

# Package ‘BrainSABER’

February 27, 2021

**Type** Package

**Title** Brain Span Atlas in Biobase Expressionset R toolset

**Version** 1.0.0

**Author** Carrie Minette and Evgeni Radichev

**Maintainer** USD Biomedical Engineering <bicbioeng@gmail.com>

**Description** The Allen Institute for Brain Science provides an RNA sequencing (RNA-Seq) data resource for studying transcriptional mechanisms involved in human brain development known as BrainSpan. BrainSABER is an R package that facilitates comparison of user data with the various developmental stages and brain structures found in the BrainSpan atlas by generating dynamic similarity heatmaps for the two data sets. It also provides a self-validating container for user data.

**License** Artistic-2.0

**Encoding** UTF-8

**Depends** R (>= 4.0), biomaRt, SummarizedExperiment

**Imports** data.table, lsa, methods, S4Vectors, utils, BiocFileCache

**RoxygenNote** 7.1.0

**biocViews** GeneExpression, Visualization, Sequencing

**VignetteBuilder** knitr

**Suggests** BiocStyle, ComplexHeatmap, fastcluster, heatmaply, knitr, plotly

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

**git\_branch** RELEASE\_3\_12

**git\_last\_commit** 378073c

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2021-02-26

## R topics documented:

|                                |   |
|--------------------------------|---|
| buildAIBSARNA . . . . .        | 2 |
| CellScabbard . . . . .         | 2 |
| CellScabbard-methods . . . . . | 4 |
| getExampleVector . . . . .     | 5 |

|                                 |    |
|---------------------------------|----|
| getExternalVector . . . . .     | 6  |
| getRelevantGenes . . . . .      | 7  |
| getSimDataFrame . . . . .       | 8  |
| getSimMatrix . . . . .          | 9  |
| getSimScores . . . . .          | 11 |
| getTrimmedExternalSet . . . . . | 12 |
| getUNDmatrix . . . . .          | 13 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>15</b> |
|--------------|-----------|

---

|               |  |
|---------------|--|
| buildAIBSARNA | <i>Function to create a SummarizedExperiment containing BrainSpan Data</i> |
|---------------|--|

---

### Description

This function is used to build the AIBSARNA SummarizedExperiment object, and must be run prior to running any other function in **BrainSABER**. This function will download the data from <http://brainspan.org> and may take several minutes, depending on internet connection speeds.

### Usage

```
buildAIBSARNA(mini = FALSE)
```

### Arguments

|      |   |
|------|---|
| mini | Default is FALSE. If mini=TRUE, build a miniature version of AIBSARNA that does not require internet connectivity and is suitable for example purposes only |
|------|---|

### Value

A SummarizedExperiment containing BrainSpan data, with the addition of RefSeq IDs via biomaRt

### Examples

```
AIBSARNA <- buildAIBSARNA(mini = TRUE)
```

---

|              |                               |
|--------------|-------------------------------|
| CellScabbard | <i>The CellScabbard class</i> |
|--------------|-------------------------------|

---

### Description

The main class used by the **BrainSABER** package to hold single cell expression data, relevant gene data, and similarity matrices. CellScabbard extends the [SummarizedExperiment::SummarizedExperiment-class] class.

**Usage**

```

CellScabbard(
  exprsData,
  phenoData = NULL,
  featureData = NULL,
  AIBSARNA = NULL,
  autoTrim = TRUE
)

```

**Arguments**

|                          |   |
|--------------------------|---|
| <code>exprsData</code>   | expression data matrix for an experiment  |
| <code>phenoData</code>   | a data frame containing attributes of individual cells  |
| <code>featureData</code> | a data frame containing attributes of features (e.g. genes)   |
| <code>AIBSARNA</code>    | an instance of the AIBSARNA dataset, built using the <code>buildAIBSARNA()</code> function  |
| <code>autoTrim</code>    | If TRUE (default), automatically trim user data to match AIBSARNA using the column names of AIBSARNA and featureData (or rownames of exprsData, if featureData is not supplied) which produce the most matched identifiers. Also automatically fills the <code>relevantGenes</code> slot, using the same column names. The column names are stored in the <code>dataSetId</code> and <code>AIBSARNAid</code> slots. |

**Details**

This class is initialized from a matrix of gene expression values and associated metadata. Methods that operate on CellScabbard objects comprise the BrainSABER workflow.

