

Package ‘SeqFeatR’

April 12, 2019

Type Package

Title A Tool to Associate FASTA Sequences and Features

Version 0.3.1

Date 2019-10-04

Author Bettina Budeus

Maintainer Bettina Budeus <bettina.budeus@stud.uni-due.de>

Depends R(>= 3.2.2),tcltk,tcltk2,Biostrings, plotrix

Imports plyr, phangorn, qvalue, widgetTools, calibrate, ggplot2,
R2jags, coda, scales, ape

Suggests xtable

Description

Provides user friendly methods for the identification of sequence patterns that are statistically significantly associated with a property of the sequence. For instance, SeqFeatR allows to identify viral immune escape mutations for hosts of given HLA types. The underlying statistical method is Fisher's exact test, with appropriate corrections for multiple testing, or Bayes. Patterns may be point mutations or n-tuple of mutations. SeqFeatR offers several ways to visualize the results of the statistical analyses, see Budeus (2016) <doi:10.1371/journal.pone.0146409>.

License GPL (>= 3)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-12 12:02:37 UTC

R topics documented:

assocpair	2
assocpairfeat	4
assocpoint	6
assocpointhierarchical	9
assocpointpair	13
assotuple	15
comparewithancestral	17

foundereffectfinder	19
getfreqs	20
orPlot	22
q-values	24
rewritetuple	25
SeqFeatRGUI	26
smallmanhattan	27
tartan	28
visualizepair	30
visualizepairfeat	31
volcanoplot	32

Index	34
--------------	-----------

assocpair	<i>Finding pairs of alignment positions that jointly mutate</i>
-----------	---

Description

Determines pairs of sequence alignment positions that mutate in a correlated fashion with respect to a consensus sequence.

Usage

```
assocpair(path_to_file_sequence_alignment = NULL,
          path_to_file_consensus = NULL, save_name_csv, dna = FALSE,
          significance_level = 0.05, multiple_testing_correction = "bonferroni")
```

Arguments

`path_to_file_sequence_alignment`
FASTA file with sequence alignment. See example file.

`path_to_file_consensus`
FASTA file with consensus sequence. See example file.

`save_name_csv` name of file to which results are written in csv format.

`dna` indicates whether sequences are DNA or amino acids.

`significance_level`
significance level for Fisher's exact test.

`multiple_testing_correction`
multiple testing correction applied to p-values. Input can be: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

Details

For every position in the sequence alignment from the FASTA file a Fisher's exact test is applied with every other position in the sequence to check whether at both positions we have correlated mutations with respect to a given consensus sequence. Significant p-values are collected in one big table. `p.adjust` from `stats` package is used for multiple testing correction; corrected values are given as extra column in csv output.

In contrast to `assocpairfeat`, `assocpair` does not use features, but uses a consensus approach. Please be sure, that this is really what you want to use. Otherwise, use `assocpairfeat` or `assoctuple` instead.

Value

A csv file with every possible co-mutation below the given p-value.

Note

For graphical output use:
[visualizepair](#).

Author(s)

Bettina Budeus

See Also

[visualizepairfeat](#), [assocpairfeat](#), [assoctuple](#)

Examples

```
#Input files
## Not run:
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
consensus_input <- system.file("extdata", "Example_Consensus_aa.fasta", package="SeqFeatR")

#Usage
assocpair(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_consensus=consensus_input,
  save_name_csv="assocpair_results.csv",
  dna=FALSE,
  significance_level=0.05,
  multiple_testing_correction="bonferroni")

## End(Not run)
```

assocpairfeat	<i>Finding pairs of sequence alignment positions associated with sequence feature.</i>
---------------	--

Description

Determines pairs of alignment positions that jointly mutate depending on sequence feature.

Usage

```
assocpairfeat(path_to_file_sequence_alignment = NULL, save_name_csv, dna = FALSE,
              patnum_threshold = 1, significance_level = 0.05,
              A11a, A12a, A21a, A22a, B11a, B12a, B21a, B22a,
              multiple_testing_correction = "bonferroni")
```

Arguments

path_to_file_sequence_alignment	FASTA file with sequence alignment. See example file.
save_name_csv	name of file to which results are written in csv format.
dna	DNA or amino acid sequences.
patnum_threshold	minimum number of patients per HLA type to consider in calculation.
significance_level	significance level in Fisher's exact test.
A11a	position of start of first HLA A allele in header line of FASTA file.
A12a	position of end of first HLA A allele in header line of FASTA file.
A21a	position of start of second HLA A allele in header line of FASTA file.
A22a	position of end of second HLA A allele in header line of FASTA file.
B11a	position of start of first HLA B allele in header line of FASTA file.
B12a	position of end of first HLA B allele in header line of FASTA file.
B21a	position of start of second HLA B allele in header line of FASTA file.
B22a	position of end of second HLA B allele in header line of FASTA file.
multiple_testing_correction	multiple testing correction applied to p-values. Input can be: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

Details

Features may be HLA types, indicated by four blocks in the FASTA comment lines. The positions of these blocks in the comment lines are defined by parameters A11, ..., B22. For patients with a homozygous HLA allele the second allele has to be "00" (without the double quotes).

For every position in the sequence alignment from the FASTA file a Fisher's exact test is applied with every other position in the sequence and every HLA-type. Significant p-values are collected in

one big table. `p.adjust` from `stats` package is used for multiple testing correction; corrected values are given as extra column in csv output.

In contrast to `assocpair`, `assocpairfeat` uses an analysis with 'features' without a consensus sequence.

Value

Table with all alignment position pairs with significant association with sequence feature.

Note

Use [visualizepairfeat](#) for graphical output.

Author(s)

Bettina Budeus

See Also

[visualizepairfeat](#), [assocpair](#)

Examples

```
#Input file
## Not run:
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")

#Usage
assocpairfeat(
  path_to_file_sequence_alignment=fasta_input,
  save_name_csv="assocpairfeat_results.csv",
  dna=FALSE,
  patnum_threshold=1,
  significance_level=0.05,
  A11=10,
  A12=11,
  A21=13,
  A22=14,
  B11=17,
  B12=18,
  B21=20,
  B22=21,
  multiple_testing_correction="bonferroni")

## End(Not run)
```

 assocpoint

Searches for associations of single alignment positions with feature(s)

Description

Searches for associations of single nucleotide or amino acid sequence alignment positions with feature(s).

