

R documentation

of all in ‘man/’

October 7, 2024

Contents

allauc	1
attenuation	2
averageandfilterexprs	4
checkexptab	5
countna	6
genesECDF	7
joinfiles	8
kneeid	9
meandifference	9
plotauc	10
plotecdf	12
plothistoknee	13
plotmetagenes	14
universegroup	16

Index	18
--------------	-----------

allauc	<i>Calculate Area Under Curve (AUC) and Differences of AUC for Transcript Data</i>
--------	--

Description

This function computes the Area Under Curve (AUC) and the differences of AUC between two conditions for a list of transcript data. It supports parallel computation for efficiency.

Usage

```
allauc(bytranslistmean, expdf, nbwindows, nbcpu = 1, dontcompare = NULL,  
controlcondname = "ctrl", stresscondname = "HS", showtime = FALSE,  
verbose = TRUE)
```

Arguments

bytranslistmean	A list of data frames, each containing transcript level data with mean values for one or more conditions.
expdf	A data frame containing experimental conditions associated with the transcript data. It should have a column named 'condition'.
nbwindows	An integer specifying the number of windows to consider for AUC calculations.
nbcpu	An integer specifying the number of CPU cores to use for parallel processing on bytranslistmean. Defaults to 1.
dontcompare	An optional parameter to specify any conditions to exclude from the comparison. Defaults to NULL.
controlcondname	A string specifying the name of the control condition Defaults to "ctrl".
stresscondname	A string specifying the name of the stress condition. Defaults to "HS".
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical value indicating whether to print progress messages Defaults to TRUE.

Details

The function first checks if exactly two conditions are present in 'expdf'. If so, it computes the differences in AUC between the two conditions using a Kolmogorov-Smirnov test and calculates the AUC for all conditions against a reference line ($y=x$). Results are merged by transcript and include adjusted p-values.

Value

A data frame containing the AUC and dAUC results for each transcript, along with associated statistical information.

See Also

[genesECDF]

Examples

```
# Example usage of allauc function
# results <- allauc(bytranslistmean, expdf, nbwindows = 100, nbcpu = 4)
```

attenuation

Calculate Attenuation from AUC and Other Transcript Features

Description

This function computes the attenuation values for each window of each transcript based on the data frames obtained with the functions 'allauc', 'kneeid', and 'countna'.

Usage

```
attenuation(allaucdf, kneedf, matnatrans, bytranslistmean, expdf, dfmeandiff,
nbcpu = 1, significant = FALSE, replaceval = NA, pval = 0.1,
showtime = FALSE, verbose = TRUE)
```

Arguments

allaucdf	A data frame containing AUC results for transcripts (see <code>allauc</code>).
kneedf	A data frame containing the inflection points (see <code>kneeid</code>).
matnatrans	A data frame containing the number of missing values per transcript (see <code>countna</code>).
bytranslistmean	A list of data frames with mean values by transcripts.
expdf	A data frame containing experimental conditions, must include a column named 'condition'.
dfmeandiff	A data frame containing means and differences in mean values, if more than one condition. (see <code>meandifference</code>).
nbcpu	An integer specifying the number of CPU cores to use for parallel processing. The parallelization is done on <code>bytranslistmean</code> whose number of elements is equal to the number of lines provided as input of 'averageandfilterexprs'. Defaults to 1.
significant	A logical indicating whether to filter out non-significant attenuation values. Defaults to FALSE.
replaceval	A value to replace non-significant attenuation values Defaults to NA.
pval	A numeric value specifying the p-value threshold for significance Defaults to 0.1.
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical value indicating whether to print progress messages Defaults to TRUE.

Details

The function merges several data frames to create a comprehensive dataset for each transcript. It computes mean values for the "up" and "down" segments of the transcript. The direction is determined by comparing the coordinates to the knee values. `up = coord < knee` and `down = coord > knee`. The up and down indexes are then retrieved and the attenuation scores are computed as: `att <- 100 - downmean / upmean * 100`

Value

A data frame containing the computed attenuation values along with associated transcript information.

