Package 'CoRe'

March 4, 2021

Type Package

Title CoRe R package

Version 0.1.0

Author Emre Karakoc, Clare Pacini, Alessandro Vinceti and Francesco Iorio

Maintainer Francesco Iorio <francesco.iorio@sanger.ac.uk>

Description The CoRe package implements algorithms for the identification of core-fitness and common-essential genes from joint analyses of multiple CRISPR-cas9 (or RNAi) viability screens

License GNU GPLv3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports mixdist

R topics documented:

`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)
```

CoRe.AdAM *Adaptive Daisy Model to compute core fitness genes*

Description

This function identifies the Core Fitness genes using the Adaptive Daisy Model [1] starting from a binary dependency matrix.

Usage

```
CoRe.AdAM(depMat,
          display=TRUE,
          main_suffix='fitness genes in at least 1 cell line',
          xlab='n. dependent cell lines',
          ntrials=1000,
          verbose=TRUE,
          TruePositives)
```

Arguments

depMat	Binary dependency matrix, rows are genes and columns are samples. 1 in position *[i,j]* indicates that inactivation of the i-th gene exerts a significant loss of fitness in the j-th sample, 0 otherwise.
display	Boolean, default is TRUE. Should bar plots of the dependency profiles be plotted.
main_suffix	If display=TRUE, title suffix to give to plot of number of genes depleted in a give number of cell lines, default is 'genes depleted in at least 1 cell line'.
xlab	label to give to x-axis of the plots, default is 'n. cell lines'.
ntrials	Integer, default =1000. How many times to randomly perturb dependency matrix to generate the null distributions.
verbose	Boolean, default is TRUE. Should the computation progress be monitored.
TruePositives	Vector of gene symbols to be used as prior known essential genes.

Details

This function calculates the Core Fitness essential genes based on the calculated minimum number of cell lines that optimizes the True positive rates with log10 odds ratios. log10 odd ratios are calculated of observed vs. expected profiles of cumulative number of fitness genes in fixed number of cell lines. Expected values are the mean of those observed across randomised version of the observed binary matrix.

Value

coreFitnessGenes
: A vector of string with Core Fitness Genes' symbols.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

[2] Hart T, Chandrashekhar M, Aregger M, Steinhart Z, Brown KR, MacLeod G, Mis M, Zimmermann M, Fradet-Turcotte A, Sun S, Mero P, Dirks P, Sidhu S, Roth FP, Rissland OS, Durocher D, Angers S, Moffat J. High-Resolution CRISPR Screens Reveal Fitness Genes and Genotype-Specific Cancer Liabilities. Cell. 2015 Dec 3;163(6):1515-26. doi: 10.1016/j.cell.2015.11.015. Epub 2015 Nov 25. PMID: 26627737.

See Also

CoRe.panessprofile CoRe.generateNullModel CoRe.empiricalOdds CoRe.truePositiveRate CoRe.tradeoffEO_TPR CoRe.coreFitnessGenes

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMap<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat)

## Loading a reference set of essential genes from
## from the CRISPRcleanR package, derived from [1] and [2]
data(curated_BAGEL_essential)

## Computing lung cancer core-fitness genes with AdAM
cfgenes <- CoRe.AdAM(LungDepMap, TruePositives = curated_BAGEL_essential)
```

CoRe.AssembleFPs *Assemble expression-based false positives*

Description

Download CCLE gene expression data [1].

Usage

```
CoRe.AssembleFPs(URL='https://ndownloader.figshare.com/files/26261476')
```

Arguments

URL	url link to download CCLE gene expression data.

Details

Download CCLE gene expression data from DepMap portal [1].

Value

LowlyExpr	Vector of lowly expressed genes used as false positive for benchmarcking function.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Barretina, J., Caponigro, G., Stransky, N. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature 483, 603–607 (2012).

See Also

CoRe.CF_Benchmark

Examples

```
FPs<-CoRe.AssembleFPs(URL='https://ndownloader.figshare.com/files/26261476')
head(FPs)
```

`CoRe.CalculateBayesianfactor`
Compute Bayesian Factors

Description

Compute Bayesian Factors based on gene score rank distribution.

Usage

```
CoRe.CalculateBayesianfactor(RankDistribution,
                             display=TRUE)
```

Arguments

`RankDistribution`	Dataframe of gene scores rank distribution.
`display`	Boolean, default is TRUE. Should probability density distribution derived from the normal mixture model be plotted.

