

# Package ‘FeatSeekR’

April 24, 2024

**Type** Package

**Title** FeatSeekR an R package for unsupervised feature selection

**Version** 1.3.0

**Description** FeatSeekR performs unsupervised feature selection using replicated measurements. It iteratively selects features with the highest reproducibility across replicates, after projecting out those dimensions from the data that are spanned by the previously selected features. The selected a set of features has a high replicate reproducibility and a high degree of uniqueness.

**License** GPL-3

**Encoding** UTF-8

**Imports** pheatmap, MASS, pracma, stats, SummarizedExperiment, methods

**RoxygenNote** 7.2.3

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

**VignetteBuilder** knitr

**BugReports** <https://github.com/tcapraz/FeatSeekR/issues>

**URL** <https://github.com/tcapraz/FeatSeekR>

**biocViews** Software, StatisticalMethod, FeatureExtraction,  
MassSpectrometry

**Config/testthat/edition** 3

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

**git\_branch** devel

**git\_last\_commit** 2b06931

**git\_last\_commit\_date** 2023-10-24

**Repository** Bioconductor 3.19

**Date/Publication** 2024-04-24

**Author** Tuemay Capraz [cre, aut] (<<https://orcid.org/0000-0002-2547-067X>>)

**Maintainer** Tuemay Capraz <tuemay.capraz@embl.de>

## Contents

|                                 |   |
|---------------------------------|---|
| calcFstat . . . . .             | 2 |
| check_input . . . . .           | 2 |
| FeatSeek . . . . .              | 3 |
| FeatSeekR . . . . .             | 4 |
| fit_lm . . . . .                | 5 |
| init_selected . . . . .         | 5 |
| plotSelectedFeatures . . . . .  | 6 |
| plotVarianceExplained . . . . . | 6 |
| simData . . . . .               | 7 |
| variance_explained . . . . .    | 8 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>9</b> |
|--------------|----------|

---

|           |                  |
|-----------|------------------|
| calcFstat | <i>calcFstat</i> |
|-----------|------------------|

---

### Description

calcFstat

### Usage

calcFstat(data, fac)

### Arguments

|      |   |
|------|---|
| data | 2 dimensional array with samples x features, where samples belongs different conditions. The function was adapted from the <i>genefilter</i> package. |
| fac  | factor of length samples, indicating which sample belongs to which condition  |

### Value

F-statistic for all features

---

|             |                    |
|-------------|--------------------|
| check_input | <i>check_input</i> |
|-------------|--------------------|

---

### Description

Checks input data. Input data should be a 2 dimensional array with features x samples or SummarizedExperiment carrying one assay named data and colData indicating which sample belongs to which condition

**Usage**

```
check_input(data, max_features, conditions = NULL)
```

**Arguments**

|                         |   |
|-------------------------|---|
| <code>data</code>       | input data provided to FeatSeek either SummarizedExperiment or 2 dimensional array with features x samples                                |
| <code>conditions</code> | if data is a 2 dimensional array with features x samples a factor indicating which sample corresponds to which condition must be provided |

**Value**

SummarizedExperiment where condition information is stored in colData

---

FeatSeek

*FeatSeek*

---

**Description**

This function ranks features of a 2 dimensional array according to their reproducibility between conditions.

**Usage**

```
FeatSeek(
  data,
  conditions = NULL,
  max_features = NULL,
  init = NULL,
  verbose = TRUE
)
```

**Arguments**

|                           |   |
|---------------------------|---|
| <code>data</code>         | SummarizedExperiment with assay named <code>data</code> , where samples belongs to different conditions. Which sample belongs to which condition should be indicated in <code>colData</code> slot <code>conditions</code> . Or <code>matrix</code> with features x samples. Each conditions have multiple samples from replicated measurements. |
| <code>conditions</code>   | factor of length samples, indicating which sample belongs to which condition. Only required if <code>data</code> is provided as <code>matrix</code> .   |
| <code>max_features</code> | integer number of features to rank  |
| <code>init</code>         | character vector with names of initial features. If NULL the feature with highest F-statistic will be used  |
| <code>verbose</code>      | logical indicating whether messages should be printed   |

**Value**

SummarizedExperiment containing one assay with the selected features. rowData stores for each selected feature the F-statistic under metric, the cumulative explained variance under explained\_variance and the feature names under selected

**Examples**

```
# run FeatSeek to select the top 20 features
data <- array(rnorm(100*30), dim=c(30, 100),
dimnames <- list(paste("feature", seq_len(30)), NULL))
conds <- rep(seq_len(50), 2)
res <- FeatSeek(data, conds, max_features=20)

# res stores the 20 selected features ranked by their replicate
# reproducibility
```

---

FeatSeekR

*FeatSeekR an R package for unsupervised feature selection*

---

**Description**

FeatSeekR performs unsupervised feature selection using replicated measurements. It iteratively selects features with the highest reproducibility across conditions, after projecting out those dimensions from the data that are spanned by the previously selected features. The selected a set of features has a high replicate reproducibility and a high degree of uniqueness.

**Details**

For information on how to use this package please type `vignette("FeatSeekR-vignette")`.