See Also

[`allauc()`], [`kneeid()`], [`countna()`], [`meandifference()`]

Examples

```
# Example usage of attenuation function
#result <- attenuation(allaucdf, kneedf, matnatrans, bytranslistmean,
#                      expdf, dfmeandiff, nbcpu = 4)
```

averageandfilterexprs *Calculate Average Expression and Filter Transcript Data*

Description

This function calculates the average expression levels for transcripts from a provided expression data frame and filters out transcripts based on a specified expression threshold. The function also renames the columns in the output data frame to include mean expression values.

Usage

```
averageandfilterexprs(expdf, alldf, expthres, showtime = FALSE,
verbose = TRUE)
```

Arguments

expdf	A data frame containing expression data that should have columns named 'condition', 'replicate', and 'strand'.
alldf	A data frame containing all transcript-related information, including biotype, chromosome, coordinates, transcript, gene, strand, window, ID and scores retrieved from the bedgraph files.
expthres	A numeric value specifying the expression threshold. Transcripts with average expression values below this threshold will be filtered out from the returned transcript vector.
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical value indicating whether to print progress messages Defaults to TRUE.

Value

A list containing:

maintable	The original data frame containing all transcript data.
exptranstab	A character vector of transcripts that meet the filtering criteria.

Examples

```
# Example usage of averageandfilterexprs
# result <- averageandfilterexprs(expdf, alldf, expthres = 10)
```

checkexptab

*Check Validity of Experiment Table***Description**

The 'checkexptab' function verifies the structure and content of an experiment table to ensure it meets specific formatting requirements. It checks for the presence of required columns, and validates that the 'direction' and 'strand' columns contain only allowable values.

Usage

```
checkexptab(exptab)
```

Arguments

exptab A data frame representing the experiment table. The table must contain the following columns: "condition", "replicate", "direction", and "strand".

Details

The function performs the following checks: - The column names of 'exptab' must match exactly: "condition", "replicate", "direction", and "strand". - The 'direction' column must contain only "forward" and "reverse". - The 'strand' column must contain only "plus" and "minus".

Value

If the experiment table is valid, the function returns 'NULL'. If the table is invalid, the function throws an error specifying the issue.

Examples

```
## Not run:
# Create a valid experiment table
exptab <- data.frame(
  condition = c("cond1", "cond2"),
  replicate = c(1, 1),
  direction = c("forward", "reverse"),
  strand = c("plus", "minus")
)
checkexptab(exptab) # Should pass without errors

# Invalid experiment table (wrong column names)
invalid_exptab <- data.frame(
  cond = c("cond1", "cond2"),
  rep = c(1, 1),
  dir = c("forward", "reverse"),
  str = c("+", "-")
)
checkexptab(invalid_exptab) # Will throw an error

## End(Not run)
```

countna	<i>Count NA values per transcript and condition</i>
---------	---

Description

This function takes a list of expression data frames, a condition information data frame, and counts the number of NA values for each transcript based on strand and condition. NA represent missing scores that were filtered out from the black list and mappability track. The function operates in parallel on transcripts to speed up the process using multiple CPU cores.

Usage

```
countna(allexprsdfs, expdf, nbcpu = 1, showtime = FALSE, verbose = TRUE)
```

Arguments

allexprsdfs	A list of data frames containing expression data. The first element is assumed to be the main table. The second element is a vector of transcript names that passed the filtering of 'averageandfilterexprs'.
expdf	A data frame containing experimental conditions and strand information. Must have columns condition and strand.
nbcpu	An integer specifying the number of CPU cores to use for parallel computation on transcripts. The number of transcripts is equal to the number of lines provided as input of 'averageandfilterexprs'. Defaults to 1.
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical flag indicating whether to print progress messages. Defaults to TRUE.

Value

A data frame where each row corresponds to a transcript, along with its associated gene, strand, and the count of NA values.