**Value**

a new CellScabbard object

**Fields**

|                                 |   |
|---------------------------------|---|
| <code>dataSetId</code>          | A character vector of length 1, containing the column name of the user's data used to trim the data to match AIBSARNA.  |
| <code>AIBSARNAid</code>         | A character vector of length 1, containing the column name of AIBSARNA used to trim AIBSARNA to match the user's data   |
| <code>relevantGenes</code>      | A [SummarizedExperiment::SummarizedExperiment-class] containing a subset of data from AIBSARNA. Generated by the <b>BrainSABER</b> workflow.  |
| <code>similarityScores</code>   | A data.frame containing similarity scores. Generated by the <b>BrainSABER</b> workflow.   |
| <code>similarityDFs</code>      | A list containing similarity data frames with age, structure, and similarity scores sorted in decreasing order. Generated by the <b>BrainSABER</b> workflow.                        |
| <code>similarityMatrices</code> | A [SummarizedExperiment::Assays-class] object containing similarity matrices, with identical dimensions, for each sample in phenoData. Generated by the <b>BrainSABER</b> workflow. |
| <code>UNDMatrices</code>        | A [SummarizedExperiment::Assays-class] object containing identical-dimension UND matrices for each sample in phenoData. Generated by the <b>BrainSABER</b> workflow.                |

## Examples

```
# construct example data set
AIBSARNA <- buildAIBSARNA(mini = TRUE)

# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)

# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]

# construct a CellScabbard data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
                        AIBSARNA = AIBSARNA, autoTrim = TRUE)
```

---

CellScabbard-methods *Methods for the CellScabbard class*

---

## Description

These methods operate on CellScabbard objects. They are used to access the results of the **BrainS-ABER** workflow stored within a CellScabbard.

## Usage

```
dataSetId(cs)

dataSetId(cs) <- value

AIBSARNAid(cs)

AIBSARNAid(cs) <- value

relevantGenes(cs)

relevantGenes(cs) <- value

similarityScores(cs)

similarityScores(cs) <- value

similarityDFs(cs)

similarityDFs(cs) <- value

similarityMatrices(cs)

similarityMatrices(cs) <- value
```

```

UNDmatrices(cs)

UNDmatrices(cs) <- value

```

### Arguments

cs                    A CellScabbard object  
value                 data type, any of matrix, data.frame,list, or SimpleList

### Value

The contents of a slot of the CellScabbard object

### Examples

```

# construct example data set
AIBSARNA <- buildAIBSARNA(mini = TRUE)

# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)

# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]

# construct a CellScabbard data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
                       AIBSARNA = AIBSARNA, autoTrim = TRUE)
relevantGenes(myGenes)

# the following fields will be empty as output must be assigned to
# them first
similarityScores(myGenes)
similarityMatrices(myGenes)
similarityDFs(myGenes)
UNDmatrices(myGenes)

```

---

getExampleVector            *Get an example vector for specified genes*

---

### Description

This function returns a named example vector of gene expression values for the specified genes, taken from the 1st row of AIBSARNA, for use in demonstrating getSimScores.

### Usage

```
getExampleVector(genes, AIBSARNA = NULL)
```

**Arguments**

|                       |  |
|-----------------------|--|
| <code>genes</code>    | a character vector of HGNC-compliant gene names  |
| <code>AIBSARNA</code> | an instance of the AIBSARNA dataset, built using the <code>buildAIBSARNA()</code> function |

**Value**

a named character vector of gene-expression values

**Examples**

```
AIBSARNA <- buildAIBSARNA(mini = TRUE)
myGenes <- c("TSPAN6", "DPM1", "C1orf112")
myExampleVector <- getExampleVector(myGenes, AIBSARNA)
```

---

|                                |                          |
|--------------------------------|--------------------------|
| <code>getExternalVector</code> | <i>getExternalVector</i> |
|--------------------------------|--------------------------|

---

**Description**

Get a named vector of gene expression values from a single sample in an outside SummarizedExperiment, for use in creating subsets of AIBSARNA with `getRelevantGenes` and comparison with that subset with `getSimScores`

