Package ‘CELLector’

March 5, 2020

Type Package

Title Genomics guided selection of cancer cell lines

Version 1.2.0

Author Hanna Najgebauer and Francesco Iorio

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

Description Functions to select the most relevant cancer cell lines to be included in a new in-vitro study, in a genomic-guided fashion. CELLector combines methods from graph theory and market basket analysis; it leverages tumour genomics data to explore, rank, and select optimal cell line models in a user-friendly way, enabling scientists to make appropriate and informed choices about model inclusion/exclusion in retrospective analyses and future studies. Additionally, it allows the selection of models within user-defined contexts, for example, by focusing on genomic alterations occurring in biological pathways of interest or considering only predetermined sub-cohorts of cancer patients. Finally, CELLector identifies combinations of molecular alterations underlying disease subtypes currently lacking representative cell lines, providing guidance for the future development of new cancer models.

License MIT

Encoding UTF-8

LazyData true

Depends arules, dplyr, stringr, data.tree, sunburstR, igraph, collapsibleTree, methods, RCurl

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

CELLector.buildModelMatrix

Mapping cell lines on the CELLector searching space

Description

This function maps cell line on the subtypes identified and assembled in the CELLector searching space, based on the collective presence/absence of the signatures of cancer functional events underlying these subtypes. The subtypes lacking representative cell lines are not considered and, in the output, the subtypes (indicated by their numerical id, which matches that in the CELLector searching space) are ranked based on the greedy algorithm described in [1] based on their covered genomic heterogeneity.

Usage

```
CELLector.buildModelMatrix(Sigs, dataset, searchSpace)
```

Arguments

`Sigs`	A vector of string, in which each element represents a signature of cancer functional events (CFEs, defined in [2]) corresponding to a node in the CELLector searching space. This is expressed as a logic formula (rule), which a cancer patient's genome must satisfy in order to be included in the sub-population represented by the node under consideration. This vector is outputted by the `CELLector.createAllSignatures` function starting from a CELLector searching space (created by the `CELLector.Build_Search_Space`) function

`dataset` A data frame in which the first two columns contain the COSMIC [3] identiefiers and names of cell lines (one per row), respectively, and then binary entries indicating the status of each CFEs (one per column) across cell lines. The format is the same of the entries of the list in the built-in `CELLector.CellLine.BEMs` object.

`searchSpace` A CELLector searching space encoded as binary tree in a navigable table, as returned by the `CELLector.Build_Search_Space` function

Value

A named binary matrix with suptypes numerical identifiers on the rows, cell line names on the column and entries specifiyng whether the cell line in the column is representative of the subtype on the row (based on the collective presence/absence of the corresponding signature of CFEs)

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Najgebauer, H. et al. Genomics Guided Selection of Cancer in vitro Models.

https://doi.org/10.1101/275032

[2] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[3] Forbes, S. A. et al. COSMIC: exploring the world’s knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

See Also

`CELLector.createAllSignatures`,

`CELLector.Build_Search_Space`,

`CELLector.CellLine.BEMs`,

`CELLector.Build_Search_Space`

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


### Change the following two lines to work with a different cancer type
tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building a CELLector searching space focusing on three pathways
### and TP53 wild-type patients only
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathwayFocused = c("RAS-RAF-MEK-ERK / JNK signaling",
                                                     "PI3K-AKT-MTOR signaling",
                                                     "WNT signaling"),
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers,
                                  subCohortDefinition='TP53',
                                  NegativeDefinition=TRUE)

### take all the signatures from the searching space
Signatures <- CELLector.createAllSignatures(CSS$navTable)

### mapping colorectal cancer cell lines onto the CELLector searching space
ModelMat<-CELLector.buildModelMatrix(Signatures$ES,CELLlineData,CSS$navTable)

head(ModelMat)
```

CELLector.Build_Search_Space

CELLector search space construction

Description

This function assembles a user defined CELLector search space analysing genomic data from a larg cohort of cancer patients (specified in input). It identifies recurrent subtypes with matched genomic signatures (as combination of cancer functional events (CFEs), defined in [1]), linking them into a hierarchical structure shaped as a a binary three with a corresponding navigable table, as detailed in [2].

Usage

