

# Package ‘proteinProfiles’

May 18, 2024

**Type** Package

**Title** Protein Profiling

**Version** 1.44.0

**Author** Julian Gehring

**Maintainer** Julian Gehring <jg-bioc@gmx.com>

**Imports** graphics, stats

**Depends** R (>= 2.15.2)

**Suggests** testthat

**Description** Significance assessment for distance measures of time-course  
protein profiles

**License** GPL-3

**LazyLoad** yes

**BiocViews** Annotation, Proteomics, TimeCourse, Clustering

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

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** ce612e2

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-17

## Contents

proteinProfiles-package . . . . .	2
annotation . . . . .	2
filterFeatures . . . . .	3
ips_sample-data . . . . .	4
profileDistance . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

proteinProfiles-package

*proteinProfiles package*


---

### Description

Significance assessment for distance measures of time-course protein profiles

### Details

The package is published under the GPL-3 license.

### Author(s)

Julian Gehring

Maintainer: Julian Gehring <julian.gehring@embl.de>

### See Also

filterFeatures, grepAnnotation, profileDistance, plotProfileDistance

---

annotation

*grepAnnotation*


---

### Description

Find protein group of interest based on pattern matching against the annotation data.

### Usage

```
grepAnnotation(anno, pattern, column, ...)
```

### Arguments

anno	Data frame with annotation to analyze. The row names have to match rownames in the matrix containing the protein abundance data. Different columns correspond to different annotation columns, e.g. from different sources.
pattern	Character string representing a (sub)string or regular expression matched against the annotation. For details, see ‘grep’.
column	Column of the annotation data ‘pattern’ is matched against.
...	Optional arguments, passed to the ‘grep’ function.

**Value**

Protein identifiers (rownames of ‘anno’) specifying the protein group of interest in the data set.

**See Also**

grep, filterRatios, proteinProfiles

**Examples**

```
data(ips_sample)

index_28S <- grepAnnotation(annotation, pattern="^28S",
column="Protein.Name")

index_ribosome <- grepAnnotation(annotation, "Ribosome", "KEGG")
```

---

filterFeatures	<i>filterFeatures</i>
----------------	-----------------------

---

**Description**

Filter out proteins exceeding a certain fraction of missing data points.

**Usage**

```
filterFeatures(values, maxNAfraction, verbose=FALSE, plot=FALSE, ...)
```

**Arguments**

values	Numeric matrix containing the data to analyze.
maxNAfraction	Numeric threshold specifying the maximum fraction of data points that can be missing (‘NA’) to still keep the protein.
verbose	Logical indicating whether to print the number of proteins before and after filtering (default: FALSE).
plot	Logical indicating whether to plot a diagnostic plot showing the distribution of the fraction of missing data points associated with the proteins (default: FALSE).
...	Arguments passed to the ‘plot’ method.

**Value**

Numeric matrix with the same structure as ‘values’, with elements not matching the filter criteria removed.

**Examples**

```
data(ips_sample)

ratios_filtered <- filterFeatures(ratios, 0.2, verbose=TRUE)
```

---

ips_sample-data	<i>IPS sample data</i>
-----------------	------------------------

---

**Description**

IPS sample data, separated in protein ratios and annotation.

**Usage**

```
data(ips_sample)
```

**Format**

**ratios** Matrix with protein profiles. Rows correspond to proteins, columns to samples.

**annotation** Data frame with annotation columns, with row names matching those of ‘ratios.’

**Examples**

```
data(ips_sample)

str(ratios)

str(annotation)
```

---

profileDistance	<i>profileDistance</i>
-----------------	------------------------

---

**Description**

Compute distance between a protein group of interest and assess its significance by comparing it to the distances of randomly selected groups of proteins.

**Usage**

```
profileDistance(values, index, nSample=1000, seed)

plotProfileDistance(z, ...)
```

**Arguments**

values	Numeric matrix containing the protein data to analyze.
index	Protein identifiers for the protein group of interest in the data set ‘values’, as returned from ‘grepAnnotation’ or an equivalent function.
nSample	Integer specifying the number of randomly selected groups of proteins with the same number of elements as specified by ‘index’, serving as a reference for the significance assessment (default: 1000).
seed	Random seed, relevant for the permutation step. For details, see the ‘seed’ function.
z	Return object from ‘profileDistance.’
...	Optional arguments passed to the ‘plot’ function.

**Value**

List with elements:

<b>d0</b> Numeric with mean distance	$d_0$
of profiles for the proteins of interest.	
<b>d1</b> Vector of numerics with mean distances	$d_1$
of randomly selected groups of proteins.	
<b>p</b> Numeric with p-value	$p$
.	

**Examples**

```
data(ips_sample)

index_28S <- grepAnnotation(annotation, pattern="^28S",
column="Protein.Name")

z <- profileDistance(ratios, index_28S)

z$p.value

plotProfileDistance(z)
```

# Index

- \* **datasets**
  - ips\_sample-data, [4](#)
- \* **distribution**
  - proteinProfiles-package, [2](#)
- \* **htest**
  - proteinProfiles-package, [2](#)
- \* **package**
  - proteinProfiles-package, [2](#)
- annotation, [2](#)
- annotation (ips\_sample-data), [4](#)
- filterFeatures, [3](#)
- grepAnnotation (annotation), [2](#)
- ips\_sample (ips\_sample-data), [4](#)
- ips\_sample-data, [4](#)
- plotProfileDistance (profileDistance), [4](#)
- profileDistance, [4](#)
- proteinProfiles
  - (proteinProfiles-package), [2](#)
- proteinProfiles-package, [2](#)
- ratios (ips\_sample-data), [4](#)