Package 'CRISPRcleanR'

October 18, 2022

Type Package

Title Unsupervised correction of gene independent cell responses to CRISPR-cas9 targeting

Version 3.0.0

Date 2021-03-15

Author Francesco Iorio, Paolo Cremaschi

Maintainer Francesco Iorio <francesco.iorio@fht.org>

License MIT + file LICENSE

Description

Unsupervised approach to identify and correct gene independent responses to CRISPRcas9 targeting, in genome-wide pooled sgRNA drop-out screens, based on the segmentation of single-guide RNA (sgRNA) fold change values across the genome, without making any assumption on the copy number status of the targeted genes. The package allows to export sgRNA fold changes and normalised sgRNA read counts, and is therefore compatible with downstream analysis tools, and works with multiple sgRNA libraries. Iorio F, Behan FM, Goncalves E, et al. (2018) <doi:10.1186/s12864-018-4989-y>.

biocViews

Depends stringr, DNAcopy, pROC, stats, utils, withr, grDevices, graphics, pracma, PRROC, tools, BiocManager, ShortRead, Biostrings, Rsubread, GenomicAlignments, Rqc, jsonlite

RoxygenNote 6.0.1

R topics documented:

AVANA_Library *Genome-wide annotation for the AVANA sgRNA library*

Description

A data frame with a named row for each sgRNA of the AVANA sgRNA library [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data(AVANA_Library)
```

Format

A a row named data frame with 71482 observations (one for each sgRNA) of the following 7 variables.

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`EXONE` exone of the targeted genomic region (string with 'ex' prefix followed by the exone number);

`CHRM` chromosome of where the targeted region resides (string)

`STRAND` targeted DNA strand ('+' or '-')

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

References

[1] Meyers RM, Bryan JG, McFarland JM, Weir BA. Computational correction of copy number effect improves specificity of CRISPR-Cas9 essentiality screens in cancer cells. Nature. 2017.

Examples

```
data(AVANA_Library)
head(AVANA_Library)
```

BAGEL_essential *Reference Core fitness essential genes*

Description

A list of reference core fitness essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [1].

Usage

```
data(BAGEL_essential)
```

Format

A vector of strings containing HGNC symbols of reference core fitness essential genes.

References

[1] BAGEL: a computational framework for identifying essential genes from pooled library screens. Traver Hart and Jason Moffat. BMC Bioinformatics, 2016 vol. 17 p. 164.

See Also

BAGEL_nonEssential

Examples

```
data(BAGEL_essential)
head(BAGEL_essential)
```

BAGEL_nonEssential *Reference set of non essential genes*

Description

A list of reference non essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [1].

Usage

```
data(BAGEL_nonEssential)
```

Format

A vector of strings containing HGNC symbols of reference non essential genes.

References

[1] BAGEL: a computational framework for identifying essential genes from pooled library screens. Traver Hart and Jason Moffat. BMC Bioinformatics, 2016 vol. 17 p. 164.

See Also

BAGEL_essential

Examples

```
data(BAGEL_nonEssential)
head(BAGEL_nonEssential)
```

Brunello_Library *Genome-wide annotation for the Brunello sgRNA library*

Description

A data frame with a named row for each sgRNA of the Brunello sgRNA library [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data(Brunello_Library)
```

Format

A a row named data frame with 76379 observations of the following variables (among others)

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`STRAND` targeted DNA strand ('sense' or 'antisense')

`EXONE` exone of the targeted genomic region (exone number);

`CHRM` chromosome of where the targeted region resides (string)

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

Source

Addgene website (catalog number: 73179; file: broadgpp-brunello-library-contents.txt, url: https://www.addgene.org/static/c
eac1-44b2-bb2f-8fea95672705/broadgpp-brunello-library-contents.txt)

References

[1] Doench JG, Fusi N, Sullender M, Hegde M, Vaimberg EW, Donovan KF, et al. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat Biotechnol. 2016;34:184-91.

[2] Ong SH, Li Y, Koike-Yusa H, Yusa K. Optimised metrics for CRISPR-KO screens with second-generation gRNA libraries [published correction appears in Sci Rep. 2018 Apr 12;8(1):6136]. Sci Rep. 2017;7(1):7384. Published 2017 Aug 7. doi:10.1038/s41598-017-07827-z

Examples

```
## Not run:
## Loading sgRNA Brunello library annotation file
data(Brunello_Library)
## Visualising first entries
head(Brunello_Library)

## Deriving the path of an example count file
## from screening the HT-29 cell line with the Brunello library
## [2]

fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/HT29-Brunello_counts.tsv',sep='')

expName<-'HT29-Brunello'

## Loading, median-normalizing and computing fold-changes
normANDfcs<-
    ccr.NormfoldChanges(filename = fn,
                        display = TRUE,
                        min_reads = 30,
                        EXPname = expName,
                        libraryAnnotation = Brunello_Library)

## Genome-sorting the fold changes
gwSortedFCs<-
    ccr.logFCs2chromPos(foldchanges = normANDfcs$logFCs,
                        libraryAnnotation = Brunello_Library)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs_and_segments<-
    ccr.GWclean(gwSortedFCs = gwSortedFCs,
                display=TRUE,
                label=expName)

## End(Not run)
```

CCLE.gisticCNA *Genome-wide copy number data for 13 human cancer cell lines.*

Description

Genome-wide Gistic [1] scores quantifying copy number status across a subset of the cell lines in `CL.subset` that are used to assess CRISPRcleaneR results in [2].

Usage

```
data(CCLE.gisticCNA)
```

Format

A data frame with one observations per gene across 13 variables (one per cell line). Row names indicate HGNC gene symbols and column names indicate cell line COSMIC identifiers [3].

Source

This data frame has been derived from the tsv file downloadable at
http://www.cbioportal.org/study?id=cellline_ccle_broad#summary.
This has been obtained by processing Affymetrix SNP array data in the Cancer Cell Line Encyclopaedia [4] repository
(https://depmap.org/portal/download/)

References

[1] Mermel CH, Schumacher SE, Hill B, et al. *GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers.* Genome Biol. 2011;12(4):R41. doi: 10.1186/gb-2011-12-4-r41.

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting.
http://doi.org/10.1101/228189

[2] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783,

[3] Barretina J, Caponigro G, Stransky N, et al. *The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity.* Nature. 2012 Mar 28;483(7391):603-7. doi: 10.1038/nature11003. Erratum in: Nature. 2012 Dec 13;492(7428):290.

Examples

```
data(CCLE.gisticCNA)
head(CCLE.gisticCNA)
```

`ccr.AnalysisPipeline` *CRISPRcleanR analysis pipeline.*

Description

This function runs sequentially all the main steps of a CRISPRscreenR analysis, adding extra control steps to ensure the consistency between the raw counts and the library used in the screen. The pipeline takes as input raw count/matrix or lists of FASTQ/BAM files and allows the user to define all the parameters used in the analysis steps. All the output files will be generated in the outdir folder and organised in two separate subfolders: "data" containing all results in the TXT format; "pdf" containing all the plots in PDF format. An optional "bam" folder containing all the BAM files generated by the alignment will be available if a list of FASTQ files is used as input.

Usage

```
ccr.AnalysisPipeline(
  # Library relaated parameters
  library_builtin = NULL,
  library_file = NULL,

  # Counts / FCs related parameters
  file_counts = NULL,

  # FASTQ / BAM options
  files_FASTQ_controls = NULL,
  files_FASTQ_samples = NULL,
  files_BAM_controls = NULL,
  files_BAM_samples = NULL,
  aligner = c("Rsubreads", "mageck"),
  maxMismatches = 0,
  nTrim5 = "0",
  nTrim3 = "0",
  nBestLocations = 2,
  strand = c("F", "R", "*"),
  duplicatedSeq = c("keep", "exclude"),
  nthreads = 1,
  indexMemory = 2000,
  fastqc_plots = FALSE,

  # Main analysis parameters
  EXPname = "",
  outdir = "./",
  ncontrols = 1,
  min_reads = 30,
  method = "ScalingByTotalReads",
  FDRth = 0.05,
```

```
    # Correction parameters
    min.ngenes = 3,
    alpha = 0.01,
    nperm = 10000,
    p.method ="hybrid",
    min.width = 2,
    kmax = 25,
    nmin = 200,
    eta = 0.05,
    trim = 0.025,
    undo.splits = "none",
    undo.prune = 0.05,
    undo.SD = 3,

    # Run MAGeCK
    run_mageck = FALSE,
    path_to_mageck = "mageck",

    # Other options undocumented
    is_web = FALSE,
    retrun_data = FALSE,
    nseed = 0xA5EED,
    verbose = -1,
    columns_map = c()
  )
```

Arguments

`library_builtin`
: A string containing the name of one of the library whose annotation is available in CRISPRcleanR.

`library_file`
: A string specifying the path to a file containing the sgRNA annotations in TXT / TSV format, with columns for sgRNA ID ("CODE"), targeted genes ("GENES"), genomic coordinates and possibly other information. This should be formatted as the `KY_Library_v1.0` data object containing the annotation of the sgRNA library presented in [1]. If FASTQ files are used as input, the annotation must include also a column "seq" with the nucleotidic sequence of the sgRNAs. This argument is ignored if a built-in library is specified.

`file_counts`
: A string specifying the path of a tsv file containing the raw sgRNA counts. This must be a tab delimited file with one row per sgRNA and the following columns/headers:

  - sgRNA: containing alphanumerical identifiers of the sgRNA under consideration;
  - gene: containing HGNC symbols of the genes targeted by the sgRNA under consideration;

  followed by the columns containing the sgRNAs' counts for the controls and columns for library transfected samples. The argument must be NULL if FASTQ

/ BAM files are specified as input.

`files_FASTQ_controls`
: List of FASTQ files used to generate the counts for the control samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name for the BAM files and in the count matrix. The argument must be NULL if counts / BAM files are specified as input.

`files_FASTQ_samples`
: List of FASTQ files used to generate the counts for the samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name for the BAM files and in the count matrix. The argument must be NULL if counts / BAM files are specified as input.

`files_BAM_controls`
: List of BAM files used to generate the counts for the control samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name in the count matrix. The argument must be NULL if counts / FASTQ files are specified as input.

`files_BAM_samples`
: List of BAM files used to generate the counts for the samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name in the count matrix. The argument must be NULL if counts / FASTQ files are specified as input.

`aligner`
: A string specifying the aligner used to map the reads to the libary. The possible options are "Rsubreads" (default) and "mageck" if the MAGeCK application [2] is installed. The parameter is ignored if the input are not FASTQ files.

`maxMismatches`
: Integer number containing the Ns allowed to count the reads. The function will consider as valid only the reads with a number of matched bases equal or greater than the length of the sgRNA sequence, provided in the annotation library, minus the maxMismatches parameter. The parameter is ignored if the input are not FASTQ / BAM files.

`nTrim5`
: Numeric value giving the number of bases trimmed off from 5' end of each read. 0 by default. If aligner = "Rsubreads" (default), the parameter accepts only numeric values. If MAGeCK is used, the parameter can specify multiple trimming lengths separated by comma (for example "7,8"), or can be set to "AUTO" to allow MAGeCK determine the trimming length. The parameter is ignored if the input are not FASTQ files.

`nTrim3`
: Numeric value giving the number of bases trimmed off from 3' end of each read. 0 by default. If aligner = "Rsubreads" (the default) the parameter accept only numeric values. The parameter is ignored if the input are not FASTQ files or if MAGeCK is used as aligner.

`nBestLocations`
: Numeric value specifying the maximal number of equally-best mapping locations that will be reported for a multi-mapping read (max 16). 2 by default. Different tabs will be included to the aligned sequecnes to specify the number of alignments reported. Please refer to the Rsubread [1] user guide for a compelete description. The parameter is ignored if the input are not FASTQ files or if MAGeCK is used as aligner.

`strand` A string specifying the strand of the alinement used to count the reads. It accepts three different options: "F" to count only the reads equal to the sgRNA sequence (default); "R" to count only the reads complementary to the sgRNA sequence; "*" all the reads without any strand filter. See the function description for details. The parameter is ignored if the input are not FASTQ files.

`duplicatedSeq` A string defining the strategy to deal with the duplicated sequences in the library index creation. See ccr.CreateLibraryIndex for details. The possible options are "keep" (the default) or "exclude". The "keep" option will maintain the first occurrence of the duplicates sequences while the "exclude" option will remove all the sgRNA whose nucleotidic sequence occur more than once. The parameter is ignored if the input are not FASTQ files.

`nthreads` Numeric value giving the number of threads used for mapping. 1 by default. The parameter is ignored if the input are not FASTQ files.

`indexMemory` A numeric value specifying the amount of memory (in megabytes) used for storing the index during read mapping. 2000 MB by default. The parameter is ignored if the input are not FASTQ files.

`fastqc_plots` A boolean value specifying if the QC plots for each FASTQ files will be generated during the alignment. The QC plots are created using the rqc function in the Rqc package. All the plots for each FASTQ file are collected in one HTML file named as the BAM file created in the alignment. The parameter is ignored if the input are not FASTQ files.

`EXPname` A string specifying the name of the experiment. This will be used to as label in the output plots.

`outdir` A string specifying folder where all the results will be saved. The function will create a "data" subfolder to store all the data in TSV format, a "pdf" subfolder to store all the plots in PRD format and an optional "bam" folder to store the BAM filed generated by the alignment of the reads if the input was based on FASTQ files.

`ncontrols` A numerical value used by the ccr.NormfoldChanges indicating the number of control replicates. It represents the columns to be considered as control counts after the first two, in the inputted tsv file. 1 by default. The parameter will not be considered when the input are FASTQ/BAM file. In this case the counts obtained by the files listed in files_FASTQ_controls / files_BAM_controls parameters will be used as controls.

`min_reads` A numerical value used by the ccr.NormfoldChanges to define a filter threshold value for sgRNAs, based on their average counts in the control sample. Specifically, it indicates the minimal number of counts that each individual sgRNA needs to have in the controls (on average) in order to be included in the output. 30 by default.

`method` A string specifying the normalisation method: 'ScalingByTotalReads' for scaling samples by total numbers or reads (default), 'MedRatios' to use the median of ratios method [1], or a gene name for scaling samples by total number of reads of the guides targeting that gene.

`FDRth` If different from NULL, will be a numerical value >=0 and <=1 specifying the false discovery rate threshold at which fixed recall will be computed. In this

case an horizontal dashed line will be added to the ROC and PrRc plots at the resulting recall and its value will be visualised in the legend. 0.05 by default

`min.ngenes` A numerical value (>0) used by the ccr.GWclean specifying the minimal number of different genes that the set of sgRNAs within a region of estimated equal logFCs should target in order for their logFCs to be corrected, i.e. mean centred. 3 by default.

`alpha` A numerical value used by the ccr.GWclean specifying the significance levels for the test to accept change-points (see DNAcopy). 0.01 by default.

`nperm` A numerical value used by the ccr.GWclean specifying the number of permutations used for p-value computation (see DNAcopy). 10000 by default.

`p.method` A string used by the ccr.GWclean specifying the method used for p-value computation. For the "perm" method the p-value is based on full permutation. For the "hybrid" method the maximum over the entire region is split into maximum of max over small segments and max over the rest. Approximation is used for the larger segment max. Default is hybrid (see DNAcopy).

`min.width` A numerical value used by the ccr.GWclean specifying the minimum number of markers for a changed segment. The default is 2 but can be made larger. Maximum possible value is set at 5 since arbitrary widths can have the undesirable effect of incorrect change-points when a true signal of narrow widths exists (see DNAcopy).

`kmax` A numerical value used by the ccr.GWclean specifying the maximum width of smaller segment for permutation in the hybrid method (see DNAcopy). 25 by default.

`nmin` A numerical value used by the ccr.GWclean specifying the minimum length of data for which the approximation of maximum statistic is used under the hybrid method. should be larger than 4*kmax (see DNAcopy). 200 by default.

`eta` A numerical value used by the ccr.GWclean specifying the probability to declare a change conditioned on the permuted statistic exceeding the observed statistic exactly j (= 1,...,nperm*alpha) times. (see DNAcopy). 0.05 by default.

`trim` A numerical value used by the ccr.GWclean specifying the proportion of data to be trimmed for variance calculation for smoothing outliers and undoing splits based on SD (see DNAcopy).

`undo.splits` A character string used by the ccr.GWclean specifying how change-points are to be undone, if at all. Default is "none". Other choices are "prune", which uses a sum of squares criterion, and "sdundo", which undoes splits that are not at least this many SDs apart. (see DNAcopy).

`undo.prune` A numerical value used by the ccr.GWclean specifying the proportional increase in sum of squares allowed when eliminating splits if undo.splits="prune" (see DNAcopy).

`undo.SD` A numerical value used by the ccr.GWclean specifying the number of SDs between means to keep a split if undo.splits="sdundo" (see DNAcopy).

`nseed` A numerical value used by the ccr.GWclean fixing the seed for the reproducibility of the results.

run_mageck	Boolean value specifying whether MAGeCK analysis [2] should be run on the counts file after the CRISPRcleanR correction [3]. This function requires python and the MAGeCK python package (v0.5.3, available at: https://sourceforge.net/projects/mageck/files/0.5/mageck-0.5.3.zip/download) to be installed.
path_to_mageck	If run_mageck is set to TRUE and the MAGeCK location is not included in the path, this option should be used to specify the MAGeCK aplication file including the full path.
is_web	Technical parameter undocumented. Used for the integration in a web architecture.
retrun_data	Technical parameter undocumented. Used for the integration in a web architecture.
verbose	Technical parameter undocumented. Used for the integration in a web architecture.
columns_map	Technical parameter undocumented. Used for the integration in a web architecture.