Usage

```
assocpoint(path_to_file_sequence_alignment = NULL,
  path_to_file_known_epitopes = NULL, path_to_file_binding_motifs = NULL, save_name_pdf,
  save_name_csv, dna = FALSE,
  patnum_threshold = 1, optical_significance_level = 0.05,
  star_significance_level = 0.001, A11, A12, A21, A22, B11, B12, B21,
  B22, C11, C12, C21, C22, has_C=FALSE, multiple_testing_correction = "bonferroni",
  bayes_factor = FALSE, constant_dirichlet_precision_parameter=FALSE,
  dirichlet_precision_parameter=20, phylo_bias_check = FALSE,
  path_to_file_reference_sequence=NULL, one_feature=FALSE,
  window_size=9, epi_plot=FALSE)
```

Arguments

path_to_file_sequence_alignment	FASTA file with sequence alignment. For reference see example file.
path_to_file_known_epitopes	csv file with known epitopes. See example file.
path_to_file_binding_motifs	csv file with HLA binding motifs if available. See example file.
save_name_pdf	name of file to which results are saved in pdf format.
save_name_csv	name of file to which results are saved in csv format.
dna	DNA or amino acid sequences.
patnum_threshold	minimum number of patients per HLA type to consider in calculation.
optical_significance_level	height of horizontal line in graphical output indicating significance level, e.g.\ 0.05.
star_significance_level	height of invisible horizontal line above which all points are marked as stars, e.g.\ 0.001 as level for high significance.
A11	position of start of first HLA A allele in header line of FASTA file.
A12	position of end of first HLA A allele in header line of FASTA file.
A21	position of start of second HLA A allele in header line of FASTA file.
A22	position of end of second HLA A allele in header line of FASTA file.

B11	position of start of first HLA B allele in header line of FASTA file.
B12	position of end of first HLA B allele in header line of FASTA file.
B21	position of start of second HLA B allele in header line of FASTA file.
B22	position of end of second HLA B allele in header line of FASTA file.
C11	position of start of first HLA C allele in header line of FASTA file.
C12	position of end of first HLA C allele in header line of FASTA file.
C21	position of start of second HLA C allele in header line of FASTA file.
C22	position of end of second HLA C allele in header line of FASTA file.
has_C	set to TRUE if there is a C allel information in header line of FASTA file.
multiple_testing_correction	multiple testing correction applied to p-values. Input can be: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
bayes_factor	if bayes factor should be applied instead of Fisher's exact test for association. See details.
constant_dirichlet_precision_parameter	if Dirichlet precision parameter K is a fixed value. See details.
dirichlet_precision_parameter	the Dirichlet precision parameter for evaluation with Bayes factor. See details.
phylo_bias_check	simple mechanism to detect phylogenetic bias in data. See details.
path_to_file_reference_sequence	Reference sequence for orientation. Will be added as extra column in the results. Not used in calculation itself.
one_feature	if there is only one feature. See details.
window_size	size of sliding window in graphical output.
epi_plot	add epitope plot to results pdf file.

Details

For each sequence alignment position (each column of sequence alignment), Fisher's exact test is evaluated for a 2-by-2 contingency table of amino acid (or nucleic acid) vs. feature. The resulting p-values are returned in a table. p-values can be corrected for multiple testing using various methods (option `multiple_testing_correction`).

If `bayes_factor` is TRUE, a simple calculation of a bayes_factor is applied instead of the Fisher's exact test. An independent model is compared with a nearly independent model (see. *Mixtures of Dirichlet Distributions and Estimation in Contingency Tables*). The user can choose if a fixed dirichlet precision parameter or a calculated one should be used. The calculated dirichlet precision parameter is dependend from the number of entries in the 2-by-2 contingency table. For more information see the mentioned paper and "Bayesian Computation with R". With Bayes Factor a usage of multiple testing is not possible (and in most of all cases not needed). If `constant_dirichlet_precision_parameter` is FALSE, the Dirichlet precision parameter K for the calculation of the Bayes Factors is calculated by the system.

The input sequence alignment may be consist either of DNA sequences (switch `dna = TRUE`) or amino acid sequences (`dna = FALSE`). Undetermined nucleotides or amino acids have to be indicated by the letter "X".

A phylogenetic bias check can be made if `bayes_factor = FALSE` and the result added as an extra column to the csv result file. For each alignment position with a p-value below a user given threshold (`optical_significance_level`), the mean distance of all sequences with the mutation, which gave rise to this p-value, is calculated and compared with the mean distance of all sequences.

Features may be either a single feature, or HLA types, the latter indicated by four blocks in the FASTA comment lines. The positions of these blocks in the comment lines are defined by parameters A11, ..., B22 (..C22 if there is information about the C allele). For patients with a homozygous HLA allele the second allele has to be "00" (without the double quotes). For single features (no HLA types), set option `one_feature=TRUE`. The value of the feature (e.g. 'yes / no', or '1 / 2 / 3') should then be given at the end of each FASTA comment, separated from the part before that by a semicolon.

Value

The function generates various types of output: a table of p-values, p-values corrected for multiple testing, z-scores, amino acid with the lowest p-value at this position, result of the phylogenetic bias test, and several plots. A Manhattan plot is generated for each feature in a pdf file with alignment positions on the x-axis and p-values on the y-axis. Two different significance levels can be indicated by a line (`optical_significance_level`) and the change from the normal dot symbol to a star (`star_significance_level`).

In case of `bayes_factor = TRUE`, the Bayes Factor and not a p-value and the estimate of the simulation standard error of the computed value of the Bayes factor is given.

Optionally (`epi_plot = TRUE`), a second plot is provided with the number of 'significant' p-values or Bayes Factors (value of significance chosen by `optical_significance_level`) in a sliding window of `x` amino acids (`x` can be any number from 1 up to the length of the alignment chosen by '`window_size`'). The typical length of an MHC I binding motif is nine). Each point in this plot is the number of 'significant' p-values or Bayes Factors in the next `x` alignment positions. Additionally, with given HLA motifs (`path_to_file_binding_motifs`), a second line is added. This line shows potential epitopes. A potential epitope is assigned if a 'significant' p-value or Bayes Factor (value of significance chosen by `optical_significance_level`) is inside one of the motifs for this HLA type.

Author(s)

Bettina Budeus

References

Albert, James H.; Gupta, Arjun K. Mixtures of Dirichlet Distributions and Estimation in Contingency Tables. *Ann. Statist.* 10 (1982), no. 4, 1261–1268. doi:10.1214/aos/1176345991. <http://projecteuclid.org/euclid.aos/1176345991>.

