Package ‘BEAMR’

July 27, 2024

Title  Bootstrap Evaluation of Association Matrices

Version  1.1.0

Description  A bootstrap-based approach to integrate multiple forms of high dimensional genomic data with multiple clinical endpoints. This method is used to find clinically meaningful groups of genomic features, such as genes or pathways. A manuscript describing this method is in preparation.

License  GPL (>= 3)

Encoding  UTF-8

RoxygenNote  7.3.2

Imports  dplyr, ggmosaic, ggplot2, ggpubr, logistf, magrittr, MASS, purrr, rlist, stats, stringr, survival, survminer

Suggests  rmarkdown

Depends  R (>= 2.10)

LazyData  true

URL  https://annaseffernick.github.io/BEAMR/,
     https://github.com/annaSeffernick/BEAMR

BugReports  https://github.com/annaSeffernick/BEAMR/issues

NeedsCompilation  no

Author  Anna Eames Seffernick [aut, cre, cph]
        (https://orcid.org/0000-0003-0848-4604),
        Stanley Pounds [aut],
        Xueyuan Cao [aut]

Maintainer  Anna Eames Seffernick <anna.seffernick@stjude.org>

Repository  CRAN

Date/Publication  2024-07-27 16:00:06 UTC
beam.dat

Pediatric T-ALL Clinical Data from COG trial AALL0434

Description

The beam.data object used in example beam analyses

Usage

beam_dat
beam_dat_sm

Format

beam_dat:
A beam.data object, which is a list with the following elements:

main.data  A data.frame with clinical/endpoint data.
mtx.data   A list of the omics data matrices.
mtx.anns   A list of omic annotation data.frames.
anns.mtch  A data.frame with information to link mtx.data and mtx.anns.
set.data   A data.frame with set.id, mtx.id, and row.id to link omic features to sets.
set.anns   Optional data.frame with set annotation data.
boot.index A matrix with bootstrap indices.

Source

NA

---

beam_dat_sm  Pediatric T-ALL Clinical Data from COG trial AALL0434

Description

The smaller beam.data object used in the example for compute_beam_stats function

Usage

beam_dat_sm

Format

beam_dat_sm:
A beam.data object, which is a list with the following elements:

main.data  A data.frame with clinical/endpoint data.
mtx.data   A list of the omics data matrices.
mtx.anns   A list of omic annotation data.frames.
anns.mtch  A data.frame with information to link mtx.data and mtx.anns.
set.data   A data.frame with set.id, mtx.id, and row.id to link omic features to sets.
set.anns   Optional data.frame with set annotation data.
boot.index A matrix with bootstrap indices.

Source

NA
## beam_specs

**Description**

The `beam.specs` object used in example beam analyses.

**Usage**

```r
beam_specs
```

**Format**

<table>
<thead>
<tr>
<th><code>beam_specs</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>A data frame with 6 rows and 3 columns:</td>
</tr>
<tr>
<td><strong>name</strong> Analysis name with omic and endpoint</td>
</tr>
<tr>
<td><strong>mtx</strong> Name of omics matrix used in the analysis</td>
</tr>
<tr>
<td><strong>mdl</strong> Regression model</td>
</tr>
</tbody>
</table>

**Source**

NA

## beam_specs_sm

**Description**

The small `beam.specs` object used in example `compute_beam_stats` function.

**Usage**

```r
beam_specs_sm
```

**Format**

<table>
<thead>
<tr>
<th><code>beam_specs_sm</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>A data frame with 2 rows and 3 columns:</td>
</tr>
<tr>
<td><strong>name</strong> Analysis name with omic and endpoint</td>
</tr>
<tr>
<td><strong>mtx</strong> Name of omics matrix used in the analysis</td>
</tr>
<tr>
<td><strong>mdl</strong> Regression model</td>
</tr>
</tbody>
</table>

**Source**

NA
# beam_stats

**Description**

The beam.stats object used in example beam analyses

**Usage**

```r
beam_stats
```

**Format**

```r
beam_stats:
  A beam.stats object, which contains the following objects
  beam.stats A list of data.frames of association statistics for each omic-endpoint pair.
  beam.specs A beam.specs object (data.frame with name, mtx, and mdl.)
  beam.data The beam.data object.
```

**Source**

NA

---

# beam_stats_sm

**Description**

The small beam.stats object used in example for compute_beam_stats function.

