Most genesets are unidirectional, meaning that most genes are either increasing or decreasing together. If this is set to `TRUE`, then the learned weights will be clipped such that the dominant directionality is kept, and the other gene weights are set to zero.

### Value

A list containing the gene weights and estimated scores of the reference samples.

`proj_vector_df` A dataframe containing the gene weights and gene ids

`dca_proj` A dataframe containing the sample scores and sample ids.

### Author(s)

Natalie R. Davidson

### References

Steven C.H. Hoi, W. Liu, M.R. Lyu and W.Y. Ma (2006). Learning Distance Metrics with Contextual Constraints for Image Retrieval. Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR2006).

### Examples

```
data(tcga_expr_df)

# transform from data.frame to SummarizedExperiment
tcga_se <- SummarizedExperiment(t(tcga_expr_df[ , -(1:4)]),
                               colData=tcga_expr_df[ , 2:4])
colnames(tcga_se) <- tcga_expr_df$tcga_id
colData(tcga_se)$sample_id <- tcga_expr_df$tcga_id

# get related genes, for us hypoxia
hypoxia_gene_ids <- get_hypoxia_genes()
hypoxia_gene_ids <- intersect(hypoxia_gene_ids, rownames(tcga_se))

# setup labels for classification
colData(tcga_se)$Y <- ifelse(colData(tcga_se)$is_normal, 0, 1)

# now we can get the gene weightings
res <- get_gene_weights(tcga_se, hypoxia_gene_ids, unidirectional=TRUE)
gene_weights_test <- res[[1]]
sample_scores <- res[[2]]
```

---

get\_hypoxia\_genes      *Get Ensembl ids of hypoxia related genes.*

---

**Description**

Returns a vector of Ensembl ids of hypoxia related genes.

**Usage**

```
get_hypoxia_genes()
```

**Value**

Vector of ensembl ids.

**Author(s)**

Natalie R. Davidson

**Examples**

```
# read in the reference expression data for hypoxia score generation
data(tcga_expr_df)

# transform from data.frame to SummarizedExperiment
tcga_se <- SummarizedExperiment(t(tcga_expr_df[ , -(1:4)]),
                               colData=tcga_expr_df[ , 2:4])
colnames(tcga_se) <- tcga_expr_df$tcga_id
colData(tcga_se)$sample_id <- tcga_expr_df$tcga_id

# let's get the expression of hypoxia associated genes
hypoxia_gene_ids <- get_hypoxia_genes()
hypoxia_gene_ids <- intersect(hypoxia_gene_ids, rownames(tcga_se))
hypoxia_se <- tcga_se[hypoxia_gene_ids,]
```

---

get\_new\_samp\_score      *Get a pathway score for an unseen sample*

---

**Description**

Using the gene weights learned from the reference cohort, we apply the weightings to new samples to estimate their pathway activity.

**Usage**

```
get_new_samp_score(gene_weights, expression_se, gene_ids, run_normalization = TRUE)
```

**Arguments**

- `gene_weights` This is a data.frame containing gene ids and gene weights, output by `get_gene_weights`. The gene ids must be in the column ids of `expression_matr`.
- `expression_se` This is an `SummarizedExperiment` object of the reference samples. Rows are genes and columns are samples. The `colData` component must contain columns `Y` and `sample_id`. The former indicates whether this is a positive or negative sample and the latter is the unique sample id. Within this method, there is a normalization step where each sample is scaled across all genes in the `SummarizedExperiment` assay. For this to be stable and consistent, we recommend that the assay contain at least 500 genes that are consistently expressed across all samples in addition to the genes in the pathway of interest.
- `gene_ids` This is a vector of strings, where each element is a `gene_id` in the pathway of interest. The `gene_ids` must be present in `rownames(expression_se)`.
- `run_normalization` Boolean value. If `TRUE`, the data will be log-transformed, centered and scaled. This is recommended since this is done to the reference set when learning the gene weights.

**Value**

A data.frame containing the sample id, sample score, and associated `Y` value if it was included in `expression_se`.

**Author(s)**

Natalie R. Davidson

**Examples**

```
data(tcga_expr_df)

# transform from data.frame to SummarizedExperiment
tcga_se <- SummarizedExperiment(t(tcga_expr_df[ , -(1:4)]),
                               colData=tcga_expr_df[ , 2:4])
colnames(tcga_se) <- tcga_expr_df$tcga_id
colData(tcga_se)$sample_id <- tcga_expr_df$tcga_id

# get the genes of interest, here hypoxia genes
hypoxia_gene_ids <- get_hypoxia_genes()
hypoxia_gene_ids <- intersect(hypoxia_gene_ids, rownames(tcga_se))

# label the samples for classification
colData(tcga_se)$Y <- ifelse(colData(tcga_se)$is_normal, 0, 1)

# now we can get the gene weightings
res <- get_gene_weights(tcga_se, hypoxia_gene_ids, unidirectional=TRUE)
gene_weights <- res[[1]]
sample_scores <- res[[2]]

# get the new data so we can apply our score to it
data(new_samp_df)
new_samp_se <- SummarizedExperiment(t(new_samp_df[ , -(1)]),
                                   colData=new_samp_df[ , 1, drop=FALSE])
colnames(colData(new_samp_se)) <- "sample_id"
```

```
new_score_df_calculated <- get_new_samp_score(gene_weights, new_samp_se)
```

---

new_samp_df	<i>Gene Expression Values for PDAC Cancer Cell Lines exposed to Hypoxia</i>
-------------	---

---

### Description

A data frame with columns:

**sample\_id** String. The name of the sample. Samples with "hyp" or "norm" in the sample id are cell lines that were exposed to hypoxic or normoxic conditions respectively. Samples with "ctrl" or "noHIF" were samples that were able to produce a HIF-mediated hypoxic response or not, respectively.

**ENSG00000074410** Int. Gene expression value for this gene.

### Usage

```
data(new_samp_df)
```

### Format

An object of class `data.frame` with 12 rows and 27 columns.

### Source

Generated by Philipp Markolin, files will be uploaded on GEO

### Examples

```
## Not run:
new_samp_df

## End(Not run)
```

---

tcga_expr_df	<i>Gene Expression Values for PDAC Cancer Cell Lines exposed to Hypoxia</i>
--------------	---

---

### Description

A data frame with columns:

**tcga\_id** String. TCGA aliquot barcode

**study** String. TCGA study abbreviation

**is\_normal** Boolean. TRUE if sample is adjacent normal, FALSE if tumor.

**libsize\_75percent** Float. Library size as estimated by the 75th quartile.

**ENSG00000070831** String. Library size normalized gene expression value for this gene.



**Usage**

```
data(tcga_expr_df)
```

**Format**

An object of class `data.frame` with 9461 rows and 54 columns.

**Source**

This data is generated by the TCGA Research Network: <https://www.cancer.gov/tcga> and downloaded from the NCI Genomic Data Commons.

**Examples**

```
## Not run:  
tcga_expr_df  
  
## End(Not run)
```

# Index

## \* datasets

- expected\_score\_output, [2](#)
- gene\_weights\_reference, [2](#)
- new\_samp\_df, [8](#)
- tcga\_expr\_df, [8](#)

expected\_score\_output, [2](#)

gene\_weights\_reference, [2](#)  
get\_classification\_accuracy, [3](#)  
get\_gene\_weights, [4](#)  
get\_hypoxia\_genes, [6](#)  
get\_new\_samp\_score, [6](#)

new\_samp\_df, [8](#)

tcga\_expr\_df, [8](#)