Package 'MVisAGe'

January 20, 2025

Title Compute and Visualize Bivariate Associations

Version 0.2.1

Description

Pearson and Spearman correlation coefficients are commonly used to quantify the strength of bivariate associations of genomic variables. For example, correlations of gene-level DNA copy number and gene expression measurements may be used to assess the impact of DNA copy number changes on gene expression in tumor tissue. 'MVisAGe' enables users to quickly compute and visualize the correlations in order to assess the effect of regional genomic events such as changes in DNA copy number or DNA methylation level. Please see Walter V, Du Y, Danilova L, Hayward MC, Hayes DN, 2018. Cancer Research <doi:10.1158/0008-5472.CAN-17-3464>.

Depends R (>= 3.3.1)

Suggests R.rsp

VignetteBuilder R.rsp

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Vonn Walter [aut, cre]

Maintainer Vonn Walter <vwalter1@pennstatehealth.psu.edu>

Repository CRAN

Date/Publication 2018-05-10 21:43:30 UTC

Contents

`cn.mat` *DNA copy number data from 98 head and neck squamous cell carcinoma (HNSC) patients*

Description

Quantitative gene-level DNA copy number measurements for 98 samples from The Cancer Genome Atlas (TCGA) HNSC cohort. The all_data_by_genes.txt dataset from the GISTIC2 output was restricted to the first 100 columns and genes that lie on chromosomes 11 and 12. Genes appear in rows; samples appear in columns (other than the first two columns described below). Gene symbols are used as row names and sample identifiers are used as column names (other than the first two columns).

Usage

```
cn.mat
```

Format

A matrix with 2719 rows and 100 columns

Locus.ID gene identifier

Cytoband cytoband containing the gene of interest

Remaining Columns quantitative DNA copy number

Column names sample identifiers (other than the first two columns)

Row names gene symbols

Source

https://gdac.broadinstitute.org/

cn.region.heatmap *A Function for Creating a Heatmap of DNA Copy Number Data*

Description

This function creates a heatmap of DNA copy number data for a given chromosomal region.

Usage

```
cn.region.heatmap(cn.mat, gene.annot, plot.chr, plot.start, plot.stop,
  plot.list, sample.annot = NULL, sample.cluster = F, low.thresh = -2,
  high.thresh = 2, num.cols = 50, collist = c("blue", "white", "red"),
  annot.colors = c("black", "red", "green", "blue", "cyan"),
  plot.sample.annot = F, cytoband.colors = c("gray90", "gray60"))
```

Arguments

`cn.mat` A matrix of gene-level DNA copy number data (rows = genes, columns = samples). DNA methylation data can also be used. Both row names (gene names) and column names (Sample IDs) must be given.

`gene.annot` A three-column matrix containing gene position information. Column 1 = chromosome number written in the form 'chr1' (note that chrX and chrY should be written chr23 and chr24), Column 2 = position (in base pairs), Column 3 = cytoband.

`plot.chr` The chromosome used to define the region of interest.

`plot.start` The genomic position (in base pairs) where the region starts.

`plot.stop` The genomic position (in base pairs) where the region stops.

`plot.list` A list produced by corr.list.compute().

`sample.annot` An optional two-column matrix of sample annotation data. Column 1 = sample IDs, Column 2 = sample annotation (e.g. tumor vs. normal). If NULL, sample annot will be created using the common sample IDs and a single group ('1'). Default = NULL.

`sample.cluster` Logical values indicating whether the samples should be clustered. Default = FALSE.

`low.thresh` Lower threshold for DNA copy number measurements. All values less than low.thresh are set equal to low.thresh. Default = -2.

`high.thresh` Upper threshold for DNA copy number measurements. All values greater than high.thresh are set equal to high.thresh. Default = 2.

`num.cols` Number of distinct colors in the heatmap. Default = 50.

`collist` Color scheme for displaying copy number values. Default = ("blue", "white", "red").

`annot.colors` Character vector used to define the color scheme for sample annotation. Default = c("black", "red", "green", "blue", "cyan").

`plot.sample.annot`
Logical value used to specify whether the sample annotation information should be plotted. Default = FALSE.