**Usage**

```r
beam_stats_sm
```

**Format**

```r
beam_stats_sm:
  A beam.stats object, which contains the following objects
  beam.stats A list of data.frames of association statistics for each omic-endpoint pair.
  beam.specs A beam.specs object (data.frame with name, mtx, and mdl.)
  beam.data The beam.data object.
```

**Source**

NA
**check_beam_specs**

*Check that beam.specs satisfies all necessary conditions*

**Description**

Check that beam.specs satisfies all necessary conditions

**Usage**

```r
check_beam_specs(beam.specs, mtx.names)
```

**Arguments**

- `beam.specs` A data.frame with column name, mtx, and mdl
- `mtx.names` A vector with the names of the data matrices (beam.data$mtx.data)

**Value**

A data.frame of beam.specs if all conditions satisfied, otherwise throws an error

**Examples**

```r
data(beam_dat)
data(beam_specs)
test_specs <- check_beam_specs(beam_specs, names(beam_dat$mtx.data))
```

**check_list_class**

*Check that each element of a list is of a required class*

**Description**

Check that each element of a list is of a required class

**Usage**

```r
check_list_class(list.object, required.class)
```

**Arguments**

- `list.object` A list used in BEAMR analysis
- `required.class` Class for list elements, e.g. matrix

**Value**

Logical TRUE if list is of required class
**clean_Bmtx**

**Examples**

```r
data(omicdat)
check_list_class(omicdat, "matrix")
```

---

**clinf**

*Pediatric T-ALL Clinical Data from COG trial AALL0434*

**Description**

A subset of clinical data from pediatric and young adult T-lineage acute lymphoblastic leukemia patients in the Children’s Oncology Group trial AALL0434, published in Liu et al., 2017 Nature Genetics

**Usage**

```r
clinf
```
compute_beam_stats

Format

  clinf:
  A data frame with 265 rows and 8 columns:
  ID   Subject ID
  MRD29 Minimal residual disease measured at day 29
  RNA.clm Key to match to RNA matrix
  Lesion.clm Key to match Lesion matrix
  Lesion.id Key to match Lesion matrix
  RNA.id  Key to match RNA matrix
  EFS   Event-free survival Surv object
  OS    Overall survival Surv object

Source

  https://www.nature.com/articles/ng.3909

compute_beam_stats  Compute bootstrap model coefficients for BEAM

Description

  Compute bootstrap model coefficients for BEAM

Usage

  compute_beam_stats(beam.data, beam.specs, stdize = TRUE)

Arguments

  beam.data  Result of prep.beam.data
  beam.specs A data.frame of strings with columns name, mtx, mdl (string with R model with mtx.row)
  stdize     Logical whether to standardize (center and scale) predictors or not. Default is TRUE.

Value

  A beam.stats object, which is a list with beam.stats (the association matrices), the beam.specs, and the beam.data

Examples

  data(beam_dat_sm)
  data(beam_specs_sm)
  test.beam.stats <- compute_beam_stats(beam.data=beam_dat_sm,
                                         beam.specs=beam_specs_sm, stdize=TRUE)
compute_feature_pvalues

*Compute feature level p-values from BEAM statistics*

**Description**

Compute feature level p-values from BEAM statistics

**Usage**

```r
compute_feature_pvalues(beam.stats)
```

**Arguments**

- `beam.stats`: A `beam.stats` object, which is a list with `beam.stats` (the association matrices), the `beam.specs`, and the `beam.data`

**Value**

A list of feature level p-values, with each entry a data frame for a different omics/endpoint association, with columns id, gene, beta, p, q

**Examples**

```r
data(beam_stats)
test.feat.pvals <- compute_feature_pvalues(beam.stats=beam_stats)
```

compute_set_pvalues

*Compute BEAMR p-values for sets*

**Description**

Compute BEAMR p-values for sets

**Usage**

```r
compute_set_pvalues(beam.stats, peel = FALSE, z = TRUE, alpha = 0.1, mess.freq = 25)
```
extend_set_data

Arguments

beam.stats  A beam.stats object from compute_beam_stats function
peel        Logical indicating whether to peel in p-value calculation
z           Logical indicating whether to z-scale each vector of one coefficient estimate
            across bootstraps before analysis
alpha       Maximum depth to peel (reduces computing time); default 0.1.
mess.freq   Message frequency; default 25.