Albert, J. *Bayesian Computation with R*. Springer; 2nd ed. 2009 edition (May 15, 2009)

See Also

[assocpointhierarchical](#), [assocpointpair](#), or [Plot](#)

Examples

```
#Input files
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
epitopes_input <- system.file("extdata", "Example_epitopes_aa.csv", package="SeqFeatR")
motifs_input <- system.file("extdata", "Example_HLA_binding_motifs_aa.csv", package="SeqFeatR")
reference_input <- system.file("extdata", "Example_reference_aa.fasta", package="SeqFeatR")

#Usage
## Not run:
assocpoint(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_known_epitopes=epitopes_input,
  path_to_file_binding_motifs=motifs_input,
  save_name_pdf="assocpoint_results.pdf",
  save_name_csv="assocpoint_results.csv",
  dna=FALSE,
  patnum_threshold=1,
  optical_significance_level=0.01,
  star_significance_level=0.5,
  A11=10,
  A12=11,
  A21=13,
  A22=14,
  B11=17,
  B12=18,
  B21=20,
  B22=21,
  C11=24,
  C12=25,
  C21=27,
  C22=28,
  has_C=FALSE,
  multiple_testing_correction="bonferroni",
  bayes_factor=FALSE,
  constant_dirichlet_precision_parameter=TRUE,
  dirichlet_precision_parameter=20,
  phylo_bias_check=FALSE,
  path_to_file_reference_sequence=reference_input,
  one_feature=FALSE,
  window_size=9,
  epi_plot=TRUE)

## End(Not run)
```

assocpointhierarchical

*Searches for associations of single alignment positions with feature(s)
with a simple hierarchical model*

Description

Searches for associations of single nucleotide or amino acid sequence alignment positions with feature(s).

Usage

```
assocpointhierarchical(path_to_file_sequence_alignment = NULL,
  path_to_file_known_epitopes = NULL, path_to_file_binding_motifs = NULL, save_name_pdf,
  save_name_csv, dna = FALSE,
  patnum_threshold = 1, optical_significance_level = 0.05,
  star_significance_level = 0.001, A11, A12, A21, A22, B11, B12, B21,
  B22, path_to_file_reference_sequence=NULL, one_feature=FALSE,
  window_size=9, epi_plot=FALSE, own_model, number_of_simulations,
  number_of_burnin, number_of_chains, response_inits,further_response_data,
  response_parameters)
```

Arguments

path_to_file_sequence_alignment	FASTA file with sequence alignment. For reference see example file.
path_to_file_known_epitopes	csv file with known epitopes. See example file.
path_to_file_binding_motifs	csv file with HLA binding motifs if available. See example file.
save_name_pdf	name of file to which results are saved in pdf format.
save_name_csv	name of file to which results are saved in csv format.
dna	DNA or amino acid sequences.
patnum_threshold	minimum number of patients per HLA type to consider in calculation.
optical_significance_level	height of horizontal line in graphical output indicating significance level, e.g.\ 0.05.
star_significance_level	height of invisible horizontal line above which all points are marked as stars, e.g.\ 0.001 as level for high significance.
A11	position of start of first HLA A allele in header line of FASTA file.
A12	position of end of first HLA A allele in header line of FASTA file.
A21	position of start of second HLA A allele in header line of FASTA file.
A22	position of end of second HLA A allele in header line of FASTA file.
B11	position of start of first HLA B allele in header line of FASTA file.
B12	position of end of first HLA B allele in header line of FASTA file.
B21	position of start of second HLA B allele in header line of FASTA file.
B22	position of end of second HLA B allele in header line of FASTA file.

<code>path_to_file_reference_sequence</code>	Reference sequence for orientation. Will be added as extra column in the results. Not used in calculation itself.
<code>one_feature</code>	if there is only one feature. See details.
<code>window_size</code>	size of sliding window in graphical output.
<code>epi_plot</code>	add epitope plot to results pdf file.
<code>own_model</code>	txt file with jags model - optional. See example file.
<code>number_of_simulations</code>	number of total iterations per chain (including burn in) - see R2jags package.
<code>number_of_burnin</code>	length of burn in, i.e. number of iterations to discard at the beginning - see R2jags package.
<code>number_of_chains</code>	number of Markov chains - see R2jags package.
<code>response_inits</code>	specification of initial value - see details and R2jags package.
<code>further_response_data</code>	additional response data - see details and rjags/R2jags package.
<code>response_parameters</code>	name of response parameters - see details and rjags/R2jags package.

Details

For a given alignment a simple hierarchical model (amino acids/ nucleotides at each position are multi-nominally distributed, features are gamma(0.1,0.1) distributed - see example file).

The input sequence alignment may be consist either of DNA sequences (switch `dna = TRUE`) or amino acid sequences (`dna = FALSE`). Undetermined nucleotides or amino acids have to be indicated by the letter "X".

Features may be either a single feature, or HLA types, the latter indicated by four blocks in the FASTA comment lines. The positions of these blocks in the comment lines are defined by parameters A11, ..., B22. For patients with a homozygous HLA allele the second allele has to be "00" (without the double quotes). For single features (no HLA types), set option `one_feature=TRUE`. The value of the feature (e.g. 'yes / no', or '1 / 2 / 3') should then be given at the end of each FASTA comment, separated from the part before that by a semicolon.

However a user can add his own model and parameter (only for EXPERTS!): The jags model file has to be given as 'own_model'. This 'activates' the following parameters: 1. Further, the user can add other response data ('further_response_data') besides the internally already given matrix with all amino acids/nucleotides at every position and every feature and the colSums of this matrix. 2. The user has to insert initial values ('response_inits') and 3. response parameters ('response_parameters'). For the definition of initial values the following variables can be used: - `all_of_letters` = all amino acids/nucleotides plus gap symbol and unknown (X) - `min_FASTA_length` = length of alignment - `allels` = all HLA-allels/features A response parameter named theta will be evaluated by the part of the program after the model, therefore the user should keep in mind to name this in his own model: The result theta has to be of the following form: 3 dimensional, with first dimension: feature vs no feature, second dimension: amino acid/nucleotide, third dimension: position (see examples file).

Value

The function generates various types of output: a table of probabilities, z-scores, amino acid/nucleotide with the highest probability at this position and several plots. A Manhattan plot is generated for each feature in a pdf file with alignment positions on the x-axis and probabilities on the y-axis. Two different significance levels can be indicated by a line (`optical_significance_level`) and the change from the normal dot symbol to a star (`star_significance_level`).

Optionally (`epi_plot = TRUE`), a second plot is provided with the number of 'significant' p-values (value of significance chosen by `optical_significance_level`) in a sliding window of `x` amino acids (`x` can be any number from 1 up to the length of the alignment chosen by `'window_size'`. The typical length of an MHC I binding motif is nine). Each point in this plot is the number of 'significant' probabilities in the next `x` alignment positions. Additionally, with given HLA motifs (`path_to_file_binding_motifs`), a second line is added. This line shows potential epitopes. A potential epitope is assigned if a 'significant' probabilities (value of significance chosen by `optical_significance_level`) is inside one of the motifs for this HLA type.