Value

A boolean value equal to TRUE if the function end without errors.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

References

[1] Tzelepis K, Koike-Yusa H, De Braekeleer E, et al *A CRISPR dropout screen identifies genetic vulnerabilities and therapeutic targets in acute myeloid leukaemia*. Cell Reports 2016 Oct 18;17(4):1193-1205

[2] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554.

[3] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

See Also

ccr.CreateLibraryIndex ccr.FASTQ2counts ccr.BAM2counts ccr.NormfoldChanges ccr.GWclean

Examples

```
## Not run:
## Define the list of FASTQ files to used for the alignment
fileCount <- file.path(
  system.file("extdata", package = "CRISPRcleanR"),
```

```
  "HT-29_counts.tsv"
)

## Run the alignment and extract the raw counts
status <- ccr.AnalysisPipeline(
  file_counts = fileCount,
  outdir='./HT29_COUNTS/',
  EXPname = 'HT29_counts',
  library_builtin = "KY_Library_v1.1",
  run_mageck = FALSE,
  ncontrols = 1
)

## End(Not run)
```

`ccr.BAM2counts` *Raw count file extraction from BAM file list.*

Description

This function takes as input a list of BAM files and reads them to generate a raw count matrix with one column for each BAM file in the list. Based on the GenomicAlignments package [1], the function requires that the BAMs were created by aligning the FASTQ files to a genome file in which the seqnames match the sgRNA ID. The same IDs should be available in the CODE column from the library annotation data frame supplied as input. The function allows the user to select which strand to read the data from. By default, the function counts only the reads with the same sequence of the sgRNA. However, depending on the strategy used to create the BAM files, the user can select to match only the reads with nsequnces complementary to the sgRNAs or to count the reads without any strand-related filter.

Usage

```
ccr.BAM2counts(
  BAMfileList,
  libraryAnnotation,
  maxMismatches = 0,
  strand = c("F", "R", "*"),
  EXPname = "",
  outdir = "./",
  export_counts = TRUE,
  overwrite = TRUE
)
```

Arguments

`BAMfileList`	List of BAM files used to generate the count file. Each file should include the path to the BAM. If present, the name of each element of the list will be used as sample name in the count matrix.

`libraryAnnotation`
: A data frame containing a sgRNAs library. This data frame must include one named row per each sgRNA and the at least following mandatory columns/headers:

  - CODE: the unique ID of the sgRNA;
  - GENES: the gene symbol related to the sgRNA;
  - seq: the nucleotidic sequence of the sgRNA without PAM.

  All the built-in libraries included in the package are already compliant with this structure.

`maxMismatches`
: Integer number containing the Ns allowed to count the reads. The function will consider as valid only the reads with a number of matched bases equal or greater than the length of the sgRNA sequence, provided in the annotation library, minus the maxMismatches parameter.

`strand`
: A string specifying the strand of the alinement used to count the reads. It accepts three different options: "F" to count only the reads equal to the sgRNA sequence (default); "R" to count conly the reads complementary to the sgRNA sequence; "*" to count all the reads without any strand filter. See the function description for details.

`EXPname`
: A string specifying the name of the experiment. This will be used to create the raw count file if the export_counts option is set to TRUE.

`outdir`
: A string specifying folder where the raw count file will be created if the export_counts option is set to TRUE.

`export_counts`
: A boolean value specifying if the raw count matrix should also be exported in a TXT file. TRUE by default.

`overwrite`
: A boolean value specifying if the raw count file will overwrite any file with the same name already present in the outdir path (only when export_counts option is set to TRUE).

Value

A dataframe with the raw counts related to each sample. The first two columns in these data frame contain sgRNAs' identifiers and HGNC symbols of target gene, respectively.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

References

[1] Lawrence, M., Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., et al. (2013). Software for Computing and Annotating Genomic Ranges. PLoS Computational Biology, 10.1371/journal.pcbi.1003118

See Also

`ccr.FASTQ2counts`

`ccr.checkCounts` *Check consistency between library annotation and count files.*

Description

This function takes as input the sgRNA library annotation and the counts matrix to ensure that they are consistent. This a utility function that runs automatically as part of the ccr.AnalysisPipeline.

Usage

```
ccr.checkCounts(
  counts,
  libraryAnnotation,
  ncontrols = 1,
  min_reads = 30
)
```

Arguments

`counts` A data frame containing the raw sgRNA counts (usable as an alternative to providing the path to a tsv file, i.e. previous argument). This must have one row per sgRNA and the following columns/headers:

- sgRNA: containing alphanumerical identifiers of the sgRNA under consideration;
- gene: containing HGNC symbols of the genes targeted by the sgRNA under consideration;

followed by the columns containing the sgRNAs' counts for the controls and columns for library trasfected samples.

`libraryAnnotation` A data frame containing a sgRNAs library. This data frame must include one named row per each sgRNA and the at least following mandatory columns/headers:

- CODE: the unique ID of the sgRNA;
- GENES: the gene symbol related to the sgRNA;
- seq: the nucleotidic sequnce of the sgRNA without PAM

All the built-in libraries included in the package are already compliant with this structure.

`ncontrols` A numerical value used by the ccr.NormfoldChanges indicating the number of control replicates (therefore columns to be considered as control counts after the first two, in the inputted tsv file). 1 by default. The parameter will not be considered when the input are FASTQ / BAM files. In this case, the counts obtained by the files listed in files_FASTQ_controls / files_BAM_controls parameters will be used as controls.

`min_reads` A numerical value used by the ccr.NormfoldChanges to define a filter threshold value for sgRNAs, based on their average counts in the control sample. Specifically, it indicates the minimal number of counts that each individual sgRNA needs to have in the controls (on average) in order to be included in the output. 30 by default.

Value

A boolean value equal to TRUE if the function ends without errors.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

See Also

`ccr.AnalysisPipeline`

`ccr.cleanChrm` *Identification and correction of genomic regions of equal log fold changes involving sgRNAs targeting a minimal number of genes within a given chromosome.*

Description

This function applies a circular binary segmentation algorithm [1, 2] to genomic-sorted log fold changes of all the sgRNAs targeting genes on the same chromosome. This procedure yields a sets of genomic regions of estimated equal sgRNAs' log fold changes, significantly differing on average from adjacent regions. If some of these regions fulfill certain criteria (detailed below) then they are deemed as responding to CRISPR-Cas9 targeting in a gene independent manner, i.e. they might be biased by local feature of the DNA) and their pattern of log fold changes is mean centered [3].

Usage

```
ccr.cleanChrm(gwSortedFCs,CHR,display=TRUE,label='',
              saveTO=NULL,min.ngenes=3,ignoredGenes=NULL,
              capped = FALSE,corrMet = 'mean',alpha = 0.01,
              nperm = 10000,p.method ="hybrid",min.width=2,
              kmax=25,nmin=200,eta=0.05,trim = 0.025,
              undo.splits = "none",undo.prune=0.05,
              undo.SD=3)
```

Arguments

`gwSortedFCs` A data frame containing genome-wide genomic-sorted sgRNAs' log fold changes. This data frame must include one named row per each sgRNAs and the following columns/headers:

- CHR: the chromosome of the gene targeted by the sgRNA under consideration;
- startp: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- endp: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- genes: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- avgFC: the log fold change of the sgRNA under consideration averaged across replicates;
- BP: the genomic coordinate of the sgRNA defined as STARTpos+(ENDpos-STARTpos)/2.

This can be generated using the `ccr.logFCs2chromPos` function, starting from a data frame containing sgRNAs' log fold changes generated by the `ccr.NormfoldChanges` function from raw sgRNAs' counts.

`CHR` Numerical value indicating the chromosome to analyse and correct. X and Y chromosome must be indicated with 23 and 24, respectively.

`display` A logical value indicating whether genomic plots showing the results of the biased regions' identification and their log fold change correction should be generated or not.

`label` A string indicating the experiment name, used in the main title of the plots and for the name of the folder where results are saved.

`saveTO` If different from NULL then it will contain the path where pdf files with then genomic plots showing the results of the biased regions' identification (and their log fold change correction) will be saved (within a folder named as defined in the `label` parameter).

`min.ngenes` A numerical value (>0) specifying the minimal number of different genes that the set of sgRNAs within a region of estimated equal log fold changes should target in order for that region to be corrected, i.e. mean centered.

`ignoredGenes` A vector of strings containing HGNC symbols of genes that should not be considered when computing the minimal number of different genes targeted by the sgRNAs in the same identified region of estimated equal log fold changes. This vector could contain, for example, a priori known essential genes. This parameter should be set to NULL (default value) for a completely unsupervised correction.

`capped` Boolean argument that if TRUE prevents the sgRNAs changing the sign of their logFC due to the correction, by capping corresponding values to 0. By default is FALSE.

`corrMet` String specifying the correction to be applied, if equal to 'mean' (its default value) than the mean of the sgRNA logFC in a biased segment is subtracted to the logFCs of all the sgRNA in the same biased segment. If different from 'mean' then the median of the sgRNA logFC in a biased segment is subtracted to the logFCs of all the sgRNA in the same biased segment.

`alpha` significance levels for the test to accept change-points (see DNAcopy).

`nperm` number of permutations used for p-value computation (see DNAcopy).

p.method	method used for p-value computation. For the "perm" method the p-value is based on full permutation. For the "hybrid" method the maximum over the entire region is split into maximum of max over small segments and max over the rest. Approximation is used for the larger segment max. Default is hybrid (see DNAcopy).
min.width	the minimum number of markers for a changed segment. The default is 2 but can be made larger. Maximum possible value is set at 5 since arbitrary widths can have the undesirable effect of incorrect change-points when a true signal of narrow widths exists (see DNAcopy).
kmax	the maximum width of smaller segment for permutation in the hybrid method (see DNAcopy).
nmin	the minimum length of data for which the approximation of maximum statistic is used under the hybrid method. should be larger than 4*kmax (see DNAcopy).
eta	the probability to declare a change conditioned on the permuted statistic exceeding the observed statistic exactly j (= 1,...,nperm*alpha) times. (see DNAcopy).
trim	proportion of data to be trimmed for variance calculation for smoothing outliers and undoing splits based on SD (see DNAcopy).
undo.splits	a character string specifying how change-points are to be undone, if at all. Default is "none". Other choices are "prune", which uses a sum of squares criterion, and "sdundo", which undoes splits that are not at least this many SDs apart. (see DNAcopy).
undo.prune	the proportional increase in sum of squares allowed when eliminating splits if undo.splits="prune" (see DNAcopy).
undo.SD	the number of SDs between means to keep a split if undo.splits="sdundo" (see DNAcopy).
	The rest of the arguments are passed to the segment function of the DNAcopy package as they are.

Value

A list containing two data frames. The first one (correctedFCs) contains a named row per each sgRNA and the following columns/header:

- CHR: the chromosome of the gene targeted by the sgRNA under consideration;
- startp: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- endp: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- genes: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- avgFC: the log fold change of the sgRNA averaged across replicates;
- correction: the type of correction: 1 = increased, -1 = decreased;
- correctedFC: the corrected log fold change of the sgRNA

.

The second one (regions) contains the identified region of estimated equal log fold changes (one region per row) and the following columns/headers:

- CHR: the chromosome of the region under consideration;
- startp: the genomic coordinate of the starting position of the region under consideration;
- endp: the genomic coordinate of the ending position of the region under consideration;
- n.sgRNAs: the number of sgRNAs targeting sequences in the region under consideration;
- avg.logFC: the average log fold change of the sgRNAs targeting the region;
- guideIdx: the indexes range of the sgRNAs targeting the region under consideration as they appear in the gwSortedF Cs provided in input.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Olshen, A. B., Venkatraman, E. S., Lucito, R., Wigler, M. (2004). *Circular binary segmentation for the analysis of array-based DNA copy number data*. Biostatistics 5: 557-572.

[2] Venkatraman, E. S., Olshen, A. B. (2007). *A faster circular binary segmentation algorithm for the analysis of array CGH data*. Bioinformatics 23: 657-63.

[3] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

See Also

ccr.logFCs2chromPos, ccr.NormfoldChanges

Examples

```
## Not run:
data(KY_Library_v1.0)

fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),'/HT-29_counts.tsv',sep='')
normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,EXPname='Example',
                                libraryAnnotation=KY_Library_v1.0)

gwSortedFCs<-ccr.logFCs2chromPos(normANDfcs$logFCs,KY_Library_v1.0)

chr8cleaned<-ccr.cleanChrm(gwSortedFCs,8,display=TRUE,label='HT-29',
                       min.ngenes=3)

## End(Not run)
```

`ccr.correctCounts` *Correction of sgRNA treatment counts for gene independent responses to CRISPR-Cas9 targeting*

Description

This function applies an inverse transformation (described in [1]) to CRISPRcleanR corrected sgRNAs' log fold changes and produces in output normalised corrected sgRNA counts (across treatments and control replicates), suitable for gene depletion/enrichment statistical testing via mean-variance modeling (for example through MAGeCK [2]*). *MAGeCK should be executed excluding initial normalisation, as the corrected sgRNA counts outputted by this function are already normalised.

Usage

```
ccr.correctCounts(CL,normalised_counts,
                  correctedFCs_and_segments,
                  libraryAnnotation,
                  minTargetedGenes=3,
                  OutDir='./',
                  ncontrols=1)
```

Arguments

`CL` A string specifying the name of the experiment. This will be used to compose names of files and folde where results will be saved.

`normalised_counts`
A data frame containing normalised sgRNAs' read counts, which can be computed using the `ccr.NormfoldChanges` function from raw sgRNAs' counts.

`correctedFCs_and_segments`
sgRNAs log fold changes corrected for gene independent responses, generated with the function `ccr.GWclean`.

`libraryAnnotation`
A data frame containing the sgRNAs' genome-wide annotations with at least a named row for each of the sgRNAs included in the foldchanges data frame provided in input. The following columns/headers should be present in this data frame (additional columns will be ignored):

- GENES: string vector containing the HGNC symbols of the genes targeted by the sgRNA under consideration;
- EXONE: string vector containing the gene exon targeted by the sgRNA under consideration (these should include the prefix "ex" followed by the exone number);
- CHRM: string vector the chromosome of the gene targeted by the sgRNA under consideration (X and Y chromosome should be specified as "X" and "Y");

- STRAND: string vector containing the strand targeted by the sgRNA under consideration ("+" or "-");
- STARTpop: numeric vector containing the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- ENDpos: numeric vector containing the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;

`minTargetedGenes`
: Minimanl number of different genes targeted by sgRNAs in a biased segment in order for the corresponding counts to be corrected (default = 3).

`OutDir`
: Path of the folder where results and plots will be saved.

`ncontrols`
: A numerical value indicating the number of control replicates (therefore columns to be considered as controls in the normalised counts).

Value

A data frame with one entry per sgRNA and individual columns for the control/treatment samples included in the normalised count data object specified by the `normalised_counts` parameter, and containing sgRNA counts corrected for gene independent responses to CRISPR-Cas9 targeting and median-ratio normalised.

Author(s)

Francesco Iorio (francesco.iorio@fht.orgfht.org)

References

[1] Iorio F, Behan FM, Goncalves E, Bhosle SG, Chen E, Shepherd R, Beaver C, Ansari R, Pooley R, Wilkinson P, Harper S, Butler AP, Stronach EA, Saez-Rodriguez J, Yusa K, Garnett MJ. Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. BMC Genomics. 2018 Aug 13;19(1):604. doi: 10.1186/s12864-018-4989-y.

[2] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554.

See Also

`ccr.NormfoldChanges`, `ccr.GWclean`

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Deriving the path of the file with the example dataset,
## from the mutagenesis of the EPLC-272H colorectal cancer cell line
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
                      '/EPLC-272H_counts.tsv',sep='')
```

```
## Loading, median-normalizing and computing fold-changes for the example dataset
normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,
                                EXPname='EPLC-272H',
                                libraryAnnotation = KY_Library_v1.0)

## Genome-sorting of the fold changes
gwSortedFCs<-ccr.logFCs2chromPos(normANDfcs$logFCs,KY_Library_v1.0)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs<-ccr.GWclean(gwSortedFCs,display=FALSE,label='EPLC-272H')

## correcting individual sgRNA treatment counts

correctedCounts<-ccr.correctCounts('EPLC-272H',normANDfcs$norm_counts,
                  correctedFCs,
                  KY_Library_v1.0,
                  minTargetedGenes=3,
                  OutDir='./')

head(correctedCounts)

## End(Not run)
```

`ccr.CreateLibraryIndex` *Library index generation based on the sgRNA library to allow FASTQ files alignment.*

Description

This function takes in input a data frame with the structure of a sgRNA library (e.g. KY) to create a library index file used for the alignment of the FASTQ files. It uses the "buildindex" function of the Rsubread [1]. The data frame must include a "seq" field containing the nucleotidic sequences of the sgRNA that will be used to create the library. The function creates in the current folder a set of files related to the library index. All the files will include the EXPname label and will be prefixed with the "Library_" tag. Moreover, the function will create an additional FA file reporting all the sgRNA sequences in the FASTA format and a TXT file that can be used for the alignment using MAGeCK [2]. The sgRNA library includes sequences that are common to more than one sgRNA to avoid multiple alignments of the same read targeting those sequences. By default, the function keeps only the first of the instances. Since the presence of a sgRNA targeting multiple genes might represent an undesirable source of noise, the function allows the user to completely exclude all the sgRNAs whose sequences appear more than once in the library.

Usage

```
ccr.CreateLibraryIndex(
  libraryAnnotation,
  duplicatedSeq = c("keep", "exclude"),
```

```
  EXPname = "",
  indexMemory = 2000,
  overwrite = FALSE
)
```

Arguments

libraryAnnotation	A data frame containing a sgRNAs library. This data frame must include one named row per each sgRNA and the at least following mandatory columns/headers: • CODE: the unique ID of the sgRNA; • GENES: the gene symbol related to the sgRNA; • seq: the nucleotidic sequence of the sgRNA without PAM. All the built-in libraries included in the package are already compliant with this structure.
duplicatedSeq	A string defining the strategy to deal with the duplicated sequences that might be present in the library. The possible options are "keep" (the default) or "exclude". The "keep" option will maintain the first occurrence of the duplicated sequences, while the "exclude" option will remove all the sgRNA whose nucleotidic sequences occur more than once.
EXPname	A string specifying the name of the experiment. This will be included in all the library files created by the function.
indexMemory	A numeric value specifying the amount of memory (in megabytes) used for storing the index during read mapping. 2000 MB by default.
overwrite	A boolean value specifying whether the index files should be overwritten, in case theyalready exist. FALSE by default.

Value

The name of the library file.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

References

[1] Liao, Y., Smyth, G.K., Shi, W. (2019). The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. Nucleic Acids Research, 47, e47 DOI:10.1093/nar/gkz114

[2] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554. [2] Hart, T., & Moffat, J. (2016). BAGEL: a computational framework for identifying essential genes from pooled library screens. BMC Bioinformatics, 17(1), 164.

See Also

ccr.FASTQ2counts

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Create the index file based on the KY 1.0 libary
fn <- ccr.CreateLibraryIndex(KY_Library_v1.0)


## End(Not run)
```

`ccr.ExecuteMageck` *Executing MAGeCK from R command line*

Description

This function executes MAGeCK [1] from the command line, taking in input the path of the file containing the sgRNA counts' file to be processed and saving the results in a user defined location. By default this function do not pre-normalise the counts. However this preliminary step can be included as specified by the corresponding argument. Additionally this function assumes that there is only one control sample, whose count values should be contained in the first column of the sgRNA counts' file. This function requires python and the MAGeCK python package (v0.5.3, available at: https://sourceforge.net/projects/mageck/files/0.5/mageck-0.5.3.zip/download) to be installed.