Value

A list with a data.frame of set p-values from BEAMR analysis, a data.frame of summary row p-
values, and a data frame of set matching.

Examples

data(beam_stats_sm)
test.pvals <- compute_set_pvalues(beam.stats=beam_stats_sm)

extend_set_data

Extend set definition data with genes on the same row separated by
commas, semicolons, slashes, etc

Description

Extend set definition data with genes on the same row separated by commas, semicolons, slashes, etc

Usage

extend_set_data(set.data, sep)

Arguments

set.data  A data frame with set definition data.
sep       Punctuation to split on.

Value

A data frame.

Examples

data(setdat)
extend_set_data(setdat, sep="","
**extract_beam_stats**  
*Extract beam stats for a specific set*

**Description**
Extract beam stats for a specific set

**Usage**
```r
extract_beam_stats(beam.stats, set.id)
```

**Arguments**
- `beam.stats`: A beam.stats object, which is a list with beam.stats (the association matrices), the beam.specs, and the beam.data
- `set.id`: A character of a set id name (an entry in in beam.data$set.data$set.id)

**Value**
A matrix with with estimated associations for each endpoint and each omic feature linked to the set

**Examples**
```r
data(beam_stats)
test.stats <- extract_beam_stats(beam_stats, set.id="ENSG00000099810")
```

**find_id_clm**  
*Find the column of mtch.data with the most rows containing an element of ids*

**Description**
Find the column of mtch.data with the most rows containing an element of ids

**Usage**
```r
find_id_clm(mtch.data, ids)
```

**Arguments**
- `mtch.data`: A data.frame
- `ids`: A vector of row ids to match

**Value**
A vector of column names with the most matches.
Examples

```r
data(omicann)
data(omicdat)
lsn.data <- omicann[[1]]
mtx.rows <- rownames(omicdat[[1]])
test <- find_id_clm(lsn.data, mtx.rows)
```

---

**gen_beam_plot_list**  
*Generate BEAM Plot List*

**Description**

Internal function: generate a list of clinical feature plots.

**Usage**

```r
gen_beam_plot_list(
  beam.result,
  beam.specs,
  beam.feat.pvals,
  number.pairs = 1,
  set.id,
  feat.id = NULL,
  title.size = 10,
  pair.order = "both",
  endpt.order = NULL
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>beam.result</td>
<td>Result of prep.beam.data</td>
</tr>
<tr>
<td>beam.specs</td>
<td>A data.frame of strings with columns name, mtx, mdl, plot</td>
</tr>
<tr>
<td>beam.feat.pvals</td>
<td>List of feature-level p-values from compute_feature_pvalues</td>
</tr>
<tr>
<td>number.pairs</td>
<td>Numeric; number of features to display in clinical plots, ordered by significance</td>
</tr>
<tr>
<td>set.id</td>
<td>A character with set name; must be in beam.result$beam.data$set.data$set.id</td>
</tr>
<tr>
<td>feat.id</td>
<td>Default NULL; a character with feature name; must be in beam.result$beam.data$set.data$row.id</td>
</tr>
<tr>
<td>title.size</td>
<td>A numeric. Specify the size of individual plot titles. Default is 10.</td>
</tr>
<tr>
<td>pair.order</td>
<td>One of c(&quot;both&quot;, &quot;omic&quot;, &quot;endpoint&quot;). Default is &quot;both.&quot; Specify how to choose feature-endpoint plots to include. If &quot;both&quot;, find the best (based on q, p, effect size) feature-omic pair for each type of omic and each endpoint separately. If &quot;omic&quot;, within each omic, find the best feature-endpoint pair and then plot this feature with all endpoints. If &quot;endpoint&quot;, need to specify endpt.order as the name of chosen endpoint. Then, within each omic, find the feature with best association with the selected endpoint, and plot this feature for all endpoints.</td>
</tr>
<tr>
<td>endpt.order</td>
<td>Default NULL. If pair.order=&quot;endpoint&quot;, specify character with endpoint name (from beam.specs$name, after the period).</td>
</tr>
</tbody>
</table>
Value

A list of plots for the specified set and/or feature.

Examples