Note

To uses this function, it is essential to install Jags. This seems to be easy for Windows and Linux systems, but complicated for MacOS. Therefore this function is not tested in installation process.

Author(s)

Bettina Budeus

References

Albert, J. Bayesian Computation with R. Springer; 2nd ed. 2009 edition (May 15, 2009)

See Also

[assocpoint](#), [assocpointpair](#)

Examples

```
#Input files
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
epitopes_input <- system.file("extdata", "Example_epitopes_aa.csv", package="SeqFeatR")
motifs_input <- system.file("extdata", "Example_HLA_binding_motifs_aa.csv", package="SeqFeatR")
own_model_input <- system.file("extdata", "Example_model_for_bayes.txt", package="SeqFeatR")
reference_input <- system.file("extdata", "Example_reference_aa.fasta", package="SeqFeatR")

#Usage
## Not run:
assocpointhierarchical(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_known_epitopes=epitopes_input,
  path_to_file_binding_motifs=motifs_input,
  save_name_pdf = "assocpointhierarchical_results.pdf",
  save_name_csv = "assocpointhierarchical_results.csv",
  dna = FALSE,
```

```

patnum_threshold = 1,
optical_significance_level = 97,
star_significance_level = 99,
A11 = 10,
A12 = 11,
A21 = 13,
A22 = 14,
B11 = 17,
B12 = 18,
B21 = 20,
B22 = 21,
path_to_file_reference_sequence = reference_input,
one_feature=FALSE,
window_size=9,
epi_plot=TRUE,
own_model=own_model_input,
number_of_simulations=20,
number_of_burnin=floor(20/2),
number_of_chains=2,
response_inits="'vcounts' = rep(1, length(all_of_letters)),
               'theta' = array(1/length(all_of_letters),
                               dim=c(2, length(all_of_letters), min_FASTA_length))",
further_response_data=c(),
response_parameters=c("theta", "vcounts")

## End(Not run)

```

assocpointpair	<i>Compare results of SeqFeatRs assocpoint with SeqFeatRs assocpair, assocpairfeat</i>
----------------	--

Description

Calculates which significant positions in result of assocpoint are also in results of assocpair/assocpairfeat.

Usage

```

assocpointpair(path_to_file_sequence_alignment = NULL,
               path_to_file_assocpoint_csv_result = NULL,
               path_to_file_assocpairfeat_csv_result = NULL,
               significance_level = 0.05, save_name_csv, save_name_pos)

```

Arguments

`path_to_file_sequence_alignment`
FASTA file with sequence alignment. For reference see example file.

`path_to_file_assocpoint_csv_result`
file with result from SeqFeatRs assocpoint. For reference see example file.

path_to_file_assocpairfeat_csv_result	file with results from SeqFeatRs assocpair, assocpairfeat. For reference see example file.
significance_level	p-value to be defined as significant.
save_name_csv	name of file to which results are saved in csv format.
save_name_pos	name of file to which results are saved in csv format (contains possible compensatory mutations).

Details

Takes the results from assocpoint and assocpair/assocpairfeat and tries to combine them into one result: we are looking for pairs of sequence alignment positions where (a) each of the positions is significantly associated with the feature (for significance level see parameter `significance_level`), and (b) the sequence states of both positions are associated. Pairs of positions fulfilling these two criteria may e.g. carry compensatory mutations. If at least one such mutation is found, a detailed output is additionally created with possible compensatory mutations

Value

A csv file with p-value for alignment position pairs and if existing a csv file with possible compensatory mutations.

Author(s)

Bettina Budeus

See Also

[assocpoint](#)
[assocpairfeat](#)
[assocpair](#)

Examples

```
#Input files
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
assocpoint_result <- system.file("extdata", "assocpoint_results.csv", package="SeqFeatR")
assocpairfeat_result <- system.file("extdata", "assocpairfeat_results.csv", package="SeqFeatR")

#Usage
assocpointpair(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_assocpoint_csv_result=assocpoint_result,
  path_to_file_assocpairfeat_csv_result=assocpairfeat_result,
  significance_level=0.05,
  save_name_csv="assocpointpair_result.csv",
  save_name_pos="possible_compensatory_mutation.csv")
```

assoctuple	<i>Searches for associations of a tuple of alignment positions with feature(s)</i>
------------	--

Description

Searches for associations of tuple of nucleotide or amino acid sequence alignment positions with feature(s).

Usage

```
assoctuple(path_to_file_sequence_alignment, path_to_file_assocpoint_csv_result, threshold,
min_number_of_elements_in_tuple, max_number_of_elements_in_tuple,
save_name_csv, column_of_feature, column_of_position, column_of_p_values,
column_of_aa, A11, A12, A21, A22, B11, B12, B21, B22, one_feature, feature)
```

Arguments

path_to_file_sequence_alignment	FASTA file with sequence alignment. For reference see example file.
path_to_file_assocpoint_csv_result	results from SeqFeatRs assocpoint
threshold	p-value threshold for sequence alignment positions to be considered.
min_number_of_elements_in_tuple	minimal number of members in tuple.
max_number_of_elements_in_tuple	maximal number of members in tuple.
save_name_csv	name of file to which results are saved in csv format.
column_of_feature	column number in which feature is located for which analysis should be done.
column_of_position	column number in which sequence position is located.
column_of_p_values	column number from which p-values should be taken. See details.
column_of_aa	column number from which amino acids should be taken. See details.
A11	position of start of first HLA A allele in header line of FASTA file.
A12	position of end of first HLA A allele in header line of FASTA file.
A21	position of start of second HLA A allele in header line of FASTA file.
A22	position of end of second HLA A allele in header line of FASTA file.
B11	position of start of first HLA B allele in header line of FASTA file.
B12	position of end of first HLA B allele in header line of FASTA file.
B21	position of start of second HLA B allele in header line of FASTA file.
B22	position of end of second HLA B allele in header line of FASTA file.
one_feature	if there is only one feature.
feature	feature identifier which should be analyzed. See details.

Details

For each tuple of sequence alignment positions, Fisher's exact test is evaluated for a 2-by-2 contingency table of amino acid tuple (or nucleic acid tuple) vs. feature. The resulting p-values are returned in a table.