**Usage**

```
getExternalVector(
  dataSet,
  index = 1,
  AIBSARNA = NULL,
  dataSetId,
  AIBSARNAid = c("gene_id", "ensembl_gene_id", "gene_symbol", "entrez_id",
    "refseq_ids")
)
```

**Arguments**

|                         |   |
|-------------------------|---|
| <code>dataSet</code>    | a CellScabbard or SummarizedExperiment object   |
| <code>index</code>      | the integer index of the sample of <code>dataSet</code> to be used  |
| <code>AIBSARNA</code>   | an instance of the AIBSARNA dataset, built using the <code>buildAIBSARNA()</code> function  |
| <code>dataSetId</code>  | the name of the column of gene identifiers in <code>rowData(dataSet)</code> to be used to compare <code>dataSet</code> to AIBSARNA.   |
| <code>AIBSARNAid</code> | the name of the column of <code>rowData(AIBSARNA)</code> that is comparable to <code>dataSetId</code> . One of "gene_id", "ensembl_gene_id", "gene_symbol", "entrez_id", "refseq_ids" |

**Value**

a named vector of gene expression values

**Examples**

```

miniAIBSARNA <- buildAIBSARNA(mini = TRUE)
myGenes <- c(4.484885, 0.121902, 0.510035)
names(myGenes) <- c("TSPAN6", "DPM1", "C1orf112")
myGeneSet <- getRelevantGenes(myGenes, "gene_symbol", miniAIBSARNA,
  AIBSARNAid = "gene_symbol")
myGeneSampleVector <- getExternalVector(myGeneSet, index = 1, miniAIBSARNA,
  dataSetId = "gene_symbol", AIBSARNAid = "gene_symbol")

```

---

|                  |  |
|------------------|--|
| getRelevantGenes | <i>Get a subset of AIBSARNA using a Gene Expression Vector</i> |
|------------------|--|

---

**Description**

This function returns a subset of the AIBSARNA dataset, containing only the genes in data, which may be a vector, a SummarizedExperiment or derivative assay() and rowData(), or a CellScabbard. If a vector is used, it must consist of numerical gene expression values with names comparable to one column of identifiers present in AIBSARNA. If data is a CellScabbard, results are stored in the relevantGenes slot of the object.

**Usage**

```

getRelevantGenes(
  data,
  dataSetId = NULL,
  AIBSARNA = NULL,
  AIBSARNAid = c("gene_id", "ensembl_gene_id", "gene_symbol", "entrez_id",
    "refseq_ids")
)

```

**Arguments**

|            |  |
|------------|--|
| data       | a vector of named gene expression values, or a compatible data set   |
| dataSetId  | (Optional) If data is not a vector, the name of the column of gene identifiers in rowData(dataSet) to be used to compare data to AIBSARNA.                 |
| AIBSARNA   | an instance of the AIBSARNA dataset, built using the buildAIBSARNA() function  |
| AIBSARNAid | the name of the column of rowData(AIBSARNA) that is comparable to dataSetId. One of "gene_id", "ensembl_gene_id", "gene_symbol", "entrez_id", "refseq_ids" |

**Value**

a SummarizedExperiment consisting of genes in data, sorted to match the order of the genes in data

**Examples**

```

AIBSARNA <- buildAIBSARNA(mini = TRUE)
# Example 1 - using CellScabbard class
# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)
# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]
# build a trimmed data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
                        AIBSARNA = AIBSARNA, autoTrim = TRUE)
relevantGenes(myGenes)

# Example 2 - manual gene selection and relevant gene extraction
myGenes <- c(4.484885, 0.121902, 0.510035)
names(myGenes) <- c("TSPAN6", "DPM1", "C1orf112")
myGeneSet <- getRelevantGenes(myGenes, AIBSARNA = AIBSARNA,
                              AIBSARNAid = "gene_symbol")

```

---

getSimDataFrame

*Get Age, Structure Acronym, and Similarity Scores Data Frame*


---

**Description**

This function takes in a CellScabbard object, or both a similarity vector or data frame as returned by `getSimScores` and a subset of AIBSARNA as returned by `getRelevantGenes`. Constructs a data.frame with columns `age`, `structure_acronym`, and either `cosine_similarity` or `euclidean_similarity`, sorted by similarity score in decreasing order. In the case of a similarity data frame, a list of data frames is returned.