```r
data(beam_stats)
test.feat.pvals <- compute_feature_pvalues(beam.stats=beam_stats)
plot.specs <- prep_beam_plot(beam.data=beam_stats$beam.data,
                             beam.specs=beam_stats$beam.specs)
plot.list <- gen_beam_plot_list(beam.result=beam_stats, beam.specs=plot.specs,
                                 beam.feat.pvals=test.feat.pvals,
                                 number.pairs=1, set.id="ENSG00000099810",
                                 feat.id=NULL, title.size=11,
                                 pair.order="omic", endpt.order=NULL)
```

get_id_index

For each row of the data.frame main.data, find the index of the matching element in vector ids

Description

For each row of the data.frame main.data, find the index of the matching element in vector ids

Usage

```r
get_id_index(mtch.data, ids, warn = TRUE)
```

Arguments

- **mtch.data**: A data.frame to be linked with the ids
- **ids**: A vector of ids to be linked in mtch.data
- **warn**: A logical value whether to include warnings with results

Value

A data.frame with matching id index

Examples

```r
data(clinf)
data(omicdat)
mtx.clms <- colnames(omicdat[[1]])
id_index <- get_id_index(clinf, mtx.clms)
```
omicann  Pediatric T-ALL Omics Annotation Data from COG trial AALL0434

Description
A subset of genomic lesion and RNA expression data from pediatric and young adult t-lineage acute lymphoblastic leukemia patients in the Children’s Oncology Group trial AALL0434, published in Liu et al., 2017 Nature Genetics. This is the annotation mapping feature id to gene name given by Ensembl ID.

Usage
omicann

Format
omicann:
A list with two data frames of omics annotation.

Lesion  A dataframe with 20 rows and 2 columns with lesion ID and Ensembl ID.
RNA     A dataframe with 20 rows and 2 columns with feature ID and Ensembl ID.

Source
https://www.nature.com/articles/ng.3909

omicdat  Pediatric T-ALL Omics Data from COG trial AALL0434

Description
A subset of genomic lesion and RNA expression data from pediatric and young adult t-lineage acute lymphoblastic leukemia patients in the Children’s Oncology Group trial AALL0434, published in Liu et al., 2017 Nature Genetics

Usage
omicdat

Format
omicdat:
A list with two data frames of omic data for each subject

Lesion  A dataframe with 20 rows and 265 columns indicating presence of lesion.
RNA     A dataframe with 20 rows and 265 columns with expression data.
plot_beam_boot

Source

https://www.nature.com/articles/ng.3909

Description

plot_beam_boot produces a pairs plot of the beam stats matrices. Default is maximum of 5 plots, ordered by most significant association direction.

Usage

plot_beam_boot(
  beam.result,
  beam.feat.pvals,
  beam.specs = NULL,
  set.id,
  max.plots = 4,
  z = TRUE
)

Arguments

beam.result A beam.stats object from compute_beam_stats
beam.feat.pvals A list containing feature-level p-values from compute_feature_pvalues.
beam.specs A data.frame. Default NULL, in which case beam.result$beam.specs is used. Otherwise can input other beam.specs data.frame that must contain name, mtx, mdl, plot columns.
set.id A character specifying the name of a set. Must be in beam.result$beam.data$set.data
max.plots A number specifying the max number of rows in the pairs plot. Default is 4, ordered by feature-level p-value.
z Logical indicating whether to z-scale each vector of one coefficient estimate across bootstraps before plotting. Default is TRUE.

Value

A pairs plot figure.

Examples

data(beam_stats)
test.pvals <- compute_set_pvalues(beam.stats=beam_stats)
test.feat.pvals <- compute_feature_pvalues(beam.stats=beam_stats)
test.boot.plot <- plot_beam_boot(beam_stats, test.feat.pvals,
  set.id="ENSG00000099810")
plot_beam_clin produces a matrix of feature level clinical plots for a set. Users can specify which omic/endpoint pairs they want to see as well as the number of features from the set. Default is all omic/endpoint pairs and the top feature (smallest feature-level p-value).

Usage

plot_beam_clin(
  beam.result,  
  beam.specs = NULL,  
  beam.set.pvals,  
  beam.feat.pvals,  
  set.id,  
  gene.name = NULL,  
  pair.type = NULL,  
  number.pairs = 1,  
  pair.order = "both",  
  endpt.order = NULL,  
  n.col = NULL,  
  n.row = NULL,  
  title.size = 10
)

Arguments

beam.result A beam.stats object from compute_beam_stats
beam.specs A data.frame. Default NULL, in which case beam.result$beam.specs is used. Otherwise can input other beam.specs data.frame that must contain name, mtx, mdl, plot columns.
beam.set.pvals A list containing BEAMR set p-values from compute_set_pvalues.
beam.feat.pvals A list containing feature-level p-values from compute_feature_pvalues.
set.id A character specifying the name of a set. Must be in beam.result$beam.data$set.data
gene.name A character specifying a Gene Name/Symbol for the set. Default is NULL
pair.type A character vector. Default NULL, in which case clinical plots for all omic/endpoint pairs are produced. Otherwise specify pairs from beam.stats$beam.specs$name
number.pairs A numeric. Default 1, in which case only feature with best simple test for each pair is plotted. If >1, show top n simple plots ordered by feature-level p-value
pair.order One of c("both", "omic", "endpoint"). Default is "both." Specify how to choose feature-endpoint plots to include. If "both", find the best (based on q, p, effect
plot_feat_clin

Description

plot_feat_clin produces a matrix of feature level clinical plots for a specific feature.

Usage