```
CELLector.Build_Search_Space<-function(ctumours,
                                       cancerType,
                                       minlen=1,
                                       verbose=TRUE,
                                       mutOnly=FALSE,
                                       cnaOnly=FALSE,
                                       includeHMS=FALSE,
                                       minGlobSupp=0.01,
                                       FeatureToExclude=NULL,
                                       pathway_CFEs = NULL,
                                       pathwayFocused=NULL,
                                       subCohortDefinition=NULL,
                                       NegativeDefinition=FALSE,
                                       cnaIdMap=NULL,
                                       cnaIdDecode=NULL,
                                       hmsIdDecode=NULL,
                                       cdg=NULL,
                                       UD_genomics=FALSE)
```

Arguments

`ctumours` A binary event matrix (BEM) modeling a cohort of cancer patients. With cancer functional events (CFEs) on the columns and sample identifers on the rows. See `CELLector.PrimTum.BEMs` for further details

`cancerType` The cancer type under consideration (specified via a TCGA label): currently available types = *BLCA, BRCA, COREAD, GBM, HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC*

`minlen` The minimal length of the genomic signatures (how many indivudal CFEs it is made of) in order to be considered in the analysis (1 by default)

`verbose` A boolean argument specifying whether step-by-step information on the algorithm progression should be displayed run-time

`mutOnly` A boolean argument specifying whether only CFEs involving somatic mutations should be considered in the analysis. If the `cnaOnly` argument is equal to `TRUE` then this must be `FALSE` (default value)

`cnaOnly` A boolean argument specifying whether only CFEs involving copy number alterations (CNAs) of chromosomal segments that are recurrently CN altered should be considered in the analysis. If the `mutOnly` argument is equal to `TRUE` then this must be `FALSE` (default value)

`includeHMS` A boolean argument specitying whether methylation data should be considered while building the searching space (`FALSE` by default).

`minGlobSupp` Minimal size of the outpputted subtypes, as ratio of the number patients included in the whole cohort (1% by default).

`FeatureToExclude`
A string (or a vector of strings) with identifiers of CFEs that should be ignored

`pathway_CFEs` A named list of string vectors, whose elements are CFEs involving genes in a biological pathway (specified by the name of the corresponding entry). A list for 14 key cancer pathways is contained in the `CELLector.Pathway_CFEs` data object (see corresponding help page for further deatails)

`pathwayFocused` If different from `NULL` (default value), it should be a vector of strings. In this case the analysis will consider only CFEs involving genes in a set of pathways, whose names are contained in this argument and must be present as names of the `pathway_CFEs` argument

`subCohortDefinition`
If different from `NULL` (default value), it should be a string containing the identifier of a CFE. In this case the analysis will consider only the primary tumour samples harbouring (or not harbouring, depending on the `NegativeDefinition` argument) the specified CFE

`NegativeDefinition`
If the `subCohortDefinition` argument is not `NULL` then this paramenter determines whether to consider primary tumour samples that harbour (if equal to `FALSE`, default value) or not (if equal to `TRUE`) the specified CFE

`cnaIdMap` A data frame mapping chromosomal regions of recurrent copy number amplifications/deletions in cancer (RACSs, as defined in [1]) identified via ADMIRE [3] in the context of specific cancer types to PanCancer RACSs. The built-in object `CELLector.CFEs.CNAid_mapping` (or an alternative data frame with the same format) should be used.

`cnaIdDecode` A table with identifiers of cancer functional events (CFEs) involving chromosomal regions of recurrent copy number alterations (RACSs, as defined by [1], i.e. identified throgh ADMIRE [3]) and their annotation. The built-in object `CELLector.CFEs.CNAid_decode` (or an alternative data frame with the same format) should be used.

`hmsIdDecode` Data frame containing annotation for the hypermethylated gene promoters CFEs. The format should be the same of the `CELLector.CFEs.HMSid_decode` object.

`UD_genomics` A boolean argument specifying whether the analysis is performed on user defined genomic data (`TRUE`) or CELLector buit-in genomic data (`FALSE`, default value).

`cdg` A list of genes that are used when decoding the identifiers of cancer functional events (CFEs) involving chromosomal regions of recurrent copy number alterations (RACSs, as defined by [1]). These will be visualised in the signatures containing the RACSs including them. A predefined list of high confidence cancer driver genes (from [1]) is provided as built-in data object (`CELLector.HCCancerDrivers`)

Details

Starting from an initial cohort of patients affected by a given cancer type and modeled by the inputted binary event matrix (BEM), the most frequent alteration or set of molecular alterations (depending on the `minlen` argument) with the largest support (the subpopulation of patients in which these alterations occur simultaneously) is identified using the `eclat` function of the `arules` R package.

Based on this, the cohort of patients is split into two subpopulations depending on the collective presence or absence of the identified alterations. This process is then executed recursively on the two resulting subpopulations and it continues until all the alteration sets (with a support of minimal size, as specified in the `minGlobSupp` argument) are identified.

Each of the alterations sets identified through this recursive process is stored in a tree node. Linking nodes identified in adjacent recursions yields a binary tree: the CELLector search space. Each individual path (from the root to a node) of this tree defines a rule (signature), represented as a logic AND of multiple terms (or their negation), one per each node in the path. If the genome of a given patient in the analysed cohort satisfies the rule then it is contained in the subpopulation represented by the terminal node of that path. Collectively, all the paths in the search space provide a representation of the spectrum of combinations of molecular alterations observed in a given cancer type, and their clinical prevalence in the analysed patient population.

Value

A named list with the CELLector search space stored as a `data.tree` object in the `TreeRoot` field and as a *navigable table*: a data frame with a row for each node of the tree and the following columns

`Idx` A numerical index for the node

`Item` The most supported CFE (or a combination of CFE), identified at the iteration in which the node has been added to the three, (i) in the whole cohort of patients (for the `Root`), (ii) in the sub population that satisfies the parent node rule (for `Left.Child` nodes) or (iii) its complement (for `Right.Child` nodes)

`ItemsDecoded` Same as `Item` but with identifiers of RACSs decoded, i.e. with loci and included driver genes (inputted in the `cdg` argument), indicated among brackets

`Type` The node type: Root (first node added), Right.Child (a node resulting from the analyses of the complementar population of patients with respect to that satisfifying the Parent node rule), Left.Child (a node resulting from refining the population of patients satisfifying the Parent node rule)

`Parent.Idx` The numerical index of the parent node (0 for the `Root`)

`AbsSupport` The number of patients satisfying the node rule

`CurrentTotal` The number of patients included in the population under consideration at the iteration time of the node inclusion in the tree, this is the same of the parent's `AbsSupport` for `Left.Child` nodes

`PercSupport` The ratio of patients collectively harbouring the combination of CFEs specified in `Items` within the subpopulation under consideration at the iteration time of the node inclusion in the tree (whose size is specified in `CurrentTotal`)

`GlobalSupport` The ratio of patients satisfying the node rule with respect to the total number of patients in the whole cohort

`Left.Child.Index` Numerical index of the left child node (0 indicates absence of a left child node)

`Right.Child.Index` Numerical index of the right child node (0 indicates absence of a right child node)

`currentPoints` The identifiers of the patients in the sub-population under consideration at the iteration time of the node inclusion in the tree

`currentFeatures` The CFEs considered at the at the iteration time of the node inclusion in the tree

`positivePoints` The identifiers of the patients satisfying the node rule

`COLORS` A vector of strings containing hexadecimal color identifiers: one for each node. These are used by the visualisation functions (`CELLector.visualiseSearchingSpace`, and `CELLector.visualiseSearchi`
and can be changed using the `CELLector.changeSScolors` function.

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Najgebauer, H. et al. Genomics Guided Selection of Cancer in vitro Models.

https://doi.org/10.1101/275032

[3] van Dyk, E., Reinders, M. J. T. & Wessels, L. F. A. A scale-space method for detecting recurrent DNA copy number changes with analytical false discovery rate control. Nucleic Acids Res. 41, e100 (2013).

See Also

`CELLector.PrimTum.BEMs`,

`CELLector.Pathway_CFEs`,

`CELLector.CFEs.CNAid_mapping`,

`CELLector.CFEs.CNAid_decode`,

CELLector.HCCancerDrivers,

CELLector.visualiseSearchingSpace,

CELLector.visualiseSearchingSpace_sunBurst,

CELLector.changeSScolors

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)

### Change the following two lines to work with a different cancer type
tumours_BEM<-CELLector.PrimTum.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building a CELLector searching space focusing on three pathways
### and TP53 wild-type patients only
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathwayFocused = c("RAS-RAF-MEK-ERK / JNK signaling",
                                                     "PI3K-AKT-MTOR signaling",
                                                     "WNT signaling"),
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers,
                                  subCohortDefinition='TP53',
                                  NegativeDefinition=TRUE)

### visualising the CELLector searching space as a binary tree
CSS$TreeRoot

### visualising the first attributes of the tree nodes
CSS$navTable[,1:11]

### visualising the sub-cohort of patients whose genome satisfies the rule of the 4th node
str_split(CSS$navTable$positivePoints[4],',')


######################################################################
### Rebuilding the search space but considering also methylation data

### important!!!: second version of primary tumours' genomic dataset
### (including methylation data should be loaded)
data(CELLector.PrimTum.BEMs_v2)

### Change the following two lines to work with a different cancer type
tumours_BEM<-CELLector.PrimTum.BEMs_v2$COREAD

### unicize the sample identifiers for the tumour data
```

```
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### loading decoding table for hypermethylation CFE identifiers
data(CELLector.CFEs.HMSid_decode)

### building a CELLector searching space
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  hmsIdDecode = CELLector.CFEs.HMSid_decode,
                                  cdg = CELLector.HCCancerDrivers,
                                  subCohortDefinition='TP53',
                                  NegativeDefinition=TRUE,
                                  includeHMS = TRUE)

### visualising the CELLector searching space as a binary tree
CSS$TreeRoot

### visualising the first attributes of the tree nodes
CSS$navTable[,1:11]

### visualising the sub-cohort of patients whose genome satisfies the rule of the 4th node
str_split(CSS$navTable$positivePoints[4],',')
```

CELLector.CellLine.BEMs

Cell Lines' Binary Genomic Event Matrices

Description

A list containing 16 data frames (one for cancer type), identified through TCGA labels. Each of these data frames contains cell lines' *binary event matrices* (BEMs) with the status (presence/absence) of *cancer functional events* (CFEs, somatic mutations in high-confidence cancer driver genes and copy number altered genomic segments) as defined in [1].

Usage

```
data(CELLector.CellLine.BEMs)
```

Format

A named list of data frames (with TCGA cancer type labels as names). Each of these data frames contains two columns with COSMIC [2] identiefiers and names of cell lines (one per row), respectively, and then binary entries indicating the status of each CFEs (one per column) across cell lines.

Details

BEMs for cell lines from the Genomics of Drug Sensitivity in Cancer (GDSC1000, [1]) panel. Data is available for cell lines matching one among 16 different TCGA cancer types: *BLCA, BRCA, COREAD, GBM, HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC*.

A decoding table for these labels is available at

https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources/Data_Summary.html.

Each data frame contains cell lines on the rows (with COSMIC identifiers and names, respectively on first and second column) and then a binary matrix with a CFE per column and entries indicating the presence/absence of a given CFE in a given cell line.

Gene symbols as column names indicate high confidence cancer driver genes and the entries in the corresponding columns indicate the presence/absence of somatic mutations. Column names with *cna* as prefix indicate chromosomal segments that are recurrently copy number altered in cancer (RACSs, defined in [1]). A list with all the considered CFEs is available in the `CELLector.CFEs` data object. A decoding table for the RACSs is available in the `CELLector.CFEs.CNAid_decode`, with the mapping realised by the values in the `CNA_identifier` column.

Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

See Also

`CELLector.PrimTum.BEMs`, `CELLector.CFEs`, `CELLector.CFEs.CNAid_decode`

Examples

```
data(CELLector.CellLine.BEMs)
CELLector.CellLine.BEMs$COREAD[1:10,
                                 c("COSMIC_identifier","CellLine",'BRAF','KRAS','cna27')]
```

`CELLector.CellLine.BEMs_v2` *Cell Lines' Binary Genomic/Epigenomic Event Matrices*

Description

A list containing 16 data frames (one for cancer type), identified through TCGA labels. Each of these data frames contains cell lines' *binary event matrices* (BEMs) with the status (presence/absence) of *cancer functional events* (CFEs, somatic mutations in high-confidence cancer driver genes, copy number altered genomic segments and hypermethylated gene promoters) as defined in [1].

Usage

```
data("CELLector.CellLine.BEMs_v2")
```

Format

A named list of data frames (with TCGA cancer type labels as names). Each of these data frames contains two columns with COSMIC [2] identiefiers and names of cell lines (one per row), respectively, and then binary entries indicating the status of each CFEs (one per column) across cell lines.

Details

BEMs for cell lines from the Genomics of Drug Sensitivity in Cancer (GDSC1000, [1]) panel. Data is available for cell lines matching one among 16 different TCGA cancer types: *BLCA, BRCA, COREAD, GBM, HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC*.

A decoding table for these labels is available at

https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources/Data_Summary.html.