Details

Compute Bayesian Factors based on gene score rank distribution using a normal mixture model.

Value

`bak`	Numeric vector of Bayesian Factors derived from mixture model.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

See Also

`CoRe.PercentileCF`

Examples

```
## Compute Bayesian Factor scores
data(curated_BAGEL_essential)
data(curated_BAGEL_nonEssential)

depMat<-CoRe.download_DepMatrix(scaled = TRUE, ess = curated_BAGEL_essential,
  noness = curated_BAGEL_nonEssential) ## Quantitative Dependency Matrix introduced in [1]

RankDistribution <- data.frame(rankings=sample(1:nrow(depMat),
  size = nrow(depMat), replace = TRUE))
rownames(RankDistribution) <- rownames(depMat)

cfBFs<-CoRe.CalculateBayesianfactor(RankDistribution,display=FALSE)
names(cfBFs) <- rownames(RankDistribution)
head(cfBFs)
```

`CoRe.CF_Benchmark` *Computes recall and other ROC indicators for identified core fitness genes*

Description

Computes recall and other ROC indicators for identified core fitness genes.

Usage

```
CoRe.CF_Benchmark(testedGenes,
                          background,
                          priorKnownSignatures,
                          falsePositives,
                          displayBar=FALSE)
```

Arguments

`testedGenes`	Vector of gene symbols that have been identified as tissue/Pan-cancer core fitness.
`background`	Vector of gene symbols that are part of the Binary Dependency Matrix.
`priorKnownSignatures`	List, signatures of independent known essential genes [1].
`falsePositives`	Lowly expressed genes used as false positive during the benchmarking [2].
`displayBar`	Boolean, default is TRUE. Should mutual exclusivity pattern of signatures be plotted.

Details

Computes recall and other ROC indicators for identified core fitness genes with respect to pre-defined signatures of essential [1] and false positive genes [2].

Value

TPRs	Dataframe listing Recall and p-values (obtained from hypergeometric distribution) associated with each signature of independent known essential genes.
PPV	Positive predicted value obtained by comparing pooled signatures against inputted tested genes.
FPR	False positive rate obtained by comparing pooled signatures against inputted false positive genes.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A. 2005;102:15545.

[2] Barretina, J., Caponigro, G., Stransky, N. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature 483, 603–607 (2012).

[3] Van der Meer D, Barthorpe S, Yang W, et al. Cell Model Passports-a hub for clinical, genetic and functional datasets of preclinical cancer models. Nucleic Acids Res. 2019;47(D1):D923–D929.

See Also

CoRe.AssembleFPs

Examples

```
# Benchmarking the identified PanCancer Core fitness genes against
# prior known essential genes [1]

data(EssGenes.DNA_REPLICATION_cons)
data(EssGenes.HISTONES)
data(EssGenes.KEGG_rna_polymerase)
data(EssGenes.PROTEASOME_cons)
data(EssGenes.SPLICEOSOME_cons)
data(EssGenes.ribosomalProteins)

data("curated_BAGEL_essential")

BinDepMat<-CoRe.download_BinaryDepMatrix()

tissues_ctypes<-c("Haematopoietic and Lymphoid",
                  "Ovary",
                  "Peripheral Nervous System",
                  "Central Nervous System",
                  "Pancreas",
                  "Head and Neck",
                  "Bone",
                  "Lung",
                  "Large Intestine",
                  "Esophagus",
```

```
                  "Endometrium",
                  "Stomach",
                  "Breast")

signatures<-list(DNA_REPLICATION=EssGenes.DNA_REPLICATION_cons,
                 HISTONES=EssGenes.HISTONES,
                 RNA_POLYMERASE=EssGenes.KEGG_rna_polymerase,
                 PROTEASOME=EssGenes.PROTEASOME_cons,
                 SPLICEOSOME=EssGenes.SPLICEOSOME_cons,
                 RIBOSOMAL_PROTS=EssGenes.ribosomalProteins)

clannotation<-
  CoRe.download_AnnotationModel(
  'https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz') ## dataset from [3]

PanCancer_CF_genes<-
  CoRe.PanCancer_AdAM(pancan_depMat = BinDepMat,
                      tissues_ctypes = tissues_ctypes,
                      clannotation = clannotation,
                      TruePositives = curated_BAGEL_essential,
                      display = FALSE)

FPs<-CoRe.AssembleFPs()
AdAMperf<-CoRe.CF_Benchmark(PanCancer_CF_genes,
  background = rownames(BinDepMat),priorKnownSignatures =
  signatures,falsePositives=FPs)
```

`CoRe.coreFitnessGenes` *Calculate the Core Fitness genes given the binary dependency matrix and the minimal number of cell line threshold.*

Description

This function identifies as Core Fitness the genes that are fitness in a number of cell lines at least equal to the inputted threshold, this is computed by the `CoRe.tradeoffEO_TPR` function.