**Usage**

```

getSimDataFrame(
  data = NULL,
  sim_score = NULL,
  relevantGenes = NULL,
  similarity_method = "cosine"
)

```

**Arguments**

|                                |  |
|--------------------------------|--|
| <code>data</code>              | a CellScabbard object with non-empty <code>relevantGenes</code> and <code>similarityScores</code> slots, or a <code>SummarizedExperiment</code> created using the <code>getRelevantGenes()</code> function |
| <code>sim_score</code>         | a vector or data frame of similarity scores  |
| <code>relevantGenes</code>     | a <code>SummarizedExperiment</code> object created using the <code>getRelevantGenes()</code> function  |
| <code>similarity_method</code> | currently supported similarity methods are "cosine" and "euclidean", defaults to "cosine"  |



**Value**

a three-column data.frame or list of data frames

**Examples**

```
AIBSARNA <- buildAIBSARNA(mini = TRUE)
# Example 1 - using CellScabbard class
# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)
# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]
# build a trimmed data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
                        AIBSARNA = AIBSARNA, autoTrim = TRUE)
# cosine similarity method
similarityScores(myGenes) <- getSimScores(data = myGenes,
                                          similarity_method = "cosine")
similarityDFs(myGenes) <- getSimDataFrame(data = myGenes,
                                          similarity_method = "cosine")

similarityDFs(myGenes)
# euclidean similarity method
similarityScores(myGenes) <- getSimScores(data = myGenes,
                                          similarity_method = "euclidean")
similarityDFs(myGenes) <- getSimDataFrame(data = myGenes,
                                          similarity_method = "euclidean")

similarityDFs(myGenes)

# Example 2 - manual gene selection and relevant gene extraction
myGenes <- c(4.484885, 0.121902, 0.510035)
names(myGenes) <- c("TSPAN6", "DPM1", "C1orf112")
myGeneSet <- getRelevantGenes(myGenes, AIBSARNA = AIBSARNA,
                              AIBSARNAid = "gene_symbol")
myCosScore <- getSimScores(myGenes, relevantGenes = myGeneSet,
                           similarity_method = "cosine")
myEucScore <- getSimScores(myGenes, relevantGenes = myGeneSet,
                           similarity_method = "euclidean")
myCosineDF <- getSimDataFrame(sim_score = myCosScore,
                              relevantGenes = myGeneSet,
                              similarity_method = "cosine")
myEuclideanDF <- getSimDataFrame(sim_score = myEucScore,
                                 relevantGenes = myGeneSet,
                                 similarity_method = "euclidean")
```

---

getSimMatrix

*Get Age, Structure Acronym, and Similarity Score Matrix*


---

**Description**

This function takes a similarity vector or data frame as returned by `getSimScores` and either a `CellScabbard` object or a subset of `AIBSARNA` as returned by `getRelevantGenes`. Returns a



getSimScores

*Get Similarity Scoring for a Gene Expression Vector***Description**

This function computes the similarity score of a gene expression vector returned by `getExternalVector` or a trimmed data set returned by `getTrimmedExternalSet`, compared to a subset of AIBSARNA, obtained by `getRelevantGenes`.

**Usage**

```
getSimScores(data, relevantGenes = NULL, similarity_method = "cosine")
```

**Arguments**

`data` a named vector of gene expression values returned by `getExternalVector`, a `SummarizedExperiment` returned by `getTrimmedExternalSet`, or a `CellScabbard` object.

`relevantGenes` a `SummarizedExperiment` object created using the `getRelevantGenes()` function

`similarity_method` currently supported similarity methods are "cosine" and "euclidean", defaults to "cosine"

**Value**

If `data` is a vector, returns a vector of similarity scores for each sample in `relevantGenes`. If `data` is a `SummarizedExperiment`, returns a data frame, with columns containing the similarity scores for and named after each sample in `data`, and rows named after each sample in `relevantGenes`. If `data` is a `CellScabbard`, the results will be stored in its `similarityScores` slot.