`cytoband.colors`
Character vector of length two used to define the color scheme for annotating the cytoband. Default = c("gray90", "gray60").

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 pd.exp = prepped.data[[”exp”]]

 pd.cn = prepped.data[[”cn”]]

 pd.ga = prepped.data[[”gene.annot”]]

 pd.sa = prepped.data[[”sample.annot”]]

 output.list = corr.list.compute(pd.exp, pd.cn, pd.ga, pd.sa)

 cn.region.heatmap(cn.mat = pd.cn, gene.annot = pd.ga, plot.chr = 11,

plot.start = 0e6, plot.stop = 135e6, sample.annot = pd.sa, plot.list = output.list)
```

corr.compute *A Function for Computing a Vector of Pearson Correlation Coefficients*

Description

This function computes Pearson correlation coefficients on a row-by-row basis for two numerical input matrices of the same size.

Usage

```
corr.compute(exp.mat, cn.mat, gene.annot, method = ”pearson”, digits = 5,
  alternative = ”greater”)
```

Arguments

`exp.mat` A matrix of gene-level expression data (rows = genes, columns = samples). Missing values are not permitted.

`cn.mat` A matrix of gene-level DNA copy number data (rows = genes, columns = samples). Both genes and samples should appear in the same order as exp.mat. Missing values are not permitted.

`gene.annot` A three-column matrix containing gene position information. Column 1 = chromosome number written in the form ’chr1’ (note that chrX and chrY should be written chr23 and chr24), Column 2 = position (in base pairs), Column 3 = cytoband. Genes should appear in the same order as exp.mat and cn.mat.

`method` A character string (either "pearson" or "spearman") specifying the method used to calculate the correlation coefficient (default = "pearson").

`digits` Used with signif() to specify the number of significant digits (default = 5).

`alternative` A character string ("greater" or "less") that specifies the direction of the alternative hypothesis, either rho > 0 or rho < 0 (default = "greater").

Value

Returns a eight-column matrix. The first three columns are the same as gene.annot. The fourth column contains gene-specific Pearson or Spearman correlation coefficients based on the entries in each row of exp.mat and cn.mat, respectively (column name = "R"). The fifth column contains squared Pearson correlation coefficients (column name = "R^2"). The sixth column contains t statistics corresponding to the correlation coefficients (column name = "tStat"). The seventh column contains the right-tailed p-value based on the t statistic (column name = "pValue"). The eighth column contains Benjamini-Hochberg q-values corresponding to the p-values. Genes with constant gene expression or DNA copy number are removed because they have zero variance.

Examples

```
corr.results = exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 corr.compute(prepped.data[["exp"]], prepped.data[["cn"]], prepped.data[["gene.annot"]])
```

corr.list.compute *A Function for Creating a List of Pearson Correlation Coefficients*

Description

This function uses the corr.compute() function to compute gene-specific Pearson correlation coefficients in each group of samples defined in a sample annotation matrix.

Usage

```
corr.list.compute(exp.mat, cn.mat, gene.annot, sample.annot = NULL,
  method = "pearson", digits = 5, alternative = "greater")
```

Arguments

`exp.mat` A matrix of gene-level expression data (rows = genes, columns = samples). Missing values are not permitted.

`cn.mat` A matrix of gene-level DNA copy number data (rows = genes, columns = samples). Both genes and samples should appear in the same order as exp.mat. Missing values are not permitted.

`gene.annot` A three-column matrix containing gene position information. Column 1 = chromosome number written in the form ’chr1’ (note that chrX and chrY should be written chr23 and chr24), Column 2 = position (in base pairs), Column 3 = cytoband. Genes should appear in the same order as exp.mat and cn.mat.

`sample.annot` An optional two-column matrix of sample annotation data. Column 1 = sample IDs, Column 2 = sample annotation (e.g. tumor vs. normal). If NULL, sample annot will be created using the common sample IDs and a single group (’1’). Default = NULL.

`method` A character string (either "pearson" or "spearman") specifying the method used to calculate the correlation coefficient (default = "pearson").

`digits` Used with signif() to specify the number of significant digits (default = 5).

`alternative` A character string ("greater" or "less") that specifies the direction of the alternative hypothesis, either rho > 0 or rho < 0 (default = "greater").

Value

Returns a list whose length is the number of unique groups defined by sample.annot. Each entry in the list is the output of corr.compute.

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 pd.exp = prepped.data[[”exp”]]

 pd.cn = prepped.data[[”cn”]]

 pd.ga = prepped.data[[”gene.annot”]]

 pd.sa = prepped.data[[”sample.annot”]]

 corr.list.compute(pd.exp, pd.cn, pd.ga, pd.sa)
```