Each data frame contains cell lines on the rows (with COSMIC identifiers and names, respectively on first and second column) and then a binary matrix with a CFE per column and entries indicating the presence/absence of a given CFE in a given cell line.

Gene symbols as row names indicate high confidence cancer driver genes and the entries in the corresponding rows indicate the presence/absence of somatic mutations. Row names with *cna* as prefix indicates chromosomal segments that are recurrently copy number altered in cancer (RACSs, defined in [1]), *hms* as a prefix indicates hypermethylated gene promoters (HMS, defined in [1]). A list with all the considered CFEs is available in the `CELLector.CFEsV2` data object. A decoding table for the RACSs is available in the `CELLector.CFEs.CNAid_decode` object, with the mapping realised by the values in the `CNA_identifier` column. A decoding table for the HMS is available in the `CELLector.CFEs.HMSid_decode` object, with the mapping realised by the values in the `hms_id` column.

Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

See Also

`CELLector.CellLine.BEMs`, `CELLector.PrimTum.BEMs_v2`, `CELLector.CFEsV2`, `CELLector.CFEs.CNAid_decode`, `CELLector.CFEs.HMSid_decode`

Examples

```
data(CELLector.CellLine.BEMs)
CELLector.CellLine.BEMs_v2$COREAD[1:10,
                               c("COSMIC_identifier","CellLine",'BRAF','hms70','cna27')]
```

`CELLector.CellLinesVarCatalog`

Built-in Cell Line variants' catalogue

Description

List of somatic variants found in cell-lines, from Iorio et al, Cell 2016 [1]

Usage

```
data("CELLector.CellLinesVarCatalog")
```

Format

A data frame with 486243 observations (one for each somatic variants) on the following 13 variables.

`SAMPLE` a character vector specifying the name of the cell line in which the variant has been found

`COSMIC_ID` a numeric vector specifying COSMIC [2] identifier of the cell line in which the variant has been found

`Cancer.Type` a character vector specifying the TCGA [3] classification of the tissue of origin of the cell line in which the variant has been found

`Gene` a character vector specifying the HUGO [4] symbol of the gene hosting the variant under consideration (from Ensemble v56)

`Transcript` a character vector specifying the identifier of the transcript affected by the variant under consideration (from Ensemble v56)

`cDNA` a character vector specifying variant position and nucleotide change relating to the cDNA

`AA` a character vector specifying aminoacid positon and alteration related to the variant under consideration

`Classification` a character vector specifying a summary of the variant type (e.g. missense, frameshift, nonsense, etc)

`Gene_list` a character vector: when non empty then the variant is within a high-confidence cancer driver gene (according to [1]), thus resulting in a cancer functional event (CFE) summarised in the Cell line Binary Event Matrices (BEMs)

`Recurrence.Filter` a character vector: if 'Yes' then the variant under consideration is observed in COSMIC [2] (v68) at a minimal threshold frequency (specified in [1])

`Subs` a numeric vector: Missense/substitution variants occurring in codons mutated in the systematic screen data in COSMIC [2] (v68) (select >=3

`Truncating` a numeric vector: Truncating variant count from the systematic screen data in COSMIC [2] (v68) (select >10)

`inframe` a numeric vector: Inframe indel alterations occurring in codons mutated in the systematic screen data in COSMIC [2] (v68) (select >=3)

Author(s)

Francesco Iorio (fi9323@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

[3] Hutter C and Zenklusen JC. The Cancer Genome Atlas: Creating Lasting Value beyond Its Data. Cell. 2018;173(2):283–285. doi:10.1016/j.cell.2018.03.042

[4] Braschi, B. et al. Genenames.org: the HGNC and VGNC resources in 2019. Nucleic Acids Res. Epub 2018 Oct 10. PMID: 30304474 DOI: 10.1093/nar/gky930

See Also

`CELLector.CellLine.BEMs`, `CELLector.CellLine.BEMs_v2`, `CELLector.CFEs`

Examples

```
data(CELLector.CellLinesVarCatalog)
head(CELLector.CellLinesVarCatalog)
```

CELLector.CELLline_buildBEM

Building a Genomic Binary Event Matrix (BEM) for in-vitro models

---

Description

This function takes in input a catalogue of somatic genomic variants observed in cancer cell lines (or it uses the most recent catalogue of somatic genomic variants from the Cell Model Passports [1]) and it converts it into a presence/absence (binary) matrix, which can be processed by the CELLector package and CELLector shiny app for mapping cell lines onto cancer-patient derived genomic subtypes.

Usage

```
CELLector.CELLline_buildBEM(varCat = NULL,
                            Tissue,
                            Cancer_Type,
                            Cancer_Type_details = NULL,
                            sample_site = NULL,
                            excludeOrganoids = FALSE,
                            humanonly = TRUE,
                            msi_status_select = NULL,
                            gender_select = NULL,
                            mutational_burden_th = NULL,
                            age_at_sampling = NULL,
                            ploidy_th = NULL,
                            ethnicity_to_exclude = NULL,
                            GenesToConsider = NULL,
                            VariantsToConsider = NULL)
```

Arguments

varCat
: A data frame containing a catalogue of somatic genomic variants observed in cancer cell lines with one row per variant. At least the following column/headers should be included: model_id (the Cell Model Passport identifier [1] of the cell line in which the variant is observed), gene_symbol (HUGO [2] symbol of the gene hosting the variant), cdna_mutation (variant specification based on the longest transcript of the gene), aa_mutation (aminoacid substitution). See format of the CELLector.RecfiltVariants object for details on the syntax of the variant and aminoacid substitution specifications. When this parameter is different from NULL all the others parameters are ignored by this function, with the exception of GenesToConsider and VariantsToConsider. By default, this parameter is set to NULL and the most up-to-date variant catalogue from the Cell Model Passports [1] is used insted. In this case the other parameters define which samples to include in the final genomic binary event matrix.

Tissue
: String specifying the tissue of origin of the cell lines that should be considering while assembling the final genomic binary event matrix. Possible values are those specified in the tissue column of the Cell Model Passport [1] annotation object, downloadable through the CELLector.CMPs_getModelAnnotation function. This parameter is used only when varCat is not set to its default NULL value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

Cancer_Type
: String (or vector of strings) specifying the cancer type(s) of origin of the cell lines that should be considering while assembling the final genomic binary event matrix. Possible values are those specified in the cancer_type column of the Cell Model Passport [1] annotation object, downloadable through the CELLector.CMPs_getModelAnn function. This parameter is used only when varCat is not set to its default NULL value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

Cancer_Type_details
: String (or vector of strings) specifying the cancer type detail(s) of the cell lines that should be considering while assembling the final genomic binary event matrix. Possible values are those specified in the cancer_type_detail column of the Cell Model Passport [1] annotation object, downloadable through the CELLector.CMPs_getModelAnnotation function. This parameter is used only when varCat is not set to its default NULL value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

sample_site
: String (or vector of strings) specifying the sample site(s) of the cell lines that should be considering while assembling the final genomic binary event matrix. Possible values are those specified in the sample_site column of the Cell Model Passport [1] annotation object, downloadable through the CELLector.CMPs_getModelAnnota function. This parameter is used only when varCat is not set to its default NULL value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

excludeOrganoids
: Boolean parameter specifying whether organoid models should not be considered while assembling the final genomic binary event matrix. This parameter is used only when varCat is not set to its default NULL value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`humanonly` Boolean parameter specifying whether human cell lines only should not be considered while assembling the final genomic binary event matrix. This parameter is used only when `varCat` is not set to its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`msi_status_select`
String parameter specifying whether microsatellite stable or instable cell lines only should not be considered while assembling the final genomic binary event matrix. Possible values for this parameters are "MSI" (for instable cell lines) and "MSS" (for stable cell lines). This parameter is used only when `varCat` is not set its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`gender_select` String (or vector of strings) specifying the gendre of the patients the cell lines that should be considering while assembling the final genomic binary event matrix were resected from. Possible values are those specified in the `gender` column of the Cell Model Passport [1] annotation object, downloadable through the `CELLector.CMPs_getModelAnnotation` function. This parameter is used only when `varCat` is not set to its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`mutational_burden_th`
A vector of two real numbers specifying an interval of number of mutations per million base pairs the mutational burden of the cell lines that should be considering while assembling the final genomic binary event matrix should fall in. This parameter is used only when `varCat` is not set to its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`age_at_sampling`
A vector of two real numbers specifying an age at sampling interval the cell lines that should be considering while assembling the final genomic binary event matrix should fall in. This parameter is used only when `varCat` is not set to its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`ploidy_th` A vector of two real numbers specifying an interval the ploidy of the cell lines that should be considering while assembling the final genomic binary event matrix should fall in. This parameter is used only when `varCat` is not set to its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`ethnicity_to_exclude`
String (or vector of strings) specifying the ethnic groups of the patients the cell lines that should not be considering while assembling the final genomic binary event matrix were resected from. Possible values are those specified in the `ethnicity` column of the Cell Model Passport [1] annotation object, downloadable through the `CELLector.CMPs_getModelAnnotation` function. This parameter is used only when `varCat` is not set to its default `NULL` value, therefore the most-up-to-date variant catalogue from the Cell Model Passports [1] is used instead of a variant catalogue provided in input.