Usage

```
ccr.ExecuteMageck(mgckInputFile,
                  expName = "expName",
                  normMethod = "none",
                  outputPath = "./")
```

Arguments

`mgckInputFile`	A string specifying the path of the (plain text) file containing the sgRNA counts' file to be processed
`expName`	A string specifying the experiment name. This is used as name prefix for all the files generated by MAGeCK.
`normMethod`	A string specifying the normalisation method to be used ('none' by default).
`outputPath`	A string specifying the folder where all the files outputted by MAGeCK will be saved.

Value

A string specifying the path to the gene summary file outputted by MAGeCK.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554. [2] Hart, T., & Moffat, J. (2016). BAGEL: a computational framework for identifying essential genes from pooled library screens. BMC Bioinformatics, 17(1), 164.

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Deriving the path of the file with the example dataset,
## from the mutagenesis of the EPLC-272H colorectal cancer cell line
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/EPLC-272H_counts.tsv',sep='')

## Loading, median-normalizing and computing fold-changes for the example dataset
normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,
                                EXPname='EPLC-272H',
                                libraryAnnotation = KY_Library_v1.0)

uncorrected_fn<-ccr.PlainTsvFile(sgRNA_count_object = normANDfcs$norm_counts,
                                 fprefix = 'EPLC-272H')

## execute MAGeCK saving files in the working directory
uncorrected_gs_fn<-ccr.ExecuteMageck(mgckInputFile = uncorrected_fn,
                                     expName = 'EPLC-272H',
                                     normMethod = 'none')

uncorrected_gs_fn


## End(Not run)
```

`ccr.FASTQ2counts` *FASTQ files alignment and raw count file extraction.*

Description

This function take as input a list of FASTQ files and align them the sequences of the sgRNA library using by default the Rsubread [1] align function. The process generates a list of BAM files that are than used to extract a raw count file with one column for each FASTQ file in the list. If Rsubread is used for the alignment, the sorted BAM files will be generated in the outdir path. If MAGeCK [2] is installed, it can be specified as alternative aligner through its count function. In this case,

some of the parameters might be unavailable and no BAM files will be created. The annotation data frame must include a "seq" field containing the nucleotidic sequences of the sgRNA that will be used to create the index library files for the alignment. By default the function allows the alignment of the reads in two different locations (nBestLocations = 2), while the duplications are removed in a second step based on the strand (strand = "F" of strand = "R" if the sequences in the library are complementary to the sgRNA sequences). This stragy allows the the identification of correct sgRNA in case of guides based on complementary sequnces. This alignment strtegy works optimally when no mismatchs (maxMismatches = 0) are allowed and the reads are trimmed to exact length of the sgRNA length. If the Ns are allowed in the alignment (maxMismatches > 0) or the reads after trimming are longer than the sgRNA sequences a more robust approach is based on the selection of only one locations (nBestLocations = 1) removing any filter based on the strand (strand = "*"). As part of the alignment process the function will generate by default quality reports based on the Rqc package [3].

Usage

```
ccr.FASTQ2counts(
  FASTQfileList,
  libraryAnnotation,
  maxMismatches = 0,
  nTrim5 = "0",
  nTrim3 = "0",
  nthreads = 1,
  nBestLocations = 2,
  strand = c("F", "R", "*"),
  indexMemory = 2000,
  duplicatedSeq = c("keep", "exclude"),
  EXPname = "",
  outdir = "./",
  aligner = c("Rsubreads", "mageck"),
  fastqc_plots = TRUE,
  export_counts = TRUE,
  overwrite = FALSE
)
```

Arguments

`FASTQfileList` List of BAM files used to generate the count file. Each file should include the path to the FASTQ. If present, the name of each element of the list will be used as sample name for the BAM files and in the count matrix.

`libraryAnnotation`
A data frame containing a sgRNAs library. This data frame must include one named row per each sgRNA and the at least following mandatory columns/headers:

- CODE: the unique ID of the sgRNA;
- GENES: the gene symbol related to the sgRNA;
- seq: the nucleotic sequnce of the sgRNA without PAM

All the built-in libraries included in the package are already compliant with this structure.

`maxMismatches` Integer number containing the Ns allowed to count the reads. The function will consider as valid only the reads with a number of matched bases equal or greater than the length of the sgRNA sequence, provided in the annotation library, minus the maxMismatches parameter.

`nTrim5` Numeric value giving the number of bases trimmed off from 5' end of each read. 0 by default. If aligner = "Rsubreads" (default), the parameter accept only numeric values. If MAGeCK is used, the parameter can specify multiple trimming lengths separated by comma (for example "7,8"), or can be set to "AUTO" to allow MAGeCK determine the trimming length.

`nTrim3` Numeric value giving the number of bases trimmed off from 3' end of each read. 0 by default. If aligner = "Rsubreads" (default), the parameter accepts only numeric values. If MAGeCK is used the parameter is ignored.

`nthreads` Numeric value giving the number of threads used for mapping. 1 by default.

`nBestLocations` Numeric value specifying the maximal number of equally-best mapping locations that will be reported for a multi-mapping read (max 16). 2 by default. Different tabs will be included to the aligned sequences to specify the number of alignments reported. Please refer to the Rsubread [1] user guide for a complelte description. If MAGeCK is used the parameter is ignored.

`strand` A string specifying the strand of the alinement used to count the reads. It accepts three different options: "F" to count only the reads equal to the sgRNA sequence (default); "R" to count only the reads complementary to the sgRNA sequence; "*" to count all the reads without any strand filter. See the function description for details.

`indexMemory` A numeric value specifying the amount of memory (in megabytes) used for storing the index during read mapping. 2000 MB by default.

`duplicatedSeq` A string defining the strategy to deal with the duplicated sequences in the library index creation. See ccr.CreateLibraryIndex for details. The possible options are "keep" (the default) or "exclude". The "keep" option will maintain the first occurrence of the duplicates sequences while the "exclude" option will remove all the sgRNA whose nucleotidic sequences occur more than once. The parameter is ignored if the input are not FASTQ files.

`EXPname` A string specifying the name of the experiment. This will be used to create the raw count file if the export_counts option is set to TRUE.

`outdir` A string specifying the folder where the raw count file will be created if the export_counts option is set to TRUE.

`aligner` A string specifying the aligner used to map the reads to the libary. The possible options are "Rsubreads" (default) and "mageck" (if the MAGeCK application [2] is installed).

`fastqc_plots` A boolean value specifying if the QC plots for each FASTQ files will be generated during the alignment. The QC plots are created using the rqc function in the Rqc package. All the plots for each FASTQ file are collected in one HTML file named as the BAM file created in the alignment.

export_counts	A boolean value specifying if the raw count matrix should also be exported in a TXT file. TRUE by default.
overwrite	A boolean value specifying if the files generated by the alignment will overwrite any file with the same name already present in the outdir path.

Value

A data.frame containing the raw counts related to each sample. The first two columns in these data frame contain sgRNAs' identifiers and HGNC symbols of target gene, respectively.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

References

[1] Liao, Y., Smyth, G.K., Shi, W. (2019). The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. Nucleic Acids Research, 47, e47 DOI:10.1093/nar/gkz114

[2] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554.

[3] de Souza, W., Carvalho, B.S., Lopes-Cendes, I. (2018). Rqc: A Bioconductor Package for Quality Control of High-Throughput Sequencing Data. Journal of Statistical Software, Code Snippets, 87(2), 1–14. DIO:10.18637/jss.v087.c02

See Also

ccr.BAM2counts ccr.CreateLibraryIndex

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Define the list of FASTQ files to used for the alignment
fileList <- file.path(
  system.file("extdata", package = "CRISPRcleanR"),
  c("test_plasmid.fq.gz", "test_sample1.fq.gz", "test_sample2.fq.gz")
)

## Run the alignment and extract the raw counts
fn <- ccr.FASTQ2counts(
  FASTQfileList = fileList,
  libraryAnnotation = KY_Library_v1.0
)

## End(Not run)
```

`ccr.geneMeanFCs` *Gene level log fold changes*

Description

This functions computes gene level log fold changes based on average log fold changes of targeting sgRNAs

Usage

```
ccr.geneMeanFCs(sgRNA_FCprofile, libraryAnnotation)
```

Arguments

`sgRNA_FCprofile`
: A named numerical vector containing the sgRNAs' log fold-changes, with names corresponding to sgRNAs identifiers.

`libraryAnnotation`
: A data frame containing the sgRNA library annotation (with same format of `KY_Library_v1.0`).

Value

A numerical vector containing gene average log fold-changes, with corresponding HGNC symbols as names.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

See Also

`KY_Library_v1.0`

Examples

```
## loading corrected sgRNAs log fold-changes and segment annotations for
## an example cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)


## loading sgRNA library annotation
data(KY_Library_v1.0)

## storing sgRNA log fold-changes in a named vector
FCs<-EPLC.272HcorrectedFCs$corrected_logFCs$avgFC
names(FCs)<-rownames(EPLC.272HcorrectedFCs$corrected_logFCs)

## computing gene level log fold-changes
```

```
geneFCs<-ccr.geneMeanFCs(FCs,KY_Library_v1.0)

head(geneFCs)
```

`ccr.genes2sgRNAs` *Targeting sgRNAs*

Description

This function returns the set of sgRNAs targeting the set of genes provided in input, in a given pooled library.

Usage

```
ccr.genes2sgRNAs(libraryAnnotation,genes)
```

Arguments

`libraryAnnotation`	A data frame with a named row for each sgRNA with the same format of `KY_Library_v1.0`
`genes`	A list of strings containing HGNC symbols

Value

A list of strings containing the identifiers of the sgRNAs targeting the inputted set of genes

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

See Also

`KY_Library_v1.0`

Examples

```
## Loading an sgRNA pooled library annotation
data(KY_Library_v1.0)
## Loading an example set of genes
data(BAGEL_essential)

ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_essential)
```

`ccr.geneSummary` *Gene level depletion summary*

Description

This function collapses single-guide RNAs (sgRNAs) depletion log fold-changes (logFCs) on a targeted gene basis, by averaging (using the `ccr.geneMeanFCs`). In addition it computes also a logFC threshold T such that when considering as significantly depleted all the genes with a depletion logFC $< T$, the false discover rate (FDR) of prior known non-essential genes is below a given threshold (specificed in input). Finally it calls significantly depleted genes according to the computed threshold.\ The significant threshold is computed using the `ccr.PrRc_Curve` function, employing a reference list of core-fitness essential genes and a list of non-essential genes, assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance (included as the built-in objects `BAGEL_essential` and `BAGEL_nonEssential`) [1].

Usage

```
ccr.geneSummary(sgRNA_FCprofile,
                libraryAnnotation,
                FDRth=0.05)
```

Arguments

`sgRNA_FCprofile`	A named numerical vector containing the sgRNAs' log fold-changes, with names corresponding to sgRNAs identifiers.
`libraryAnnotation`	A data frame containing the sgRNA library annotation (with same format of `KY_Library_v1.0`)
`FDRth`	The FDR threshold to consider in order to derive the significance threshold (FDR 5% by default)

Value

A data frame with gene symbols as row names and two columns: the first one indicating the gene depletion logFC and the second one including a boolean value specifying if the gene under consideration is significantly depleted at the indicated FDR level.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] BAGEL: a computational framework for identifying essential genes from pooled library screens. Traver Hart and Jason Moffat. BMC Bioinformatics, 2016 vol. 17 p. 164.

See Also

KY_Library_v1.0, ccr.geneMeanFCs, ccr.PrRc_Curve, ccr.PrRc_Curve, BAGEL_essential, BAGEL_nonEssential

Examples

```
## loading corrected sgRNAs log fold-changes and segment annotations for
## an example cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)

## loading sgRNA library annotation
data(KY_Library_v1.0)

## storing sgRNA log fold-changes in a named vector
FCs<-EPLC.272HcorrectedFCs$corrected_logFCs$avgFC
names(FCs)<-rownames(EPLC.272HcorrectedFCs$corrected_logFCs)

## computing gene level log fold-changes
geneFCs<-ccr.geneSummary(FCs,KY_Library_v1.0)
```

---

ccr.get.CCLEgisticSets

CCLE gistic score gene sets

Description

This function splits all the genes into 5 classes (-2, -1, 0, +1 and +2) based on the CNA Gistic [1] score observed in a given cell line.

Usage

```
ccr.get.CCLEgisticSets(cellLine,CCLE.gisticCNA=NULL,GDSC.CL_annotation=NULL)
```

Arguments

cellLine	A string specifying the name of a cell line (or a COSMIC identifier [2]);
CCLE.gisticCNA	Genome-wide Gistic [1] scores quantifying copy number status across cell lines with the same format of CCLE.gisticCNA. If NULL then this function uses the CCLE.gisticCNA builtin data frame, containing data for 13 cell lines of the 15 used in [3] to assess the performances of CRISPRcleanR.
GDSC.CL_annotation	Cell lines annotation dataframe with the same structure of the GDSC.CL_annotation. If NULL then the GDSC.CL_annotation is used.

Value

A named list of vectors with the following fields:

gm2	A vector of strings containing identifiers of sgRNAs targeting genes whit a Gistic score = -2 in the cell line under consideration;
gm1	A vector of strings containing identifiers of sgRNAs targeting genes whit a Gistic score = -1 in the cell line under consideration;
gz	A vector of strings containing identifiers of sgRNAs targeting genes whit a Gistic score = 0 in the cell line under consideration;
gp1	A vector of strings containing identifiers of sgRNAs targeting genes whit a Gistic score = +1 in the cell line under consideration;
gp2	A vector of strings containing identifiers of sgRNAs targeting genes whit a Gistic score = +2 in the cell line under consideration;

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Mermel CH, Schumacher SE, Hill B, et al. *GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers.* Genome Biol. 2011;12(4):R41. doi: 10.1186/gb-2011-12-4-r41.

[2] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783,

[3] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

See Also

ccr.get.gdsc1000.AMPgenes

Examples

```
GS<-ccr.get.CCLEgisticSets('HT-29')

head(GS$gm2)
head(GS$gm1)
head(GS$gz)
head(GS$gp1)
head(GS$gp2)
```

`ccr.get.gdsc1000.AMPgenes` *Copy number amplified genes in a given cell line from the GDSC1000*

Description

This function takes in input the name (or the COSMIC identifier [1]) of a cell line included in the GDSC1000 project [2] and it identifies the genes that are copy number amplified (according to a user defined minimal copy number value) in that cell line, using gene level copy number data from the Genomics of Drug Sensitivity in 1,000 Cancer Cell lines (GDSC1000) [2].

Usage

```
ccr.get.gdsc1000.AMPgenes(cellLine, minCN = 8, exact = FALSE,
                          GDSC.geneLevCNA=NULL,GDSC.CL_annotation=NULL)
```

Arguments

`cellLine`	A string specifying the name of a cell line (or a COSMIC identifier [1]);
`minCN`	Lower threshold for the minimum copy number of any genomic segment containing coding sequence of a gene in order for it to be considered as copy number amplified.
`exact`	If TRUE, then those genes for which any genomic segment containing coding sequence has a minimum copy number equal to `minCN` are considered as copy number amplified.
`GDSC.geneLevCNA`	Genome-wide copy number data with the same format of `GDSC.geneLevCNA`. This can be assembled from the xls sheet specified in the source section [a] (containing data for the GDSC1000 cell lines). If NULL, then this function uses the data in the built in `GDSC.geneLevCNA` data frame, containing data derived from [a] for 15 cell lines used in [3] to assess the performances of CRISPRcleanR.
`GDSC.CL_annotation`	Cell lines annotation dataframe with the same structure of the `GDSC.CL_annotation`. If NULL then the `GDSC.CL_annotation` is used.

Value

A data frame, containing one row for each copy number amplified gene with the following columns:

`Gene`	HGNC symbol of the gene;
`minCN`	Minimum copy number of any genomic segment containing coding sequence of the gene in the cell line under consideration.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

Source

[a] ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-6.0/Gene_level_CN.xlsx.