data.prep *A Function for Preparing mRNAseq and Copy Number Data Matrices*

Description

This function prepares mRNAseq and copy number data matrices for use in other mVisAGe functions.

Usage

```
data.prep(exp.mat, cn.mat, gene.annot, sample.annot = NULL, log.exp = FALSE,
  gene.list = NULL)
```

Arguments

`exp.mat`	A matrix of gene-level expression data (rows = genes, columns = samples). Both row names (gene names) and column names (sample IDs) must be given.
`cn.mat`	A matrix of gene-level DNA copy number data (rows = genes, columns = samples). DNA methylation data can also be used. Both row names (gene names) and column names (Sample IDs) must be given.
`gene.annot`	A three-column matrix containing gene position information. Column 1 = chromosome number written in the form 'chr1' (note that chrX and chrY should be written chr23 and chr24), Column 2 = position (in base pairs), Column 3 = cytoband.
`sample.annot`	An optional two-column matrix of sample annotation data. Column 1 = sample IDs, Column 2 = categorical sample annotation (e.g. tumor vs. normal). If NULL, sample annot will be created using the common sample IDs and a single group ('1'). Default = NULL.
`log.exp`	A logical value indicating whether or not the expression values have been log-transformed. Default = FALSE.
`gene.list`	Used to restrict the output to a set of genes of interest, e.g. genes identified by GISTIC as having recurrent copy number alterations. Default = NULL, and in this case all genes are used.

Value

Returns a list with four components: cn, exp, gene.annot, and sample.annot. Each of cn, exp, and gene.annot have been restricted to a common set of genes, and these appear in the same order. Similarly, cn, exp, and sample.annot have been restricted to a common set of subjects that appear in the same order.

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)
```

```
data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)
```

`exp.mat` *Gene expression data from 100 head and neck squamous cell carcinoma (HNSC) patients*

Description

RSEM gene expression measurements for 100 samples from The Cancer Genome Atlas (TCGA) HNSC cohort after restricting to genes that lie in chromosomes 11 and 12. Genes appear in rows; samples appear in columns (other than the first two columns described below). Gene symbols are used as row names and sample identifiers are used as column names (other than the first two columns).

Usage

```
exp.mat
```

Format

A matrix with 2161 rows and 100 columns

Columns RSEM gene expression measurements

Column names sample identifiers

Row names gene symbols

Source

https://gdc.cancer.gov/

`gene.annot` *Gene annotation data (hg38)*

Description

Genomic position and cytoband annotation data.

Usage

```
gene.annot
```

Format

A matrix with 26417 rows and 3 columns

chr chromosome containing the gene of interest

pos genomic position (base pairs) for the gene of interest

cytoband cytoband containing the gene of interest

Row names gene symbols

Source

`https://usegalaxy.org/`

perm.significance *A Function for Computing a Vector of Pearson Correlation Coefficients*

Description

This function computes Pearson correlation coefficients on a row-by-row basis for two numerical input matrices of the same size.

Usage