`GenesToConsider`
A list of strings with HGNC symbols [2] for genes hosting the variants to be extracted from the catalogue and assembled into the final matrix. When set to its default `NULL` value, all genes hosting at least one variants are considered.

`VariantsToConsider`

A list of individual somatic variants to be extracted from the catalogue and assembled into the final matrix. The format should be the same of the `CELLector.RecfiltVariants`.

Value

A data frame with one row per cell lines, Cell model passport identifiers and cell line names in the first two columns followed by a presence/absence (binary) matrix with gene symbols on the columns, specifying in the $i, j-$entry the status of the jth gene in the ith cell line, i.e. 0 = wild-type, 1 = mutated.

Author(s)

Francesco Iorio (fi9323@gmail.com)

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. doi:10.1093/nar/gky872

[2] Braschi, B. et al. Genenames.org: the HGNC and VGNC resources in 2019. Nucleic Acids Res. Epub 2018 Oct 10. PMID: 30304474 DOI: 10.1093/nar/gky930

[3] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.CMPs_getModelAnnotation`, `CELLector.Tumours_buildBEM`

Examples

```
## loading high-confidence cancer driver genes from [1]
data(CELLector.HCCancerDrivers)

## loading COSMIC [3] variants observed it at least two patients from [3]
data(CELLector.RecfiltVariants)

## Assembling a genomic binary event matrix (BEM) for human derived microsatellite stable diploid colorectal car
resected from male patients using genomic data from the Cell Model Passports [1]

COREAD_cl_BEM <- CELLector.CELLline_buildBEM(
                  Tissue='Large Intestine',
                  Cancer_Type = 'Colorectal Carcinoma',
                  msi_status_select = 'MSI',
                  ploidy_th = c(2,2),
                  GenesToConsider = CELLector.HCCancerDrivers,
                  VariantsToConsider = CELLector.RecfiltVariants)

## showing first entries of the BEM
head(COREAD_cl_BEM)

## showing a bar diagram with mutation frequencies in the BEM
barplot(sort(colSums(COREAD_cl_BEM[,3:ncol(COREAD_cl_BEM)]),decreasing=TRUE)[1:20],
        las=2,ylab='n. mutated cell lines')
```

`CELLector.CFEs` *Cancer Functional Events*

Description

Identifiers of cancer functional events (CFEs, i.e. somatic mutations in high confidence cancer driver genes or chromosomal regions of recurrent copy number amplification/deletion) from [1], which are also present in the binary event matrices of the cell lines and the primary tumours considered in this version of CELLector.

Usage

```
data("CELLector.CFEs")
```

Format

A vector of strings with one entry per identifier.

Details

Gene symbols indicate somatic mutations in high confidence cancer driver genes and entries with *cna* prefix indicate chromosomal segments that are recurrently copy number altered in cancer (RACSs), both defined in [1].

A decoding table for the RACSs is available in the `CELLector.CFEs.CNAid_decode`, with the mapping realised by the values in the `CNA_identifier` column.

Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.PrimTum.BEMs`, `CELLector.CellLine.BEMs`, `CELLector.CFEs`, `CELLector.CFEs.CNAid_decode`

Examples

```
data(CELLector.CFEs)
head(CELLector.CFEs)
```

CELLector.CFEs.CNAid_decode

Decoding table for copy number alteration cancer functional events

Description

A table with identifiers of cancer functional events (CFEs) involving chromosomal regions of recurrent copy number alterations (RACSs, as defined by [1], i.e. identified throgh ADMIRE [2]) and their annotation.

Usage

```
data("CELLector.CFEs.CNAid_decode")
```

Format

A data frame with 731 observations (one for each CNA CFE) on the following 15 variables.

`Identifier` The RACS identifer, as defined in [1]

`CancerType` A TCGA label indicating the cancer type where the RACS has been identified (via ADMIRE [2])

`Recurrent` A string specifying whether the RACS under consideration is frequently amplified (value = `Amplification`) or deleted) (value = `deleted`)

`chr` Chromosome number of the RACS

`start` Starting position of the RACS

`stop` Ending position of the RACS

`nGenes` Number of protein coding genes included in the RACS

`locus` Genomic locus of the RACS

`ContainedGenes` A string with comma separated symbols of the genes included in the RACS

`CNA_Identifier` A string containing the identifer of the RACS as it appears in the Binary Event Matrix (BEM) of the cancer type specified in the `CancerType` field included in the `CELLector.CellLine.BEMs` and the `CELLector.PrimTum.BEMs` data objects

Details

This data frame contains a comprehensive annotation of the CFEs involving RACSs appearing in the BEMs of cell lines and primary tumours, contained in the `CELLector.CellLine.BEMs` and the `CELLector.PrimTum.BEMs` data objects. Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

This table is used by the `CELLector.cna_look_up` function to decode the identifier of CFE involving a RACS.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] van Dyk, E., Reinders, M. J. T. & Wessels, L. F. A. A scale-space method for detecting recurrent DNA copy number changes with analytical false discovery rate control. Nucleic Acids Res. 41, e100 (2013).

See Also

`CELLector.CellLine.BEMs`, `CELLector.PrimTum.BEMs`, `CELLector.cna_look_up`

Examples

```
data(CELLector.CFEs.CNAid_decode)
head(CELLector.CFEs.CNAid_decode)

data(CELLector.CellLine.BEMs)
colnames(CELLector.CellLine.BEMs$COREAD)[8]

CELLector.cna_look_up(cna_ID = colnames(CELLector.CellLine.BEMs$COREAD)[8],
                      TCGALabel = 'COREAD',
                      cnaId_decode = CELLector.CFEs.CNAid_decode)
```

`CELLector.CFEs.CNAid_mapping` *Pan-Cancer/Cancer-Specific RACSs map.*

Description

A data frame mapping chromosomal regions of recurrent copy number amplifications/deletions in cancer (RACSs, as defined in [1]) identified via ADMIRE [2] in the context of specific cancer types to PanCancer RACSs.

Usage

```
data("CELLector.CFEs.CNAid_mapping")
```

Format

A data frame with 425 observations (one for each PanCancer RACS) and a column for each of 27 different cancer types (specified by TCGA labels). The entry in position *i,j* contains the identifier of the *i*th PanCancer RACS in the context of the *j*th cancer type (where available).

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] van Dyk, E., Reinders, M. J. T. & Wessels, L. F. A. A scale-space method for detecting recurrent DNA copy number changes with analytical false discovery rate control. Nucleic Acids Res. 41, e100 (2013).

Examples

```
data(CELLector.CFEs.CNAid_mapping)
head(CELLector.CFEs.CNAid_mapping)
```

CELLector.CFEs.HMSid_decode

Decoding table for hypermethylated gene promoter cancer functional events

Description

A table with identifiers of cancer functional events (CFEs) involving informative hypermehtylated gene promoters (hms) defined by [1].

Usage

```
data("CELLector.CFEs.HMSid_decode")
```

Format

A data frame with 731 observations (one for each hms CFE) on the following 7 variables.

`Cancer.Types` A TCGA label indicating the cancer type where the hms has been identified as informative (as explained in [1])

`Genomic.Coordinates` The genomic coordinates of the CgP island whose methylation signal was considered in [1] for this hms

`Discretization.threshold` Threshold used to determine the status of this CgP island (hypermethylated or non hypermethylated) defined as the best separative ones between distributions of hypermethylated and non hypermethylated samples when looking at the methylation (beta) signal of the CgP island under consideration in cancer patients, whole procedure described in [1]

`dip.statistic` Dip statistic resulting from testing the bimodality of the methylation (beta) signal of the CgP island under consideration in cancer patients. Whole procedure described in [1]

`dip.pval` Dip p-value from testing the bimodality of the methylation (beta) signal of the CgP island under consideration in cancer patients. Whole procedure described in [1]

`GN` Gene whose promoter is considered by the hms

`hms_id` Alphanumerical identifiers of the hms

Details

This data frame contains the annotation of the CFEs involving the hms appearing in the BEMs of cell lines and primary tumours, contained in the `CELLector.CellLine.BEMs_v2` and the `CELLector.PrimTum.BEMs_v2` data objects (version 2).