See Also

[averageandfilterexprs]

Examples

```
# Assuming allexprsdfs is a list of data frames and expdf contains the  
# conditions:  
# result <- countna(allexprsdfs, expdf, nbcpu = 4)
```

genesECDF*Compute ECDF for Genes Based on Expression Data*

Description

This function calculates the empirical cumulative distribution function (ECDF) for expressed genes across multiple transcripts. It processes the expression data to filter out non-expressed transcripts, compute ECDF values for each transcript, and combine the results into a unified data frame. The function operates in parallel for speed optimization.

Usage

```
genesECDF(allexprsdfts, expdf, nbcpu = 1, rounding = 10,  
showtime = FALSE, verbose = TRUE)
```

Arguments

allexprsdfts	A list of data frames where the first element is the main expression data frame and the second element contains the names of the expressed transcripts (see 'averageandfilterexprs').
expdf	A data frame containing experimental conditions and other relevant information.
nbcpu	An integer specifying the number of CPU cores to use for parallel computation. Default is 1.
rounding	An integer specifying the rounding factor for computing ECDF. Default is 10.
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical flag indicating whether to print progress messages. Default is TRUE.

Details

The function performs several steps:

1. Filters the main expression table to retain only the expressed transcripts.
2. Splits the data by each transcript.
3. For each transcript, computes ECDF values for the score columns while respecting the strand orientation ("plus" or "minus").
4. Combines the ECDF results into a final data frame.

The function uses parallel processing to compute ECDF for each transcript simultaneously, making it faster on systems with multiple CPU cores.

Value

A list containing two elements:

concatdf	A data frame with ECDF results for each transcript.
nbrows	An integer indicating the number of rows in each transcript table.

See Also

[averageandfilterexprs]

Examples

```
# Assuming allexprsdFs is a list of data frames and expdf contains the
# conditions:
# result <- genesECDF(allexprsdFs, expdf, rounding = 10, nbcpu = 4,
#   verbose = TRUE)
```

joinfiles

Join Bedgraph Files for Protein-Coding and lncRNA Data

Description

The ‘joinfiles’ function processes bedgraph files located in the specified working directory, joins data related to protein-coding and lncRNA annotations, and outputs a combined result as a TSV file. The function retrieves bedgraph files matching a specific pattern, processes the files to create windows-based summaries, and merges annotations for protein-coding and lncRNA biotypes. The resulting data is written to an output file.

Usage

```
joinfiles(workingdir = ".", window = 200, bgpattern = "*.bg",
  protscoreDir = "protein_coding_score", lncscoreDir = "lncRNA_score",
  outtsv = "dTAG_Cugusi_stranded_20230810.tsv", verbose = TRUE)
```

Arguments

workingdir	The directory containing bedgraph files. Defaults to the current working directory (“.”).
window	The window size used for joining the score files. Defaults to 200.
bgpattern	A file pattern to identify bedgraph files. Defaults to “*.bg”.
protscoreDir	Directory containing the protein-coding score files. Defaults to “protein_coding_score”.
lncscoreDir	Directory containing the lncRNA score files. Defaults to “lncRNA_score”.
outtsv	The output TSV filename where the merged data will be saved. Defaults to “dTAG_Cugusi_stranded_20230810.tsv”.
nbcpu	An integer specifying the number of CPU cores to use for parallel computation. Default is 1.
verbose	Logical flag to enable verbose output during the function execution. Defaults to ‘TRUE’.

Value

The data.frame with the complete set of annotations and scores.

Examples

```
## Not run:
joinfiles(workingdir = "data", window = 100, bgpattern = "*.bedgraph",
  protscoreDir = "prot_scores", lncscoreDir = "lnc_scores",
  outtsv = "results.tsv")

## End(Not run)
```

kneeid	<i>Identify the Knee and Max ECDF Differences for Each Transcript</i>
--------	---

Description

This function identifies the knee point (i.e., point of maximum change) and the maximum difference in the empirical cumulative distribution function (ECDF) for each transcript, across different experimental conditions.

