Package 'CoRe'

March 22, 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 GPL (>=3)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports magrittr, RCurl, readr, pheatmap, stringr

R topics documented:

Index   **36**

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] 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.

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] 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

See Also

BAGEL_essential

Examples

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

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

Description

This function estimates Core Fitness essential genes using the Adaptive Daisy Model [1] starting from a binary gene 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 (screens, cell-cell lines). 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 bars indicating dependency profiles and boxes for estimated null models be plotte.
main_suffix	If display=TRUE, title suffix to be give to the plots.
xlab	label to be used in the x-axis of the plots, default is 'n. cell lines'.
ntrials	Integer, default =1000. How many times to randomly perturb dependency matrix to generate null distributions of number of genes called essentials in fixed number of cell lines.
verbose	Boolean, default is TRUE. Should the computation progress be monitored.
TruePositives	Vector of gene symbols to be used as reference prior known essential genes.

Details

This function identifies Core Fitness essential genes from the joint analysis of multiple CRISPR-Cas9 viability screens performed on different cell-lines / models. It works with binary gene x cell-line essantial/non-essential matrices and it estimates the minimal number n of cell-lines in which a gene should be called as essential in order to be considered as a core-fitness essential gene for the tissue of origin of the screened cell-lines. This threshold is computed in a semi-supervised way and it is defined as that maximising the deviance from expectation of the number of genes that are essential in n cell-lines and their true positive rates computed with respect to a set of prior known core-fitness essential genes (to be provided in input).

Value

coreFitnessGenes
          A vector of strings with estimated 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
LungDepMat<-CoRe.extract_tissueType_SubMatrix(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(LungDepMat, TruePositives = curated_BAGEL_essential)
```

CoRe.AssembleFPs   *Assembling expression-based false positives*

Description

Assembling a set of lowly expressed genes cancer cell lines from the CCLE [1].

Usage

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

Arguments

URL   URL of the CCLE gene expression dataset.

Details

This function download CCLE gene expression data from DepMap portal [1] then it estimates a set of overall lowly expressed genes as detailed in [2].

Value

LowlyExpr   A vector of strings with symbols of genes that are lowly expressed across ~1,300
                   cell lines to be used as false positive by the benchmarking function.

Author(s)

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).

[2] Pacini, Dempster et al, Integrated cross-study datasets of genetic dependencies in cancer. https://doi.org/10.1101/2020
.

See Also

CoRe.CF_Benchmark

Examples

```
FPs<-CoRe.AssembleFPs()
head(FPs)
```

CoRe.CF_Benchmark   *Recall of known essential genes and ROC indicators*

Description

This function assesses the set of predicted core fitness genes by computing the recall (and other ROC indicators) of prior known essential genes and false positives.

Usage

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

Arguments

testedGenes	Vector of gene symbols that have been identified as tissue-specific or Pan-cancer core fitness genes.
background	Vector of gene symbols included in the Dependency Matrix used to make the prediction (the background population).
priorKnownSignatures	
	A List of string vectors containing each a signature of prior known essential genes (their symbol)[1].
falsePositives	Genes to be used to compute false positive rates, this can be for example lowly expressed genes from the CCLE [2], assembled through the CoRe.AssembleFPs function.
displayBar	Boolean, default is TRUE. Should a heatmap of the signatures' coverage be plotted.

Details

Computes recall and other ROC indicators for identified core fitness genes with respect to pre-defined signatures of essential and false positive genes defined in input.

Value

TPRs	Dataframe listing Recall and enrichment p-values (obtained from hypergeometric distribution) associated with each signature of prior known essential genes.
PPV	Positive predicted value obtained pooling all inputed signatures together and using them as positive cases.
FPR	False positive rate of the inputed 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] Behan FM, Iorio F, Picco G, Gonçalves E, et al. Prioritization of cancer therapeutic targets using CRISPR-Cas9 screens. Nature. 2019 Apr;568(7753):511-516.

[4] 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]

# loading signtures of prior known essential genes
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)

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)

# downloading binary dependency matrix from project Score [3]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Running ADaM [3] to identify Pan-Cancer core fitness genes

## defining the cell line tissues to be used in the first step of ADaM
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")

## Downloading cell line model annotations from the Cell Model Passports [3]
clannotation<-
  CoRe.download_AnnotationModel(
  'https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz')

## Running ADaM [2]
PanCancer_CF_genes<-
  CoRe.PanCancer_ADaM(pancan_depMat = BinDepMat,
                      tissues_ctypes = tissues_ctypes,
                      clannotation = clannotation,
                      TruePositives = curated_BAGEL_essential,
                      display = FALSE)

## Assemling lowly expressed genes from the CCLE [2]
FPs<-CoRe.AssembleFPs()

## benchmarking the core fitness genes predicted by ADaM
## plotting a heatmap highlighting the recalled prior known essential genes
## with barplots and enrichhment pvalues
ADaMperf<-CoRe.CF_Benchmark(PanCancer_CF_genes,
  background = rownames(BinDepMat),priorKnownSignatures =
  signatures,falsePositives=FPs)

## Inspecting TPRs, PPV and FPR
ADaMper$
```