Usage

```
CoRe.coreFitnessGenes(depMat,
                      crossoverpoint)
```

Arguments

`depMat`	Binary dependency matrix, rows are genes and columns are samples. 1 in position *[i,j]* indicates that inactivation of the i-th gene exerts a significant loss of fitness in the j-th sample, 0 otherwise.
`crossoverpoint`	minimum number of cell lines in which a gene needs to be fitness in order to be called core-fitness

Value

A vector that containing the Core Fitness Genes:

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

See Also

CoRe.tradeoffEO_TPR

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMap<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat)

## Compute as core-fitness genes those that are fitness
## in at least 20 lung cance cell lines
cfgenes <- CoRe.coreFitnessGenes(depMat=LungDepMap,crossoverpoint=3800)
```

CoRe.CS_AdAM *Execute AdAM on a specific tissue/cancer type*

Description

Execute AdAM on tissue or cancer type specific dependency submatrix.

Usage

```
CoRe.CS_AdAM(pancan_depMat,
                    tissue_ctype = 'Non-Small Cell Lung Carcinoma',
                    clannotation = NULL,
                    display=TRUE,
                    main_suffix='fitness genes in at least 1 cell line',
                    xlab='n. dependent cell lines',
                    ntrials=1000,
                    verbose=TRUE,
                    TruePositives)
```

Arguments

pancan_depMat	Binary Dependency Matrix containing all cell models.
tissue_ctype	Character, name of tissue/cancer type of interest.
clannotation	Cancer cell lines latest model annotation file on the cell model passports.
display	Boolean, default is TRUE. Should bar plots of the dependency profiles be plotted.

main_suffix	If display=TRUE, title suffix to give to plot of number of genes depleted in a give number of cell lines, default is 'genes depleted in at least 1 cell line'.
xlab	label to give to x-axis of the plots, default is 'n. cell lines'.
ntrials	Integer, default =1000. How many times to randomly perturb dependency matrix to generate the null distributions.
verbose	Boolean, default is TRUE. Should the computation progress be monitored.
TruePositives	Vector of gene symbols to be used as prior known essential genes.

Details

Execute sequentially the whole AdAM pipeline on tissue or cancer type specific dependency submatrix.

Value

coreFitnessGenes

A vector of string with Core Fitness Genes' symbols for the tissue/cancer type of interest.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

[2] Van der Meer D, Barthorpe S, Yang W, et al. Cell Model Passports-a hub for clinical, genetic and functional datasets of preclinical cancer models. Nucleic Acids Res. 2019;47(D1):D923–D929.

See Also

CoRe.AdAM

Examples

```
## Downloading Quantitative Dependency Matrix
## for > 300 cancer cell lines from [1]

data(curated_BAGEL_essential)

## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Perform all the analyses but on different tissues or cancer-types
clannotation<-
 CoRe.download_AnnotationModel('https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz') ## dataset

SNCLC_cf_genes<-CoRe.CS_AdAM(BinDepMat,tissue_ctype = 'Non-Small Cell Lung Carcinoma',
                             clannotation = clannotation,
                             TruePositives = curated_BAGEL_essential)
```

`CoRe.download_AnnotationModel` *Download Cell Passport models annotation file*

Description

Downloading Cell Model Passport annotation file [1].

Usage

```
CoRe.download_AnnotationModel(
URL='https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz')
```

Arguments

`URL`	url link to download Cell Model Passport annotation file.

Details

If URL link exists download Binary Dependency Matrix.

Value

`X`	Cell Model Passport annotation file.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Van der Meer D, Barthorpe S, Yang W, et al. Cell Model Passports-a hub for clinical, genetic and functional datasets of preclinical cancer models. Nucleic Acids Res. 2019;47(D1):D923–D929.

Examples

```
## Downloading Cell Model Passport annotation file
modelAnn<-CoRe.download_AnnotationModel('https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz')

head(modelAnn)
```

CoRe.download_BinaryDepMatrix
Download Binary Dependency Matrix

Description

Downloading Binary Dependency Matrix introduced in Behan 2019 from Project Score [1].

Usage