For this to work properly the result from SeqFeatRs `assocpoint` can be used, but also a user generated csv file in which at least one column describes the sequence position and another the p-value at this position, as well as a FASTA file from which those file originated.

assoctuple takes only those sequence positions which have a p-value lower than the given threshold ('threshold'). Be aware that for big datasets the calculation time can be high and a calculation of every position with every other position will most definitely result in low quality data.

Please be also aware that it just uses the position in your csv file. If this is NOT the correct position, because of removal of empty or near empty alignment positions, correct the csv file before starting.

Use the same FASTA file you had used for `assocpoint`!

The sequence positions to be included in this analysis are normally chosen from the corrected p-values (but can be anything else as long as it is between 0 and 1, even an own added column). The size of the tuples can be anything from 2 to number of rows (= number of alignment positions) in the csv input file. The input sequence alignment may be consist either of DNA sequences (switch `dna = TRUE`) or amino acid sequences (`dna = FALSE`). Undetermined nucleotides or amino acids have to be indicated by the letter "X".

Features may be HLA types, indicated by four blocks in the FASTA comment lines. The positions of these blocks in the comment lines are defined by parameters A11, ..., B22. For patients with a homozygous HLA allele the second allele has to be "00" (without the double quotes). For non-HLA-type features, set option `one_feature=TRUE`. The value of the feature (e.g. 'yes / no', or '1 / 2 / 3') should then be given at the end of each FASTA comment, separated from the part before that by a semicolon.

The analysis is done only for one single feature. This is chosen by either 'feature' if there is only 'one_feature', or `column_of_feature` if there are HLA types.

Value

A csv list of tuple positions and the p-value of there association.

Author(s)

Bettina Budeus

See Also

[assocpoint](#)

Examples

```
#Input files
## Not run:
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
assocpoint_result <- system.file("extdata", "assocpoint_results.csv", package="SeqFeatR")
```

```

#Usage
assotuple(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_assocpoint_csv_result=assocpoint_result,
  threshold=0.2,
  min_number_of_elements_in_tuple=2,
  max_number_of_elements_in_tuple=2,
  save_name_csv="assotuple_result.csv",
  column_of_feature=9,
  column_of_position=1,
  column_of_p_values=9,
  column_of_aa=12,
  A11=10,
  A12=11,
  A21=13,
  A22=14,
  B11=17,
  B12=18,
  B21=20,
  B22=21,
  one_feature=FALSE,
  feature="")

## End(Not run)

```

comparewithancestral *Compares ancestral sequence with sequence alignment*

Description

Generates from a nexus tree file the possible ancestral sequence and compares this with the sequence alignment.

Usage

```

comparewithancestral(path_to_file_sequence_alignment = NULL,
  path_to_file_nexus_tree = NULL, save_name_csv,
  patnum_threshold = 1, significance_level = 0.05,
  A11, A12, A21, A22, B11, B12, B21, B22)

```

Arguments

`path_to_file_sequence_alignment`
FASTA file with sequence alignment. For reference see example file.

`path_to_file_nexus_tree`
nexus tree file of sequences alignment. For reference see example file.

`save_name_csv` name of file to which results are saved in csv format.

`patnum_threshold`
minimum number of patients per HLA type to consider in calculation.

significance_level	significance level in Fisher's exact test.
A11	position of start of first HLA A allele in header line of FASTA file.
A12	position of end of first HLA A allele in header line of FASTA file.
A21	position of start of second HLA A allele in header line of FASTA file.
A22	position of end of second HLA A allele in header line of FASTA file.
B11	position of start of first HLA B allele in header line of FASTA file.
B12	position of end of first HLA B allele in header line of FASTA file.
B21	position of start of second HLA B allele in header line of FASTA file.
B22	position of end of second HLA B allele in header line of FASTA file.

Details

Features may be HLA types, indicated by four blocks in the FASTA comment lines. The positions of these blocks in the comment lines are defined by parameters A11, ..., B22. For patients with a homozygous HLA allele the second allele has to be "00" (without the double quotes).

Uses the R-packages ape and phangorn to create from a given nexus tree file the ancestral sequence. For every sequence position the ancestral amino acid is compared with the amino acid of every HLA type and with a Fisher's exact test the significant differences are calculated.

Value

Generates a csv file with p-values and q-values (generated by the R-package qvalue) for every HLA type and sequence position with a p-value below given threshold (significance_level).

Note

BEWARE! You have to name the FASTA sequences (= comment line in the file) so that the R-packages ape and phangorn can read and write them correctly. It will replace any space character with "_" (without the double quotes) and ignore ":" (without the double quotes)!

Author(s)

Bettina Budeus

Examples

```
#Input files
## Not run:
fasta_input <- system.file("extdata", "Example_aa_for_cwa.fasta", package="SeqFeatR")
tree_input <- system.file("extdata", "Example_tree.nh", package="SeqFeatR")

#Usage
comparewithancestral(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_nexus_tree=tree_input,
  save_name_csv="hlaTree_results.csv",
  patnum_threshold=1,
```

```

significance_level=0.05,
A11=10,
A12=11,
A21=13,
A22=14,
B11=17,
B12=18,
B21=20,
B22=21)

## End(Not run)

```

foundereffectfinder *Tries to identify a founder effect in results from SeqFeatRs assocpair*

Description

Takes the result of co-mutation analysis like SeqFeatRs assocpair function and adds a column with a note if the row is in the same branch.

Usage

```

foundereffectfinder(path_to_file_sequence_alignment = NULL,
  path_to_file_assocpair_csv_result = NULL, path_to_file_nexus_tree = NULL,
  save_name_csv, threshold = 7)

```

Arguments

path_to_file_sequence_alignment FASTA file with sequence alignment. For reference see example file.

path_to_file_assocpair_csv_result csv file with results from SeqFeatRs assocpair function. For reference see example file.

path_to_file_nexus_tree nexus file with tree data for the given sequence alignment. For reference see example file.

save_name_csv name of file to which results are saved in csv format.

threshold threshold for the number of allowed occurrences of tips in one branch.

Details

Uses the results from SeqFeatRs assocpair analysis and a generated tree for the same sequences in nexus format (for tree generation, e.g <http://mafft.cbrc.jp/alignment/server/> can be used). For every pair of amino acids the number of sequences with both amino acids inside one branch is calculated. If this number exceeds 'threshold' times the number of found associations, this association is considered to be due to a founder effect.

Value

Input file with an added column with founder effect information.

Note

Only use files generated with SeqFeatRs assocpair or assoctuple.