```r
plot_feat_clin(
  feat.id,
  beam.result, = NULL,
  beam.specs = NULL,
  beam.set.pvals,
  beam.feat.pvals,
  endpt.order,
  n.col,
  n.row,
  title.size
)
```
prep_beam_data

prep_beam_data


def prep_beam_data(beam.data, beam.specs = NULL, beam.set.pvals = NULL, beam.feat.pvals = NULL, feat.id, n.row = NULL, n.col = NULL)
{
  ...
}

Arguments

feat.id A character specifying the name of a feature. Must be in beam.result$beam.data$set.data
beam.result A beam.stats object from compute_beam_stats
beam.specs A data.frame. Default NULL, in which case beam.result$beam.specs is used. Otherwise can input other beam.specs data.frame that must contain name, mtx, mdl, plot columns.
beam.set.pvals A list containing BEAMR set p-values from compute_set_pvalues.
beam.feat.pvals A list containing feature-level p-values from compute_feature_pvalues.
n.row A numeric. Specify the number of rows for the plot layout; default NULL will automatically define the number of rows after number of columns specified.
n.col A numeric. Specify the number of columns for the plot layout; default NULL will use the number of omics types.

Value

A figure (ggarrange object)

Examples

data(beam_stats)
test.pvals <- compute_set_pvalues(beam.stats=beam_stats)
test.feat.pvals <- compute_feature_pvalues(beam.stats=beam_stats)
plot.specs <- prep_beam_plot(beam.data=beam.stats$beam.data, beam.specs=beam.stats$beam.specs)
test.plot <- plot_feat_clin(beam.result=beam_stats, beam.specs=plot.specs, beam.set.pvals=test.pvals, beam.feat.pvals=test.feat.pvals, feat.id="ENSG00000227443_loss", n.col=2, n.row=NULL)

prep_beam_data Prepare data for BEAM analysis

Description

Prepare data for BEAM analysis
**Usage**

```r
prep_beam_data(
  main.data,
  mtx.data,
  mtx.anns = NULL,
  set.data = NULL,
  set.anns = NULL,
  n.boot = 1000,
  seed = NULL
)
```

**Arguments**

- `main.data`: A data.frame
- `mtx.data`: A list, each element is a matrix
- `mtx.anns`: A list, each element is a data.frame
- `set.data`: A data.frame with columns `set.id`, `mtx.id`, `row.id`
- `set.anns`: A data frame with `set.id` and other columns
- `n.boot`: Number of bootstraps
- `seed`: Initial seed for random number generation

**Value**

A `beam.data` object, which is a list with `main.data`, `mtx.data`, `mtx.anns`, `anns.mtch`, `set.data`, `set.anns`, and `boot.index`

**Examples**