```
CoRe.download_BinaryDepMatrix(
URL='https://cog.sanger.ac.uk/cmp/download/binaryDepScores.tsv.zip')
```

Arguments

URL	url link to download binary matrix.

Details

If URL link exists download Binary Dependency Matrix.

Value

Binary matrix

X	Binary Dependency Matrix introduced in Behan 2019 from Project Score.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

Examples

```
## Downloading Binary Dependency Matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

head(BinDepMat)
```

CoRe.download_DepMatrix

Download Quantitative Dependency Matrix

Description

Downloading Quantitative Dependency Matrix introduced in Behan 2019 from Project Score [1].

Usage

```
CoRe.download_DepMatrix(
              URL='https://cog.sanger.ac.uk/cmp/download/essentiality_matrices.zip',
                     scaled=FALSE,
                     ess,
                     noness)
```

Arguments

URL	url link to download quantitative dependency matrix.
scaled	Boolean, default is FALSE. Should the Quantitative Dependency matrix be scaled using CERES strategy.
ess	Vector of gene symbols to be used as prior known essential genes.
noness	Vector of gene symbols to be used as prior known non-essential genes.

Details

If URL link exists download Quantitative Dependency Matrix.

Value

Quantitative Dependency Matrix

X	Quantitative Dependency Matrix introduced in Behan 2019 from Project Score.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

Examples

```
## Downloading Quantitative Dependency Matrix
## for > 300 cancer cell lines from [1]
data(curated_BAGEL_essential)
data(curated_BAGEL_nonEssential)

DepMat<-CoRe.download_DepMatrix(ess = curated_BAGEL_essential, noness = curated_BAGEL_nonEssential)

head(DepMat)
```

CoRe.empiricalOdds *Empirical odds of number of fitness genes per number of cell lines*

Description

This function calculates log10 odd ratios of observed vs expected profiles of cumulative number of fitness genes in fixed number of cell lines.

Usage

```
CoRe.empiricalOdds(observedCumSum,
                   simulatedCumSum)
```

Arguments

observedCumSum	Observed profile of cumulative sum of numbers of fitness genes in fixed number of cell lines. This is generated by the ADAM.panessprofile function.
simulatedCumSum	Random profiles of cumulative sum of fitness genes in fixed number of cell lines. This is generated by the function ADAM.generateNullModel.

Details

This function calculates log10 odd ratios of observed vs expected profiles of cumulative number of fitness genes in fixed number of cell lines. Expected values are the mean of those observed across randomised version of the observed binary matrix.

Value

A named vector:

odds	log base 10 odd ratios of observed versus expected cumulative sums of number of fitness genes across fixed numbers of cell lines.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

See Also

CoRe.panessprofile, CoRe.generateNullModel

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMap<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat)

## Computing number of fitness genes across fixed numbers
## of cell lines and its cumulative sums
observed <- CoRe.panessprofile(depMat = LungDepMap)

## Simulating Null model for the number of fitness genes
## across numbers of cell lines and their cumulative sums
null_m<-CoRe.generateNullModel(depMat = LungDepMap)

## Computing empirical odds of number of fitness genes per number of cell lines
logOdds <- CoRe.empiricalOdds(observedCumSum=observed$CUMsums,simulatedCumSum=null_m$nullCumSUM)
logOdds
```

CoRe.extract_tissueType_BinDepMatrix

Subset Binary Dependency Matrix based on tissue/cancer type of interest

---

Description

Extracting Binary Dependency SubMatrix for a given tissue or cancer type.

Usage

```
CoRe.extract_tissueType_BinDepMatrix(fullBinDepMat,
                                     tissue_type="Non-Small Cell Lung Carcinoma")
```

Arguments

fullBinDepMat	Binary Dependency Matrix containing all cell models.
tissue_type	Character, name of tissue/cancer type of interest.

Details

Extracting Binary Dependency SubMatrix for a given tissue or cancer type, among those included in the latest model annotation file on the cell model passports [1][2].

Value

Binary Dependency SubMatrix

cs_depmat	Subset of Binary Dependency Matrix containing cell lines of interest.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

[2] Van der Meer D, Barthorpe S, Yang W, et al. Cell Model Passports—a hub for clinical, genetic and functional datasets of preclinical cancer models. Nucleic Acids Research 2019; 47(D1):D923–D929. doi:10.1093/nar/gky872.