References

[1] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783,

[2] Iorio F, Knijnenburg TA, Vis DJ, Bignell GR, Menden MP, et al. *A landscape of pharmacogenomic interactions in cancer* Cell 2016 Jul 28;166(3):740-54

[3] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

See Also

ccr.get.CCLEgisticSets

Examples

```
CNAgenes<-
    ccr.get.gdsc1000.AMPgenes('HT-29')
head(CNAgenes)
```

---

ccr.get.nonExpGenes *Non expressed genes in a given cell line*

Description

This function takes in input the name (or the COSMIC identifier [1]) of a cell line and it identifies genes that are not expressed (according to a user defined FPKM threshold) using a collection of RNAseq profile from [2].

Usage

```
ccr.get.nonExpGenes(cellLine, th = 0.05,
                    amplified = FALSE, minCN = 8,
                    RNAseq.fpkms=NULL,GDSC.CL_annotation=NULL)
```

Arguments

cellLine	A string specifying the name of a cell line (or a COSMIC identifier [1]);
th	Minimum FPKM value for a gene to be considered as expressed;
amplified	A logic value specifying whether the selected not expressed genes should be also copy number amplified function;
minCN	If amplified = TRUE, this parameter defines a lower threshold for the minimum copy number of any genomic segment containing coding sequence of a gene in order for it to be considered as copy number amplified.
RNAseq.fpkms	Genome-wide substitute reads with fragments per kilobase of exon per million reads mapped (FPKM) across cell lines. These can be derived from a comprehensive collection of RNAseq profiles described in [2]. The format must be the same of the RNAseq.fpkms builtin data frame. If NULL then this function uses the RNAseq.fpkms builtin data fram containing data for 15 cell lines used in [3] to assess CRISPRcleaneR results.
GDSC.CL_annotation	Cell lines annotation dataframe with the same structure of the GDSC.CL_annotation. If NULL then the GDSC.CL_annotation is used.

Value

A vector of string containing the HGNC symbols of non expressed (optionally copy number amplified) genes in the cell line under consideration.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783.

[2] Garcia-Alonso L, Iorio F, Matchan A, et al. *Transcription factor activities enhance markers of drug response in cancer* doi: https://doi.org/10.1101/129478

[3] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

See Also

ccr.get.gdsc1000.AMPgenes

Examples

```
ccr.get.nonExpGenes('HT-29',amplified = TRUE)
```

`ccr.getCounts` *Convert count inputs in a standard count matrix format.*

Description

The pipeline takes as input raw count/matrix or lists of FASTQ/BAM files to return a data frame containing all samples counts. This is a utility function that runs automatically as part of the ccr.AnalysisPipeline.

Usage

```
ccr.getCounts(
  file_counts,
  files_FASTQ_controls,
  files_FASTQ_samples,
  files_BAM_controls,
  files_BAM_samples,
  libraryAnnotation,
  maxMismatches,
  nTrim5,
  nTrim3,
  nthreads,
  nBestLocations,
  duplicatedSeq,
  indexMemory,
  strand,
  EXPname,
  outdir_data,
  outdir_bam,
  aligner,
  fastqc_plots,
  verbose
)
```

Arguments

`file_counts` A string specifying the path to a tsv file containing the raw sgRNA counts. The file must be tab delimited with one row per sgRNA and the following columns/headers:

- sgRNA: containing alphanumerical identifiers of the sgRNA under consideration;
- gene: containing HGNC symbols of the genes targeted by the sgRNA under consideration;

followed by the columns containing the sgRNAs' counts for the controls and columns for library transfected samples. The argument must be NULL if FASTQ / BAM files are specified as input.

`files_FASTQ_controls`
: List of FASTQ files used to generate the counts for the control samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name for the BAM files and in the count matrix. The argument must be NULL if counts/BAM files are specified as input.

`files_FASTQ_samples`
: List of FASTQ files used to generate the counts for the samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name for the BAM files and in the count matrix. The argument must be NULL if counts/BAM files are specified as input.

`files_BAM_controls`
: List of BAM files used to generate the counts for the control samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name in the count matrix. The argument must be NULL if counts/FASTQ files are specified as input.

`files_BAM_samples`
: List of BAM files used to generate the counts for the samples. Each file name should include the path. If present, the name of each element of the list will be used as sample name in the count matrix. The argument must be NULL if counts/FASTQ files are specified as input.

`libraryAnnotation`
: A data frame containing a sgRNAs library. This data frame must include one named row per each sgRNA and the at least following mandatory columns/headers:

  - CODE: the unique ID of the sgRNA;
  - GENES: the gene symbol related to the sgRNA;
  - seq: the nucleotidic sequence of the sgRNA without PAM

  All the built-in libraries included in the package are compliant with this structure.

`maxMismatches`
: Integer number containing the Ns allowed counting the reads. The function will consider valid only the reads with a number of matched bases equal to or greater than the length of the sgRNA sequence, provided in the annotation library, minus the maxMismatches parameter. The parameter is ignored if the input are not FASTQ / BAM files.

`nTrim5`
: Numeric value giving the number of bases trimmed off from 5' end of each read. 0 by default. If aligner = "Rsubreads" (the default) the parameter accept only numeric values. If MAGeCK is used the parameter can specify multiple trimming lengths, separated by comma (,); for example, "7,8" or can be set to "AUTO" to allow MAGeCK to determine the trimming length. The parameter is ignored if the input are not FASTQ files.

`nTrim3`
: Numeric value giving the number of bases trimmed off from 3' end of each read. 0 by default. If aligner = "Rsubreads" (the default) the parameter accept only numeric values. The parameter is ignored if the input are not FASTQ files or if MAGeCK is used as aligner.

`nBestLocations`
: Numeric value specifying the maximal number of equally-best mapping locations that will be reported for a multi-mapping read (max 16). 2 by default. Different tags will be included to the aligned sequences to specify the number

of alignments reported. Please refer to the Rsubread [1] user guide for a complete description. The parameter is ignored if the input are not FASTQ files or if MAGeCK is used as aligner.

`duplicatedSeq` A string defining the strategy to deal with the duplicated sequences in the library index creation. See ccr.CreateLibraryIndex for details. The possible options are "keep" (the default) or "exclude". The "keep" option will maintain the first occurrence of the duplicated sequences while the "exclude" option will remove all the sgRNA whose nucleotidic sequences occur more than once. The parameter is ignored if the input are not FASTQ files.

`indexMemory` A numeric value specifying the amount of memory (in megabytes) used for storing the index during read mapping. 2000 MB by default. The parameter is ignored if the input are not FASTQ files.

`strand` A string specifying the strand of the alinement used to count the reads. It accepts three different options: "F" to count only the reads equal to the sgRNA sequence (default); "R" the read only the reads complementary to the sgRNA sequence; "*" all the reads without any strand filter. See the function description for details. The parameter is ignored if the input are not FASTQ files.

`EXPname` A string specifying the name of the experiment. This will be used as label in the output plots.

`outdir_data` A string specifying the folder where all the results will be saved.

`outdir_bam` A string specifying the folder where all the BAM files created by the alignment process will be saved. The parameter is not considered in the case of BAM/counts file input.

`aligner` A string specifying the aligner used to map the reads to the library. The possible options are "Rsubreads" (default) and "mageck" if the MAGeCK application [2] is installed. The parameter is ignored if the input are not FASTQ files.

`fastqc_plots` A boolean value specifying if the QC plots for each FASTQ files will be generated during the alignment. The QC plots are created using the rqc function in the Rqc package. All the plots for each FASTQ file are collected in one HTML file named as the BAM file created in the alignment. The parameter is ignored if the input are not FASTQ files.

`verbose` Technical parameter undocumented. Used for the integration in web architecture.

Value

A boolean value equal to TRUE if the function ends without errors.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

See Also

`ccr.AnalysisPipeline`

`ccr.getLibrary` *Convert library inputs in a standard library annotaion.*

Description

This function takes as input a string identifying one of the sgRNA built-in library or a file name (comprehensive of the full path) to create a data frame with the standard format of the annotaion library requested by CRISPRcleanR. This is a utility function that runs automatically as part of the ccr.AnalysisPipeline.

Usage

```
ccr.getLibrary(
  library_builtin,
  library_file,
  verbose = FALSE
)
```

Arguments

`library_builtin`	A string containing the name of one of the libraries whose annotation is available in CRISPRcleanR.
`library_file`	A string specifying the path to a file containing the sgRNA annotations in TXT / TSV format, with columns for sgRNA ID ("CODE"), targeted genes ("GENES"), genomic coordinates, and possibly other information. This should be formatted as the `KY_Library_v1.0` data object containing the annotation of the sgRNA library presented in [1]. If FASTQ files are used as input the annotation must include also a column "seq" with the nucleotidic sequence of the sgRNAs. This argument is ignored if a built-in library is specified.
`verbose`	Technical parameter undocumented. Used for the integration in web architecture.

Value

A data frame compliant with the annotation library format requested by CRIPRcleanR.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

See Also

`ccr.AnalysisPipeline`

`ccr.GWclean` *Unsupervised identification and correction of gene independent cell responses to CRISPR-Cas9 targeting.*

Description

This function takes in input a genome-wide essentiality profile derived from a CRISPR-Cas9 experiment employing a pooled library of single guide RNAs (sgRNAs) targeting protein coding genes, which are transfected in an *in vitro* model stably expressing Cas9. The essentiality profile quantifies the loss/gain-of-fitness caused by each sgRNA-targeting, and it is expressed as log fold changes (logFCs) between the aboundance of the sgRNAs at an end point after cell purification and their aboundance in the plasmid pool used for viral production, or at an initial time point, or in any other control condition. A circular binary segmentation algorithm [1, 2] is applied by this function to the genome-wide pattern of logFCs provided in input, in order to identify genomic regions including sgRNAs with sufficiently equal logFC (and mean logFC sufficiently different from background) and targeting a minimal number of different genes. Assuming that it is very unlikely to observe the same loss/gain-of-fitness effect when targeting a large number of contiguous genes, if certain user-defined condition (detailed below) are met then the logFCs of such regions are deemed as biased by some local feature of the involved genomic segment (which could be, for example, copy number amplified [3]), and they are corrected, i.e. mean centered [4].

Usage

```
ccr.GWclean(gwSortedFCs,label='',display=TRUE,
            saveTO=NULL,ignoredGenes=NULL,min.ngenes=3,
            alpha = 0.01,
            nperm = 10000,
            p.method ="hybrid",
            min.width=2,
            kmax=25,
            nmin=200,
            eta=0.05,
            trim = 0.025,
            undo.splits = "none",
            undo.prune=0.05,
            undo.SD=3)
```

Arguments

`gwSortedFCs` A data frame containing genome-wide genomic-sorted sgRNAs' log fold changes. This data frame must include one named row per each sgRNA and the following columns/headers:

- CHR: the chromosome of the gene targeted by the sgRNA under consideration;
- startp: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;

- endp: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- genes: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- avgFC: the log fold change of the sgRNA under consideration averaged across replicates;
- BP: the genomic coordinate of the sgRNA defined as STARTpos+(ENDpos-STARTpos)/2.

This can be generated using the `ccr.logFCs2chromPos` function, starting from a data frame containing sgRNAs' log fold changes generated by the `ccr.NormfoldChanges` function (from raw sgRNAs' counts), from raw sgRNAs' counts.

`label` A string indicating the experiment name. This is used to compose the main title of the plots generated by this function and the name of the folder where the results are saved.

`display` A logical value indicating whether genomic plots showing the results of the biased regions' identification and their log fold change correction should be generated or not.

`saveTO` If different from NULL then this parameter will contain the path where pdf files with then genomic plots showing the results of the biased regions' identification (and their log fold change correction) will be saved (within a folder named as defined in the `label` parameter).

`ignoredGenes` A vector of strings containing HGNC symbols of genes that should not be considered when computing the minimal number of different genes targeted by sgRNAs in the same identified region of estimated equal log fold changes. This could contain, for example, a-priori known essential genes.

`min.ngenes` A numerical value (>0) specifying the minimal number of different genes that the set of sgRNAs within a region of estimated equal logFCs should target in order for theri logFCs to be corrected, i.e. mean centered.

`alpha` significance levels for the test to accept change-points (see DNAcopy).

`nperm` number of permutations used for p-value computation (see DNAcopy).

`p.method` method used for p-value computation. For the "perm" method the p-value is based on full permutation. For the "hybrid" method the maximum over the entire region is split into maximum of max over small segments and max over the rest. Approximation is used for the larger segment max. Default is hybrid (see DNAcopy).

`min.width` the minimum number of markers for a changed segment. The default is 2 but can be made larger. Maximum possible value is set at 5 since arbitrary widths can have the undesirable effect of incorrect change-points when a true signal of narrow widths exists (see DNAcopy).

`kmax` the maximum width of smaller segment for permutation in the hybrid method (see DNAcopy).

`nmin` the minimum length of data for which the approximation of maximum statistic is used under the hybrid method. should be larger than 4*kmax (see DNAcopy).

eta	the probability to declare a change conditioned on the permuted statistic exceeding the observed statistic exactly j (= 1,...,nperm*alpha) times. (see DNAcopy).
trim	proportion of data to be trimmed for variance calculation for smoothing outliers and undoing splits based on SD (see DNAcopy).
undo.splits	a character string specifying how change-points are to be undone, if at all. Default is "none". Other choices are "prune", which uses a sum of squares criterion, and "sdundo", which undoes splits that are not at least this many SDs apart. (see DNAcopy).
undo.prune	the proportional increase in sum of squares allowed when eliminating splits if undo.splits="prune" (see DNAcopy).
undo.SD	the number of SDs between means to keep a split if undo.splits="sdundo" (see DNAcopy).
	The rest of the arguments are passed to the segment function of the DNAcopy package as they are.

Value

A list containing two data frames and a vector of strings. The first data frame (corrected_logFCs) contains a named row per each sgRNA and the following columns/header:

- CHR: the chromosome of the gene targeted by the sgRNA under consideration;
- startp: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- endp: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- genes: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- avgFC: the log fold change of the sgRNA averaged across replicates;
- correction: the type of correction: 1 = increased log fold change, -1 = decreased log fold change. 0 indicates no correction;
- correctedFC: the corrected log fold change of the sgRNA

.
The second data frame (segments) contains the identified region of estimated equal log fold changes (one region per row) and the following columns/headers:

- CHR: the chromosome of the region under consideration;
- startp: the genomic coordinate of the starting position of the region under consideration;
- endp: the genomic coordinate of the ending position of the region under consideration;
- n.sgRNAs: the number of sgRNAs targeting sequences in the region under consideration;
- avg.logFC: the average log fold change of the sgRNAs in the region;
- guideIdx: the indexes range of the sgRNAs targeting the region under consideration as they appear in the gwSortedF Cs provided in input.

The string of vectors (SORTED_sgRNAs) contains the sgRNAs' identifiers in the same order as they are reported in the gwSortedFCs input data frame, i.e. genome sorted.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Olshen, A. B., Venkatraman, E. S., Lucito, R., Wigler, M. (2004). *Circular binary segmentation for the analysis of array-based DNA copy number data*. Biostatistics 5: 557-572. \

[2] Venkatraman, E. S., Olshen, A. B. (2007). *A faster circular binary segmentation algorithm for the analysis of array CGH data*. Bioinformatics 23: 657-63. \

[3] Andrew J. Aguirre, Robin M. Meyers, Barbara A. Weir, Francisca Vazquez, Cheng-Zhong Zhang, Uri Ben-David, April Cook, Gavin Ha, William F. Harrington, Mihir B. Doshi, Maria Kost-Alimova, Stanley Gill, Han Xu, Levi D. Ali, Guozhi Jiang, Sasha Pantel, Yenarae Lee, Amy Goodale, Andrew D. Cherniack, Coyin Oh, Gregory Kryukov, Glenn S. Cowley, Levi A. Garraway, Kimberly Stegmaier, Charles W. Roberts, Todd R. Golub, Matthew Meyerson, David E. Root, Aviad Tsherniak and William C. Hahn. *Genomic copy number dictates a gene-independent cell response to CRISPR-Cas9 targeting*. Cancer Discov June 3 2016 DOI: 10.1158/2159-8290.CD-16-0154

[4] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

See Also

ccr.cleanChrm

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Deriving the path of the file with the example dataset,
## from the mutagenesis of the HT-29 colorectal cancer cell line
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),'/HT-29_counts.tsv',sep='')

## Loading, median-normalizing and computing fold-changes for the example dataset
normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,EXPname='HT-29',
                                libraryAnnotation = KY_Library_v1.0)

## Genome-sorting of the fold changes
gwSortedFCs<-ccr.logFCs2chromPos(normANDfcs$logFCs,KY_Library_v1.0)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs<-ccr.GWclean(gwSortedFCs,display=TRUE,label='HT-29')

## Visualising first five entries of the corrected fold changes
head(correctedFCs$corrected_logFCs)


## End(Not run)
```

`ccr.impactOnPhenotype` *Assessing the impact and potential distortion introduced by the CRISPRcleanR correction on the genes showing loss/gain-of-fitness effect.*

Description

This function compares two MAGeCK [1] gene summaries (obtained from sgRNA count files pre/post CRISPRcleanR correction) and it computes the percentages of genes whose loss/gain-of-fitness effect is attenuated post CRISPRcleanR correction or potentially distorted (i.e. loss-of-fitness genes are detected post CRISPRcleanR correction as gain-of-fitness genes, and viceversa). Results are returned in output and optionally plotted as bar/pie charts.

Usage