```r
data(clinf)
data(omicdat)
data(omicann)
data(setdat)
test.beam.data <- prep_beam_data(main.data=clinf, mtx.data=omicdat,
                                 mtx.anns=omicann, set.data=setdat,
                                 set.anns=NULL, n.boot=10, seed=123)
```

---

**prep_beam_plot**

Prepare for BEAM plotting

**Description**

Add a "plot" column to `beam.specs`, which includes string of plot commands.

**Usage**

```r
prep_beam_plot(beam.data, beam.specs)
```
prep_beam_specs

Arguments

beam.data         Result of prep.beam.data
beam.specs        A data.frame of strings with columns name, mtx, mdl (string with R model with mtx.row)

Value

An updated beam.specs object that includes the column "plot"

Examples

data(clinf)
data(omicdat)
data(omicann)
data(setdat)
test.beam.data <- prep_beam_data(main.data=clinf, mtx.data=omicdat,
                                 mtx.anns=omicann, set.data=setdat,
                                 set.anns=NULL, n.boot=10, seed=123)
specs <- prep_beam_specs(beam.data=test.beam.data, endpts=c("MRD29", "EFS", "OS"),
                         firth=TRUE)
plot.specs <- prep_beam_plot(beam.data=test.beam.data, beam.specs=specs)

Description

Prepare the beam.specs data.frame for BEAM model fitting. Specifies the univariate models needed to compute the BEAMR set p-values.

Usage

prep_beam_specs(
  beam.data,
  endpts,
  firth = TRUE,
  adjvars = NULL,
  endptmdl = NULL
)

Arguments

beam.data         A beam.data object from prep_beam_data
endpts            A vector of endpoint variable names in main.data
firth             A logical value. If TRUE (default) fit Firth penalized Cox model to account for monotone likelihood in the presence of rare events or predictors. If FALSE fit usual Cox model.
adjvars  Default NULL, optional vector of adjustment variable names in main.data
endptmdl  Optional model specification data.frame with endpoint name column called "endpt" and model string column called "mdl"

Value

The beam.specs object, a data.frame specifying the omics-endpoint association models to be fit

Examples

data(clinf)
data(omicdat)
data(omicann)
data(setdat)
test.beam.data <- prep_beam_data(main.data=clinf, mtx.data=omicdat, mtx.anns=omicann, set.data=setdat, set.anns=NULL, n.boot=10, seed=123)
  #Without adjustment
  prep_beam_specs(beam.data=test.beam.data, endpts=c("MRD29", "OS", "EFS"), firth=TRUE)
  # With adjustment
  prep_beam_specs(beam.data=test.beam.data, endpts=c("OS", "EFS"), adjvars=c("MRD29"), firth=TRUE)

print.beam.data  Print summary information about a beam.data object

Description

Print summary information about a beam.data object

Usage

## S3 method for class 'beam.data'
print(x, ...)

Arguments

x  An object of class "beam.data"
...
Other arguments passed to or from other methods

Value

Messages about the beam.data object

Examples

data(beam_dat)
print(beam_dat)
print.beam.stats  

Print summary information about beam.stats object

Description

Print summary information about beam.stats object

Usage

```r
## S3 method for class 'beam.stats'
print(x, ...)
```

Arguments

- `x` An object of class "beam.stats"
- `...` Other arguments passed to or from other methods

Value

Messages about the beam.data object

Examples

```r
data(beam_stats)
print(beam_stats)
```

setdat  

Map of Pediatric Data from COG trial AALL0434

Description

Map between annotation and omic data for a subset of clinical data from pediatric and young adult t-lineage acute lymphoblastic leukemia patients in the Children’s Oncology Group trial AALL0434, published in Liu et al., 2017 Nature Genetics

Usage

setdat

Format

- `setdat`: A data frame with 40 rows and 3 columns
- `set.id` Ensembl ID that defines gene-feature set
- `mtx.id` Name of omic matrix where corresponding feature data can be found
- `row.id` Feature name in corresponding omic matrix
### specs

**Source**

[https://www.nature.com/articles/ng.3909](https://www.nature.com/articles/ng.3909)

<table>
<thead>
<tr>
<th>specs</th>
<th>Pediatric T-ALL BEAMR Analysis Specs Data from COG trial AALL0434</th>
</tr>
</thead>
</table>

**Description**

The beam.specs object used in example beam analyses.

**Usage**

```r
specs
```

**Format**

```r
specs:
A data frame with 6 rows and 3 columns:

- **name**: Analysis name with omic and endpoint
- **mtx**: Name of omics matrix used in the analysis
- **mdl**: Regression model
```

**Source**

NA

---

<table>
<thead>
<tr>
<th>subset_beam_result</th>
<th>Subset beam.stats Result</th>
</tr>
</thead>
</table>

**Description**

Filter the beam.stats object from compute_beam_stats with various filtering criteria. Default is to filter to top 50 sets with smallest q-value. At least one filtering criteria must be specified. Can also use intersection or union of multiple criteria.
Usage

subset_beam_result(
  beam.result,
  beam.set.pvals = NULL,
  beam.feat.pvals = NULL,
  mtx.rows = NULL,
  set.ids = NULL,
  endpts = NULL,
  omics = NULL,
  p.limit = NULL,
  q.limit = NULL,
  p.feat.limit = NULL,
  q.feat.limit = NULL,
  intersect = TRUE,
  recalc = FALSE
)

Arguments

beam.result  A beam.stats object from compute_beam_stats
beam.set.pvals  A list containing BEAMR set p-values from compute_set_pvalues; required if p.limit or q.limit are specified.
beam.feat.pvals  A list containing feature-level p-values from compute_feature_pvalues; required if p.feat.limit or q.feat.limit are specified.
mtx.rows  A list of vectors of feature names corresponding to row.id in set.data. List names correspond to mtx.id in set.data. If specified, filter to all sets containing at least one of these features.
set.ids  A character vector of set.ids. If specified, filter to these sets.
endpts  A character vector of endpoint names. If specified, filter to sets that correspond to these endpoints.
omics  A character vector of omics names. If specified, filter to sets that correspond to these omics.
p.limit  A numeric value. If specified, determine mtx.rows that are below this threshold if p<1 or top p sets if p>1.
q.limit  A numeric value. If specified, determine mtx.rows that are below this threshold if q<1 or top q sets if q>1.
p.feat.limit  A numeric value. If specified, determine mtx.rows that are below this threshold if p.feat<1 or top p.feat sets if p.feat>1 (feature p-values).
q.feat.limit  A numeric value. If specified, determine mtx.rows that are below this threshold if q.feat<1 or top q.feat sets if q.feat>1.
intersect  A logical value. Default is TRUE. If TRUE, use intersection of all specified criteria. If FALSE use union of all specified criteria.
recalc  A logical value. Default is FALSE. If TRUE, recalculate p-values. If FALSE use original set p-values.
**Value**

A list with filtered beam.stats object, updated beam.set.pvals, and filtered beam.feat.pvals.

**Examples**

```r
data(beam_stats)
test.pvals <- compute_set_pvalues(beam.stats=beam_stats)
test.feat.pvals <- compute_feature_pvalues(beam.stats=beam_stats)
filt.beam.stats <- subset_beam_result(beam_stats, test.pvals, test.feat.pvals,
    endpts=c("EFS","OS"), q.limit=10, intersect=TRUE, recalc=FALSE)
```
Index

* datasets
  - beam_dat, 2
  - beam_dat_sm, 3
  - beam_specs, 4
  - beam_specs_sm, 4
  - beam_stats, 5
  - beam_stats_sm, 5
  - clinf, 7
  - omicann, 14
  - omicdat, 14
  - setdat, 22
  - specs, 23

  beam_dat, 2
  beam_dat_sm, 3
  beam_specs, 4
  beam_specs_sm, 4
  beam_stats, 5
  beam_stats_sm, 5

  check_beam_specs, 6
  check_list_class, 6
  clean_Bmtx, 7
  clinf, 7
  compute_beam_stats, 8
  compute_feature_pvalues, 9
  compute_set_pvalues, 9

  extend_set_data, 10
  extract_beam_stats, 11

  find_id_clm, 11

  gen_beam_plot_list, 12
  get_id_index, 13

  omicann, 14
  omicdat, 14

  plot_beam_boot, 15
  plot_beam_clin, 16
  plot_feat_clin, 17
  prep_beam_data, 18
  prep_beam_plot, 19
  prep_beam_specs, 20
  print.beam.data, 21
  print.beam.stats, 22
  setdat, 22
  specs, 23
  subset_beam_result, 23
  print.beam.data, 21
  print.beam.stats, 22

26