```
perm.significance(exp.mat, cn.mat, gene.annot, method = "pearson",
  digits = 5, num.perms = 100, random.seed = NULL,
  alternative = "greater")
```

Arguments

`exp.mat`	A matrix of gene-level expression data (rows = genes, columns = samples). Missing values are not permitted.
`cn.mat`	A matrix of gene-level DNA copy number data (rows = genes, columns = samples). Both genes and samples should appear in the same order as exp.mat. Missing values are not permitted.
`gene.annot`	A three-column matrix containing gene position information. Column 1 = chromosome number written in the form 'chr1' (note that chrX and chrY should be written chr23 and chr24), Column 2 = position (in base pairs), Column 3 = cytoband. Genes should appear in the same order as exp.mat and cn.mat.
`method`	A character string (either "pearson" or "spearman") specifying the method used to calculate the correlation coefficient (default = "pearson").
`digits`	Used with signif() to specify the number of significant digits (default = 5).
`num.perms`	Number of permutations used to assess significance (default = 1e2).
`random.seed`	Random seed (default = NULL).
`alternative`	A character string ("greater" or "less") that specifies the direction of the alternative hypothesis, either rho > 0 or rho < 0 (default = "greater").

Value

Returns a five-column matrix. The first three columns are the same as gene.annot. The fourth column contains gene-specific Pearson or Spearman correlation coefficients based on the entries in each row of exp.mat and cn.mat, respectively (column name = "R"). The fifth column contains squared Pearson correlation coefficients (column name = "R^2"). The sixth column contains the permutation-based right-tailed p-value of the correlation coefficient (column name = "perm_pValue"). The seventh column contains Benjamini-Hochberg q-values corresponding to the p-values. Genes with constant gene expression or DNA copy number are removed because they have zero variance.

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 perm.significance(prepped.data[["exp"]], prepped.data[["cn"]], prepped.data[["gene.annot"]])
```

perm.significance.list.compute

A Function for Creating a List of Pearson Correlation Coefficients

Description

This function uses the corr.compute() function to compute gene-specific Pearson correlation coefficients in each group of samples defined in a sample annotation matrix.

Usage

```
perm.significance.list.compute(exp.mat, cn.mat, gene.annot,
  sample.annot = NULL, method = "pearson", digits = 5, num.perms = 100,
  random.seed = NULL, alternative = "greater")
```

Arguments

exp.mat	A matrix of gene-level expression data (rows = genes, columns = samples). Missing values are not permitted.
cn.mat	A matrix of gene-level DNA copy number data (rows = genes, columns = samples). Both genes and samples should appear in the same order as exp.mat. Missing values are not permitted.
gene.annot	A three-column matrix containing gene position information. Column 1 = chromosome number written in the form 'chr1' (note that chrX and chrY should be written chr23 and chr24), Column 2 = position (in base pairs), Column 3 = cytoband. Genes should appear in the same order as exp.mat and cn.mat.

`sample.annot` An optional two-column matrix of sample annotation data. Column 1 = sample IDs, Column 2 = sample annotation (e.g. tumor vs. normal). If NULL, sample annot will be created using the common sample IDs and a single group ('1'). Default = NULL.

`method` A character string (either "pearson" or "spearman") specifying the method used to calculate the correlation coefficient (default = "pearson").

`digits` Used with signif() to specify the number of significant digits (default = 5).

`num.perms` Number of permutations used to assess significance (default = 1e2).

`random.seed` Random seed (default = NULL).

`alternative` A character string ("greater" or "less") that specifies the direction of the alternative hypothesis, either rho > 0 or rho < 0 (default = "greater").

Value

Returns a list whose length is the number of unique groups defined by sample.annot. Each entry in the list is the output of perm.significance.

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 pd.exp = prepped.data[[”exp”]]

 pd.cn = prepped.data[[”cn”]]

 pd.ga = prepped.data[[”gene.annot”]]

 pd.sa = prepped.data[[”sample.annot”]]

 perm.significance.list.compute(pd.exp, pd.cn, pd.ga, pd.sa)
```

`sample.annot` *Sample annotation data*

Description

Human papillomavirus (HPV) infection status for the n = 279 patients with head and neck squamous cell carcinoma in the The Cancer Genoma Atlas cohort.

Usage

```
sample.annot
```

Format

A matrix with 26417 rows and 3 columns

Barcode sample identifier

New.HPV.Status HPV infection status

Source

http://www.nature.com/nature/journal/v517/n7536/full/nature14129.html

smooth.genome.plot *A Function for Plotting Smoothed Pearson Correlation Coefficients Across Multiple Chromosomes*

Description

This function plots smoothed R or R^2 values produced by corr.list.compute() across multiple chromosomes or genomewide.

Usage