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Subset Binary Dependency Matrix based on "Non-Small Cell Lung Carcinoma" cancer type
LungDepMat<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat,tissue_type="Non-Small Cell Lung Carcinoma")

head(LungDepMat)
```

CoRe.generateNullModel

Null model of number of fitness genes across numbers of cell lines and their cumulative sums

---

Description

This function randomly perturbs the binary dependency matrix to generate a null distribution of profiles of fitness genes across fixed number of cell lines, and corresponding null distribution of cumulative sums.

Usage

```
CoRe.generateNullModel(depMat,
                       ntrials=1000,
                       display=TRUE,
                       verbose=TRUE)
```

Arguments

depMat	Binary dependency matrix, rows are genes and columns are samples. 1 in position *[i,j]* indicates that inactivation of the i-th gene exerts a significant loss of fitness in the j-th sample, 0 otherwise.
ntrials	Integer, default = 1000. How many times to randomly perturb dependency matrix to generate the null distributions.
display	Boolean, default is TRUE. Should bar plots of the dependency profiles be plotted
verbose	Boolean, default is TRUE. Should a progress bar be displayed

Details

For a number of trials specified in (ntrials) the inputted binary dependency matrix is randomised, keeping its column marginal sums. The profiles of fitness genes across fixed number of cell lines, and corresponding cumulative sums, are returned for each random perturbation.

Value

A list with the following two named vectors:

`nullProf`	Matrix of number of fitness genes for fixed number of cell lines from. Each rows of matrix corresponds to a randomisation trial.
`nullCumSum`	Matrix of profile of cumulative number of fitness genes in fixed number of cell lines. Each row of matrix is one randomisation trial.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

See Also

`CoRe.randomisedepMat` `CoRe.panessprofile`

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMap<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat)

## Simulating Null model for the number of fitness genes
## across numbers of cell lines and their cumulative sums
pprofile <- CoRe.generateNullModel(depMat = LungDepMap)
```

CoRe.PanCancer_AdAM *Execute AdAM at the Pan-cancer level*

Description

Execute AdAM on tissue or cancer type specific dependency submatrix.

Usage

```
CoRe.PanCancer_AdAM(pancan_depMat,
                            tissues_ctypes,
                            clannotation = NULL,
                            display=TRUE,
                            ntrials=1000,
                            verbose=TRUE,
                            TruePositives)
```

Arguments

`pancan_depMat`	Binary Dependency Matrix containing all cell models.
`tissues_ctypes`	Vector of string with tissue/cancer type names of interest.
`clannotation`	Cancer cell lines latest model annotation file on the cell model passports.
`display`	Boolean, default is TRUE. Should bar plots of the dependency profiles be plotted.
`ntrials`	Integer, default =1000. How many times to randomly perturb dependency matrix to generate the null distributions.
`verbose`	Boolean, default is TRUE. Should the computation progress be monitored.
`TruePositives`	Vector of gene symbols to be used as prior known essential genes.

Details

Execute sequentially the whole AdAM pipeline on every tissue and identify Pan-cancer core fitness genes.

Value

`PanCancer_CF_genes`
: A vector of string with Core Fitness Genes' symbols for the tissue/cancer type of interest.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

[2] Van der Meer D, Barthorpe S, Yang W, et al. Cell Model Passports-a hub for clinical, genetic and functional datasets of preclinical cancer models. Nucleic Acids Res. 2019;47(D1):D923–D929.

See Also

`CoRe.CS_AdAM` `CoRe.AdAM`

Examples

```
# Identifying pan-cancer core-fitness genes with the AdAM model, as
# described in Behan et al 2019, i.e. performing analyses at individual
# tissues/cancer-type level then collapsing results at pan-cancer level

## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

tissues_ctypes<-c("Haematopoietic and Lymphoid",
                  "Ovary",
                  "Peripheral Nervous System",
                  "Central Nervous System",
```

```
                  "Pancreas",
                  "Head and Neck",
                  "Bone",
                  "Lung",
                  "Large Intestine",
                  "Esophagus",
                  "Endometrium",
                  "Stomach",
                  "Breast")

clannotation<-
 CoRe.download_AnnotationModel('https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz') ## dataset

data(curated_BAGEL_essential)
PanCancer_CF_genes<-
  CoRe.PanCancer_AdAM(pancan_depMat = BinDepMat,
                      tissues_ctypes = tissues_ctypes,
                      clannotation = clannotation,
                      TruePositives = curated_BAGEL_essential,
                      display = FALSE)
```

CoRe.panessprofile *Profile of number of fitness genes across fixed numbers of cell lines and its cumulative sums*

Description

This function calculates the numbers (and cumulative numbers) of genes whose inactivation exerts a fitness effect in n cell lines, varying n from 1 to the number of cell lines in the dataset in input.