Please post questions regarding the package to the Bioconductor Support Site:

<https://support.bioconductor.org>

**Author(s)**

Tümay Capraz

---

|        |               |
|--------|---------------|
| fit_lm | <i>fit_lm</i> |
|--------|---------------|

---

**Description**

Fit linear model for each feature as a function of the previously selected features S. The dimensions spanned by the selected features are projected out of the data by setting each feature to its residuals from the linear model fit.

**Usage**

```
fit_lm(data, S, k)
```

**Arguments**

|      |   |
|------|---|
| data | data (2 dimensional array samples x features) |
| S    | set of selected features                      |
| k    | current iteration                             |

**Value**

data with previously selected features projected out

---

|               |                      |
|---------------|----------------------|
| init_selected | <i>init_selected</i> |
|---------------|----------------------|

---

**Description**

Checks if preselected init features are in input data. If init is NULL, it is set to feature with highest condition correlation.

**Usage**

```
init_selected(init, se)
```

**Arguments**

|      |                                      |
|------|--------------------------------------|
| init | preselected starting set of features |
| data | input data as SummarizedExperiment   |

**Value**

names of initial set of feature

---

`plotSelectedFeatures` *plotSelectedFeatures*

---

**Description**

plot correlation matrix of selected feature sets

**Usage**

```
plotSelectedFeatures(res, n_features = NULL, assay = "selected")
```

**Arguments**

|                         |  |
|-------------------------|--|
| <code>res</code>        | result SummarizedExperiment from FeatSeek function   |
| <code>n_features</code> | top <code>n_features</code> to plot. if NULL then the maximum number of features in <code>res</code> will be plotted |
| <code>assay</code>      | assay slot to plot from result SummarizedExperiment object, default is the selected features slot                    |

**Value**

returns heatmap of selected features

**Examples**

```
# run FeatSeek to select the top 20 features
data <- array(rnorm(100*30), dim=c(30,100),
             dimnames = list(paste("feature", seq_len(30)), NULL))
conds <- rep(seq_len(50), 2)
res <- FeatSeek(data, conds, max_features=20)

# res stores the 20 selected features ranked by their replicate
# reproducibility
# plot the top 5 features
plotSelectedFeatures(res, n_features=5)
```

---

`plotVarianceExplained` *plotVarianceExplained*

---

**Description**

plot variance explained from 1 to `max_features` in `res`

**Usage**

```
plotVarianceExplained(res)
```

**Arguments**

res                    result SummarizedExperiment from FeatSeek function

**Value**

returns plot of variance explained vs number of features

**Examples**

```
# run FeatSeek to select the top 20 features
data <- array(rnorm(100*30), dim=c(30,100),
             dimnames = list(paste("feature", seq_len(30)), NULL))
conds <- rep(seq_len(50), 2)
res <- FeatSeek(data, conds, max_features=20)

# res stores the 20 selected features ranked by their replicate
# reproducibility
plotVarianceExplained(res)
```

---

simData

*simData*


---

**Description**

simulate Data with orthogonal feature clusters and replicated samples. Each feature cluster corresponds to a different latent factor and contains 10 redundant features. E.g. choosing samples = 100, n\_latent\_factors = 5 and replicates = 2 will simulate a 50 x 200 data matrix, where the first 100 samples belong to replicate 1 and sample 101-200 belong to replicate 2.

**Usage**

```
simData(conditions, n_latent_factors, replicates)
```

**Arguments**

conditions            number of conditions to generate samples from  
n\_latent\_factors        number of latent factors to generate  
replicates             number of replicates to generate

**Details**

simData constructs n\_latent\_factors by generating a random matrix  $\mathbf{Q}$  whose row vectors  $\mathbf{Q}_i \sim \mathcal{N}(0, 1)$  with  $n$  samples and  $i \in \{1, \dots, n\_latent\_factors\}$  are orthonormal, each corresponding to a different latent factor. To simulate a set of redundant feature groups, it generates 10 features  $X_j$  for each latent factor  $\mathbf{Q}_i$  by scaling each latent factor by a random factor  $\delta_j \sim \mathcal{N}(0, 1)$  and adding replicate specific noise  $\epsilon_c \sim \mathcal{N}(0, 0.1)$  with  $c \in \{1, \dots, replicates\}$  preserving orthogonality.

**Value**

SummarizedExperiment object carrying simulated data, with colData indicating which sample belongs to which replicate

**Examples**

```
# simulate data 100 samples from 100 conditions, 20 features generated by 2
# latent factors and 2 replicates
simData(conditions=100, n_latent_factors=2, replicates=2)
```

---

variance\_explained     *variance\_explained*

---

**Description**

variance\_explained

**Usage**

```
variance_explained(data, selected)
```

**Arguments**

data                    2 dimensional array samples x features  
selected                character vector of selected features

**Value**

average variance explained by selected features



# Index

## \* **internal**

- calcFstat, 2
- check\_input, 2
- fit\_lm, 5
- init\_selected, 5
- variance\_explained, 8

## \* **package**

- FeatSeekR, 4

- calcFstat, 2
- check\_input, 2

- FeatSeek, 3
- FeatSeekR, 4
- fit\_lm, 5

- init\_selected, 5

- plotSelectedFeatures, 6
- plotVarianceExplained, 6

- simData, 7

- variance\_explained, 8