```
ccr.impactOnPhenotype(MO_uncorrectedFile,
                      MO_correctedFile,
                      sigFDR = 0.05,
                      expName = "expName",
                      display = TRUE)
```

Arguments

`MO_uncorrectedFile`	String specifying the path to a MAGeCK gene summary file produced by MAGeCK from non corrected sgRNA counts.
`MO_correctedFile`	String specifying the path to a MAGeCK gene summary file produced by MAGeCK from CRISPRcleanR corrected sgRNA counts.
`sigFDR`	A numerical value in [0,1] False discovery rate threshold at which genes are called as significantly exerting a loss/gain-of-fitness effect.
`expName`	A string specifying the experiment name, used as main title in the figures (ignored if the `display` argument is set to `FALSE`).
`display`	Boolean value specifying whether figures sumarising the comparison results should be plotted.

Details

For each of the considered MAGeCK gene summaries, this function calls loss/gain-of-fitness based on the MAGeCK negative/positive false discovery rate and the user defined threshold (as specified by the `sigFDR` argument). Particularly, are called as significant loss-of-fitness genes those with a negative fdr < `sigFDR` and a positive fdr >= `sigFDR`, and as significant gain-of-fitness genes those those with a positive fdr < `sigFDR` and a negative fdr >= `sigFDR`. All the other genes are deemed as not exerting any effect on cellular fitness.

Value

A list containing the following four numerical values and two data frames:

- `GW_impact %`: Percentage of genes impacted by the CRISPRcleanR correction, i.e. showing a gain/loss-of-fitness genes effect in the MAGeCK gene summary obtained from uncorrected sgRNA counts, over the total number of screened genes;
- `Phenotype_G_impact %`: Percentage of genes impacted by the CRISPRcleanR correction, i.e. showing a gain/loss-of-fitness genes effect in the MAGeCK gene summary obtained from uncorrected sgRNA counts, over the total number of genes showing a gain/loss of fitness effect in the MAGeCK gene summary obtained from uncorrected sgRNA counts;
- `GW_distortion %`: Percentage of genes distorted by the CRISPRcleanR correction, i.e. showing a gain/loss-of-fitness effect in the MAGeCK gene summary obtained from corrected sgRNA counts that is opposite to the effect in that obtained from uncorrected sgRNA counts, over the total number of screened genes;
- `Phenotype_G_distortion %`: Percentage of genes distorted by the CRISPRcleanR correction, i.e. showing a gain/loss-of-fitness effect in the MAGeCK gene summary obtained from corrected sgRNA counts that is opposite to the effect in that obtained from uncorrected sgRNA counts, over the total number of screened genes,over the total number of genes showing a gain/loss of fitness effect in the MAGeCK gene summary obtained from uncorrected sgRNA countsl;
- `geneCounts`: A contingency table with gene counts as entries, with data referring to the original (uncorrected) sgRNA counts on the columns, and to the corrected sgRNA counts on the rows. There are three vectors for each dimensions, respectively for number of genes showing a significant loss of fitness effect (dep.), number of genes not showing any fitness effect (or with a not clear effect, i.e. showing both gain and loss of fitness effect, null), and number of genes showing a significant gain of fitness effect (enr.);
- `distortion`: a data frame showing genes whose fitness effect has been distorted by the CRISPRcleanR correction: one row per gene (as specified by the row names), with two column per condition (i.e. prior/post correction), indicating the loss of fitness effect fdr (neg.fdr and ccr.neg.fdr) and the gain of fitness effect fdr (pos.fdr and ccr.pos.fdr) as outputted by MAGeCK;
- `attenuation`: a data frame showing genes whose fitness effect has been attenuated by the CRISPRcleanR correction: one row per gene (as specified by the row names), with two column per condition (i.e. prior/post correction), indicating the loss of fitness effect fdr (neg.fdr and ccr.neg.fdr) and the gain of fitness effect fdr (pos.fdr and ccr.pos.fdr) as outputted by MAGeCK;

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554. [2] Hart, T., & Moffat, J. (2016). BAGEL: a computational framework for identifying essential genes from pooled library screens. BMC Bioinformatics, 17(1), 164.

See Also

ccr.ExecuteMageck

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Deriving the path of the file with the example dataset,
## from the mutagenesis of the EPLC-272H colorectal cancer cell line
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/EPLC-272H_counts.tsv',sep='')

## Loading, median-normalizing and computing fold-changes for the example dataset
normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,
                                EXPname='EPLC-272H',
                                libraryAnnotation = KY_Library_v1.0)

uncorrected_fn<-ccr.PlainTsvFile(sgRNA_count_object = normANDfcs$norm_counts,
                                 fprefix = 'EPLC-272H')

## execute MAGeCK on uncorrected normalised counts
uncorrected_gs_fn<-ccr.ExecuteMageck(mgckInputFile = uncorrected_fn,
                                     expName = 'EPLC-272H',
                                     normMethod = 'none')

## Genome-sorting of the fold changes
gwSortedFCs<-ccr.logFCs2chromPos(normANDfcs$logFCs,KY_Library_v1.0)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs<-ccr.GWclean(gwSortedFCs,display=FALSE,label='EPLC-272H')

## correcting individual sgRNA treatment counts
correctedCounts<-ccr.correctCounts('EPLC-272H',normANDfcs$norm_counts,
                                   correctedFCs,
                                   KY_Library_v1.0,
                                   minTargetedGenes=3,
                                   OutDir='./')

## saving corrected/uncorrected sgRNA count files as plain tsv files
corrected_fn<-ccr.PlainTsvFile(sgRNA_count_object = correctedCounts,
                               fprefix = 'EPLC-272H_ccleaned')

## execute MAGeCK on corrected normalised counts
## - it requires MAGeCK to be pre-installed -
corrected_gs_fn<-ccr.ExecuteMageck(mgckInputFile = corrected_fn,
                                   expName = 'EPLC-272H_ccleaned')

## If MAGeCK is installed and correctly executed then
## Assessing the impact of CRISPcleanR correction on gain/loss-of-fitness genes
```

```
RES<-ccr.impactOnPhenotype(MO_uncorrectedFile = uncorrected_gs_fn,
                      MO_correctedFile = corrected_gs_fn,
                      expName = 'EPLC-272H')

## Percentage of genes whose gain/loss-of fitness effect is impacted by CRISPRcleanR
## over the total number of screened genes
RES[1]

## Percentage of genes whose gain/loss-of fitness effect is impacted by CRISPRcleanR
## over the total number of genes with a significant gain/loss-of fitness effect when
## using uncorrected sgRNA counts
RES[2]

## Percentage of genes whose gain/loss-of fitness effect is distorted by CRISPRcleanR
## over the total number of screened genes
RES[3]

## Percentage of genes whose gain/loss-of fitness effect is distorted by CRISPRcleanR
## over the total number of genes with a significant gain/loss-of fitness effect when
## using uncorrected sgRNA counts
RES[4]

## Contingency table showing the impact of the CRISPRcleanR correction on the phenotype
RES$geneCounts

## Genes whose gain/loss-of-fitness effect has been distorted by the CRISPRcleanR correction
RES$distortion

## End(Not run)
```

`ccr.logFCs2chromPos` *Genomic sorting of sgRNAs' log fold changes.*

Description

This function maps genome-wide sgRNAs' log fold changes (averaged across replicates) on the genome and returns them sorted according to the position of their targeted region on the chromosomes.

Usage

```
ccr.logFCs2chromPos(foldchanges, libraryAnnotation)
```

Arguments

`foldchanges`	A data frame containing genome-wide sgRNAs' log fold changes, one column per library transfection replicate, with first and second column containing the sgRNAs' identifiers and the HGNC symbols of the targeted genes, respectively. This can be generated from raw count files using the `ccr.NormfoldChanges` function.

`libraryAnnotation`

A data frame containing the sgRNAs' genome-wide annotations with at least a named row for each of the sgRNAs included in the foldchanges data frame provided in input. The following columns/headers should be present in this data frame (additional columns will be ignored):

- GENES: string vector containing the HGNC symbols of the genes targeted by the sgRNA under consideration;
- EXONE: string vector containing the gene exon targeted by the sgRNA under consideration (these should include the prefix "ex" followed by the exone number);
- CHRM: string vector the chromosome of the gene targeted by the sgRNA under consideration (X and Y chromosome should be specified as "X" and "Y");
- STRAND: string vector containing the strand targeted by the sgRNA under consideration ("+" or "-");
- STARTpop: numeric vector containing the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- ENDpos: numeric vector containing the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;

Additiol columns can be optionally included and will be ignored by this function. The annation for the genome-wide sgRNA library presented in [1] is included in the `KY_Library_v1.0` data object, formatted as described above.

Value

A data frame with a named row per each sgRNA and the following columns/headers:

- `CHR`: the chromosome where the gene targeted by the sgRNA under consideration resides;
- `startp`: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- `endp`: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- `avgFC`: the log fold change of the sgRNA averaged across replicates;
- `BP`: the genomic coordinate of the sgRNA defined as STARTpos+(ENDpos-STARTpos)/2.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Tzelepis K, Koike-Yusa H, De Braekeleer E, Li Y, Metzakopian E, Dovey OM, Mupo A, Grinkevich V, Li M, Mazan M, Gozdecka M, Onishi S, Cooper J, Patel M, McKerrell T, Chen B, Domingues AF, Gallipoli P, Teichmann S, Ponstingl H, McDermott U, Saez-Rodriguez J, Huntly BJP, Iorio F, Pina C, Vassiliou GS, Yusa K. *A CRISPR dropout screen identifies genetic vulnerabilities and therapeutic targets in acute myeloid leukaemia*. Cell Reports 2016 Oct 18;17(4):1193-1205

See Also

ccr.NormfoldChanges, KY_Library_v1.0

Examples

```
## Not run:
    data(KY_Library_v1.0)
   fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),'/A2058_counts.tsv',sep='')
    normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,
                                    EXPname='Example',
                                    libraryAnnotation=KY_Library_v1.0)
    mappedLogFCs<-ccr.logFCs2chromPos(normANDfcs$logFCs,KY_Library_v1.0)
    head(mappedLogFCs)

## End(Not run)
```

---

ccr.multDensPlot *Mutiple shaded density plot*

Description

This functions plots multiple distribution densities with solid colors for the curves and shaded colors for underlying areas.

Usage

```
ccr.multDensPlot(TOPLOT, COLS,
                 XLIMS, TITLE, LEGentries, XLAB)
```

Arguments

TOPLOT	A list of density object computed using the density function of the stats package.
COLS	A vector of colors of the same length of TOPLOT that are used to plot the density curves. Alpha-reduced versions of these colors are used to fill the underlying areas.
XLIMS	A vector of two numerical values optionally specifying x-axis limits (NULL by default).
TITLE	A string containing the plot title.
LEGentries	A vector of strings (one per each density in TOPLOT) specifying corresponding legend entries.
XLAB	A string containing the x-axis label.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

Examples

```
## generating random data
x <- rnorm(1000, 0, 0.5)
y <- rnorm(1000, 2, 0.4)
z <- rnorm(1000, -1, 1.5)


## assembling kernel estimated distributions into a list
ToPlot<-list(x=density(x),y=density(y),z=density(z))

## density visualisation
ccr.multDensPlot(ToPlot,COLS = c('red','blue','gray'),
    TITLE = 'example',LEGentries = c('x','y','z'),
    XLIMS = c(-5,3))
```

`ccr.NormfoldChanges` *Normalisation of sgRNA counts and fold change computation*

Description

This function normalises sgRNAs' counts stored in a tsv file whose path is provided in input, to adjust for the effect of library size and read count distributions, scaling by the total number of reads per sample or using the gene wise median of ratios method [1]. It computes log fold changes of transfected library replicates versus controls (tipically the sgRNA counts in the plasmid). The output of this function is returned as a list, and it is also saved into two tsv files.

Usage

```
ccr.NormfoldChanges(filename,
                    Dframe=NULL, display=TRUE,
                    saveToFig=FALSE,
                    outdir='./', min_reads=30, EXPname='',
                    libraryAnnotation, ncontrols=1,
                    method='ScalingByTotalReads')
```

Arguments

`filename` A string specifying the path of a tsv file containing the raw sgRNA counts. This must be a tab delimited file with one row per sgRNA and the following columns/headers:

- sgRNA: containing alphanumerical identifiers of the sgRNA under consideration;
- gene: containing HGNC symbols of the genes targeted by the sgRNA under consideration;

followed by the columns containing the sgRNAs' counts for the controls and columns for library trasfected samples. The argument is ignored if `Dframe` is not NULL.

`Dframe` A data frame containing the raw sgRNA counts (usable as alternative to providing the path to a tsv file, i.e. previous argument). This must have one row per sgRNA and the following columns/headers:

- sgRNA: containing alphanumerical identifiers of the sgRNA under consideration;
- gene: containing HGNC symbols of the genes targeted by the sgRNA under consideration;

followed by the columns containing the sgRNAs' counts for the controls and columns for library trasfected samples. If set to its default NULL value, then the function will try to load and use the file specified in `filename`.

`display` A logic value specifying whether figures containing boxplots with the count values pre/post normalisation and log fold-changes should be visualised (TRUE, by default).

`saveToFig` A logic value specifying whether figures containing boxplots with the count values pre/post normalisation and log fold-changes should be saved as pdf files (FALSE, by default). Setting this parameter to TRUE overrides the value of the display parameter.

`outdir` Path of the directory where the normalised sgRNAs' counts and the log fold changes, as well as the pdf files (if the parameter saveToFig is set to TRUE), must be saved.

`min_reads` This parameter defines a filter threshold value for sgRNAs, based on their average counts in the control sample. Specifically, it indicates the minimal number of counts that each individual sgRNA needs to have in the controls (on average) in order to be included in the output.

`EXPname` A string specifying the name of the experiment. This will be used to compose main title of the generated figures and file names.

`libraryAnnotation` A data frame containing the sgRNA annotations, with a named row for each sgRNA, and columns for targeted genes, genomic coordinates and possibly other informations. This should be formatted as the `KY_Library_v1.0` data object containing the annotation of the sgRNA library presented in [2].

`ncontrols` A numerical value indicating the number of control replicates (therefore columns to be considered as control counts after the first two, in the inputted tsv file).

`method` A string specifying the normalisation method: 'ScalingByTotalReads' for scaling samples by total numbers or reads, 'MedRatios' to use the median of ratios method [1], or a gene name for scaling samples by total number of reads of the guides targeting that gene.

Value

A list containing two data frames: for the normalised sgRNAs' counts (norm_counts) and the sgRNAs' log fold changes (logFCs) respectively. First two columns in these data frames contain sgRNAs' identifiers and HGNC symbols of targete gene, respectively.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Anders S, Huber W. *Differential expression analysis for sequence count data*. Genome Biol. 2010, 11: R106
[2] Tzelepis K, Koike-Yusa H, De Braekeleer E, et al *A CRISPR dropout screen identifies genetic vulnerabilities and therapeutic targets in acute myeloid leukaemia*. Cell Reports 2016 Oct 18;17(4):1193-1205

See Also

KY_Library_v1.0

Examples

```
## Not run:
    ## loading sgRNA library annotation
    data(KY_Library_v1.0)

    ## derive path for an example dataset
   fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),'/HT-29_counts.tsv',sep='')

    ## sgRNAs' normalisation and computation of log fold-changes
    normANDfcs<-ccr.NormfoldChanges(fn,
                                    min_reads=30,
                                    EXPname='Example',
                                    libraryAnnotation=KY_Library_v1.0)

    ## inspecting first 5 entries of the data frames containing the
    ## normalised counts and the log fold-changes
    head(normANDfcs$norm_counts)
    head(normANDfcs$logFCs)

## End(Not run)
```

---

ccr.perf_distributions

CRISPRcleanR correction assessment: inspection of sgRNA log fold changes distributions

---

Description

This function creates distributions density plots of sgRNA log fold changes for defined sets of targeted genes prior/post CRISPRcleanR correction.

Usage

```
ccr.perf_distributions(cellLine, correctedFCs,
                       GDSC.geneLevCNA = NULL,
                       CCLE.gisticCNA = NULL,
                       RNAseq.fpkms = NULL,
                       minCNs = c(8, 10),
                       libraryAnnotation,
                       GDSC.CL_annotation=NULL)
```

Arguments

cellLine	A string specifying the name of a cell line (or a COSMIC identifier [1]);
correctedFCs	sgRNAs log fold changes corrected for gene independent responses to CRISPR-Cas9 targeting, generated with the function ccr.GWclean (first data frame included in the list outputted by ccr.GWclean, i.e. corrected_logFCs).
GDSC.geneLevCNA	Genome-wide copy number data with the same format of GDSC.geneLevCNA. This can be assembled from the xls sheet specified in the source section [a] (containing data for the GDSC1000 cell lines). If NULL, then this function uses the built in GDSC.geneLevCNA data frame, containing data derived from [a] for 15 cell lines used in [2] to assess the performances of CRISPRcleanR.
CCLE.gisticCNA	Genome-wide Gistic [3] scores quantifying copy number status across cell lines with the same format of CCLE.gisticCNA. If NULL then this function uses the CCLE.gisticCNA builtin data frame, containing data for 13 cell lines of the 15 used in [2] to assess the performances of CRISPRcleanR.
RNAseq.fpkms	Genome-wide substitute reads with fragments per kilobase of exon per million reads mapped (FPKM) across cell lines. These can be derived from a comprehensive collection of RNAseq profiles described in [4]. The format must be the same of the RNAseq.fpkms builtin data frame. If NULL then this function uses the RNAseq.fpkms builtin data fram containing data for 15 cell lines used in [2] to assess CRISPRcleaneR results.
minCNs	A numerical vector with two entries specifying the minimal copy number for a gene in order to be considered amplified based on the data in GDSC.geneLevCNA. These two values can be 2, 4, 8 or 10.
libraryAnnotation	The sgRNA library annotations formatted as specified in the reference manual entry of the KY_Library_v1.0 built in library.
GDSC.CL_annotation	Cell lines annotation dataframe with the same structure of the GDSC.CL_annotation. If NULL then the GDSC.CL_annotation is used.