Usage

```
CoRe.panessprofile(depMat,
                   display=TRUE,
                   main_suffix='fitness genes in at least 1 cell line',
                   xlab='n. dependent cell lines')
```

Arguments

depMat	A binary dependency matrix, i.e. a binary matrix with genes on rows and samples on columns. A 1 in position *[i,j]* indicates that inactivation of the i-th gene exerts a significant loss of fitness in the j-th sample, 0 otherwise.
display	Boolean, default is TRUE. Should bar plots of the dependency profiles be plotted
main_suffix	If display=TRUE, title suffix to give to plot showing number of genes depleted across fixed number of cell lines, default is 'genes depleted in at least 1 cell line'
xlab	If display=TRUE, label to be given to the x-axis of the plots, default is 'n. cell lines'

Value

A list with the following two named vectors:

`panessprof`	Number of genes that are depleted for a number of cell lines
`CUMsums`	Cumulative number of genes depleted in at least x cell lines

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMap<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat)

## Computing number of fitness genes across fixed numbers
## of cell lines and its cumulative sums
pprofile <- CoRe.panessprofile(depMat = LungDepMap)
head(pprofile)
```

`CoRe.PercentileCF` *Execute 90-th percentile method*

Description

Execute 90-th percentile method [1] on Quantitative Dependency Matrix.

Usage

```
CoRe.PercentileCF(depMat,
                  display=TRUE,
                  percentile=0.9,
                  method='fixed',
                  thresholding='localMin')
```

Arguments

`depMat`	Quantitative Dependency Matrix containing Pan-cancer or tissue/cancer types specific models.
`display`	Boolean, default is TRUE. Should gene score rank distribution of the dependency be plotted.

`percentile` Numerical value in range [0,1], default is 0.9. Percentile to be used as threshold.

`method` Character, default is 'fixed'. Specify which version of the 90-th percentile use, options are:
- fixed: calculates the rank distribution of genes at their least dependent 90-th percentile cell line.
- average: calculates the average rank distribution of the least dependent 90-th percentile cell lines.
- slope: employs all the cell lines to fit a linear model to generate a gene score rank distribution for each gene. The slope distribution of the genes score ranks form a bimodal distribution similar to the percentile gene score rank methods of the first two versions. Using this distribution it is possible to determine the point of minimum density between two peaks and predict the essential genes.

For each version, the resulting bimodal distribution is used for the identification of essential genes under the considered cell lines.

`thresholding` Character, default is 'localMin'. Specify the thresholding strategy to adopt, options are:
- localMin: use local minimum for setting threshold.
- second option set threshold using Bayesian Factor (BF) scores. Genes having a BF score ¿= 10 are considered as core fitness.

Details

Calculate the Core Fitness genes using the 90th-percentile [1] least dependent cell line from Quantative knockout screen dependency matrix containing Pan-cancer or tissue/cancer types specific models.

Value

List of the following items:

`cfgenes` A vector of string with Core Fitness Genes' symbols for the tissue/cancer type of interest.

`geneRanks` Dataframe containing rank scores for each gene.

`LocalMinRank` Discriminative threshold estimated using the 'localMin' thresholding option.

`cfBFs` Numeric vector containing Bayesian Factor scores for each gene based on gene score rank distribution.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Dempster, J.M., Pacini, C., Pantel, S. et al. Agreement between two large pan-cancer CRISPR-Cas9 gene dependency data sets. Nat Commun 10, 5817 (2019).

See Also

`CoRe.CalculateBayesianfactor`

Examples

```
## Execute all possible combination of 90-th percentile method on Quantitative
## knockout screen dependency
data(curated_BAGEL_essential)
data(curated_BAGEL_nonEssential)

depMat<-CoRe.download_DepMatrix(scaled = TRUE, ess = curated_BAGEL_essential, noness = curated_BAGEL_nonEssen

CFgenes<-CoRe.PercentileCF(depMat,method = 'fixed',thresholding='localMin')
CFgenesAVG<-CoRe.PercentileCF(depMat,method = 'average',thresholding='localMin')
CFgenesSLOPE<-CoRe.PercentileCF(depMat,method = 'slope',thresholding='localMin')