This table is used by the `CELLector.hms_look_up` function to decode the identifier of CFE involving an hms.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.CellLine.BEMs_v2`, `CELLector.PrimTum.BEMs_v2`, `CELLector.hms_look_up`

Examples

```
data(CELLector.CFEs.HMSid_decode)
head(CELLector.CFEs.HMSid_decode)

data(CELLector.CellLine.BEMs_v2)
colnames(CELLector.CellLine.BEMs_v2$COREAD)[162]

CELLector.hms_look_up(hms_ID = colnames(CELLector.CellLine.BEMs_v2$COREAD)[162],
                      TCGALabel = 'COREAD',
                      hmsId_decode = CELLector.CFEs.HMSid_decode)
```

`CELLector.CFEsV2` *Cancer Functional Events (version 2)*

Description

Identifiers of cancer functional events (CFEs, i.e. somatic mutations in high confidence cancer driver genes, chromosomal regions of recurrent copy number amplification/deletion, and informative hypermethylated gene promoters) from [1], which are also present in the binary event matrices (version 2) of the cell lines and the primary tumours considered in this version of CELLector.

Usage

```
data("CELLector.CFEsV2")
```

Format

A vector of strings with one entry per identifier.

Details

Gene symbols indicate somatic mutations in high confidence cancer driver genes, entries with *cna* prefix indicate chromosomal segments that are recurrently copy number altered in cancer (RACSs), entries with *hms* prefix indicate hypermehylated gene promoters, all defined in [1].

A decoding table for the HMS is available in the `CELLector.CFEs.HMSid_decode` object, with the mapping realised by the values in the `hms_id` column.

Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.PrimTum.BEMs_v2`, `CELLector.CellLine.BEMs_v2`, `CELLector.CFEs`, `CELLector.CFEs.CNAid_decode`, `CELLector.CFEs.HMSid_decode`

Examples

```
data(CELLector.CFEsV2)
head(CELLector.CFEsV2)
```

`CELLector.changeSScolors`

Changing CELLector searching space color scheme

Description

This function reassigns a set of distinc random colors to the cancer patients' subtypes identified and stored in the CELLector searching space (compute by the `CELLector.Build_Search_Space` function). These colors are used by the visualisation functions `CELLector.visualiseSearchingSpace` and `CELLector.visualiseSearchingSpace_sunBurst`

Usage

```
CELLector.changeSScolors(searchSpace)
```

Arguments

`searchSpace`	A CELLector searching space as returned by the CELLector.Build_Search_Space function

Value

A CELLector searching space as returned by the CELLector.Build_Search_Space function, with new colors

Author(s)

Hanna Najgebauer and Francesco Iorio

See Also

`CELLector.Build_Search_Space`,

`CELLector.Build_Search_Space`,

`CELLector.visualiseSearchingSpace_sunBurst`

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building the CELLector searching space
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  mutOnly = FALSE,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers)

### visualising the Searching space as collapsible tree
CELLector.visualiseSearchingSpace(searchSpace = CSS,CLdata = CELLlineData)
### visualising the Searching space as sunburst
CELLector.visualiseSearchingSpace_sunBurst(searchSpace = CSS)

### changing the searching space colors
CSS<-CELLector.changeSScolors(CSS)

### visualising the Searching space as collapsible tree
CELLector.visualiseSearchingSpace(searchSpace = CSS,CLdata = CELLlineData)
### visualising the Searching space as sunburst
CELLector.visualiseSearchingSpace_sunBurst(searchSpace = CSS)
```

CELLector.CMPs_getDriverGenes

List of high-confidence cancer driver genes from the Cell Model Passports

Usage

```
CELLector.CMPs_getDriverGenes
(URL = "https://cog.sanger.ac.uk/cmp/download/cancer_genes_latest.csv.gz")
```

Arguments

URL	The URL specifying the online location of the file containing the high-confidence cancer driver genes. Default value points to the most up-to-date version of this file (which is keptupdated).

Value

A vector of strings (containinng HGNC symbols) with one entry per high-confidence cancer driver genes.

Author(s)

Francesco Iorio (fi9323@gmail.com)

References

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. doi:10.1093/nar/gky872

Examples

```
CMP_cdgenes<-CELLector.CMPs_getDriverGenes()
  head(CMP_cdgenes)
```

CELLector.CMPs_getModelAnnotation

In-vitro models' annotation from the Cell Model Passports

Usage

```
CELLector.CMPs_getModelAnnotation
(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 this file (which is keptupdated).

Value

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

Author(s)

Francesco Iorio (fi9323@gmail.com)

References

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. doi:10.1093/nar/gky872

Examples

```
 CMP_annotation<-CELLector.CMPs_getModelAnnotation()
 head(CMP_annotation)
```

CELLector.CMPs_getVariants

Somatic variants of in-visto models from the Cell Model Passports

---

Usage

```
CELLector.CMPs_getVariants
(URL = "https://cog.sanger.ac.uk/cmp/download/mutations_2018-08-01_1640.csv.gz")
```

Arguments

URL	The URL specifying the online location of the file containing the somatic variants. Default value points to the most up-to-date version of this file (which is keptupdated).

Value

A data frame with one entry per variant and columns specifying different annotation entries.

Author(s)

Francesco Iorio (fi9323@gmail.com)

References

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. doi:10.1093/nar/gky872

Examples

```
CMP_variants<-CELLector.CMPs_getVariants()
  head(CMP_variants)
```

CELLector.cna_look_up *Decoding identifers of chromosomal regions of recurrent Copy Number Alterations*

Description

This functions shows the annotation for a chromosomal region of recurrent copy number alterations (RACS) as defined in [1].

Usage

```
CELLector.cna_look_up(cna_ID, cnaId_decode, TCGALabel)
```

Arguments

`cna_ID`	A string containin the RACS identifier. Full list available in the `CELLector.CFEs` object.
`cnaId_decode`	A data frame containing the RACSs' annotation, available in the `CELLector.CFEs.CNAid_decode` object
`TCGALabel`	A TCGA label indicating the cancer type under consideration: *BLCA, BRCA, COREAD, GBM, HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC* available in this version.

Value

A data frame with a single line containing the annotation of the RACS indicated in input.

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.CFEs`,

`CELLector.CFEs.CNAid_decode`

Examples

```
data(CELLector.CFEs.CNAid_decode)
CELLector.cna_look_up(cna_ID='cna26',
                      cnaId_decode = CELLector.CFEs.CNAid_decode,
                      TCGALabel = 'BRCA')
```

CELLector.createAllSignatures *Derive signatures underlying the cancer patients' subtypes in the CELLector search space*

Description

This function takes in input the CELLector search space encoded as a binary tree in a *navigable table*. Then, for each individual path (from the root to a node) of this tree it derives a rule (signature), represented as a logic AND of multiple terms (or their negation), one per each node in the path. Negations are added when right branches are encountered.

Usage

```
CELLector.createAllSignatures(NavTab)
```

Arguments

`NavTab`	A CELLector searching space encoded as binary tree in a navigable table, as returned by the `CELLector.Build_Search_Space` function.

Value

A list with two vectors of strings and a numerical vector. Each element of the first two vectors represent a signature of cancer functional events (CFEs, defined in [1]) corresponding to a node in the CELLector searching space. This is expressed as a logic formula (rule), which a cancer patient's genome must satisfy in order to be included in the sub-population represented by the node under consideration. The first vector (`S`) contains decoded signatures, i.e. where the CFEs involving copy number alterations are represented by a genomic loci and contained cancer driver genes. The second vector (`ES`) contains signatures of CFEs as they are represented in the binary event matrix containing the patients genomic data used to build the CELLector searching space. Further deatils are provided in [2]. The third vector (`STS`) contains the percentage of cancer patients belonging to the subtype represented by the signatures.