Details

This function generates 4 sets of plots. They contains log fold change distributions density plots prior/post CRISPRcleanR correction respectively for

- (i) Copy number amplified genes according to the data in GDSC.geneLevCNA based on the two threshold values specified in minCNs;

- (ii) Copy number amplified genes according to the data in `CCLE.gisticCNA` (gistic score = +2);
- (iii) Copy number amplified non expressed genes according to the data in `GDSC.geneLevCNA` based on the two threshold values specified in `minCNs`, and the data in `RNAseq.fpkms` (FPKM < 0.05);
- (iv) reference sets of core fitness essential genes from MSigDB [5] (included in the builtin vectors `EssGenes.DNA_REPLICATION_cons`, `EssGenes.KEGG_rna_polymerase`, `EssGenes.PROTEASOME_cons`, `EssGenes.ribosomalProteins`, `EssGenes.SPLICEOSOME_cons`, and reference core-fitness-essential and non-essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [6] (`BAGEL_essential`, `BAGEL_nonEssential`).

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

Source

[a] ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-6.0/Gene_level_CN.xlsx.

References

[1] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783.

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

[3] Mermel CH, Schumacher SE, Hill B, et al. *GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers.* Genome Biol. 2011;12(4):R41. doi: 10.1186/gb-2011-12-4-r41.

[4] Garcia-Alonso L, Iorio F, Matchan A, et al. *Transcription factor activities enhance markers of drug response in cancer* doi: https://doi.org/10.1101/129478

[5] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 15545-15550. http://doi.org/10.1073/pnas.0506580102

[6] BAGEL: a computational framework for identifying essential genes from pooled library screens. Traver Hart and Jason Moffat. BMC Bioinformatics, 2016 vol. 17 p. 164.

See Also

KY_Library_v1.0, ccr.GWclean,
GDSC.geneLevCNA, CCLE.gisticCNA, RNAseq.fpkms,
EssGenes.DNA_REPLICATION_cons, EssGenes.KEGG_rna_polymerase, EssGenes.PROTEASOME_cons,
EssGenes.ribosomalProteins, EssGenes.SPLICEOSOME_cons
BAGEL_essential, BAGEL_nonEssential

Examples

```
## Not run:
## loading corrected sgRNAs log fold-changes and segment annotations for an example
## cell line (HT-29)
data(HT.29correctedFCs)

## loading library annotation
data(KY_Library_v1.0)

## inpecting sgRNA log fold change distributions prior/post CRISPRcleanR correction
ccr.perf_distributions('HT-29',HT.29correctedFCs$corrected_logFCs,
                       libraryAnnotation = KY_Library_v1.0)


## End(Not run)
```

---

ccr.perf_statTests *CRISPRcleanR correction assessment: Statistical tests*

Description

This function tests the log fold changes of sgRNAs targeting different sets of genes for statistically significant differences with respect to background pre and post CRISPRcleanR correction, creating two sets of boxplots with outcomes and outputting statistical indicators.

Usage

```
ccr.perf_statTests(cellLine, libraryAnnotation, correctedFCs,
                   outDir = "./",
                   GDSC.geneLevCNA = NULL,
                   CCLE.gisticCNA = NULL,
                   RNAseq.fpkms = NULL,
                   GDSC.CL_annotation=NULL,
                   verbose = c(-1, 0, 1))
```

Arguments

cellLine	A string specifying the name of a cell line (or a COSMIC identifier [1]);

`libraryAnnotation`
: The sgRNA library annotations formatted as specified in the reference manual entry of the `KY_Library_v1.0` built in library.

`correctedFCs`
: sgRNAs log fold changes corrected for gene independent responses to CRISPR-Cas9 targeting, generated with the function `ccr.GWclean` (first data frame included in the list outputted by `ccr.GWclean`, i.e. `corrected_logFCs`).

`outDir`
: The path of the folder where the boxplot will be saved.

`GDSC.geneLevCNA`
: Genome-wide copy number data with the same format of `GDSC.geneLevCNA`. This can be assembled from the xls sheet specified in the source section [a] (containing data for the GDSC1000 cell lines). If NULL, then this function uses the built in `GDSC.geneLevCNA` data frame, containing data derived from [a] for 15 cell lines used in [2] to assess the performances of CRISPRcleanR.

`CCLE.gisticCNA`
: Genome-wide Gistic [3] scores quantifying copy number status across cell lines with the same format of `CCLE.gisticCNA`. If NULL then this function uses the `CCLE.gisticCNA` builtin data frame, containing data for 13 cell lines of the 15 used in [2] to assess the performances of CRISPRcleanR.

`RNAseq.fpkms`
: Genome-wide substitute reads with fragments per kilobase of exon per million reads mapped (FPKM) across cell lines. These can be derived from a comprehensive collection of RNAseq profiles described in [4]. The format must be the same of the `RNAseq.fpkms` builtin data frame. If NULL then this function uses the `RNAseq.fpkms` builtin data fram containing data for 15 cell lines used in [2] to assess CRISPRcleaneR results.

`GDSC.CL_annotation`
: Cell lines annotation dataframe with the same structure of the `GDSC.CL_annotation`. If NULL then the `GDSC.CL_annotation` is used.

`verbose`
: Numeric value. In determine the details in the level of details in the messages displeyed running the function: -1 suppres all the messages, 0 display a minimal set of messages, 1 dsiplay all messages (default).

Details

This functions assess the statistical difference pre/post CRISPRcleanR correction of log fold changes for sgRNAs targeting respectively:

- copy number (CN) deleted genes according to the GDSC1000 repository
- CN deleted genes (gistic score = -2) according to the CCLE repository
- non expressed genes (FPKM < 0.05)
- genes with gistic score = 1
- genes with gistic score = 2
- non espressed genes (FPKM < 0.05) with gistic score = 1
- non espressed genes (FPKM < 0.05) with gistic score = 2
- genes with minimal CN = 2, according to the GDSC1000
- genes with minimal CN = 4, according to the GDSC1000
- genes with minimal CN = 8, according to the GDSC1000

- genes with minimal CN = 10, according to the GDSC1000
- non expressed genes (FPKM < 0.05) with minimal CN = 2, according to the GDSC1000
- non expressed genes (FPKM < 0.05) with minimal CN = 4, according to the GDSC1000
- non expressed genes (FPKM < 0.05) with minimal CN = 8, according to the GDSC1000
- non expressed genes (FPKM < 0.05) with minimal CN = 10, according to the GDSC1000
- core fitness essential genes, assembling signatures from MsigDB [5], included in the builtin vectors `EssGenes.DNA_REPLICATION_cons`, `EssGenes.KEGG_rna_polymerase`, `EssGenes.PROTEASOME_cons`, `EssGenes.ribosomalProteins`, `EssGenes.SPLICEOSOME_cons`
- Reference core fitness essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [6] (`BAGEL_essential`)
- Reference core fitness essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [6] after the removal core fitness essential genes from MsigDB [5]
- Reference non essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [6] (`BAGEL_nonEssential`)

Value

A list of three named 2x19 matrices, with one entry per statistical test, rows indicating pre/post CRISPRcleanR correction sgRNAs' log fold changes and one column per each tested gene set. In each matrix the entries contains, respectively

`PVALS`	Pvalue resulting from a Student's t-test assessing the differences between sgRNAs log fold changes pre (first row) and post (second row) CRISPRcleanR correction with respect to background
`SIGNS`	The sign of the difference (1 = mean log fold change of the tested set larger that the mean of the background population, -1 = mean log fold change of the tested set smaller than the mean of the background population)
`EFFsizes`	Effect size (computing via the Cohen's D): difference of the means / pooled standard deviation.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

Source

[a] `ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-6.0/Gene_level_CN.xlsx`.

References

[1] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783.

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

[3] Mermel CH, Schumacher SE, Hill B, et al. *GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers.* Genome Biol. 2011;12(4):R41. doi: 10.1186/gb-2011-12-4-r41.

[4] Garcia-Alonso L, Iorio F, Matchan A, et al. *Transcription factor activities enhance markers of drug response in cancer* doi: https://doi.org/10.1101/129478

[5] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 15545-15550. http://doi.org/10.1073/pnas.0506580102

[6] BAGEL: a computational framework for identifying essential genes from pooled library screens. Traver Hart and Jason Moffat. BMC Bioinformatics, 2016 vol. 17 p. 164.

See Also

KY_Library_v1.0, ccr.GWclean,
GDSC.geneLevCNA, CCLE.gisticCNA, RNAseq.fpkms,
EssGenes.DNA_REPLICATION_cons, EssGenes.KEGG_rna_polymerase, EssGenes.PROTEASOME_cons,
EssGenes.ribosomalProteins, EssGenes.SPLICEOSOME_cons
BAGEL_essential, BAGEL_nonEssential

Examples

```
## Not run:
## loading corrected sgRNAs log fold-changes and segment annotations for an example
## cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)

## loading library annotation
data(KY_Library_v1.0)

## Evaluate correction effects. Boxplots will be saved in EPLC-272H.pdf
## in the current directory
RES<-ccr.perf_statTests('EPLC-272H',libraryAnnotation = KY_Library_v1.0,
                  correctedFCs = EPLC.272HcorrectedFCs$corrected_logFCs)
RES$PVALS
RES$EFFsizes

## End(Not run)
```

`ccr.PlainTsvFile` *Saving a sgRNA counts' object in plain tsv file*

Description

This function takes in input a sgRNA counts' object, as outputted (for example) by the `ccr.NormfoldChanges` function and saves it as plaing tab delimited text file (which can be processed by MAGeCK [1]).

Usage

```
ccr.PlainTsvFile(sgRNA_count_object,
                 fprefix = "",
                 path = "./")
```

Arguments

`sgRNA_count_object`	sgRNA counts data object.
`fprefix`	A string specifying a name prefix of the tsv file which will contain the inputted sgRNA counts data object.
`path`	A string specifying the location where the tsv file will be saved.

Value

A string specifying the complete path of the saves tsv file.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

References

[1] Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., et al. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology, 15(12), 554. [2] Hart, T., & Moffat, J. (2016). BAGEL: a computational framework for identifying essential genes from pooled library screens. BMC Bioinformatics, 17(1), 164.

See Also

`ccr.NormfoldChanges`

Examples

```
## Not run:
## Loading sgRNA library annotation file
data(KY_Library_v1.0)

## Deriving the path of the file with the example dataset,
```

```
## from the mutagenesis of the EPLC-272H colorectal cancer cell line
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/EPLC-272H_counts.tsv',sep='')

## Loading, median-normalizing and computing fold-changes for the example dataset
normANDfcs<-ccr.NormfoldChanges(fn,min_reads=30,
                                EXPname='EPLC-272H',
                                libraryAnnotation = KY_Library_v1.0,
                                display=FALSE)

## saving median-normalised sgRNA counts' as a plain tsv file in ./EPLC-272H_sgRNA_count.tsv
uncorrected_fn<-ccr.PlainTsvFile(sgRNA_count_object = normANDfcs$norm_counts,fprefix = 'EPLC-272H')

uncorrected_fn

## End(Not run)
```

`ccr.PrRc_Curve` *Classification performances of reference sets of genes (or sgRNAs) based on depletion log fold-changes*

Description

This functions computes Precision/Recall (or PPV/Sensitivity, PrRc) curve, area under the PrRc curve and (optionally) Recall (i.e. TPR) at fixed false discovery rate (computed as 1 - Precision (or PPV)) and corresponding log fold change threshold) when classifying reference sets of genes (or sgRNAs) based on their depletion log fold-changes

Usage

```
ccr.PrRc_Curve(FCsprofile,
                         positives,
                         negatives,
                         display = TRUE,
                         FDRth = NULL,
                         expName = NULL)
```

Arguments

`FCsprofile`	A numerical vector containing gene average depletion log fold changes (or sgRNAs' depletion log fold changes) with names corresponding to HGNC symbols (or sgRNAs' identifiers).
`positives`	A vector of strings containing a reference set of positive cases: HGNC symbols of essential genes or identifiers of their targeting sgRNAs. This must be a subset of `FCsprofile` names, disjointed from `negatives`.
`negatives`	A vector of strings containing a reference set of negative cases: HGNC symbols of essential genes or identifiers of their targeting sgRNAs. This must be a subset of `FCsprofile` names, disjointed from `positives`.

display	A logical parameter specifying if a plot containing the computed precision/recall curve with ROC indicators should be plotted (default = TRUE).
FDRth	If different from NULL, will be a numerical value >=0 and <=1 specifying the false discovery rate threshold at which fixed recall will be computed. In this case, if the display parameter is TRUE, an orizontal dashed line will be added to the plot at the resulting recall and its value will be visualised in the legend.
expName	If different from NULL and display parameter is TRUE this parameter should be a string specifying the title of the plot with the computed precision/recall curve.

Value

A list containint three numerical variable AUC, Recall, and sigthreshold indicating the area under PrRc curve and (if FDRth is not NULL) the recall at the specifying false discovery rate and the corresponding log fold change threshold (both equal to NULL, if FDRth is NULL), respectively.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

See Also

BAGEL_essential, BAGEL_nonEssential,

ccr.genes2sgRNAs, ccr.VisDepAndSig,

ccr.ROC_Curve

Examples

```
## Not run:
## loading corrected sgRNAs log fold-changes and segment annotations for an example
## cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)

## loading reference sets of essential and non-essential genes
data(BAGEL_essential)
data(BAGEL_nonEssential)

## loading library annotation
data(KY_Library_v1.0)

## storing sgRNA log fold-changes in a named vector
FCs<-EPLC.272HcorrectedFCs$corrected_logFCs$avgFC
names(FCs)<-rownames(EPLC.272HcorrectedFCs$corrected_logFCs)

## deriving sgRNAs targeting essential and non-essential genes (respectively)
BAGEL_essential_sgRNAs<-ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_essential)
BAGEL_nonEssential_sgRNAs<-ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_nonEssential)

## computing classification performances at the sgRNA level
ccr.PrRc_Curve(FCs,BAGEL_essential_sgRNAs,BAGEL_nonEssential_sgRNAs)
```

```
## computing gene level log fold-changes
geneFCs<-ccr.geneMeanFCs(FCs,KY_Library_v1.0)

## computing classification performances at the sgRNA level, with Recall at 5% FDR
ccr.PrRc_Curve(geneFCs,BAGEL_essential,BAGEL_nonEssential,FDRth = 0.05)

## End(Not run)
```

`ccr.RecallCurves` *CRISPRcleanR correction assessment: Recall curve inspection*

Description

This function creates plots with Recall curve outcomes (a it computes areas under the Recall curves) resulting from classifying defined sets of sgRNAs (respectively genes) based on their log fold change (respectively log fold changes averaged across targeting sgRNAs).

Usage

```
ccr.RecallCurves(cellLine, correctedFCs, GDSC.geneLevCNA = NULL,
                 RNAseq.fpkms = NULL, minCN = 8, libraryAnnotation,
                 GeneLev = FALSE,GDSC.CL_annotation=NULL)
```

Arguments

`cellLine`	A string specifying the name of a cell line (or a COSMIC identifier [1]);
`correctedFCs`	sgRNAs log fold changes corrected for gene independent responses to CRISPR-Cas9 targeting, generated with the function `ccr.GWclean` (first data frame included in the list outputted by `ccr.GWclean`, i.e. `corrected_logFCs`).
`GDSC.geneLevCNA`	Genome-wide copy number data with the same format of `GDSC.geneLevCNA`. This can be assembled from the xls sheet specified in the source section [a] (containing data for the GDSC1000 cell lines). If NULL, then this function uses the built in `GDSC.geneLevCNA` data frame, containing data derived from [a] for 15 cell lines used in [2] to assess the performances of CRISPRcleanR.
`RNAseq.fpkms`	Genome-wide substitute reads with fragments per kilobase of exon per million reads mapped (FPKM) across cell lines. These can be derived from a comprehensive collection of RNAseq profiles described in [4]. The format must be the same of the `RNAseq.fpkms` builtin data frame. If NULL then this function uses the `RNAseq.fpkms` builtin data fram containing data for 15 cell lines used in [2] to assess CRISPRcleaneR results.
`minCN`	A numerical value specifying the minimal copy number for a gene in order to be considered amplified based on the data in `GDSC.geneLevCNA`. This value can be 2, 4, 8 or 10.

`libraryAnnotation`
: The sgRNA library annotations formatted as specified in the reference manual entry of the `KY_Library_v1.0` built in library.

`GeneLev`
: A logical value specifying if the Recall should be computed at level of genes. In this case average gene log fold changes are computed from the inputted corrected log fold changes across targeting sgRNAs.

`GDSC.CL_annotation`
: Cell lines annotation dataframe with the same structure of the `GDSC.CL_annotation`. If NULL then the `GDSC.CL_annotation` is used.

Details

This function generates 2 plots, showing Recall curves resulting from classifying the following 4 sets of sgRNAs (or Genes, depending on the parameter `GeneLev`, based on their log fold changes (or log fold changes averaged across targeting guides):

- (i) Copy number amplified genes according to the data in `GDSC.geneLevCNA` based on the threshold value specified in `minCNs`;
- (ii) Copy number amplified non expressed genes according to the data in `GDSC.geneLevCNA` based on the threshold value specified in `minCNs`, and the data in `RNAseq.fpkms` (FPKM < 0.05);
- (iv) reference sets of core-fitness-essential and non-essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [5]
(`BAGEL_essential`, `BAGEL_nonEssential`).

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

Source

[a] `ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-6.0/Gene_level_CN.xlsx`.

References

[1] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783.

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

[3] Mermel CH, Schumacher SE, Hill B, et al. *GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers.* Genome Biol. 2011;12(4):R41. doi: 10.1186/gb-2011-12-4-r41.

[4] Garcia-Alonso L, Iorio F, Matchan A, et al. *Transcription factor activities enhance markers of drug response in cancer* doi: https://doi.org/10.1101/129478

[5] BAGEL: a computational framework for identifying essential genes from pooled library screens. Traver Hart and Jason Moffat. BMC Bioinformatics, 2016 vol. 17 p. 164.