Author(s)

Bettina Budeus

References

Mayr, Ernst (1954). "Change of genetic environment and evolution". In Julian Huxley. Evolution as a Process. London: George Allen & Unwin. OCLC 974739

See Also

[assocpair](#)

Examples

```
#Input files
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
assocpair_result <- system.file("extdata", "assocpair_results.csv", package="SeqFeatR")
tree_input <- system.file("extdata", "Example_tree.nh", package="SeqFeatR")

#Usage
foundereffectfinder(
  path_to_file_sequence_alignment=fasta_input,
  path_to_file_assocpair_csv_result=assocpair_result,
  path_to_file_nexus_tree=tree_input,
  save_name_csv="foundereffectfinder_result.csv",
  threshold=7)
```

getfreqs

Get frequencies for amino acids/nucleotides in alignment window (epitope)

Description

Gets frequencies of amino acids/nucleotides inside an epitope out of an alignment. Epitope start and end positions must be in the range of the length of the alignment, starting with the first position of the alignment as one. You can add the whole sequence consensus, but also a part of it.

Usage

```
getfreqs(path_to_file_sequence_alignment = NULL, save_csv, save_png,
         epitope_position_start, epitope_position_end,
         path_to_file_consensus, patnum_threshold = 1, A11, A12, A21, A22,
         B11, B12, B21, B22, one_feature=FALSE)
```

Arguments

path_to_file_sequence_alignment	a FASTA file with sequence data. For reference please look in example file.
save_csv	name for the csv output.
save_png	name for the png/svg output.
epitope_position_start	starting position of the epitope in the alignment.
epitope_position_end	ending position of the epitope in the alignment.
path_to_file_consensus	a FASTA file with the consensus sequence from epi_pos_start till epi_pos_end.
patnum_threshold	the minimum number of patients of one HLA type to consider in the calculation.
A11	the position of the start of the first HLA A allele in the description block of the FASTA file.
A12	the position of the end of the first HLA A allele in the description block of the FASTA file.
A21	the position of the start of the second HLA A allele in the description block of the FASTA file.
A22	the position of the end of the second HLA A allele in the description block of the FASTA file.
B11	the position of the start of the first HLA B allele in the description block of the FASTA file.
B12	the position of the end of the first HLA B allele in the description block of the FASTA file.
B21	the position of the start of the second HLA B allele in the description block of the FASTA file.
B22	the position of the end of the second HLA B allele in the description block of the FASTA file.
one_feature	if there is only one feature.

Details

This function counts the number of amino acids NOT being the same as the consensus and generates an output graphic as well as a result csv file.

The features may be HLA types, indicated by four blocks in the FASTA comment lines. The positions of these blocks in the comment lines are defined by parameters A11, ..., B22. For patients

with a homozygous HLA allele the second allele has to be "00" (without the double quotes). For non-HLA-type features, set option `one_feature=TRUE`. The value of the feature (e.g. 'yes / no', or '1 / 2 / 3') should then be given at the end of each FASTA comment, separated from the part before that by a semicolon.

Value

A csv file with a row for each feature and two columns (absolute and relative) for the frequencies of the amino acids in the epitope.

Author(s)

Bettina Budeus

See Also

[assocpoint](#)

Examples

```
#Input files
fasta_input <- system.file("extdata", "Example_aa.fasta", package="SeqFeatR")
consensus_input <- system.file("extdata", "Example_Consensus_aa.fasta", package="SeqFeatR")

#Usage
getfreqs(
  path_to_file_sequence_alignment=fasta_input,
  save_csv="getfreqs_result.csv",
  save_png="getfreqs_result.png",
  epitope_position_start=1,
  epitope_position_end=40,
  path_to_file_consensus=consensus_input,
  patnum_threshold=1,
  A11=10,
  A12=11,
  A21=13,
  A22=14,
  B11=17,
  B12=18,
  B21=20,
  B22=21,
  one_feature=FALSE)
```

orPlot

Visualize odds ratios and p-values from SeqFeatRs assocpoint

Description

Creates odds ratio/p-value plots for every feature and sequence alignment position.

Usage

```
orPlot(path_to_file_assocpoint_results = NULL,
       save_name_pdf, separator = ";",
       number_of_cases = 2, odds_column_position = c(6),
       p_value_column_position = c(2), name_column_position = c(2), freq = 7,
       sequence_column_position = NULL, max_y_axis = 50, interval = 10,
       has_color = 0, bias = 5, p_or_od = "P")
```

Arguments

`path_to_file_assocpoint_results`
 csv file with results from SeqFeatRs assocpoint function. For reference see example file.

`save_name_pdf` name of file to which results are saved in csv format.

`separator` letter with which csv is separated (usually ";" or ",", or "\t").

`number_of_cases`
 number of different features in input file.

`odds_column_position`
 column number of first odds ratios.

`p_value_column_position`
 column number of first p-value.

`name_column_position`
 column number of feature name in first row.

`freq` frequency of repeat in columns with more than two cases.

`sequence_column_position`
 column number of first amino acid/nucleotide sequence.

`max_y_axis` estimated guess of the best p-value/ highest Odds ratio. Usual the number behind the e as positive number.

`interval` interval on the y axis.

`has_color` if there should be presented another kind of information as a color feature. See details.

`bias` bias of colors. Select with integer values if you want the first color further down. See details.

`p_or_od` if p-values or OR should be plotted as height. "P" for p-value, "OR" for Odds ratios.

Details

Creates from SeqFeatRs assocpoint results a graphical output where each position in the sequence alignment is shown as bar. The height of this bar can be the log₁₀ of p-value at this position or the odds ratio, the direction information (up or down) based upon odds ratio (below one is down, above one is up). If the user adds a column number for an amino acid or DNA base, this sequence is added above each bar.

The bar plot can be additionally colored by another type of information (e.g. display p-values as height, odds ratio as direction and sequence entropy as color). The source of this color information

has to be added in 'has_color' as a column number. This column can be added manually before creating this plot. The column should contain only numbers between zero and one. A bias can be given to change the color distribution (see R-function colorRampPalette).

Author(s)

Bettina Budeus

See Also

[assocpoint](#)

Examples

```
#Input file
assocpoint_result <- system.file("extdata", "assocpoint_results.csv", package="SeqFeatR")

#Usage
orPlot(
  path_to_file_assocpoint_results=assocpoint_result,
  save_name_pdf="orPlot.pdf",
  separator=";",
  number_of_cases=8,
  odds_column_position=6,
  p_value_column_position=2,
  name_column_position=2,
  freq=7,
  sequence_column_position=NULL,
  max_y_axis=4,
  interval=0.2,
  has_color=0,
  bias=5,
  p_or_od="P")
```

q-values

Calculate q-values from p-values

Description

Calculates and adds a column of q-values to a list of given p-values.