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Najgebauer, H. et al. Genomics Guided Selection of Cancer in vitro Models.

https://doi.org/10.1101/275032

See Also

`CELLector.Build_Search_Space`

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


### Change the following two lines to work with a different cancer type
tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building a CELLector searching space focusing on three pathways
### and TP53 wild-type patients only
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathwayFocused = c("RAS-RAF-MEK-ERK / JNK signaling",
                                                     "PI3K-AKT-MTOR signaling",
                                                     "WNT signaling"),
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers,
                                  subCohortDefinition='TP53',
                                  NegativeDefinition=TRUE)

### derive signatures from searching space
Signatures <- CELLector.createAllSignatures(CSS$navTable)

data.frame(Signatures = Signatures$S,'SubType Size'=Signatures$STS)
```

CELLector.HCCancerDrivers

High Confidence Cancer Driver genes

Description

A list of high confidence cancer driver genes from [1]

Usage

```
data("CELLector.HCCancerDrivers")
```

Format

A vector of strings with one entry per cancer gene.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

Examples

```
data(CELLector.HCCancerDrivers)
## maybe str(CELLector.HCCancerDrivers) ; plot(CELLector.HCCancerDrivers) ...
```

`CELLector.hms_look_up` *Decoding identifers of hypermethylated gene promoters*

Description

This functions shows the annotation for coded hypermethylated gene promoters as defined in [1].

Usage

```
CELLector.hms_look_up(hms_ID, hmsId_decode, TCGALabel)
```

Arguments

`hms_ID`	A string containin the identifier of the hypermethylated gene promoters. Full list available in the `CELLector.CFEsV2` object.
`hmsId_decode`	A data frame containing the hypermethylated gene promoter annotations, available in the `CELLector.CFEs.HMSid_decode` object
`TCGALabel`	A TCGA label indicating the cancer type under consideration: *BLCA, BRCA, COREAD, GBM , HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC* available in this version.

Value

A data frame with a single line containing the hypermethylated gene promoter indicated in input.

Author(s)

Francesco Iorio (fi9323@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.CFEsV2`

Examples

```
data(CELLector.CFEs.HMSid_decode)
CELLector.hms_look_up('hms26',
                      CELLector.CFEs.HMSid_decode,
                      TCGALabel = 'COREAD')
```

`CELLector.makeSelection` *Genomics guided selection of Cancer Cell lines*

Description

Given a CELLector searching space

(outputted by the `CELLector.Build_Search_Space` function) with tumour genomic subtypes and matched underlying signatures of cancer functional events (CFEs, as defined in [1]), and a map of human cancer cell lines on it (outputted by the `CELLector.buildModelMatrix` function), this function selects *n* most representative cell lines by applying a greedy strategy described in [2] in order to maximise the covered genomic heterogeneity of primary tumours.

Usage

```
CELLector.makeSelection(modelMat, n, searchSpace)
```

Arguments

`modelMat` A named binary matrix with tumour suptypes numerical identifiers on the rows, cell line names on the column and entries specifiyng whether the cell line in the column is representative of the subtype on the row (based on the collective presence/absence of the corresponding signature of CFEs). This is outputted by the `CELLector.buildModelMatrix` function starting from a CELLector search space (outputted by the `CELLector.Build_Search_Space`) and a cell line binary event matrix (BEM): a data frame in which the first two columns contain the COSMIC [3] identiefiers and names of cell lines (one per row), respectively, and then binary entries indicating the status of each CFEs (one per column) across cell lines. The format is the same of the entries of the list in the built-in `CELLector.CellLine.BEMs` object

`n` An integer specifying the number of cell lines to select

`searchSpace` A CELLector searching space, outputted by the `CELLector.Build_Search_Space` from a BEM modeling a cohort of cancer patients. With cancer functional events (CFEs) on the columns and sample identifers on the rows. See CELLector.PrimTum.BEMs for further details

Value

A data frame with one row per selected cell line and the following columns:

`Tumour.SubType.Index` The numerical index of the represented tumour subtype (this is the same index that the subtype has in the inputted CELLector searching space)

`Representative.Cell.Line` The name of the selected cell line

`Signature` The signature of CFEs underlying the subtype under consideration and collectively present in the selected cell line

`percentage.patients` The size of the considered represented subtype with respect the whole cohort of cancer patients

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Najgebauer, H. et al. Genomics Guided Selection of Cancer in vitro Models.

https://doi.org/10.1101/275032

[3] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

See Also

`CELLector.Build_Search_Space`,

`CELLector.buildModelMatrix`,

`CELLector.CellLine.BEMs`,

`CELLector.PrimTum.BEMs`

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


### Change the following two lines to work with a different cancer type
tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building a CELLector searching space focusing on three pathways
### and TP53 wild-type patients only
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathwayFocused = c("RAS-RAF-MEK-ERK / JNK signaling",
                                                     "PI3K-AKT-MTOR signaling",
                                                     "WNT signaling"),
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                        cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                        cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                        cdg = CELLector.HCCancerDrivers,
                                        subCohortDefinition='TP53',
                                        NegativeDefinition=TRUE)


### take all the signatures from the searching space
Signatures <- CELLector.createAllSignatures(CSS$navTable)

### mapping the cell lines on the CELLector searching space
ModelMat<-CELLector.buildModelMatrix(Signatures$ES,CELLlineData,CSS$navTable)


### selecting 10 cell lines
selectedCellLines<-CELLector.makeSelection(modelMat = ModelMat,
                                           n=10,
                                           searchSpace = CSS$navTable)

selectedCellLines
```

CELLector.mostSupported_CFEs

Most recurrent combinations of Cancer Functional Events

Description

This function identifies the most frequent combination of cancer functional events (CFEs) in a large cohort of cancer patients.

Usage

```
CELLector.mostSupported_CFEs(transactions,
                             minSupport = 0.05,
                             minlen = 1,
                             maxLen = 10)
```

Arguments

transactions	A named binary matrix with CFEs on the rows, samples on the columns and entries specifying the presence/absence of a given CFE in a given sample: the *transactions* object.
minSupport	The minimal support that a combination of CFEs must have, i.e. the minimal ratio of samples in which the CFEs must be observed simoultanously, in order to be considered in the analysis.
minlen	The minimal length of a combination of CFEs (of how many indivudal CFE it needs to be composed) in order to be considered in the analysis (1 by default).
maxLen	The maximal length of a combination of CFEs (the maximal number of indivudal CFEs) in order to be considered in the analysis (10 by default).

Details

This function uses the *eclat* function from the R package *arules*.

Value

A list with the following fields:

`MSIS` A string or a vector of strings (depending on the argument `minlen`) specifying the CFE (or the combination of individual CFEs) that is the most frequently observed (simultaneously across the samples in input)

`SUPPORT` The ratio of samples where the combination of CFEs in `MSIS` is obaserved on the total number of samples, i.e. number of columns in the `transactions` argument

`absSUPPORT` The number of samples where the combination of CFEs in `MSIS` is obaserved

`supportingSamples` The identifiers of the samples supporting `MSIS`, i.e. the names of the columns of *transactions*, in which the entries corresponding to `MSIS` rows are equal to 1.

Author(s)

Hanna Najgebauer and Francesco Iorio

References

Najgebauer et al., CELLector: Genomics Guided Selection of Cancer in vitro Models.

doi:10.1101/275032

Examples

```
data(CELLector.PrimTum.BEMs)
RES<-CELLector.mostSupported_CFEs(transactions = t(CELLector.PrimTum.BEMs$COREAD),
                               minlen = 2)
```

`CELLector.MSIstatus` *Cell lines' Microsatellite status*

Description

The microsatellite status of the cell lines in the CELLector collection, which can be stable (MSI-S), lowly instable (MSI-L), or highly instable (MSI-H) from [1]

Usage

```
data("CELLector.MSIstatus")
```

Format

A named vector of string with one entry per cell lines (with COSMIC [2] identifiers as names) specigying the MSI status of each cell line as detailed in the description above.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015

Examples

```
data(CELLector.MSIstatus)
head(CELLector.MSIstatus)
```

CELLector.Pathway_CFEs *Cancer functional events in biological pathways*

Description

Lists of cancer functional events (CFEs) from [1] involving genes in 14 key cancer biological pathways

Usage

```
data("CELLector.Pathway_CFEs")
```

Format

Named list of string vectors, whose elements are CFEs involving genes in a fixed biological pathway.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

Examples

```
data(CELLector.Pathway_CFEs)
CELLector.Pathway_CFEs$`RAS-RAF-MEK-ERK / JNK signaling`
```

`CELLector.PrimTum.BEMs`

Primary Tumours' Binary Genomic Event Matrices

Description

A list containing 16 data frames (one for cancer type), identified through TCGA labels. Each of these data frames contains primary tumours' *binary event matrices* (BEMs) with the status (presence/absence) of *cancer functional events* (CFEs, somatic mutations in high-confidence cancer driver genes and copy number altered genomic segments) as defined in [1].

Usage

```
data("CELLector.PrimTum.BEMs")
```

Format

A named list of binary matrices (with TCGA cancer type labels as names). The entries of each of these matrices indicate the status (Present/Absent) of each CFE (one per row) across primary tumors samples (one per column).

Details

BEMs of primary tumours from the Genomics of Drug Sensitivity in Cancer (GDSC1000, [1]) study. Data is available for 16 different TCGA cancer types: *BLCA, BRCA, COREAD, GBM, HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC*.

A decoding table for these labels is available at

https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources/Data_Summary.html

Each data frame contains primary tumour samples on the columns and CFEs on the rows, with entries indicating the presence/absence of a given CFE in a given primary tumour sample.