```
smooth.genome.plot(plot.list, plot.column = "R^2", annot.colors = c("black",
  "red", "green", "blue", "cyan"), vert.pad = 0.05, ylim.low = NULL,
  ylim.high = NULL, plot.legend = TRUE, legend.loc = "bottomright",
  lty.vec = NULL, lwd.vec = NULL, loess.span = 250, expand.size = 50,
  rect.colors = c("light gray", "gray"), chr.label = TRUE,
  xaxis.label = "Chromosome", yaxis.label = NULL, main.label = NULL,
  axis.cex = 1, label.cex = 1, xaxis.line = 1.5, yaxis.line = 2.5,
  main.line = 0, margin.vec = rep(1, 4))
```

Arguments

`plot.list` A list produced by corr.list.compute().

`plot.column` "R" or "R^2" depending on whether Pearson correlation coefficients or squared Pearson correlation coefficients will be plotted. Default = "R^2".

`annot.colors` A vector of colors used for plotting values in different entries of plot.list. Default = c("black", "red", "green", "blue", "cyan").

`vert.pad` Amount of vertical white space in the plot. Default = 0.

`ylim.low` Smallest value on the y-axis (used to control the range of values on the y-axis). Default = NULL.

`ylim.high` Largest value on the y-axis (used to control the range of values on the y-axis). Default = NULL.

`plot.legend` Logical value specifying whether a legend should be included. Default = FALSE.

`legend.loc` Character value specifing the location of the legend. Default = "topright". See See `legend`.

`lty.vec` Vector specifying line types for plotting values in different entries of plot.list. Default = NULL. See `par`.

`lwd.vec` Vector specifying line widths for plotting values in different entries of plot.list. Default = NULL. See `par`.

`loess.span` A numerical value used to control the level of smoothing. Smoothing is performed separately for each chromosome, and loess.span effectively defines the number of genes in each smoothing window. Default = 250.

`expand.size` A numerical value used to control smoothing at the ends of chromosomes. Both ends of each chromosome are artificially extended by expand.size genes, smoothing is performed on the expanded chromosome, and then the smoothed values are restricted to the size of the original chromosome. Default = 50.

`rect.colors` A character vector of length two that controls the background color for each alternating chromosome. Default = c("light gray", "gray").

`chr.label` Logical value specifying whether chromosome numbers should appear on the plot. Default = FALSE.

`xaxis.label` Text used to label the x-axis of the plot. Default = "Chromosome". See `plot`.

`yaxis.label` Text used to label the y-axis of the plot. Default = NULL. See `plot`.

`main.label` Text used to label the plot header. Default = NULL. See `par`.

`axis.cex` Numerical value used to specify the font size on the axes. Default = 1. See `par`.

`label.cex` Numerical value used to specify the font size for the axis labels. Default = 1. See `par`.

`xaxis.line` Numerical value used to specify location of xaxis.label. Default = 0. See `mtext`.

`yaxis.line` Numerical value used to specify location of yaxis.label. Default = 0. See `mtext`.

`main.line` Numerical value used to specify location of main.label. Default = 0. See `mtext`.

`margin.vec` Numerical vector specifying margin sizes. Default = rep(1, 4). See `par`.

Value

Creates a plot of gene-level R or R^2 values produced by corr.list.compute(). Values of R

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 pd.exp = prepped.data[[”exp”]]

 pd.cn = prepped.data[[”cn”]]

 pd.ga = prepped.data[[”gene.annot”]]

 pd.sa = prepped.data[[”sample.annot”]]
```

```
output.list = corr.list.compute(pd.exp, pd.cn, pd.ga, pd.sa)

smooth.genome.plot(plot.list = output.list, lwd.vec = c(3, 3), lty.vec = c(1, 2))
```

smooth.region.plot *A Function for Plotting Smoothed Pearson Correlation Coefficients Genomewide*

Description

This function plots smoothed R or R^2 values produced by corr.list.compute() genomewide.