Usage

```
qvalues(path_to_file_csv = NULL, save_name_csv)
```

Arguments

path_to_file_csv

file with a column with p-values. Has to contain a column called "p_value" (without the double quotes). For reference see example file.

save_name_csv name of file to which results are saved in csv format.

Details

Takes a csv file with a column which is called 'p_values' and uses the qvalues package to calculate from this column the corresponding q-values.

Author(s)

Bettina Budeus

Examples

```
#Input file
file_input <- system.file("extdata", "assocpair_results.csv", package="SeqFeatR")

#Usage
qvalues(
  path_to_file_csv=file_input,
  save_name_csv="q-values_result.csv")
```

rewritetuple

Rewrite result from SeqFeatRs assoctuple

Description

Function to rewrite output from SeqFeatRs assoctuple into an usable form for SeqFeatRs tartan.

Usage

```
rewritetuple(path_to_file_assoctuple_csv_result = NULL, save_name_csv,
  first_position, second_position, value_position, separator, threshold)
```

Arguments

```
path_to_file_assoctuple_csv_result
    csv file with results from SeqFeatRs assoctuple function. For reference see ex-
    ample file.
save_name_csv    name of file to which results are saved in csv format.
first_position   column position of first sequence position.
second_position  column position of second sequence position.
value_position   column position of p-value.
separator        separator of csv input file (usually ";", ",", "\t").
threshold        p-value threshold, below which data is to be included in result.
```

Details

Extracts the two sequence positions and p-value if it is below the threshold and generates output csv file for SeqFeatRs tartan with these values.

Author(s)

Bettina Budeus

See Also

[assoctuple](#)

Examples

```
#Input file
assoctuple_result <- system.file("extdata", "assoctuple_result.csv", package="SeqFeatR")

#Usage
rewritetuple(
  path_to_file_assoctuple_csv_result=assoctuple_result,
  save_name_csv="rewritetuple_result.csv",
  first_position=3,
  second_position=4,
  value_position=10,
  separator="\t",
  threshold=0.1)
```

SeqFeatRGUI

GUI for SeqFeatR

Description

A GUI for all program features of SeqFeatR

Usage

```
SeqFeatR_GUI()
```

Details

This is a GUI for the program SeqFeatR. The first tab of the GUI deals with the association of alignment positions with features (e.g. HLA types); a subset of the functionality is provided by a web-server at <http://seqfeatr.zmb.uni-due>. See also video tutorial at <https://www.youtube.com/watch?v=CYidGPE6dw> and vignette in this package.

Author(s)

Bettina Budeus

Examples

```
## Not run: SeqFeatR_GUI()
```

smallmanhattan	<i>A small version of manhattan plot</i>
----------------	--

Description

Creates a small version of the manhattan plot from assocpoint with annotated alignment positions.

Usage

```
smallmanhattan(path_to_file_csv, save_name_png, save_name_svg, feature,  
               corrected=FALSE, x_axis_breaks)
```

Arguments

path_to_file_csv	csv file with results from assocpoint. For reference see example file.
save_name_png	name of file to which graphic is saved in png format.
save_name_svg	name of file to which graphic is saved in svg format.
feature	column number in input csv file with the values. See details.
corrected	if the corrected p-values should be used.
x_axis_breaks	x axis breaks to be displayed. See details.

Details

The input file should be a result file from SeqFeatRs assocpoint.

The feature, that should be displayed as an extra manhattan plot, must be given just like it is displayed in the csv file, e.g for HLA-A2, insert "A2" as value for feature. If you want to show the corrected p-values, change corrected to TRUE.

The x_axis_breaks will mark ticks on the x axis, so that a special alignment position can be displayed. Those axis breaks have to be given in the following format only:

- starting and ending with quotes
- between those quotes comma separated ascending alignment position numbers (please keep in mind, that the plot will only display a little more than the highest alignment number, whatever big number you have given here)
- no blanks
- can be left empty (""), then breaks with 50 positions will be displayed

Value

A small version of the manhattan plot as png/svg picture file. svg files can be changed more easily than png files.

Author(s)

Bettina Budeus

See Also[assocpoint](#)**Examples**

```

#Input files
## Not run:
assocpoint_result <- system.file("extdata", "assocpoint_results.csv", package="SeqFeatR")

#Usage
smallmanhattan(
  assocpoint_result,
  save_name_png = "small_manhattan.png",
  save_name_svg = "small_manhattan.svg",
  feature = "A2",
  corrected = FALSE,
  x_axis_breaks = "1,10,16,20,30,40")

## End(Not run)

```

tartan	<i>Visualize association of mutation pairs and features from two (comparable) analyses.</i>
--------	---

Description

Combines the results of two of SeqFeatRs assocpair or assoctuple analyses in one plot.

Usage

```

tartan(path_to_file_assocpair_csv_result=NULL,
  path_to_file_assocpair_csv_result2=NULL, save_name_pdf, space,
  colors, name_positions, names, ticks, first_position_1,
  second_position_1, value_1, first_position_2, second_position_2,
  value_2, with_distance_matrix, path_to_distance_matrix)

```

Arguments

`path_to_file_assocpair_csv_result` csv file with results from SeqFeatRs assocpair or assoctuple function. For reference see example file.

`path_to_file_assocpair_csv_result2` csv file with results from SeqFeatRs assocpair or assoctuple function. For reference see example file.

`save_name_pdf` name of file to which results are saved in pdf format.

`space` space between blocks. See details.

`colors` the colors of the tartan plot. See details.

<code>name_positions</code>	Positions on x and y axis where the labels should be inserted.
<code>names</code>	labels to be inserted on x and y axis. Beware! <code>name_position</code> and <code>names</code> must have the same length!
<code>ticks</code>	ticks which should mark interesting spots and are also the breakpoints on which the blocks are separated. See details.
<code>first_position_1</code>	column in which first alignment position is located in the first csv file.
<code>second_position_1</code>	column in which second alignment position is located in the first csv file.
<code>value_1</code>	column in which value to be shown is located in first csv file.
<code>first_position_2</code>	column in which first alignment position is located in the second csv file.
<code>second_position_2</code>	column in which second alignment position is located in the second csv file.
<code>value_2</code>	column in which value to be shown is located in csv second file.
<code>with_distance_matrix</code>	if there is a distance matrix available and should be used.
<code>path_to_distance_matrix</code>	path to the distance matrix. Can be zero or empty.

Details

For each sequence x sequence position for one feature the corresponding (p-)values are added as a colored dot.