CoRe.coreFitnessGenes   *Determining Core Fitness from a binary dependency matrix and re-*
                         quired minimal number of dependent cell lines.

Description

This function identifies as Core Fitness those genes that are fitness genes in at least *n* cell lines
(defined in input), according to the binary dependency matrix defined in input. This minimal *n* is
estimated through the ADaM method [1] by the CoRe.tradeoffEO_TPR function.

Usage

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

Arguments

depMat   Binary dependency matrix where rows are genes and columns are cell-lines/samples.
                A 1 in position *[i,j]* indicates that the inactivation of the *i*-th gene exerts a sig-
                nificant loss of fitness in the *j*-th sample, i.e. that gene is a fitness gene for that
                cell line

crossoverpoint The estimated minimum number of cell lines in which a gene should be a sig-
nificant fitness gene in order to be called a core-fitness gene.

Value

A vector of string containing the predicted 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 a binary 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_SubMatrix(BinDepMat)

## Compute as core-fitness genes those that are fitness
## in at least 20 Non-Small Cell Lung Carcinoma cell lines
cfgenes <- CoRe.coreFitnessGenes(depMat=LungDepMap,crossoverpoint=20)
```

CoRe.CS_ADaM   *Execute ADaM on a specific tissue/cancer type*

Description

Execute ADaM on a tissue- or cancer-type-specific binary 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	A string specifying the tissue/cancer type of interest, this must be compliant with the Cell Model Passports annotation [1].
clannotation	Cancer cell line models' annotation from the cell model passports. This can be downloaded using the `CoRe.download_AnnotationModel` function
display	Boolean, default is TRUE. Should bar plots of dependency profiles and boxplots of estimated empirical distribution be visualised.
main_suffix	If display=TRUE, title suffix to be given to plots of number of genes that are essential/fitness in a give number of cell lines, default is 'genes depleted in at least 1 cell line'.
xlab	x-axis label of the plots, default is 'n. cell lines'.
ntrials	Integer, default =1000. How many times the dependency matrix shouldd be suffled in order to generate null distributions of number of genes that are essential in fixed numbers of cell lines
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 a tissue or cancer type specific dependency submatrix.

Value

coreFitnessGenes

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

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.

[2] 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.

[3] Hart T, Chandrashekhar M, Aregger M, et al. High-Resolution CRISPR Screens Reveal Fitness Genes and Genotype-Specific Cancer Liabilities. Cell. 2015 Dec 3;163(6):1515-26.

See Also

`CoRe.ADaM`

Examples

```
## downloading a reference set of prior known essential genes from [3]
## curated as detailed in [2]

data(curated_BAGEL_essential)

## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [2]
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   The URL specifying the online location of the annotation file. Default value
                points to the most up-to-date version of the Cell Model Passports annotation file
                (which is kept updated).

Value

A data frame with one row per model and one colum per annotation entry.