Usage

```
kneeid(transdflist, expdf, nbcpu = 1, showtime = FALSE, verbose = TRUE)
```

Arguments

transdflist	A list of data frames where each data frame contains transcript data with ECDF values for each condition.
expdf	A data frame containing experimental information including a condition column.
nbcpu	An integer specifying the number of CPU cores to use for parallel computation. The parallelization is performed on the elements of transdflist. Defaults to 1.
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical flag indicating whether to print progress messages. Defaults to TRUE.

Value

A data frame where each row corresponds to a transcript and contains the coordinates of the knee point and the maximum ECDF difference for each condition.

Examples

```
# Assuming transdflist is a list of transcript data frames and expdf contains
# conditions for each experiment:
# result <- kneeid(transdflist, expdf, nbcpu = 4, verbose = TRUE)
```

meandifference	<i>Compute Mean and Differences of Scores for Each Condition</i>
----------------	--

Description

This function calculates the mean values, mean Fx (ECDF) and ECDF differences (Fx) for expression data, across different experimental conditions.

Usage

```
meandifference(resultsecdf, expdf, nbwindows, showtime = FALSE,
  verbose = TRUE)
```

Arguments

resultsecdf	A data frame containing ECDF results for each transcript and condition (see <code>genesECDF</code>).
expdf	A data frame containing experimental information including a condition column.
nbwindows	An integer representing the number of windows (or segments) in each transcript.
showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical flag indicating whether to print progress messages. Defaults to TRUE.

Value

A data frame that contains, for each condition:

- Mean values for the "value" and "Fx" columns (e.g., `mean_value_ctrl`, `mean_Fx_ctrl`).
- Differences between the Fx column and coordinate ratios (e.g., `diff_Fx_ctrl`).

Examples

```
# Assuming resultsecdf is a data frame with ECDF results and expdf contains
# conditions:
# result <- meandifference(resultsecdf, expdf, nbwindows = 200,
# verbose = TRUE)
```

plotauc

Plot AUC Comparison Between Conditions

Description

This function generates scatterplots comparing the area under the curve (AUC) for control and stress conditions, with an option to highlight specific genes or groups. The plot can be saved as a file or displayed interactively.

Usage

```
plotauc(tab, genevec = NA, auc_ctrlname = "AUC_ctrl",
        auc_stressname = "AUC_HS",
        pvalkstestcolname = "adjFDR_p_dAUC_Diff_meanFx_HS_ctrl",
        labelx = "AUC in Control", labely = "AUC in Stress", axismin_x = -10,
        axismax_x = 100, axismin_y = -10, axismax_y = 100, maintitle = "",
        subtitle = "", legendpos = "bottom", formatname = "pdf", outfold = ".",
        outfile = "AUCcompare_pval", plotype = "pval", plot = FALSE,
        universename = "Universe", groupname = "Group", verbose = TRUE)
```

Arguments

tab	A data frame containing the AUC values for control and stress conditions, and other columns required for plotting (e.g., p-values or group memberships, see <code>allauc</code>).
genevec	A vector of gene names to highlight on the plot, applicable when <code>plotttype</code> is set to "pval". Default is NA.
auc_ctrlname	The column name in <code>tab</code> for the AUC under control conditions. Default is "AUC_ctrl".
auc_stressname	The column name in <code>tab</code> for the AUC under stress conditions. Default is "AUC_HS".
pvalkstestcolname	The column name in <code>tab</code> for the adjusted FDR p-values from the KS test. Default is "adjFDR_p_dAUC_Diff_meanFx_HS_ctrl".
labelx	Label for the x-axis. Default is "AUC in Control".
labeley	Label for the y-axis. Default is "AUC in Stress".
axismin_x	Minimum value for the x-axis. Default is -10.
axismax_x	Maximum value for the x-axis. Default is 100.
axismin_y	Minimum value for the y-axis. Default is -10.
axismax_y	Maximum value for the y-axis. Default is 100.
maintitle	Main title of the plot. Default is an empty string.
subtitle	Subtitle of the plot. Default is an empty string.
legendpos	Position of the legend. Default is "bottom".
formatname	Format of the saved plot (e.g., "pdf", "png"). Default is "pdf".
outfold	Output folder where the plot will be saved. Default is ".".
outfile	Name of the output file. Default is "AUCcompare_pval".
plotttype	Type of plot to generate. Can be "pval" for p-value based plots or "groups" for group-based plots. Default is "pval".
plot	A logical flag indicating whether to display the plot interactively (TRUE) or save it to a file (FALSE). Default is FALSE.
universename	Column name in <code>tab</code> representing the universe group in group-based plots. Default is "Universe".
groupname	Column name in <code>tab</code> representing specific groups in group-based plots. Default is "Group".
verbose	A logical flag indicating whether to display detailed messages about the function's progress. Default is TRUE.