See Also

KY_Library_v1.0, ccr.GWclean, GDSC.geneLevCNA, RNAseq.fpkms, BAGEL_essential, BAGEL_nonEssential

Examples

```
## Not run:
## loading corrected sgRNAs log fold-changes and segment annotations for an example
## cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)

## loading library annotation
data(KY_Library_v1.0)

## Creating recall curve plots and computing corresponding underlying area
## at the level of sgRNAs
ccr.RecallCurves('EPLC-272H',EPLC.272HcorrectedFCs$corrected_logFCs,
                 libraryAnnotation=KY_Library_v1.0)

## Creating recall curve plots and computing corresponding underlying area
## at the gene level
ccr.RecallCurves('EPLC-272H',EPLC.272HcorrectedFCs$corrected_logFCs,
                 libraryAnnotation=KY_Library_v1.0,GeneLev = TRUE)

## End(Not run)
```

---

ccr.RemoveExtraFiles *Clean intermediate files after pipeline execution.*

Description

This function removes from the output data folder intermediate files created by the analysis pipeline run. This is a utility function that runs automatically as part of the ccr.AnalysisPipeline.

Usage

```
ccr.RemoveExtraFiles(
  is_web = FALSE,
  file_counts = NULL,
  files_FASTQ_controls = NULL,
  files_FASTQ_samples = NULL,
```

```
    files_BAM_controls = NULL,
    files_BAM_samples = NULL,
    outdir_data = NULL
  )
```

Arguments

is_web	Boolen flag that indicates if the piepline run was part of a web application back-end process.
file_counts	A string specifying the path of a tsv file containing the raw sgRNA counts.
files_FASTQ_controls	List of FASTQ files used to generate the counts for the control samples. Each file name should include the path. The argument must be NULL if counts / BAM files are specified as input.
files_FASTQ_samples	List of FASTQ files used to generate the counts for the samples. Each file name should include the path. The argument must be NULL if counts / BAM files are specified as input.
files_BAM_controls	List of BAM files used to generate the counts for the control samples. Each file name should include the path. The argument must be NULL if counts / FASTQ files are specified as input.
files_BAM_samples	List of BAM files used to generate the counts for the samples. Each file name should include the path. The argument must be NULL if counts / FASTQ files are specified as input.
outdir_data	A string specifying folder where all the results data files are stored.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

See Also

ccr.AnalysisPipeline

---

ccr.ROC_Curve *Classification performances of reference sets of genes (or sgRNAs) based on depletion log fold-changes*

Description

This functions computes Specificity/Sensitivity (or TNR/TPR, or ROC) curve, area under the ROC curve and (optionally) Recall (i.e. TPR) at fixed false discovery rate (computed as 1 - Precision (or Positive Predicted Value)) and corresponding log fold change threshold) when classifying reference sets of genes (or sgRNAs) based on their depletion log fold-changes

Usage

```
ccr.ROC_Curve(FCsprofile,
                          positives,
                          negatives,
                          display = TRUE,
                          FDRth = NULL,
                          expName = NULL)
```

Arguments

FCsprofile	A numerical vector containing gene average depletion log fold changes (or sgRNAs' depletion log fold changes) with names corresponding to HGNC symbols (or sgRNAs' identifiers).
positives	A vector of strings containing a reference set of positive cases: HGNC symbols of essential genes or identifiers of their targeting sgRNAs. This must be a subset of FCsprofile names, disjointed from negatives.
negatives	A vector of strings containing a reference set of negative cases: HGNC symbols of essential genes or identifiers of their targeting sgRNAs. This must be a subset of FCsprofile names, disjointed from positives.
display	A logical parameter specifying if a plot containing the computed ROC curve with ROC indicators should be plotted (default = TRUE).
FDRth	If different from NULL, will be a numerical value >=0 and <=1 specifying the false discovery rate threshold at which fixed recall will be computed. In this case, if the display parameter is TRUE, an orizontal dashed line will be added to the plot at the resulting recall and its value will be visualised in the legend.
expName	If different from NULL and display parameter is TRUE this parameter should be a string specifying the title of the plot with the computed ROC curve.

Value

A list containint three numerical variable AUC, Recall, and sigthreshold indicating the area under ROC curve and (if FDRth is not NULL) the recall at the specifying false discovery rate and the corresponding log fold change threshold (both equal to NULL, if FDRth is NULL), respectively.

Author(s)

Francesco Iorio (francesco.iorio@fht.org)

See Also

BAGEL_essential, BAGEL_nonEssential,

ccr.genes2sgRNAs, ccr.VisDepAndSig,

ccr.PrRc_Curve

Examples

```
## Not run:
## loading corrected sgRNAs log fold-changes and segment annotations for an example
## cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)

## loading reference sets of essential and non-essential genes
data(BAGEL_essential)
data(BAGEL_nonEssential)

## loading library annotation
data(KY_Library_v1.0)

## storing sgRNA log fold-changes in a named vector
FCs<-EPLC.272HcorrectedFCs$corrected_logFCs$avgFC
names(FCs)<-rownames(EPLC.272HcorrectedFCs$corrected_logFCs)

## deriving sgRNAs targeting essential and non-essential genes (respectively)
BAGEL_essential_sgRNAs<-ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_essential)
BAGEL_nonEssential_sgRNAs<-ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_nonEssential)

## computing classification performances at the sgRNA level
ccr.ROC_Curve(FCs,BAGEL_essential_sgRNAs,BAGEL_nonEssential_sgRNAs)

## computing gene level log fold-changes
geneFCs<-ccr.geneMeanFCs(FCs,KY_Library_v1.0)

## computing classification performances at the sgRNA level, with Recall at 5% FDR
ccr.ROC_Curve(geneFCs,BAGEL_essential,BAGEL_nonEssential,FDRth = 0.05)

## End(Not run)
```

`ccr.sgRNAmeanFCs` *Extract corrected logFC in vectorial format.*

Description

This function takes as input a data frame containing the corrected logFC in the format generated by the ccr.GWclean function. This is a utility function that runs automatically as part of the ccr.AnalysisPipeline.

Usage

```
ccr.sgRNAmeanFCs(
  foldchanges
)
```

Arguments

foldchanges	A data frame with a "correctedFC" column storing the CN corrected logFC and row names equal to the sgRNA associated to the foldchages.

Value

A named vector containing the corrected logFC and names equal to the related sgRNAs.

Author(s)

Paolo Cremaschi (paolo.crmeaschi@fht.org)

See Also

ccr.AnalysisPipeline ccr.GWclean

---

ccr.VisDepAndSig *Depletion profile visualisation with genes signatures superimposed and recall*

Description

This functions ranks the gene (or sgRNAs) log fold changes. Based on this it determines a log fold change threshold based on a user defined false discovery rate when classifying two gene (sgRNA) positive/negative references sets (tipically core-fitness-essential and non-essential genes), and it computes the Recall (or True Positive Rate) of genes in other user defined sets at the determined threshold. It produces a plot where the log fold changes are visualised alongside the rank positions of the genes included in the inputted sets and, their recall and the determined FDR threshold.

Usage

```
ccr.VisDepAndSig(FCsprofile,SIGNATURES,TITLE='',
                 pIs=NULL,nIs=NULL,
                 th=0.05,plotFCprofile=TRUE)
```

Arguments

FCsprofile	A numerical vector containing gene average depletion log fold changes (or sgRNAs' depletion log fold changes) with names corresponding to HGNC symbols (or sgRNAs' identifiers).
SIGNATURES	A named list of vectors containing HGNC gene symbols. Two of these lists are used as classification template (respectively for positive and negative cases) to determine a log fold-change threshold providing a user defined classification false discovery rate.
TITLE	A string specifiying the title of the plot.
pIs	The index position of the signature that contains the positive cases of the classification template.

nIs	The index position of the signature that contains the negative cases of the classification template.
th	A numerical value specifying the desired classification false discovery rate (this must be a real number between 0 and 1).
plotFCprofile	A logic value specifying whether the log fold changes should be plotted.

Value

A named numerical vector containing recall scores for all the inputted signatures at the computed false discovery rate threshold for log fold-changes.

Author(s)

Francesco Iorio (iorio@gmail.com)

See Also

ccr.ROC_Curve, ccr.PrRc_Curve

Examples

```
## loading corrected sgRNAs log fold-changes and segment annotations
## for an example cell line (EPLC-272H)
data(EPLC.272HcorrectedFCs)

## loading reference sets of essential and non-essential genes
data(BAGEL_essential)
data(BAGEL_nonEssential)

## loading other sets of core fitness genes
data(EssGenes.ribosomalProteins)
data(EssGenes.DNA_REPLICATION_cons)
data(EssGenes.KEGG_rna_polymerase)
data(EssGenes.PROTEASOME_cons)
data(EssGenes.SPLICEOSOME_cons)

## storing the sgRNA log fold changes into a name vector
FCs<-EPLC.272HcorrectedFCs$corrected_logFCs$avgFC
names(FCs)<-rownames(EPLC.272HcorrectedFCs$corrected_logFCs)

## loading sgRNA library annotation
data(KY_Library_v1.0)

## computing gene average log fold changes
FCs<-ccr.geneMeanFCs(FCs,KY_Library_v1.0)


## Assembling a named list with all the considered gene sets
SIGNATURES<-list(Ribosomal_Proteins=EssGenes.ribosomalProteins,
                 DNA_Replication = EssGenes.DNA_REPLICATION_cons,
                 RNA_polymerase = EssGenes.KEGG_rna_polymerase,
```

```
                  Proteasome = EssGenes.PROTEASOME_cons,
                  Spliceosome = EssGenes.SPLICEOSOME_cons,
                  CFE=BAGEL_essential,
                  non_essential=BAGEL_nonEssential)

## Visualising log fold change profile with superimposed signatures specifying
## that the reference gene sets are in positions 6 and 7
Recall_scores<-ccr.VisDepAndSig(FCsprofile = FCs,
                            SIGNATURES = SIGNATURES,
                            TITLE = 'EPLC-272H',
                            pIs = 6,
                            nIs = 7)

Recall_scores
```

CL.subset *COSMIC identifiers of 15 immortalised human cancer cell lines*

Description

COSMIC identifiers [1] of 15 cell lines included in the GDSC1000 panel [2] that are used in [3] to assess CRISPRcleaneR results.

Usage

```
data(CL.subset)
```

Format

A vector of strings.

References

[1] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783,

[2] Iorio F, Knijnenburg TA, Vis DJ, Bignell GR, Menden MP, et al. *A landscape of pharmacogenomic interactions in cancer* Cell 2016 Jul 28;166(3):740-54

[3] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(CL.subset)

## Loading annotation for the GDSC1000 cell lines
data(GDSC.CL_annotation)
```

```
## Visualising annotation
GDSC.CL_annotation[CL.subset,]
```

EPLC.272HcorrectedFCs *CRISPRcleanR corrected data for an example cell line*

Description

This list contains corrected sgRNAs log fold-changes and segment annotations for an example cell line (EPLC-272H), obtained using the `ccr.GWclean` function, as detailed in its reference manual entry `ccr.GWclean`.

Usage

```
data("EPLC.272HcorrectedFCs")
```

Format

A list containing two data frames and a vector of strings. The first data frame (corrected_logFCs) contains a named row per each sgRNA and the following columns/header:

- `CHR`: the chromosome of the gene targeted by the sgRNA under consideration;
- `startp`: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- `endp`: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- `genes`: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- `avgFC`: the log fold change of the sgRNA averaged across replicates;
- `correction`: the type of correction: 1 = increased log fold change, -1 = decreased log fold change. 0 indicates no correction;
- `correctedFC`: the corrected log fold change of the sgRNA

.
The second data frame (segments) contains the identified region of estimated equal log fold changes (one region per row) and the following columns/headers:

- `CHR`: the chromosome of the gene targeted by the sgRNA under consideration;
- `startp`: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- `endp`: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- `genes`: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- `avgFC`: the log fold change of the sgRNA averaged across replicates;

- correction: the type of correction: 1 = increased log fold change, -1 = decreased log fold change. 0 indicates no correction;
- correctedFC: the corrected log fold change of the sgRNA

.

The second data frame (segments) contains the identified region of estimated equal log fold changes (one region per row) and the following columns/headers:

- CHR: the chromosome of the region under consideration;
- startp: the genomic coordinate of the starting position of the region under consideration;
- endp: the genomic coordinate of the ending position of the region under consideration;
- n.sgRNAs: the number of sgRNAs targeting sequences in the region under consideration;
- avg.logFC: the average log fold change of the sgRNAs in the region;
- guideIdx: the indexes range of the sgRNAs targeting the region under consideration as they appear in the gwSortedF Cs provided in input.

The string of vectors (SORTED_sgRNAs) contains the sgRNAs' identifiers in the same order as they are reported in the gwSortedFCs data frame inputted to the ccr.GWclean function.

Examples

```
data(EPLC.272HcorrectedFCs)
head(EPLC.272HcorrectedFCs$corrected_logFCs)
head(EPLC.272HcorrectedFCs$segments)
head(EPLC.272HcorrectedFCs$SORTED_sgRNAs)
```

EssGenes.DNA_REPLICATION_cons

Core Fitness essential genes involved in DNA replication

Description

List of core fitness essential genes involved in DNA replication assembled by merging together multilpe DNA replication signatures from MSigDB [1] as detailed in [2].

Usage

```
data("EssGenes.DNA_REPLICATION_cons")
```

Format

A vector of strings containing HGNC symbols.

References

[1] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 15545-15550. http://doi.org/10.1073/pnas.0506580102

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(EssGenes.DNA_REPLICATION_cons)
head(EssGenes.DNA_REPLICATION_cons)
```

EssGenes.HISTONES *Core Fitness essential histone genes*

Description

List of core fitness essential histone genes assembled by merging together multilpe signatures from MSigDB [1] as detailed in [2].

Usage

```
data("EssGenes.HISTONES")
```

Format

A vector of strings containing HGNC symbols.

References

[1] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 15545-15550. http://doi.org/10.1073/pnas.0506580102

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(EssGenes.HISTONES)
head(EssGenes.HISTONES)
```

EssGenes.KEGG_rna_polymerase

Core Fitness essential rna polymerase genes

Description

List of core fitness essential rna polymerase genes downloaded from MSigDB [1].

Usage

```
data("EssGenes.KEGG_rna_polymerase")
```

Format

A vector of strings containing HGNC symbols.

References

[1] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 15545-15550. http://doi.org/10.1073/pnas.0506580102

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(EssGenes.KEGG_rna_polymerase)
head(EssGenes.KEGG_rna_polymerase)
```

EssGenes.PROTEASOME_cons

Core Fitness essential proteasome genes

Description

List of core fitness essential proteasome genes assembled by merging together multilpe DNA replication signatures from MSigDB [1] as detailed in [2].

Usage

```
data("EssGenes.PROTEASOME_cons")
```

Format

A vector of strings containing HGNC symbols.

References

[1] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 15545-15550. http://doi.org/10.1073/pnas.0506580102

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(EssGenes.PROTEASOME_cons)
head(EssGenes.PROTEASOME_cons)
```

EssGenes.ribosomalProteins

Core Fitness essential genes coding for ribosomal proteins

Description

List of core fitness essential coding for ribosomal proteins curated from [1].

Usage

```
data("EssGenes.KEGG_rna_polymerase")
```

Format

A vector of strings containing HGNC symbols.

References

[1] Yoshihama, M. et al. The human ribosomal protein genes: sequencing and comparative analysis of 73 genes. Genome Res. 12, 379-390 (2002)

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(EssGenes.ribosomalProteins)
head(EssGenes.ribosomalProteins)
```

EssGenes.SPLICEOSOME_cons *Core Fitness essential spliceosome genes*

Description

List of core fitness essential spliceosome genes assembled by merging together multilpe DNA replication signatures from MSigDB [1] as detailed in [2].

Usage

```
data("EssGenes.SPLICEOSOME_cons")
```

Format

A vector of strings containing HGNC symbols.