Gene symbols as row names indicate high confidence cancer driver genes and the entries in the corresponding rows indicate the presence/absence of somatic mutations. Row names with *cna* as prefix indicate chromosomal segments that are recurrently copy number altered in cancer (RACSs, defined in [1]). A list with all the considered CFEs is available in the `CELLector.CFEs` data object. A decoding table for the RACSs is available in the `CELLector.CFEs.CNAid_decode`, with the mapping realised by the values in the `CNA_identifier` column.

Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

CELLector.PrimTum.BEMs_v2, CELLector.CellLine.BEMs, CELLector.CFEs, CELLector.CFEs.CNAid_decode

Examples

```
data(CELLector.PrimTum.BEMs)
CELLector.PrimTum.BEMs$COREAD[c('BRAF','KRAS','cna27'),1:10]
```

CELLector.PrimTum.BEMs_v2 *Primary Tumours' Binary Genomic/Epigenomic Event Matrices*

Description

A list containing 16 data frames (one for cancer type), identified through TCGA labels. Each of these data frames contains primary tumours' *binary event matrices* (BEMs) with the status (presence/absence) of *cancer functional events* (CFEs, somatic mutations in high-confidence cancer driver genes, copy number altered genomic segments and hypermethylated gene promoters) as defined in [1].

Usage

```
data("CELLector.PrimTum.BEMs_v2")
```

Format

A named list of binary matrices (with TCGA cancer type labels as names). The entries of each of these matrices indicate the status (Present/Absent) of each CFE (one per row) across primary tumors samples (one per column).

Details

BEMs of primary tumours from the Genomics of Drug Sensitivity in Cancer (GDSC1000, [1]) study. Data is available for 16 different TCGA cancer types: *BLCA, BRCA, COREAD, GBM, HNSC, KIRC, LAML, LGG, LUAD, LUSC, OV, PRAD, SKCM, STAD, THCA, UCEC*.

A decoding table for these labels is available at

https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources/Data_Summary.html

Each data frame contains primary tumour samples on the columns and CFEs on the rows, with entries indicating the presence/absence of a given CFE in a given primary tumour sample.

Gene symbols as row names indicate high confidence cancer driver genes and the entries in the corresponding rows indicate the presence/absence of somatic mutations. Row names with *cna* as prefix indicates chromosomal segments that are recurrently copy number altered in cancer (RACSs, defined in [1]), \emphms as a prefix indicates hypermethylated gene promoters (HMS, defined in [1]). A list with all the considered CFEs is available in the `CELLector.CFEsV2` data object. A decoding table for the RACSs is available in the `CELLector.CFEs.CNAid_decode` object, with the mapping realised by the values in the `CNA_identifier` column. A decoding table for the HMS is available in the `CELLector.CFEs.HMSid_decode` object, with the mapping realised by the values in the `hms_id` column.

Please note that the same RACS identifier across multiple cancer types might indicate different chromosomal regions, therefore in order to be decode it should be considered jointly with the TCGA label of the data frame it has been extracted from.

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

`CELLector.PrimTum.BEMs`, `CELLector.CellLine.BEMs_v2`, `CELLector.CFEsV2`, `CELLector.CFEs.CNAid_decode`, `CELLector.CFEs.HMSid_decode`

Examples

```
data(CELLector.PrimTum.BEMs_v2)
CELLector.PrimTum.BEMs_v2$COREAD[c('BRAF','KRAS','hms67','hms70'),]
```

`CELLector.PrimTumVarCatalog` *Built-in Primary Tumour variants' catalogue*

Description

List of somatic variants found in primary tumours, from Iorio et al, Cell 2016 [1]

Usage

```
data("CELLector.PrimTumVarCatalog")
```

Format

A data frame with 727228 observations (one for each somatic variants) on the following 12 variables.

`SAMPLE` a character vector specifying the identifier of the sample in which the variant has been found

`Cancer.Type` a character vector specifying the TCGA [2] classification of the tissue of origin of the tumour in which the variant has been found

`Gene` a character vector specifying the HUGO [3] symbol of the gene hosting the variant under consideration

`Transcript` a character vector specifying the identifier of the transcript affected by the variant under consideration

`cDNA` a character vector specifying variant position and nucleotide change relating to the cDNA

`AA` a character vector specifying aminoacid positon and alteration related to the variant under consideration

`Classification` a character vector specifying a summary of the variant type (e.g. missense, frameshift, nonsense, etc)

`Gene.List` a character vector: when non empty then the variant is within a high-confidence cancer driver gene (according to [1]), thus resulting in a cancer functional event (CFE) summarised in the Primary tumours' Binary Event Matrices (BEMs)

Recurrence.Filter a character vector: if 'Yes' then the variant under consideration is observed in COSMIC [4] (v68) at a minimal threshold frequency (specified in [1])

Subs a numeric vector: Missense/substitution variants occurring in codons mutated in the systematic screen data in COSMIC [2] (v68) (select >=3

Truncating a numeric vector: Truncating variant count from the systematic screen data in COSMIC [2] (v68) (select >10)

inframe a numeric vector: Inframe indel alterations occurring in codons mutated in the systematic screen data in COSMIC [2] (v68) (select >=3)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Hutter C and Zenklusen JC. The Cancer Genome Atlas: Creating Lasting Value beyond Its Data. Cell. 2018;173(2):283–285. doi:10.1016/j.cell.2018.03.042

[3] Braschi, B. et al. Genenames.org: the HGNC and VGNC resources in 2019. Nucleic Acids Res. Epub 2018 Oct 10. PMID: 30304474 DOI: 10.1093/nar/gky930

[4] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

Examples

```
data(CELLector.PrimTumVarCatalog)
head(CELLector.PrimTumVarCatalog)
```

CELLector.RecfiltVariants

Somatic mutations recurrently observed in cancer

Description

Somatic variant passing a recurrence based threshold (defined in [1]) in COSMIC [2]

Usage

```
data("CELLector.RecfiltVariants")
```

Format

A vector of string with one entry per variant. Each string is a space delimited list of three elements: The HUGO symbol of the gene hosting the variant, a string codifying the effect of the variant on cDNA, a string codifying the effect of the variant in terms of aminoacid substitutions.

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

Examples

```
data(CELLector.RecfiltVariants)
head(CELLector.RecfiltVariants)
```

`CELLector.Score` *Cell line quality score*

Description

This function computes a quality score of each cell line in terms of its ability to represent an entire cohort of considered disease matching patients. This is quantified as a trade-off between two factors. The first factor is the length of the CELLector signatures (in terms of number of composing individual alterations) that are present in the cell line under consideration. This is proportional to the granularity of the representative ability of the cell line, i.e. the longest the signature the more precisely defined is the represented sub-cohort of patients. The second factor is the size of the patient subpopulation represented by the signatures that can be observed in the cell line under consideration, thus accounting for the prevalence of the sub-cohort modeled by that cell line. An input parameter allows for these two factors to be weighted equally or differently.

Usage

```
CELLector.Score(NavTab, CELLlineData, alpha = 0.75)
```

Arguments

`NavTab` A CELLector searching space encoded as binary tree in a navigable table, as returned by the `CELLector.Build_Search_Space` function.

`CELLlineData` A data frame in which the first two columns contain the COSMIC [1] identiefiers (or any other alphanumeric identifier) and names of cell lines (one per row), respectively, and then binary entries indicating the status of cancer functional events (CFEs, as defined in [2], one per column) across cell lines. The format is the same of the entries of the list in the built-in `CELLector.CellLine.BEMs`. Based on this data the cell lines are mapped onto the CELLector searching space and their quality assessed.

`alpha` A parameter that weights the contribution of two factors accounted to compute the quality score of a cell line: respectively, the length of a signature that can be observed in a given cell line (weight = alpha) and the ratio of the sub-cohort of patients represented by that signature (1 - alpha). A cell line can be positive for multiple signatures, thus resulting in multiple quality scores. The largest is selected for each cell line.

Value

A data frame with one row per cell line and the following columns:

`CellLines` containing an alphanumerical identifier of each cell line. These are the same as in `CELLlineData`;

`GlobalSupport` the ratio of patients represented by the signature yielding the best `CELLectorScore` for the cell line under consideration;

SignatureLength
: the length of the signature (in terms of number of individual CFEs [2]) yielding the best CELLectorScore for the cell line under consideration;

CELLectorScores
: the CELLector quality scores;

Signature
: The signature yielding the best CELLector quality score for the cell line under consideration.