Details

The function supports two plot types:

- "pval": The plot highlights genes based on adjusted FDR p-values and can highlight specific genes provided in `genevec`.
- "groups": The plot highlights predefined groups, such as "Attenuated" and "Outgroup", within the data.

If `plot = TRUE`, the plot is displayed interactively. If `plot = FALSE`, the plot is saved to a file in the specified format and output folder.

Value

A plot comparing AUC values between control and stress conditions, either displayed or saved to a file.

See Also

[allauc]

Examples

```
# Assuming `tab` contains AUC values and p-values:
# plotauc(tab, genevec = c("Gene1", "Gene2"), plottype = "pval")
```

plotecdf

Plot Empirical Cumulative Distribution Function (ECDF)

Description

This function generates an ECDF plot to analyze transcription density relative to the distance from the transcription start site (TSS) across different conditions. The plot displays AUC values, Kolmogorov-Smirnov (KS) statistics, and knee points, with options to display or save the plot.

Usage

```
plotecdf(dfmeandiff, unigroupdf, expdf, genename, colvec, outfold = NA,
digits = 2, middlewind = 100, pval = 0.01, plot = FALSE, verbose = TRUE)
```

Arguments

dfmeandiff	A data frame containing the mean differences of transcription levels and cumulative distribution values (Fx) for different windows around the TSS (see meandifference).
ungroupdf	A data frame containing gene-specific statistics, including their belonging to Universe or Group (see universegroup).
expdf	A data frame with experimental conditions and replicates.
genename	A string specifying the name of the gene of interest to plot.
colvec	A vector of colors used to distinguish different conditions in the plot.
outfold	A string specifying the output folder where the plot will be saved if plot = FALSE. Default is NA.
digits	The number of decimal places to round the AUC and KS values. Default is 2.
middlewind	The index of the middle window representing the region centered around the TSS. Default is 100.
pval	A numeric value for the p-value threshold to determine the significance of the KS test. Default is 0.01.
plot	A logical flag indicating whether to display the plot interactively (TRUE) or save it to a file (FALSE). Default is FALSE.
formatname	String of the format of the saved plot. Possible values are "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", and "svg". Default is "pdf".
verbose	A logical flag indicating whether to display detailed messages about the function's progress. Default is TRUE.

Details

The function processes data related to transcription levels and cumulative transcription density for a given gene across multiple experimental conditions. The ECDF plot is constructed with optional annotation of key statistics such as AUC values and significant KS test results. Knee points, representing significant changes in transcription density, are also displayed if the KS test passes the specified p-value threshold.

Value

An ECDF plot showing the transcription density across windows around the TSS, with highlights for significant KS test results and knee points. The plot can either be displayed or saved as a file.

See Also

[meandifference], [universegroup]

Examples

```
# Assuming `dfmeandiff`, `unigroupdf`, and `expdf` contain the necessary
# data:
# plotecdf(dfmeandiff, unigroupdf, expdf, genename = "GeneX",
# colvec = c("blue", "red"))
```

plothistoknee

Plot Histogram of Distance from TSS to Knee Point

Description

This function generates a histogram showing the distribution of the distance from the transcription start site (TSS) to the knee point for attenuated genes. The distance can be plotted either as a percentage of the gene length or in kilobases (kb).