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()

head(modelAnn)
```

CoRe.download_BinaryDepMatrix
                          Download Binary Dependency Matrix

Description

Downloading Binary Dependency Matrix introduced in [1] from the Project Score portal [2].

Usage

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

Arguments

URL   URL pointing to the online location of a zipped folder containing a binary de-
                pendency matrix. By default this will point to an entry on the data download
                page of the Project Score portal with data from [1].

Details

If the URL points to a valid online location this function downloads a Binary Dependency Matrix.

Value

A binary Dependency Matrix (from [1]) where rows are genes and columns are cell lines. A 1 in
position [i,j] indicates that the inactivation of the i-th gene exerts a significant loss of fitness in the
j-th sample, 0 otherwise.

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] Dwane L, Behan FM, Gonçalves E, et al. Project Score database: a resource for investigat-
ing cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan
8;49(D1):D1365-D1372.

Examples

```
## Downloading Binary Dependency Matrix
## for > 300 cancer cell lines from [1,2]
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 [1,3].

Usage

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

Arguments

URL	URL pointing to the online location of a zipped folder containing a quantitative dependency matrix. By default this will point to an entry on the data download page of the Project Score portal with data from [1].
scaled	Boolean, default is FALSE. Should the Quantitative Dependency matrix be scaled using reference set of essential and non essential genes (provided in input), as detailed in [2].
ess	Vector of gene symbols to be used as reference essential genes. Ignored if scaled is set to FALSE.
noness	Vector of gene symbols to be used as reference non-essential genes. Ignored if scaled is set to FALSE.

Details

If the URL points to a valid online location this function downloads a quantitative Dependency Matrix.

Value

A Dependency Matrix (from [1,3]) where rows are genes and columns are cell lines. The entry in position [i,j] quantifies the effect of the inactivation of the i-th gene on the survival of the j-th cell line (its fitness).

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] Meyers RM, Bryan JG, McFarland JM, et al. Computational correction of copy number effect improves specificity of CRISPR-Cas9 essentiality screens in cancer cells. Nat Genet. 2017 Dec;49(12):1779-1784. doi: 10.1038/ng.3984.

[3] Dwane L, Behan FM, Gonçalves E, et al. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

Examples

```
## loading reference set of essential/non-essential genes
data(curated_BAGEL_essential)
data(curated_BAGEL_nonEssential)

## Downloading and scaling Quantitative Dependency Matrix
## for > 300 cancer cell lines from [1]
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 `CoRe.panessprofile` function.

simulatedCumSum

Random profiles of cumulative sum of fitness genes in fixed number of cell lines. This is generated by the function `CoRe.generateNullModel`.

Details

This function is used by the ADaM method [1] to calculate 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 initial binary matrix. This function is used in cascade after calls to `CoRe.generateNullModel` and `CoRe.panessprofile`.

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

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] Dwane L, Behan FM, Gonçalves E, et al. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

See Also

CoRe.panessprofile, CoRe.generateNullModel

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1,2]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Extracting dependency submatrix for
## Non-Small Cell Lung Carcinoma cell lines only
LungDepMap<-CoRe.extract_tissueType_SubMatrix(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_SubMatrix

Subset Dependency Matrix based on tissue/cancer type of interest

---

Description

Extracting Dependency SubMatrix for a given tissue or cancer type.

Usage

```
CoRe.extract_tissueType_SubMatrix(fullDepMat,
                                  tissue_type="Non-Small Cell Lung Carcinoma")
```

Arguments

fullDepMat   Dependency Matrix containing all cell models. For example, downloadable
                from Project Score [1,2] using the function `CoRe.download_BinaryDepMatrix`
                or `CoRe.download_DepMatrix`.

tissue_type   Character, name of tissue/cancer type of interest. If the dependency matrix is
                from Project Score [1,2] this needs to be compatible with the nomeclature used
                in the Cell Model Passports [3]. Cell line annotations can be downloaded from
                the Cell Model Passports using the function `CoRe.download_AnnotationModel`.

Details

Extracting 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

Dependency SubMatrix

cs_depmat   Dependency Submatrix containing data only for the cell lines from 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] Dwane L, Behan FM, Gonçalves E, et al. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

[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 Research 2019; 47(D1):D923–D929. doi:10.1093/nar/gky872.

Examples

```
## Downloading dependency matrix from [1,2]
## for > 300 cancer cell lines from
BinDepMat<-CoRe.download_BinaryDepMatrix()