**Examples**

```
AIBSARNA <- buildAIBSARNA(mini = TRUE)
# Example 1 - using CellScabbard class
# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)
# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]
# build a trimmed data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
  AIBSARNA = AIBSARNA, autoTrim = TRUE)
similarityScores(myGenes) <- getSimScores(data = myGenes,
  similarity_method = "cosine")

similarityScores(myGenes)
similarityScores(myGenes) <- getSimScores(data = myGenes,
  similarity_method = "euclidean")

similarityScores(myGenes)
```

```
# Example 2 - manual gene selection and relevant gene extraction
myGenes <- c(4.484885, 0.121902, 0.510035)
names(myGenes) <- c("TSPAN6", "DPM1", "C1orf112")
myGeneSet <- getRelevantGenes(myGenes, AIBSARNA = AIBSARNA,
  AIBSARNAid = "gene_symbol")
CosScores <- getSimScores(myGenes, myGeneSet,
  similarity_method = "cosine")
EucScores <- getSimScores(myGenes, myGeneSet,
  similarity_method = "euclidean")
```

---

getTrimmedExternalSet *Get a trimmed version of an external data set*

---

### Description

Returns a SummarizedExperiment that is a subset of dataSet containing only genes that are present in AIBSARNA, for use in getSimScores or getUNDmatrix.

### Usage

```
getTrimmedExternalSet(
  dataSet,
  dataSetId = "gene_symbol",
  AIBSARNA = NULL,
  AIBSARNAid = c("gene_id", "ensembl_gene_id", "gene_symbol", "entrez_id",
    "refseq_ids")
)
```

### Arguments

|            |  |
|------------|--|
| dataSet    | a CellScabbard or SummarizedExperiment object  |
| dataSetId  | the name of the column of gene identifiers in rowData(dataSet) to be used to compare dataSet to AIBSARNA.  |
| AIBSARNA   | an instance of the AIBSARNA dataset, built using the buildAIBSARNA() function  |
| AIBSARNAid | the name of the column of rowData(AIBSARNA) that is comparable to dataSetId. One of "gene_id", "ensembl_gene_id", "gene_symbol", "entrez_id", "refseq_ids" |

### Value

a SummarizedExperiment object containing trimmed data set

### Examples

```
AIBSARNA <- buildAIBSARNA(mini = TRUE)
# Example 1 - using CellScabbard class
# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)
# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
```

```

fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]
# build a trimmed data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
                       AIBSARNA = AIBSARNA, autoTrim = TRUE)
# use the appropriate id's to extract the trimmed gene set from the data
dataSetId = dataSetId(myGenes)
AIBSARNAid = AIBSARNAid(myGenes)
myTrimmedGeneSet <- getTrimmedExternalSet(myGenes,
                                          dataSetId = dataSetId, AIBSARNA, AIBSARNAid = AIBSARNAid)

# Example 2 - manual gene selection and relevant gene extraction
myGenes <- c(4.484885, 0.121902, 0.510035)
names(myGenes) <- c("TSPAN6", "DPM1", "C1orf112")
myGeneSet <- getRelevantGenes(myGenes, AIBSARNA = AIBSARNA,
                              AIBSARNAid = "gene_symbol")
myTrimmedGeneSet <- getTrimmedExternalSet(myGeneSet,
                                          dataSetId = "gene_symbol", AIBSARNA, AIBSARNAid = "gene_symbol")

```

---

getUNDMatrix

getUNDMatrix

---

## Description

This function returns a matrix showing whether gene expression values in `dataSet` are up-regulated, down-regulated, or normal. `method = "discrete"` will function on any `CellScabbard` object, while `method = "log2FC"` requires a trimmed data set as returned by `getTrimmedExternalSet` and a matching subset of `AIBSARNA` as returned by `getRelevantGenes`. Results are stored in the 'UNDMatrices' slot of the `dataSet` if it's a `CellScabbard` object.

## Usage

```

getUNDMatrix(
  dataSet,
  relevantGenes = NULL,
  method = c("discrete", "log2FC"),
  up_threshold = 0.5,
  down_threshold = -0.5,
  matrix_type = c("num", "char")
)

```

## Arguments

|                             |  |
|-----------------------------|--|
| <code>dataSet</code>        | a <code>CellScabbard</code> or <code>SummarizedExperiment</code> object  |
| <code>relevantGenes</code>  | (optional) a <code>SummarizedExperiment</code> and subset of <code>AIBSARNA</code>   |
| <code>method</code>         | "discrete" applies thresholds directly to expression data. "log2FC" applies thresholds to the log2 fold-change between the expression data of each sample from <code>dataSet</code> and <code>relevantGenes</code> . |
| <code>up_threshold</code>   | a numerical value defining the lower bound (inclusive) by which to consider a gene up-regulated, defaults to 0.5   |
| <code>down_threshold</code> | a numerical value defining the upper bound (inclusive) by which to consider a gene down-regulated, defaults to -0.5  |