Usage

```
plothistoknee(unigroupdf, plottype = "percent", xlimvec = NA,
binwidthval = NA, kneename = "knee_AUC_HS", plot = FALSE, outfold = ".",
formatname = "pdf", universename = "Universe", groupname = "Group",
verbose = TRUE)
```

Arguments

unigroupdf	A data frame containing gene-level statistics, including knee point data and group classification (see universegroup).
plottype	A string specifying the type of distance to plot. Options are "percent" for the percentage of the gene or "kb" for distance in kilobases. Default is "percent".
xlimvec	A numeric vector of length 2 specifying the limits of the x-axis. Default is NA, which automatically sets the limits based on plottype.
binwidthval	A numeric value for the width of the bins in the histogram. Default is NA, which automatically selects a bin width based on plottype.

kneename	A string specifying the name of the column in unigroupdf that contains the knee point data. Default is "knee_AUC_HS".
plot	A logical flag indicating whether to display the plot interactively (TRUE) or save it to a file (FALSE). Default is FALSE.
outfold	A string specifying the output folder where the plot will be saved if plot = FALSE. Default is the current directory.
formatname	A string specifying the format of the saved plot file. Default is "pdf".
universe	A string specifying the name of the column in unigroupdf that defines the universe of genes. Default is "Universe".
group	A string specifying the name of the column in unigroupdf that defines the group classification of genes. Default is "Group".
verbose	A logical flag indicating whether to display detailed messages about the function's progress. Default is TRUE.

Value

A histogram showing the distribution of the distance from the TSS to the knee point for attenuated genes. The plot can either be displayed interactively or saved to a file.

See Also

[universegroup]

Examples

```
# Assuming `unigroupdf` contains the necessary data:
# plohistoknee(unigroupdf, plotype = "kb", xlimvec = c(0, 300),
# binwidthval = 10)
```

plotmetagenes

Plot Metagenes for Gene Groups

Description

This function plots metagene profiles based on transcript data, comparing transcription density across conditions (e.g., control vs. stress). The function allows the user to plot metagenes for different gene groups such as attenuated genes, outgroup genes, the entire universe of genes, or all genes.

Usage

```
plotmetagenes(unigroupdf, dfmeandiff, plotype = "attenuation",
daucname = "dAUC_Diff_meanFx_HS_ctrl", auc_ctrlname = "AUC_ctrl",
auc_stressname = "AUC_HS", plot = FALSE, formatname = "pdf", outfold = ".",
verbose = TRUE)
```

Arguments

unigroupdf	A data frame containing gene-level information, including group classifications and dAUC data for different conditions (see universegroup).
dfmeandiff	A data frame containing mean transcription values and coordinates for each transcript (see meandifference).
plottype	A string specifying the group of genes to plot. Options are "attenuation", "outgroup", "universe", or "all". Default is "attenuation".
daucname	A string specifying the column name for the delta AUC value (difference between conditions). Default is "dAUC_Diff_meanFx_HS_ctrl".
auc_ctrlname	A string specifying the column name for the control condition AUC values. Default is "AUC_ctrl".
auc_stressname	A string specifying the column name for the stress condition AUC values. Default is "AUC_HS".
plot	A logical flag indicating whether to display the plot interactively (TRUE) or save it to a file (FALSE). Default is FALSE.
formatname	A string specifying the format of the saved plot file. Default is "pdf".
outfold	A string specifying the output folder where the plot will be saved if plot = FALSE. Default is the current directory.
verbose	A logical flag indicating whether to display detailed messages about the function's progress. Default is TRUE.

Details

This function summarizes mean transcription levels across genomic coordinates for different gene groups and plots the transcription density from the transcription start site (TSS) to the transcription termination site (TTS). The function can generate metagene plots for different gene groups such as attenuated, outgroup, or all genes, and compares transcription profiles between conditions (e.g., control vs. stress). The resulting plot helps visualize differences in transcriptional response between groups of genes under different conditions.

Value

A metagene plot comparing transcription density across conditions (e.g., control vs. stress) for the selected group of genes. The plot can either be displayed interactively or saved to a file.