CFgenes_BFs<-CoRe.PercentileCF(depMat,method = 'fixed',thresholding='BFs')
CFgenesAVG_BFs<-CoRe.PercentileCF(depMat,method = 'average',thresholding='BFs')
CFgenesSLOPE_BFs<-CoRe.PercentileCF(depMat,method = 'slope',thresholding='BFs')
```

CoRe.randomisedepMat *Column-wise matrix randomisation*

Description

This function takes in input a matrix and shuffles its entries column-wise. Then matrix resulting from this shuffling will have the same column marginal totals of the inputted one.

Usage

```
CoRe.randomisedepMat(depMat)
```

Arguments

depMat	A numeric matrix

Value

The matrix given in input with entries shuffled column wisely.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
```

```
LungDepMap<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat)

## Randomising the colums of the submatrix
rnd_exampleDepMat<-CoRe.randomisedepMat(LungDepMap)
```

`CoRe.tradeoffEO_TPR` *Calculate AdAM threshold*

Description

This function finds the minimum number of cell lines in which a gene needs to be fitness in order to be called core-fitness for all the considered cell lines. This is defined as the n providing the best trade-off between i) coverage of priori-known essential genes in the resulting set of predicted core-fitness genes, i.e. fitness in at least n cell lines, and ii) deviance from expectation of the number of fitness genes in n cell lines.

Usage

```
CoRe.tradeoffEO_TPR(EO,
                    TPR,
                    test_set_name,
                    display = TRUE)
```

Arguments

`EO`	Profile of empirical odds values. Computed with the `CoRe.empiricalOdds` function.
`TPR`	Profile of True positive rates for across number of cell line. Computed with the `CoRe.truePositiveRate` function.
`test_set_name`	Name to give to the analysis, used for plotting titles.
`display`	Boolean, default is TRUE. Should AdAM tradeoff strategy be plotted.

Details

Compare and plot the log10 odds ratios with the true positive rates to find the cross over point where the true positive rate falls below the odds ratio.

Value

ADAM model threshold:

`point`	Number of cell lines for which a gene needs to be a fitness gene in order to be predicted as core-fitness gene.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

[2] Hart T, Chandrashekhar M, Aregger M, Steinhart Z, Brown KR, MacLeod G, Mis M, Zimmermann M, Fradet-Turcotte A, Sun S, Mero P, Dirks P, Sidhu S, Roth FP, Rissland OS, Durocher D, Angers S, Moffat J. High-Resolution CRISPR Screens Reveal Fitness Genes and Genotype-Specific Cancer Liabilities. Cell. 2015 Dec 3;163(6):1515-26. doi: 10.1016/j.cell.2015.11.015. Epub 2015 Nov 25. PMID: 26627737.

See Also

CoRe.empiricalOdds, CoRe.truePositiveRate

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMat<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat,tissue_type="Non-Small Cell Lung Carcinoma")

## Loading a reference set of essential genes from
## from the CRISPRcleanR package, derived from [2]
data(BAGEL_essential)

# Generate the profiles of number of fitness genes across number of cell lines from
# observed data and corresponding comulative sums.
pprofile<-CoRe.panessprofile(depMat=LungDepMat)

# Generate a set of random profiles of number of genes depleted for a number of cell lines
# and corresponding cumulative sums by perturbing observed data.
nullmodel<-CoRe.generateNullModel(depMat=LungDepMat,ntrials = 1000)

# Calculate log10 odd ratios of observed/expected profiles of cumulative number of fitness
# genes in fixed number of cell lines.
# Observed values are from the ADAM.panessprofile function and expected are the average of
# random set from CoRe.generateNullModel
EO<-CoRe.empiricalOdds(observedCumSum = pprofile$CUMsums,simulatedCumSum =nullmodel$nullCumSUM )

# Calculate True positive rates for fitness genes in at least n cell lines in the observed
# dependency matrix, with positive cases from a reference set of essential genes
TPR<-CoRe.truePositiveRate(LungDepMat,BAGEL_essential)

# Calculate minimum number of cell lines a gene needs to be a fitness gene in order to
# be considered as a core-fitness gene
crossoverpoint<-CoRe.tradeoffEO_TPR(EO,TPR$TPR,test_set_name = 'BAGEL essential')
crossoverpoint
```

CoRe.truePositiveRate *Profile of True Positive Rates*

Description

This function calculates a profile of True Positive Rates for fitness genes in at least n cell lines, with positive cases from a reference set of essential genes.

Usage