`matrix_type` either "num" for a numerical matrix with -1 indicating down-regulation, 1 indicating up-regulation, and 0 indicating normal, or "char" for a character matrix with "D" indicating down-regulation, "U" indicating up-regulation, and "N" indicating normal

### Value

a list containing as many numerical or character matrices as samples in `dataSet`, with each matrix having as many rows as genes in `dataSet` and as many columns as samples in `relevantGenes`

### Examples

```
AIBSARNA <- buildAIBSARNA(mini = TRUE)
# Example 1 - using CellScabbard class
# get a random sample of 3 genes
totalGenes <- nrow(AIBSARNA)
gene_idx <- sample.int(totalGenes, 3)
sample_idx <- c(1,3,5)
# Subset AIBSARNA
exprs <- assay(AIBSARNA)[gene_idx, sample_idx]
fd <- rowData(AIBSARNA)[gene_idx, ]
pd <- colData(AIBSARNA)[sample_idx, ]
# build a trimmed data set
myGenes <- CellScabbard(exprsData = exprs, phenoData = pd, featureData = fd,
                        AIBSARNA = AIBSARNA, autoTrim = TRUE)
UNDmatrices(myGenes) <- getUNDmatrix(myGenes, method = "discrete",
                                     up_threshold = 3,
                                     down_threshold = 1, matrix_type = "char")
UNDmatrices(myGenes)
UNDmatrices(myGenes) <- getUNDmatrix(myGenes, method = "log2FC",
                                     up_threshold = 3,
                                     down_threshold = 1, matrix_type = "num")
UNDmatrices(myGenes)

# Example 2 - manual gene selection and relevant gene extraction
myGenes <- c(4.484885, 0.121902, 0.510035)
names(myGenes) <- c("TSPAN6", "DPM1", "C1orf112")
myGeneSet <- getRelevantGenes(myGenes, AIBSARNA = AIBSARNA,
                              AIBSARNAid = "gene_symbol")
myTrimmedGeneSet <- getTrimmedExternalSet(myGeneSet,
                                          dataSetId = "gene_symbol", AIBSARNA, AIBSARNAid = "gene_symbol")
myUNDnumericalMatrix <- getUNDmatrix(myTrimmedGeneSet, method = "discrete",
                                     up_threshold = 3, down_threshold = 1, matrix_type = "num")
myUNDcharacterMatrix <- getUNDmatrix(myTrimmedGeneSet, myGeneSet,
                                     method = "log2FC",
                                     up_threshold = 3, down_threshold = 1, matrix_type = "char")
```

# Index

.CellScabbard (CellScabbard), 2

AIBSARNAid (CellScabbard-methods), 4  
AIBSARNAid<- (CellScabbard-methods), 4

buildAIBSARNA, 2

CellScabbard, 2  
CellScabbard, ANY, ANY-method  
    (CellScabbard-methods), 4  
CellScabbard-class (CellScabbard), 2  
CellScabbard-methods, 4

dataSetId (CellScabbard-methods), 4  
dataSetId<- (CellScabbard-methods), 4

getExampleVector, 5  
getExternalVector, 6  
getRelevantGenes, 7  
getSimDataFrame, 8  
getSimMatrix, 9  
getSimScores, 11  
getTrimmedExternalSet, 12  
getUNDmatrix, 13

relevantGenes (CellScabbard-methods), 4  
relevantGenes<- (CellScabbard-methods),  
    4

similarityDFs (CellScabbard-methods), 4  
similarityDFs<- (CellScabbard-methods),  
    4

similarityMatrices  
    (CellScabbard-methods), 4  
similarityMatrices<-  
    (CellScabbard-methods), 4  
similarityScores  
    (CellScabbard-methods), 4  
similarityScores<-  
    (CellScabbard-methods), 4

UNDmatrices (CellScabbard-methods), 4  
UNDmatrices<- (CellScabbard-methods), 4