The input files may be the results from SeqFeatRs `assocpair` or `assotuple` analyses, but any csv file with two different columns of sequence positions and one column with values is sufficient. The values should be in a comparable range, but they can be other information as p-values (e.g. mutual or direct information or distance matrix given as an extra input file).

The displayed colors are chosen by the user. The format for `'colors'` must be color names in quotes (e.g. "red", "blue", etc.) separated by "," (without the double quotes). More colors will give a better distinction.

To distinguish different parts of the sequence (e.g. proteins, markers, etc.) the user may add sequence positions as ticks (`'ticks'`) at which the graphical display is split and a little space (`'space'`) is inserted. The user can also add names to be displayed on the x-axis/y-axis (`'name_positions'`, `'names'`).

Author(s)

Bettina Budeus

See Also

[assocpair](#), [assotuple](#)

Examples

```
#Input files
assocpair_result <- system.file(
  "extdata", "shared_mutations_result_for_tartan.csv", package="SeqFeatR")
assocpair_result2 <- system.file(
  "extdata", "assocpair_results.csv", package="SeqFeatR")

#Usage
tartan(
  path_to_file_assocpair_csv_result=assocpair_result,
  path_to_file_assocpair_csv_result2=assocpair_result2,
  save_name_pdf="tartan_plot.pdf",
  space=5,
  colors=c("wheat", "darkblue", "black", "green"),
  name_positions=c(1,30),
  names=c("S","F"),
  ticks=c(10,30),
  first_position_1=2,
  second_position_1=3,
  value_1=4,
  first_position_2=2,
  second_position_2=3,
  value_2=11,
  with_distance_matrix=FALSE,
  path_to_distance_matrix=NULL)
```

 visualizepair

Visualize pairs of alignment positions that jointly mutate

Description

Creates a pdf output with a visualization of the results from the analysis from SeqFeatRs assocpair function.

Usage

```
visualizepair(path_to_file_assocpair_csv_result = NULL,
  save_name_pdf, significance_level = 0.01)
```

Arguments

path_to_file_assocpair_csv_result
 csv file with results from SeqFeatRs assocpair function. For reference see example file.

save_name_pdf name of file to which results are saved in pdf format.

significance_level
 significance value below which the results of the analysis are considered to be relevant enough to be plotted.

Details

An output is created with a sequence position versus sequence position graphic. Every dot color(one position with the other position) is based on its p-value. The color code is on the right side. In contrast to visualizepairfeat, no feature information will be used.

Note

Only use files generated from SeqFeatRs assocpair function.

Author(s)

Bettina Budeus

See Also

[assocpair](#)

Examples

```
#Input file
assocpair_result <- system.file("extdata", "assocpair_results.csv", package="SeqFeatR")

#Usage
visualizepair(
  path_to_file_assocpair_csv_result=assocpair_result,
  save_name_pdf="vispair_plot.pdf",
  significance_level=0.05)
```

visualizepairfeat	<i>Visualize pairs of sequence alignment positions associated with sequence feature.</i>
-------------------	--

Description

Creates a pdf output with a visualization of the results from the analysis from SeqFeatRs assocpairfeat function.

Usage

```
visualizepairfeat(path_to_file_assocpairfeat_csv_result = NULL, save_name_pdf,
  significance_level = 0.01)
```

Arguments

`path_to_file_assocpairfeat_csv_result`
csv file with results from SeqFeatRs assocpairfeat function. For reference see example file.

`save_name_pdf` name of file to which results are saved in pdf format.

`significance_level`
significance value below which the results of the analysis are considered to be relevant enough to be plotted.

Details

For every feature (if only one p-value is below given threshold ('significance_level')) an output page (pdf page) is created with a sequence position versus sequence position graphic. Every dot color(one position with the other position) is based on its p-value. The color code is on the right side.

Note

Only use files generated from SeqFeatRs assocpairfeat function.

Author(s)

Bettina Budeus

See Also

[assocpairfeat](#)

Examples

```
#Input file
assocpairfeat_result <- system.file("extdata", "assocpairfeat_results.csv", package="SeqFeatR")

#Usage
visualizepairfeat(
  path_to_file_assocpairfeat_csv_result=assocpairfeat_result,
  save_name_pdf="vispairfeat_plot.pdf",
  significance_level=0.05)
```

volcanoplot

Create volcano plot

Description

Creates a pdf output file with a volcano plot out of the results from SeqFeatRs assocpoint.

Usage

```
volcanoplot(path_to_file_assocpoint_csv_result,  
            save_name_pdf, p_values_pos, odds_pos, pos_pos,  
            level_of_sig_p, level_of_sig_odds)
```

Arguments

```
path_to_file_assocpoint_csv_result  
    csv file with results from SeqFeatRs assocpoint. For reference see example file.  
save_name_pdf    name of file to which results are saved in pdf format.  
p_values_pos     column number with p-values.  
odds_pos         column number with odds ratios.  
pos_pos          column number with positions.  
level_of_sig_p   level of significance p values.  
level_of_sig_odds  
    level of significance odds ratios.
```

Details

Plots \log_{10} Odds ratios versus $-\log_{10}$ p-values and annotates the dots with sequence positions.

Author(s)

Bettina Budeus

Examples

```
#Input file  
assocpoint_result <- system.file("extdata", "assocpoint_results.csv", package="SeqFeatR")  
  
#Usage  
volcanoplot(  
  path_to_file_assocpoint_csv_result=assocpoint_result,  
  save_name_pdf="volcano_plot.pdf",  
  p_values_pos = 3,  
  odds_pos = 6,  
  pos_pos = 1,  
  level_of_sig_p = 0.05,  
  level_of_sig_odds = 1)
```

Index

assocpair, [2](#), [5](#), [14](#), [20](#), [29](#), [31](#)
assocpairfeat, [3](#), [4](#), [14](#), [32](#)
assocpoint, [6](#), [12](#), [14](#), [16](#), [22](#), [24](#), [28](#)
assocpointhierarchical, [8](#), [9](#)
assocpointpair, [8](#), [12](#), [13](#)
assoctuple, [3](#), [15](#), [26](#), [29](#)

comparewithancestral, [17](#)

foundereffectfinder, [19](#)

getfreqs, [20](#)

orPlot, [8](#), [22](#)

q-values, [24](#)
qvalues (q-values), [24](#)

rewritetuple, [25](#)

SeqFeatR_GUI (SeqFeatRGUI), [26](#)
SeqFeatRGUI, [26](#)
smallmanhattan, [27](#)

tartan, [28](#)

visualizepair, [3](#), [30](#)
visualizepairfeat, [3](#), [5](#), [31](#)
volcanoplot, [32](#)