```
CoRe.truePositiveRate(depMat,
                      essentialGeneSet)
```

Arguments

depMat	Binary dependency matrix, rows are genes and columns are samples. 1 in position *[i,j]* indicates that inactivation of the i-th gene exerts a significant loss of fitness in the j-th sample, i.e. the i-th gene is a fitness gene for the j-th cell line, 0 otherwise.
essentialGeneSet	Reference set of predefined essential genes. This is used to define positive cases.

Details

This function calculates true positive rates for fitness genes in at least n cell lines (for each n). First, this function calculates the number of cell lines for which each gene is a fitness gene. Second, for a given number of cell lines, the set of genes that are fitness genes in at least that number of cell lines is determined. Finally, this set of genes is then compared to the reference set of essential genes to calculate a true positive rate.

Value

A list of the following vectors:

P	Vector of number of genes that are fitness genes in a given number of cell lines.
TP	Vector of number of genes that are fitness genes in a given number of cell lines and are true positives, i.e. in the reference set of essential genes provided in input.
TPR	TP divided by number of genes in set reference set of essential genes provided in input.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

[2] Hart T, Chandrashekhar M, Aregger M, Steinhart Z, Brown KR, MacLeod G, Mis M, Zimmermann M, Fradet-Turcotte A, Sun S, Mero P, Dirks P, Sidhu S, Roth FP, Rissland OS, Durocher D, Angers S, Moffat J. High-Resolution CRISPR Screens Reveal Fitness Genes and Genotype-Specific Cancer Liabilities. Cell. 2015 Dec 3;163(6):1515-26. doi: 10.1016/j.cell.2015.11.015. Epub 2015 Nov 25. PMID: 26627737.

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMat<-CoRe.extract_tissueType_BinDepMatrix(BinDepMat,tissue_type="Non-Small Cell Lung Carcinoma")

## Loading a reference set of essential genes from
## from the CRISPRcleanR package, derived from [2]
data(BAGEL_essential)

TPR<-CoRe.truePositiveRate(LungDepMat,BAGEL_essential)
head(TPR)
```

CoRe.VisCFness *Visualization of CFness of a gene*

Description

Visualization of CFness of a gene and comparison to positive and negative control gene.

Usage

```
CoRe.VisCFness(depMat,
                        gene,
                        percentile=0.9,
                        posControl='RPL12',
                        negControl='MAP2K1')
```

Arguments

depMat	Quantitative Dependency Matrix containing all cell models.
gene	Character, name of gene of interest.
percentile	Numerical value in range [0,1], default is 0.9. Percentile to be used as threshold.
posControl	Name of gene used as positive control for the visualization.
negControl	Name of gene used as negative control for the visualization.

Author(s)

C. Pacini, E. Karakoc, A. Vinceti & F. Iorio

References

[1] Behan FM, Iorio F, Picco G, Gonçalves E, Beaver CM, Migliardi G, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019;568:511–6.

Examples

```
## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [1]
DepMat<-CoRe.download_DepMatrix()

gene<-"A1BG"

## CFness visualization
CoRe.VisCFness(DepMat,
               gene,
               percentile=0.9,
               posControl='RPL12',
               negControl='MAP2K1')
```

`curated_BAGEL_essential` *Curated 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] and filtered as shown in [2].

Usage

```
data(curated_BAGEL_essential)
```

Format

A vector of strings containing HGNC symbols of curated 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.

[2] Behan, F.M., Iorio, F., Picco, G. et al. Prioritization of cancer therapeutic targets using CRISPR–Cas9 screens. Nature 568, 511–516 (2019).

See Also

curated_BAGEL_nonEssential

Examples

```
data(curated_BAGEL_essential)
head(curated_BAGEL_essential)
```

curated_BAGEL_nonEssential
Curated 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] and filtered as shown in [2].

Usage

```
data(curated_BAGEL_nonEssential)
```

Format

A vector of strings containing HGNC symbols of curated 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.

[2] Behan, F.M., Iorio, F., Picco, G. et al. Prioritization of cancer therapeutic targets using CRISPR–Cas9 screens. Nature 568, 511–516 (2019).

See Also

curated_BAGEL_essential

Examples

```
data(curated_BAGEL_nonEssential)
head(curated_BAGEL_nonEssential)
```

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)
```

Index