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

head(LungDepMat)
```

CoRe.FiPer   *Fitness Percentile method to estimate common essential genes*

Description

Fitness percentile method to identify common essential genes from the joint analysis of multiple gene dependency profiles.

Usage

```
CoRe.FiPer(depMat,
                display=TRUE,
                percentile=0.9,
                method='fixed')
```

Arguments

depMat Quantitative Dependency Matrix containing Pan-cancer or tissue/cancer-types specific gene fitness/dependency scores across cell-lines/samples. The value in position $[i, j]$ of such matrix quantifies the fitness/dependency score of the i-th gene in the j-th cell line.

display Boolean, default is TRUE. Should gene score rank distributions of dependency scores be plotted.

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

method Character, default is 'fixed'. This parameter specifies which variant of the Fitness Percentile method should be used. Admissimble values are:
- fixed: a distribution of gene fitness-rank-positions in their least dependent n-th (determined by the percentile parameter) percentile cell line is used in the subsequent step of the Fitness Percentile method.
- average: a distribution of gene average fitness-rank-position across all cell lines at or over the n-th percentile of least dependent cell lines (where n is determined by the percentile parameter) is used in the subsequent step of the Fitness Percentile method.
- slope: for each gene, a linear model is fit on the sequence of gene fitness-rank-positions across all cell lines sorted according to their dependency on that gene, then a distrubution of models' slopes is used in the subsequent step of the Fitness Percentile method.
- AUC: for each gene, the area under the curve resulting from considering the sequence of gene fitness-rank-positions across all cell lines sorted according to their dependency on that gene is used in the subsequent step of the Fitness Percentile method.

Details

This function implements the Fitness Percentile method to estimate common essential genes from multiple gene dependency screens introduced in [1]. For each gene in the quantitative dependency matrix provided in input a score is computed using three possible variants of the method. This can

ben its fitness/essentiality rank when considering all gene essentiality scores in its n-th percentile least dependent cell line, or the average rank when considering all cell lines falling within or over the n-th percentile of least dependent cell lines, or the slope of the curve obtained when fitting a linear model on the sequence of fitness-rank-positions across cell lines sorted according to the dependency on that gene (where both variant and n are user defined.)

The density of these gene scores (generally bimodal) is then estimated using a Gaussian kernel and the central point of minimum density is identified. Genes whose score falls below the point of minimum density are classified as common essential. This method is fully detailed in [1]. Observed histogram, estimated density distributions and discriminative threshold are also plotted.

Value

List of the following items:

cfgenes	A vector of strings with common essential genes' symbols for the tissue/cancer type of interest.
geneRanks	Dataframe containing rank scores for each gene.
LocalMinRank	Discriminative rank threshold.

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).

[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.

[3] 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.

[4] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

Examples

```
## Downloading a set of priori known essential genes to be used as true positives from [2] and manually
## curated as detailed in [3]
data(curated_BAGEL_essential)
data(curated_BAGEL_nonEssential)

## Downloading and scaling quantitative dependency matrix from project score [3,4]
depMat<-CoRe.download_DepMatrix(scaled = TRUE, ess = curated_BAGEL_essential, noness = curated_BAGEL_nonEssen

## Executing the three variants of the Fitness percentile method
CFgenes<-CoRe.FiPer(depMat,method = 'fixed')
CFgenesAVG<-CoRe.FiPer(depMat,method = 'average')
CFgenesSLOPE<-CoRe.FiPer(depMat,method = 'slope')
```

```
CFgenesAUC<-CoRe.FiPer(depMat,method = 'AUC')