References

[1] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America, 102(43), 1554515550. http://doi.org/10.1073/pnas.0506580102

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(EssGenes.SPLICEOSOME_cons)
head(EssGenes.SPLICEOSOME_cons)
```

GDSC.CL_annotation *Tissue type and other annotations for 1,001 human cancer cell lines*

Description

Tissue type and other annotations for 1,001 human cancer cell lines

Usage

```
data(GDSC.CL_annotation)
```

Format

A data frame with 1,001 observations of the following 7 variables.

`CL.name` Cell line name;

`COSMIC.ID` Cosmic identifier of the cell line;

`GDSC.description_1` Tissue descriptor (Genomics of Drug Sensitivity in Cancer - Level 1);

`GDSC_description_2` Tissue descriptor (Genomics of Drug Sensitivity in Cancer - Level 2);

`'TCGA type'` Manaually curated matched TCGA cancer type;

`MMR` Microsatellite instability status (MSI-S = Stable, MSI-L = Instable, MSI-H = highly-Instable).

Source

This data frame has been derived from the xls table available at https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources//Data/suppData/TableS1E.xlsx.

References

[1] Iorio F, Knijnenburg TA, Vis DJ, Bignell GR, Menden MP, et al. A landscape of pharmacogenomic interactions in cancer Cell 2016 Jul 28;166(3):740-54

Examples

```
data(GDSC.CL_annotation)
head(GDSC.CL_annotation)
```

GDSC.geneLevCNA *Genome-wide copy number data for 15 human cancer cell lines.*

Description

Genome-wide copy number data derived from PICNIC analysis of Affymetrix SNP6 segmentation data (EGAS00001000978, part of the Genomics of Drug Sensitivity in 1,000 Cancer Cell Lines (GDSC1000) panel [1]) for 15 cell lines used in [2] to assess CRISPRcleaneR results.

Usage

```
data(GDSC.geneLevCNA)
```

Format

A data frame with HGNC gene symbols on the row cancer cell lines' cosmic identifiers on the columns. The entry in position *i,j* indicates the copy number status of gene *i* in cell line *j*.

Details

Each entry of the data frame is a string made of four comma seperated peices of data (`n1,n2,n3,n4`), hyphen (`-`) is used when the corresponding data is unknown.

The four values indicate:

- `n1`: Maximum copy number of any genomic segment containing coding sequence of the gene (`-1` indicates a value could not be assigned).
- `n2`: Minimum copy number of any genomic segment containing coding sequence of the gene (`-1` indicates a value could not be assigned).
- `n3`: Zygosity - (`H`) if all segments containing gene sequence are heterozygous, (`L`) if any segment containing coding sequence has LOH, (`0`) if the complete coding sequence of the gene falls within a homozygous deletion.
- `n4`: Disruption (`D`) if the gene spans more than 1 genomic segment (`-`) if no disruption occures.

Source

This data frame has been derived from the xls table available at `ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-6.0/Gene_level_CN.xlsx`.

References

[1] Iorio F, Knijnenburg TA, Vis DJ, Bignell GR, Menden MP, et al. *A landscape of pharmacogenomic interactions in cancer* Cell 2016 Jul 28;166(3):740-54

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

Examples

```
data(GDSC.geneLevCNA)
GDSC.geneLevCNA[1:10,1:10]
```

`GeCKO_Library_v2` *Genome-wide annotation for the GeCKO (v2) sgRNA library*

Description

A data frame with a named row for each sgRNA of the GeCKO sgRNA library [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data(GeCKO_Library_v2)
```

Format

A a row named data frame with 121327 observations of the following variables (among others)

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`STRAND` targeted DNA strand ('+' or '-')

`EXONE` exone of the targeted genomic region (exone number);

`CHRM` chromosome of where the targeted region resides (string)

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

Details

GeCKO v2 library was developed with the aim of targeting all genes with a uniform number of sgRNAs, and included 6 sgRNAs per gene distributed over 3-4 constitutively expressed exons. Minimization of off-target effects was based on a specificity analysis. In addition the library included a number of sgRNAs targeting microRNAs (miRNAs) and 2,000 non targeting sgRNAs, for a total number of 123411 sgRNAs.

Genomic coordinates of the sgRNAs (required by CRISPRcleanR) of the GeCKO v2 library were not available on the annotation file avaialable on AddGene [2], although some partial mappings are provided.

We generated the locations of these mapping positions on the reference genome using the sequence content of the sgRNAs available in the library annotation, using the latest human reference genome (GRCh38), using multiple tools, as detailed in the following steps:

Step 1 The sgRNAs were mapped onto the human reference sequence using the bwa short read mapper. Only the reads that were mapped to the reference genome uniquely were selected and their positions of mappings (start/and end positions) were superimposed to those of the intended targeted gene in Ensembl gene annotation v100. From these mappings all the sgRNAs that were mapped to the correct corresponding target genes were identified and retained. Although bwa is an efficient mapper, due to multiple mapping locations and some small insertions and deletions, some sgRNAs were not mapped to the reference sequence.

Step 2 All the sgRNAs that were mapped to the reference genome in multiple locations were selected and overlapped with the intended targeted gene locations. The sgRNAs mapped onto at least one gene-matching location were selected and retained.

Step 3 All the sgRNAs that were not mapped to their intended/declared target gene were selected and the intended gene symbol/name checked for alternative/more-recent gene symbols/names. All possible alternative gene names were identified and checked for overlap. After correction some of the mappings were corrected and the corresponding sgRNA retained.

Step 4 All the remaining sgRNAs (missing or not mapped) were selected and mapped to the reference genome using the blast tool. Here the mapping is slower but more accurate. The results of the blast psl files including all possible mappings of sgRNAs were parsed. The positions were similarly compared to the reference gene annotations and corrected for most recent gene names/symbols. The sgRNAs correctly mapped were retained.

Step 5 All the remaining sgRNAs were compared against miRBase for non-coding RNAS and for the consistency of the naming of these miRNAs. The matching sgRNAs were identified and retained.

Step 6 The remaining sgRNAs, matching many locations in the human reference genome or with an intendend target name different from that in the annotation file, were mapped to their targeted region using the Waterman-Smith local alignment manually. All the remaining sgRNAs were manually curated retained.

Step 7 Some of the sgRNAs were not added to the final annotation data object. The main reason for this is that these genes were removed from the primary human reference in the GRCh38 version. Also, some miRNAs are retracted as well as some genes. Finally some sgRNAs did not map to the gene that they are intended to target.

These removed sgRNAs were declared to target:

GTF2H2D, LILRA3, LOC391322, LOC653486, PRAMEF16, SMCR9, hsa-mir-1273a, hsa-mir-1273d, hsa-mir-1273g, hsa-mir-1302-5, hsa-mir-3118-5, hsa-mir-3118-6, hsa-mir-320d-1, hsa-mir-3669, hsa-mir-3673, hsa-mir-3910-1, hsa-mir-3910-2, hsa-mir-4419a, hsa-mir-4419b, hsa-mir-4459, hsa-mir-4472-2, hsa-mir-5096, hsa-mir-548aa-2, hsa-mir-548d-2, hsa-mir-6087, GTF2H2D, LILRA3, LOC391322, LOC653486, PRAMEF16, SMCR9

The final list of retained sgRNAs included 121320 (out of 123411). Note that 2000 of the excluded sgRNAs were Non-targeting.

Source

Source of the Library: AddGene, https://www.addgene.org/pooled-library/zhang-human-gecko-v2
Source of the annotation file used for the sgRNA remapping: https://sourceforge.net/projects/mageck/files/libraries/Human_C

Sources for the tools:

blat Standalone BLAT v. 36x5 fast sequence search command line tool. The executable is downloadable from http://hgdownload.soe.ucsc.edu/admin/exe/

blast blastn: 2.6.0+ Package: blast 2.6.0, build Jan 15 2017 17:12:27 https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=

miRbase http://www.mirbase.org/ From available databases, we used has.gff3 which is from human reference sequence hg38

Smith-Waterman https://www.ebi.ac.uk/Tools/psa/emboss_water/ The web interface is used for the local alignment as well as the command line via REST API.

References

[1] Sanjana NE, Shalem O, Zhang F. Improved vectors and genome-wide libraries for CRISPR screening. Nat Methods. 2014;11(8):783-784. doi:10.1038/nmeth.3047

[2] Aguirre AJ, Meyers RM, Weir BA, et al. Genomic Copy Number Dictates a Gene-Independent Cell Response to CRISPR/Cas9 Targeting. Cancer Discov. 2016;6(8):914-929. doi:10.1158/2159-8290.CD-16-0154

Examples

```
## Not run:
## Loading sgRNA GeCKO library annotation file
```

```
data(GeCKO_Library_v2)
## Visualising first entries
head(GeCKO_Library_v2)

## Deriving the path of an example count file
## from screening the HT-29 cell line with the GeCKO v2
## library [2]
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/HT-29-GeCKOv2_counts.tsv',sep='')

expName<-'HT29-GeCKOv2'

## Loading, median-normalizing and computing fold-changes
normANDfcs<-
    ccr.NormfoldChanges(filename = fn,
                        display = TRUE,
                        min_reads = 30,
                        EXPname = expName,
                        libraryAnnotation = GeCKO_Library_v2)

## Genome-sorting the fold changes
gwSortedFCs<-
    ccr.logFCs2chromPos(foldchanges = normANDfcs$logFCs,
                        libraryAnnotation = GeCKO_Library_v2)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs_and_segments<-
    ccr.GWclean(gwSortedFCs = gwSortedFCs,
                display=TRUE,
                label=expName)

## End(Not run)
```

`HT.29correctedFCs` *CRISPRcleanR corrected data for an example cell line*

Description

This list contains corrected sgRNAs log fold-changes and segment annotations for an example cell line (HT-29), obtained using the `ccr.GWclean` function, as detailed in its reference manual entry `ccr.GWclean`.

Usage

```
data("HT.29correctedFCs")
```

Format

A list containing two data frames and a vector of strings. The first data frame (corrected_logFCs) contains a named row per each sgRNA and the following columns/header:

- CHR: the chromosome of the gene targeted by the sgRNA under consideration;
- startp: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- endp: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- genes: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- avgFC: the log fold change of the sgRNA averaged across replicates;
- correction: the type of correction: 1 = increased log fold change, -1 = decreased log fold change. 0 indicates no correction;
- correctedFC: the corrected log fold change of the sgRNA

.
The second data frame (segments) contains the identified region of estimated equal log fold changes (one region per row) and the following columns/headers:

- CHR: the chromosome of the gene targeted by the sgRNA under consideration;
- startp: the genomic coordinate of the starting position of the region targeted by the sgRNA under consideration;
- endp: the genomic coordinate of the ending position of the region targeted by the sgRNA under consideration;
- genes: the HGNC symbol of the gene targeted by the sgRNA under consideration;
- avgFC: the log fold change of the sgRNA averaged across replicates;
- correction: the type of correction: 1 = increased log fold change, -1 = decreased log fold change. 0 indicates no correction;
- correctedFC: the corrected log fold change of the sgRNA

.
The second data frame (segments) contains the identified region of estimated equal log fold changes (one region per row) and the following columns/headers:

- CHR: the chromosome of the region under consideration;
- startp: the genomic coordinate of the starting position of the region under consideration;
- endp: the genomic coordinate of the ending position of the region under consideration;
- n.sgRNAs: the number of sgRNAs targeting sequences in the region under consideration;
- avg.logFC: the average log fold change of the sgRNAs in the region;
- guideIdx: the indexes range of the sgRNAs targeting the region under consideration as they appear in the gwSortedF Cs provided in input.

The string of vectors (SORTED_sgRNAs) contains the sgRNAs' identifiers in the same order as they are reported in the gwSortedFCs data frame inputted to the ccr.GWclean function.

Examples

```
data(HT.29correctedFCs)
head(HT.29correctedFCs$corrected_logFCs)
head(HT.29correctedFCs$segments)
head(HT.29correctedFCs$SORTED_sgRNAs)
```

KY_Library_v1.0 *Genome-wide annotation for the Sanger sgRNA Library v1.0*

Description

A data frame with a named row for each sgRNA of the Sanger sgRNA library presented in [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data(KY_Library_v1.0)
```

Format

A a row named data frame with 90709 observations (one for each sgRNA) of the following 7 variables.

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`EXONE` exone of the targeted genomic region (string with 'ex' prefix followed by the exone number);

`CHRM` chromosome of where the targeted region resides (string)

`STRAND` targeted DNA strand ('+' or '-')

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

References

[1] Tzelepis K, Koike-Yusa H, De Braekeleer E, Li Y, Metzakopian E, Dovey OM, Mupo A, Grinkevich V, Li M, Mazan M, Gozdecka M, Onishi S, Cooper J, Patel M, McKerrell T, Chen B, Domingues AF, Gallipoli P, Teichmann S, Ponstingl H, McDermott U, Saez-Rodriguez J, Huntly BJP, Iorio F, Pina C, Vassiliou GS, Yusa K. *A CRISPR dropout screen identifies genetic vulnerabilities and therapeutic targets in acute myeloid leukaemia*. Cell Reports 2016 Oct 18;17(4):1193-1205

Examples

```
data(KY_Library_v1.0)
head(KY_Library_v1.0)
```

KY_Library_v1.1 *Genome-wide annotation for the Sanger sgRNA Library v1.1*

Description

A data frame with a named row for each sgRNA of the updated Sanger sgRNA library presented in [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data(KY_Library_v1.1)
```

Format

A a row named data frame with 90709 observations (one for each sgRNA) of the following 7 variables.

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`EXONE` exone of the targeted genomic region (string with 'ex' prefix followed by the exone number);

`CHRM` chromosome of where the targeted region resides (string)

`STRAND` targeted DNA strand ('+' or '-')

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

References

[1] Tzelepis K, Koike-Yusa H, De Braekeleer E, Li Y, Metzakopian E, Dovey OM, Mupo A, Grinkevich V, Li M, Mazan M, Gozdecka M, Onishi S, Cooper J, Patel M, McKerrell T, Chen B, Domingues AF, Gallipoli P, Teichmann S, Ponstingl H, McDermott U, Saez-Rodriguez J, Huntly BJP, Iorio F, Pina C, Vassiliou GS, Yusa K. *A CRISPR dropout screen identifies genetic vulnerabilities and therapeutic targets in acute myeloid leukaemia*. Cell Reports 2016 Oct 18;17(4):1193-1205

Examples

```
data(KY_Library_v1.1)
head(KY_Library_v1.1)
```

MiniLibCas9_Library *Genome-wide annotation for the MiniLibCas9 sgRNA library*

Description

A data frame with a named row for each sgRNA of the MiniLibCas9 sgRNA library [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data("MiniLibCas9_Library")
```

Format

A data frame with 37701 observations on the following variables (among others).

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`STRAND` targeted DNA strand ('+' or '-')

`CHRM` chromosome of where the targeted region resides (string)

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

Source

https://github.com/EmanuelGoncalves/crispy/blob/master/notebooks/minlib/libraries/MinLibCas9.csv.gz

References

[1] Goncalves E, Thomas M, Behan FM, Picco G, Pacini C, Allen F, Parry-Smith D, et al. 2019. Minimal Genome-Wide Human CRISPR-Cas9 Library. BioRxiv, January, 848895. https://doi.org/10.1101/848895

Examples

```
## Not run:
## Loading sgRNA MiniLibCas9 library annotation file
data(MiniLibCas9_Library)
## Visualising first entries
head(MiniLibCas9_Library)

## Deriving the path of an example count file
## from screening the HT-29 cell line with the Brunello library
## [1]

fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/HT29-MiniLibCas9_counts.tsv',sep='')
```

```
expName<-'HT29-MiniLibCas9'

## Loading, median-normalizing and computing fold-changes
normANDfcs<-
    ccr.NormfoldChanges(filename = fn,
                        display = TRUE,
                        min_reads = 30,
                        EXPname = expName,
                        libraryAnnotation = MiniLibCas9_Library)

## Genome-sorting the fold changes
gwSortedFCs<-
    ccr.logFCs2chromPos(foldchanges = normANDfcs$logFCs,
                        libraryAnnotation = MiniLibCas9_Library)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs_and_segments<-
    ccr.GWclean(gwSortedFCs = gwSortedFCs,
                display=TRUE,
                label=expName)

## End(Not run)
```

`RNAseq.fpkms` *RNAseq derived genome-wide basal expression profiles for 15 cell lines.*

Description

Genome-wide substitute reads with fragments per kilobase of exon per million reads mapped (FPKM) for the 15 cell lines specified in `CL.subset`, derived from a comprehensive collection of RNAseq profiles described in [1] and used in [2] to assess CRISPRcleaneR results.

Usage

```
data(RNAseq.fpkms)
```

Format

A data frame with one bservations per gene and one variable per cell line. Row names indicates HGNC symbols and column names indicate cell line COSMIC identifiers [3].

References

[1] Garcia-Alonso L, Iorio F, Matchan A, et al. *Transcription factor activities enhance markers of drug response in cancer* doi: https://doi.org/10.1101/129478

[2] Iorio, F., Behan, F. M., Goncalves, E., Beaver, C., Ansari, R., Pooley, R., et al. (n.d.). Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting. http://doi.org/10.1101/228189

[3] Forbes SA, Beare D, Boutselakis H, et al. *COSMIC: somatic cancer genetics at high-resolution* Nucleic Acids Research, Volume 45, Issue D1, 4 January 2017, Pages D777-D783,

See Also

CL.subset

Examples

```
data(RNAseq.fpkms)
head(RNAseq.fpkms)
```

Whitehead_Library *Genome-wide annotation for the Whitehead sgRNA library*

Description

A data frame with a named row for each sgRNA of the Whitehead sgRNA library [1] including annotations such as targeted genes, and genomic coordinates.

Usage

```
data(Whitehead_Library)
```

Format

A a row named data frame with 181131 observations of the following variables (among others)

`CODE` alphanumerical identifier of the sgRNAs;

`GENES` targeted gene;

`STARTpos` starting genomic coordinate of the targeted genomic region (numeric);

`STRAND` targeted DNA strand ('sense' or 'antisense')

`EXONE` exone of the targeted genomic region (exone number);

`CHRM` chromosome of where the targeted region resides (string)

`ENDpos` ending genomic coordinate of the targeted genomic region (numeric).

`seq` nucelotidic sequence of the sgRNAs without the PAM. (string).

Source

Discontinued by Addgene

References

[1] Wang T, Birsoy K, Hughes NW, et al. Identification and characterization of essential genes in the human genome. Science. 2015;350(6264):1096-1101. doi:10.1126/science.aac7041

Examples

```
## Not run:
## Loading sgRNA Whitehead library annotation file
data(Whitehead_Library)
## Visualising first entries
head(Whitehead_Library)

## Deriving the path of an example count file
## from screening the HT-29 cell line with the Whitehead library
## [2]
fn<-paste(system.file('extdata', package = 'CRISPRcleanR'),
          '/HT-29-Whitehead_counts.tsv',sep='')

expName<-'HT29-Whitehead'

## Loading, median-normalizing and computing fold-changes
normANDfcs<-
    ccr.NormfoldChanges(filename = fn,
                        display = TRUE,
                        min_reads = 30,
                        EXPname = expName,
                        libraryAnnotation = Whitehead_Library)

## Genome-sorting the fold changes
gwSortedFCs<-
    ccr.logFCs2chromPos(foldchanges = normANDfcs$logFCs,
                        libraryAnnotation = Whitehead_Library)

## Identifying and correcting biased sgRNAs' fold changes
correctedFCs_and_segments<-
    ccr.GWclean(gwSortedFCs = gwSortedFCs,
                display=TRUE,
                label=expName)

## End(Not run)
```

Index