See Also

[universegroup], [meandifference]

Examples

```
# Assuming `unigroupdf` and `dfmeandiff` contain the necessary data:
# plotmetagenes(unigroupdf, dfmeandiff, plottype = "universe", plot = TRUE)
```

universegroup

*Define Universe and Group of Genes Based on Expression Data***Description**

This function categorizes genes into a "Universe" and assigns them into groups such as "Attenuated" or "Outgroup" based on transcription data and thresholds. The universe is defined by thresholds for window size, missing data count, mean transcription levels, and p-values. Genes are further classified into groups based on conditions related to AUC and p-value thresholds.

Usage

```
universegroup(completedf, controlname = "ctrl", stressname = "HS",
windsizethres = 50, countnathres = 20, meanctrlthres = 0.5,
meanstressthres = 0.5, pvaltheorythres = 0.1, aucctrlthreshhigher = -10,
aucctrlthreslower = 15, aucstressthres = 15, attenuatedpvalksthres = 2,
outgrouppvalksthres = 0.2, showtime = FALSE, verbose = TRUE)
```

Arguments

completedf	A data frame obtained with the function attenuation.
controlname	A string representing the control condition name. Default is "ctrl".
stressname	A string representing the stress condition name. Default is "HS".
windsizethres	A numeric threshold for the minimum window size. Default is 50.
countnathres	A numeric threshold for the maximum number of missing data points (NA values). Default is 20.
meanctrlthres	A numeric threshold for the minimum mean transcription value in the control condition. Default is 0.5.
meanstressthres	A numeric threshold for the minimum mean transcription value in the stress condition. Default is 0.5.
pvaltheorythres	A numeric threshold for the minimum p-value used to define the universe of genes. Default is 0.1.
aucctrlthreshhigher	A numeric threshold for the lower bound of the control AUC value in the out-group classification. Default is -10.
aucctrlthreslower	A numeric threshold for the upper bound of the control AUC value in the out-group classification. Default is 15.
aucstressthres	A numeric threshold for the minimum stress AUC value used to classify attenuated genes. Default is 15.
attenuatedpvalksthres	A numeric threshold for the negative log10 of the p-value (from KS test) for defining attenuated genes. Default is 2.
outgrouppvalksthres	A numeric threshold for the maximum KS p-value used to define the outgroup. Default is 0.2.

showtime	A logical value indicating if the duration of the function processing should be indicated before ending. Defaults to FALSE.
verbose	A logical flag indicating whether to print progress messages. Defaults to TRUE.

Details

A transcript belongs to "Universe" if: `window_size > windsizethres & Count_NA < countnathres & meanctrl > meanctrlthres & meanstress > meanstressthres & pvaltheory > pvaltheorythres`

A transcript belongs to the groups: - **Attenuated**: if `Universe == TRUE & aucstress > aucstressthres & -log10(pvals) > attenuatedpvalsthres` - **Outgroup**: if `Universe == TRUE & pvals > outgroup-pvalsthres & aucctrl > aucctrlthreshigher & aucctrl < aucctrlthreslower`

This function is useful for classifying genes in transcriptomics data based on their transcriptional response to different experimental conditions.

Value

A modified data frame with two additional columns: `Universe`, indicating whether each gene is part of the universe, and `Group`, classifying the genes into groups such as "Attenuated", "Outgroup", or NA.

See Also

[attenuation]

Examples

```
# Example usage:
# classified_df <- universegroup(completedf, controlname = "ctrl",
# stressname = "HS")
```

Index

`allauc`, [1](#)
`attenuation`, [2](#)
`averageandfilterexprs`, [4](#)

`checkexptab`, [5](#)
`countna`, [6](#)

`genesECDF`, [7](#)

`joinfiles`, [8](#)

`kneeid`, [9](#)

`meandifference`, [9](#)

`plotauc`, [10](#)
`plotecdf`, [12](#)
`plothistoknee`, [13](#)
`plotmetagenes`, [14](#)

`universegroup`, [16](#)