## Inspect the identified common essential genes
CFgenes$cfgenes
CFgenesAVG$cfgenes
CFgenesSLOPE$cfgenes
CFgenesAUC$cfgenes
```

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   A binary dependency matrix where rows are genes and columns are cell-lines/samples
            with a 1 in position *[i,j]* indicating that the inactivation of the *i*-th gene exerts a
            significant loss of fitness in the *j*-th cell-line/sample.

ntrials   Integer, default = 1000. How many randomly permuted versions of the inputted
            dependency matrix should be generated to simulate null distributions of number
            of fitenss genes across number of cell lines and their cumulative sums.

display   Boolean, default is TRUE. Should bar plots of the dependency profiles be plot-
            ted.

verbose   Boolean, default is TRUE. Should a progress bar be displayed during the execu-
            tion of the function.

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 executed random perturbation.

Value

A list with the following two named vectors:

nullProf   Matrix of number of fitness genes across fixed number of cell lines from. Each
            rows of matrix corresponds to a randomised version of the inputted matrix.

nullCumSum   Matrix of profile of cumulative number of fitness genes in fixed number of cell lines. Each rows of matrix corresponds to a randomised version of the inputted matrix.

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] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

See Also

CoRe.randomisedepMat CoRe.panessprofile

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1,2]
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)

## Inspecting output
head(pprofile$nullProf)

## Inspecting output
head(pprofile$nullCumSUM)
```

CoRe.PanCancer_ADaM	*Execute ADaM at the Pan-cancer level*

Description

Execute ADaM at PanCancer level.

Usage

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

Arguments

pancan_depMat   A binary dependency matrix derived from screening (ideally 100s of) cell-lines from multiple tissue lineages and where rows are genes and columns are cell-lines/samples, with a 1 in position *[i,j]* indicating that the inactivation of the *i*-th gene exerts a significant loss of fitness in the *j*-th cell-line/sample.

tissues_ctypes   Vector of strings with tissue/cancer type names of interest. These should be compatible with the cell model annotations of the Cell Model Passports [2] (downloadable through the function CoRe.download_AnnotationModel).

clannotation   Data frame containing the Cancer cell lines' annotations, derived from the cell model passports [2] (downloadable through the function CoRe.download_AnnotationModel).

display   Boolean, default is TRUE. Should bar plots of the dependency profiles be plotted.

ntrials   Integer, default =1000. How many times to randomly perturb the dependency matrix in order to generate null distributions of number of fitness genes across fixed number of cell lines.

verbose   Boolean, default is TRUE. Should the computation progress be monitored.

TruePositives   Vector of gene symbols to be used as prior known essential genes by the ADaM algorithm.

Details

This function executes ADaM on every tissue in cascade to identify Cancer Type specific Core Fitness genes, then iterates the procedure as detailed in [1] to identify a set of Pan-cancer core fitness genes.

Value

PanCancer_CF_genes
                A vector of string with predicted PanCancer 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] 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.

[3] 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.

[4] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

See Also

CoRe.CS_ADaM CoRe.ADaM CoRe.download_AnnotationModel

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 iterating the proceudre at pan-cancer level

## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [1,4]
BinDepMat<-CoRe.download_BinaryDepMatrix()

## Defining tissues/cancer-types that should be considered in the
## first phase of ADaM executions
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")

## Downloading cell line annotations from the Cell Model Passports [2]
clannotation<-
 CoRe.download_AnnotationModel('https://cog.sanger.ac.uk/cmp/download/model_list_latest.csv.gz') ## dataset

## Downloading a set of priori known essential genes to be used as true positives from [3] and manually
## curated as detailed in [1]
data(curated_BAGEL_essential)

## Execute ADaM at the pancancer level
PanCancer_CF_genes<-
  CoRe.PanCancer_ADaM(pancan_depMat = BinDepMat,
                      tissues_ctypes = tissues_ctypes,
                      clannotation = clannotation,
                      TruePositives = curated_BAGEL_essential,
                      display = FALSE)
```

```
## Inspect output
PanCancer_CF_genes
```

| 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 dependency map given 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 cell-lines/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 across fixed number of cell lines in the inputted dependency matrix
CUMsums	Cumulative number of genes depleted in at least n cell lines in the inputted dependency matrix

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] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

Examples

```
## Downloading dependency matrix
## for > 300 cancer cell lines from [1,2]
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.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.