Usage

```
smooth.region.plot(plot.list, plot.chr, plot.start, plot.stop,
  plot.column = "R^2", annot.colors = c("black", "red", "green", "blue",
  "cyan"), vert.pad = 0.05, ylim.low = NULL, ylim.high = NULL,
  plot.legend = TRUE, legend.loc = "topleft", lty.vec = NULL,
  lwd.vec = NULL, loess.span = 50, expand.size = 50,
  xaxis.label = "Position (Mb)", yaxis.label = NULL, main.label = NULL,
  axis.cex = 1, label.cex = 1, xaxis.line = 1.5, yaxis.line = 2.5,
  main.line = 0)
```

Arguments

`plot.list` A list produced by corr.list.compute().

`plot.chr` The chromosome for which gene-level R or R^2 values will be plotted.

`plot.start` The genomic position (in base pairs) where the plot will start.

`plot.stop` The genomic position (in base pairs) where the plot will stop.

`plot.column` "R" or "R^2" depending on whether Pearson correlation coefficients or squared Pearson correlation coefficients will be plotted. Default = "R^2".

`annot.colors` A vector of colors used for plotting values in different entries of plot.list. Default = c("black", "red", "green", "blue", "cyan").

`vert.pad` Amount of vertical white space in the plot. Default = 0.

`ylim.low` Smallest value on the y-axis (used to control the range of values on the y-axis). Default = NULL.

`ylim.high` Largest value on the y-axis (used to control the range of values on the y-axis). Default = NULL.

`plot.legend` Logical value specifying whether a legend should be included. Default = FALSE.

`legend.loc` Character value specifing the location of the legend. Default = "topright". See See `legend`.

`lty.vec` Vector specifying line types for plotting values in different entries of plot.list. Default = NULL. See `par`.

lwd.vec Vector specifying line widths for plotting values in different entries of plot.list. Default = NULL. See par.

loess.span A numerical value used to control the level of smoothing. Smoothing is performed separately for each chromosome, and loess.span effectively defines the number of genes in each smoothing window. Default = 250.

expand.size A numerical value used to control smoothing at the ends of the region of interest. Both ends of the region are artificially extended by expand.size genes, smoothing is performed on the expanded region, and then the smoothed values are restricted to the size of the original region. Default = 50.

xaxis.label Text used to label the x-axis of the plot. Default = "Chromosome". See plot.

yaxis.label Text used to label the y-axis of the plot. Default = NULL. See plot.

main.label Text used to label the plot header. Default = NULL. See plot.

axis.cex Numerical value used to specify the font size on the axes. Default = 1. See par.

label.cex Numerical value used to specify the font size for the axis labels. Default = 1. See par.

xaxis.line Numerical value used to specify location of xaxis.label. Default = 0. See mtext.

yaxis.line Numerical value used to specify location of yaxis.label. Default = 0. See mtext.

main.line Numerical value used to specify location of main.label. Default = 0. See mtext.

Value

Creates a plot of gene-level R or R^2 values produced by corr.list.compute().

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 pd.exp = prepped.data[[”exp”]]

 pd.cn = prepped.data[[”cn”]]

 pd.ga = prepped.data[[”gene.annot”]]

 pd.sa = prepped.data[[”sample.annot”]]

 output.list = corr.list.compute(pd.exp, pd.cn, pd.ga, pd.sa)

 smooth.region.plot(plot.list = output.list, plot.chr = 11, plot.start = 0e6, plot.stop = 135e6)
```

tcga.cn.convert *A Function for Reformatting TCGA DNA Copy Number Matrices*

Description

This function reformats DNA copy number matrices obtained from the Broad Institute's Firehose GDAC (https://gdac.broadinstitute.org/) so they can be used as input for mVisAGe functions.

Usage

```
tcga.cn.convert(cn.mat)
```

Arguments

`cn.mat` A matrix of DNA copy number data included in the GISTIC2 output. Typically all_data_by_genes.txt, or a subset thereof, including the Locus.ID and Cytoband columns.

Value

A matrix of DNA copy number data (rows = genes, columns = samples) that is suitable for input to mVisAGe functions.

Examples