[2] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

Examples

```
## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [1,2]
BinDepMat<-CoRe.download_BinaryDepMatrix()

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

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

CoRe.tradeoffEO_TPR   *Calculate ADaM threshold*

Description

This function is used by the ADaM method [1] to identify the minimum number of cell lines in which a gene needs to be fitness in order to be called core-fitness for all the cell lines analysed by ADaM. 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 at least *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 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 plots be produced.

Details

Compare and plot the curve of log10 odds ratios with the true positive rate curve, across all tested *n* value, to find the cross over point.

Value

ADAM model threshold:

point	Number of cell lines for which a gene needs to be a significant 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] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

[3] 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 CoRe.ADaM

Examples

```
## Downloading binary dependency matrix
## for > 300 cancer cell lines from Project Score [1,2]
BinDepMat<-CoRe.download_BinaryDepMatrix()

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

## Loading a reference set of essential genes from
## from the CRISPRcleanR package, derived from [3]
data(BAGEL_essential)

# Generate the profiles of number of fitness genes across number of cell lines from
# observed data and corresponding comulative sums, plotting the results
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.
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 is used by the ADaM method [1] 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] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell

dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

[3] 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 binary dependency matrix
## for > 300 cancer cell lines from Project Score [1,2]
BinDepMat<-CoRe.download_BinaryDepMatrix()

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

## Loading a reference set of essential genes from
## from the CRISPRcleanR package, derived from [3]
data(BAGEL_essential)

TPR<-CoRe.truePositiveRate(LungDepMat,BAGEL_essential)

## inspect results
TPR
```

CoRe.VisCFness   *Common-essentiality tendency visualisation*

Description

This function visualises the tendency of a given gene to be common essential (in the dependency matrix provided in input according the the Fitness Percentile method described in [1]) and compares this tendency to that of a positive and a negative control. It also visualises the distribution of scores computed by the Fitness percentile methods (according to the selected variant) with the discriminative threshold and rank position of the input gene as well as the control genes.

Usage

```
CoRe.VisCFness(depMat,gene,method='fixed',
                       percentile=0.9,
                       posControl='RPL8',
                       negControl='MAP2K1')
```

Arguments

depMat   Quantitative Dependency Matrix containing Pan-cancer or tissue/cancer-types specific gene fitness/dependency scores across cell-lines/samples. The value in position $[i, j]$ of such matrix quantifies the fitness/dependency score of the i-th gene in the j-th cell line.

gene	Character, name of the gene whose common-essentiality tendency should be visualised.
method	Character, default is 'fixed'. This parameter specifies which variant of the Fitness Percentile method should be used. Admissimble values are: - fixed: a distribution of gene fitness-rank-positions in their least dependent n-th (determined by the `percentile` parameter) percentile cell line is used in the subsequent step of the Fitness Percentile method. - average: a distribution of gene average fitness-rank-position across all cell lines at or over the n-th percentile of least dependent cell lines (where n is determined by the `percentile` parameter) is used in the subsequent step of the Fitness Percentile method. - slope: for each gene, a linear model is fit on the sequence of gene fitness-rank-positions across all cell lines sorted according to their dependency on that gene, then a distrubution of models' slopes is used in the subsequent step of the Fitness Percentile method. - AUC: for each gene, the area under the curve resulting from considering the sequence of gene fitness-rank-positions across all cell lines sorted according to their dependency on that gene is used in the subsequent step of the Fitness Percentile method.
percentile	Numerical value in range [0,1], default is 0.9. Percentile to be used as a comparative threshold.
posControl	Character, name of a gene used as positive control for the visualization.
negControl	Character, name of gene used as negative control for the visualization.

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).

[2] 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.

[3] Dwane L, Behan FM, Gonçalves E, Lightfoot H, Yang W, van der Meer D, Shepherd R, Pignatelli M, Iorio F, Garnett MJ. Project Score database: a resource for investigating cancer cell dependencies and prioritizing therapeutic targets. Nucleic Acids Res. 2021 Jan 8;49(D1):D1365-D1372.

Examples

```
## Downloading and scaling quantitative dependency matrix from project score [2,3]
DepMat<-CoRe.download_DepMatrix()

## CFness visualisation of BRAF (a context specific essential gene)
CoRe.VisCFness(DepMat,'BRAF')

## CFness visualisation of RPL22 (a common essential gene)
CoRe.VisCFness(DepMat,'RPL22')
```

---

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