```
cn.mat = tcga.cn.convert(cn.mat)
```

tcga.exp.convert *A Function for Reformatting TCGA mRNA Expression Matrices*

Description

This function reformats mRNA expression matrices obtained from the Broad Institute's Firehose GDAC (https://gdac.broadinstitute.org/) so they can be used as input for mVisAGe functions.

Usage

```
tcga.exp.convert(exp.mat)
```

Arguments

`exp.mat` A matrix of mRNA expression data. Typically illuminahiseq_rnaseqv2-RSEM_genes_normalized, or a subset thereof, including the header rows.

Value

A matrix of mRNA expression data (rows = genes, columns = samples) that is suitable for input to mVisAGe functions.

Examples

```
exp.mat = tcga.exp.convert(exp.mat)
```

---

unsmooth.region.plot *A Function for Plotting Pearson Correlation Coefficients in a Given Genomic Region*

Description

This function plots unsmoothed R or R^2 values produced by corr.list.compute() in a specified genomic region.

Usage

```
unsmooth.region.plot(plot.list, plot.chr, plot.start, plot.stop,
  plot.column = "R", plot.points = TRUE, plot.lines = TRUE,
  gene.names = TRUE, annot.colors = c("black", "red", "green", "blue",
  "cyan"), vert.pad = 0.05, num.ticks = 5, ylim.low = NULL,
  ylim.high = NULL, pch.vec = NULL, lty.vec = NULL, lwd.vec = NULL,
  plot.legend = TRUE, legend.loc = "topright")
```

Arguments

`plot.list`	A list produced by corr.list.compute().
`plot.chr`	The chromosome for which gene-level R or R^2 values will be plotted.
`plot.start`	The genomic position (in base pairs) where the plot will start.
`plot.stop`	The genomic position (in base pairs) where the plot will stop.
`plot.column`	"R" or "R^2" depending on whether Pearson correlation coefficients or squared Pearson correlation coefficients will be plotted. Default = "R^2".
`plot.points`	Logical value specifying whether points should be included in the plot. Default = TRUE.
`plot.lines`	Logical values specifying whether points should be connected by lines. Default = FALSE.
`gene.names`	Logical value specifying whether gene names should appear on the plot. Default = FALSE.
`annot.colors`	A vector of colors used for plotting values in different entries of plot.list. Default = c("black", "red", "green", "blue", "cyan").
`vert.pad`	Amount of vertical white space in the plot. Default = 0.05.

`num.ticks` Number of ticks on the x-axis. Default = 5.

`ylim.low` Smallest value on the y-axis (used to control the range of values on the y-axis). Default = NULL.

`ylim.high` Largest value on the y-axis (used to control the range of values on the y-axis). Default = NULL.

`pch.vec` Vector specifying point characters for plotting values in different entries of plot.list. Default = NULL. See `par`.

`lty.vec` Vector specifying line types for plotting values in different entries of plot.list. Default = NULL. See `par`.

`lwd.vec` Vector specifying line widths for plotting values in different entries of plot.list. Default = NULL. See `par`.

`plot.legend` Logical value specifying whether a legend should be included. Default = FALSE.

`legend.loc` Character value specifing the location of the legend. Default = "topright". See See `legend`

Value

Creates a plot of gene-level R or R^2 values produced by corr.list.compute().

Examples

```
exp.mat = tcga.exp.convert(exp.mat)

 cn.mat = tcga.cn.convert(cn.mat)

 prepped.data = data.prep(exp.mat, cn.mat, gene.annot, sample.annot, log.exp = FALSE)

 pd.exp = prepped.data[[”exp”]]

 pd.cn = prepped.data[[”cn”]]

 pd.ga = prepped.data[[”gene.annot”]]

 pd.sa = prepped.data[[”sample.annot”]]

 output.list = corr.list.compute(pd.exp, pd.cn, pd.ga, pd.sa)

 unsmooth.region.plot(plot.list = output.list, plot.chr = 11, plot.start = 69e6, plot.stop = 70.5e6)
```

Index