Author(s)

Francesco Iorio (fi9232@gmail.com)

References

[1] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

[2] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

CELLector.Build_Search_Space

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


### Change the following two lines to work with a different cancer type
tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD


### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building a CELLector searching space focusing on three pathways
### and TP53 wild-type patients only
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathwayFocused = c("RAS-RAF-MEK-ERK / JNK signaling",
                                                     "PI3K-AKT-MTOR signaling",
                                                     "WNT signaling"),
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers,
                                  subCohortDefinition='TP53',
                                        NegativeDefinition=TRUE)

### Evaluating cell line quality
CScores<-CELLector.Score(NavTab=CSS$navTable,CELLlineData = CELLlineData)

### Visualising best cell lines
head(CScores)
```

CELLector.selectionVisit

Visiting the CELLector searching space

Description

This function visits the CELLector searching space using the greedy algorithm described in [1] and it returns the cancer patients subtypes identifiers sorted for cell line selection, maximising the genomic heterogeneity observed in the analysed cohort of cancer patients when considering the first *n* subtypes.

Usage

```
CELLector.selectionVisit(TAV)
```

Arguments

`TAV`	A CELLector searching space stored as a *Navigable Table*, outputted by the `CELLector.Build_Search_Space` function

Value

A numerical vector containing the subtypes identifers sorted as explained above

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Najgebauer, H. et al. Genomics Guided Selection of Cancer in vitro Models.

https://doi.org/10.1101/275032

See Also

`CELLector.Build_Search_Space`

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building the CELLector searching space
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  mutOnly = FALSE,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers)

CELLector.selectionVisit(CSS$navTable)
```

CELLector.solveFormula

Identify cell lines harbouring a signature of Cancer Functional Events

Description

This function takes in input a signature of Cancer Functional Events (CFEs, defined in [1]) as outputted by the `CELLector.createAllSignatures` function, and a binary event matrix (BEM) modeling the presence/absence of all the CFEs across a set of immortalised human cancer cell lines. It returns the set of cell line collectively harbouring the inputted signature.

Usage

```
CELLector.solveFormula(RULE, dataset, To_beExcluded = NULL)
```

Arguments

`RULE` A string representing a signature of cancer functional events (CFEs, defined in [1]), i.e. names of CFEs space separated (and possibly negated ~)

`dataset` A data frame in which the first two columns contain the COSMIC [2] identiefiers and names of cell lines (one per row), respectively, and then binary entries indicating the status of each CFEs (one per column) across cell lines. The format is the same of the entries of the list in the built-in `CELLector.CellLine.BEMs` object.

`To_beExcluded` If different from NULL (default value), then this must be a list of strings with cell line names that should be excluded a priori from the output.

Value

A list with the following entries:

`PS` Positive samples: names of the cell lines collectively harbouring the signature of CFEs provided in input

`N` The number of cell lines in `PS`

`PERC` The number of cell lines collectively harbouring the signatures of CFEs provided in input as ratio of the total number of cell lines in the inputted BEM

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

See Also

`CELLector.createAllSignatures`, `CELLector.CellLine.BEMs`

Examples

```
data(CELLector.CellLine.BEMs)

### Selecting colorectal cancer cell lines that are
### APC wild type and BRAF mutant
CellLines<-CELLector.solveFormula('~APC BRAF',
                                  CELLector.CellLine.BEMs$COREAD)

CellLines
```

`CELLector.Tumours_buildBEM` *Building a Genomic Binary Event Matrix (BEM) for primary tumours*

Description

This function takes in input a catalogue of somatic genomic variants observed in primary tumours (or it uses a built in catalogue from TCGA, presented in [1]) and it converts it into a presence/absence (binary) matrix, which can be processed by the CELLector package and CELLector shiny app for identifying patient subtypes, and map in vitro models onto these.

Usage

```
CELLector.Tumours_buildBEM(varCat = NULL,
                           Cancer_Type,
                           GenesToConsider = NULL,
                           VariantsToConsider = NULL)
```

Arguments

- `varCat` A data frame containing a catalogue of somatic genomic variants observed in primary tumours, with one row per variant. The format should be the same of the `CELLector.PrimTumVarCatalog` data object (which is used when this parameter is set to its default `NULL` value) or at least contain the following column headers `SAMPLE`, `Cancer.Type`, `Gene`, `cDNA`, `AA`, `Recurrence.Filter`
- `Cancer_Type` A string specifying the cancer type for which individual variants should be extracted from the catalogue and assembled into the final matrix. It must be a value included in the `Cancer.Type` column of the `varCat` data object
- `GenesToConsider` A list of strings with HGNC symbols [2] for genes hosting the variants to be extracted from the catalogue and assembled into the final matrix. When set to its default `NULL` value, all genes hosting at least one variants are considered.
- `VariantsToConsider` A list of individual somatic variants to be extracted from the catalogue and assembled into the final matrix. The format should be the same of the `CELLector.RecfiltVariants` (which is used when this parameter is set to its default `NULL` value)

Value

A presence/absence (binary) matrix with gene symbols on the rows and patient sample ids on the columns, specifying in the $i, j-$entry the status of the ith gene in the jth patient sample, i.e. 0 = wild-type, 1 = mutated.

Author(s)

Francesco Iorio (fi9323@gmail.com)

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

[2] Braschi, B. et al. Genenames.org: the HGNC and VGNC resources in 2019. Nucleic Acids Res. Epub 2018 Oct 10. PMID: 30304474 DOI: 10.1093/nar/gky930

[3] Tate JG, Bamford S, Jubb HC, et al. COSMIC: the Catalogue Of Somatic Mutations In Cancer. Nucleic Acids Res. 2019;47(D1):D941–D947. doi:10.1093/nar/gky1015

See Also

`CELLector.PrimTumVarCatalog`, `CELLector.CELLline_buildBEM`

Examples

```
## loading high-confidence cancer driver genes from [1]
data(CELLector.HCCancerDrivers)

## loading COSMIC [3] variants observed it at least two patients from [1]
data(CELLector.RecfiltVariants)

## Assembling a BRCA primary tumour binary event matrix (BEM)
BRCA_tum_BEM<-
  CELLector.Tumours_buildBEM(Cancer_Type = 'BRCA',
                                VariantsToConsider =
                                CELLector.RecfiltVariants)

## showing first 100 entries of the BEM
BRCA_tum_BEM[1:10,1:10]

## showing a bar diagram with mutation frequency of 30 top frequently altered genes
barplot(100*sort(rowSums(BRCA_tum_BEM),
                 decreasing=TRUE)[1:30]/ncol(BRCA_tum_BEM),
                 las=2,ylab='% patients')
```

CELLector.unicizeSamples

Unicize patient samples' identifiers

Description

This function checks if there are multiple samples derived from the same patients in the binary event matrix (BEM) modeling the presence/absence of the cancer functional events (CFEs, defined in [1]), in the cancer patients. These can be mantained (and in this case their identifier will be made unique) or discarded

Usage

```
CELLector.unicizeSamples(ctumours,
                         keepReplicates = TRUE)
```

Arguments

`ctumours`	A binary matrix with entries indicating the status (Present/Absent) of each CFE (one per row) across primary tumors samples (one per column).
`keepReplicates`	A boolean value indicating whether the duplicated samples should be kept (and their identifier made unique, by adding a progressive numerical suffix) or discarded (in this case only one sample per patient will be kept and identifiers unchanged).

Value

A binary matrix with entries indicating the status (Present/Absent) of each CFE (one per row) across primary tumors samples (one per column), and with unique patients' (column) identifiers

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

CELLector.PrimTum.BEMs

Examples

```
data(CELLector.PrimTum.BEMs)

tumours_data<-CELLector.PrimTum.BEMs$COREAD

dim(tumours_data)
length(unique(colnames(tumours_data)))

tumours_data<-CELLector.unicizeSamples(tumours_data)

dim(tumours_data)
length(unique(colnames(tumours_data)))
```

CELLector.visualiseSearchingSpace

Visualising the CELLector searching space as a collapsibe tree

---

Description

This function visualise the CELLector searching space as an interactive collapsible tree. Moving the mouse over the nodes of this tree shows patients' subtype details and (optionally) matched cell lines

Usage

```
CELLector.visualiseSearchingSpace(searchSpace, CLdata = NULL)
```

Arguments

searchSpace: A CELLector searching space as returned by the CELLector.Build_Search_Space function

CLdata: If different from NULL (default) this argument must contain a data frame in which the first two columns contain the COSMIC [1] identiefiers and names of cell lines (one per row), respectively, and then binary entries indicating the status of cancer functional events (CFEs, as defined in [2], one per column) across cell lines. The format is the same of the entries of the list in the built-in CELLector.CellLine.BEMs. Based on this data the cell lines are mapped onto the CELLector searching space and the number of cell lines mapped in each subtype showed when moving the mouse on the corresponding node.

Author(s)

Hanna Najgebauer and Francesco Iorio

References

[1] Forbes, S. A. et al. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. Nucleic Acids Res. 43, D805–11 (2015).

[2] Iorio, F. et al. A Landscape of Pharmacogenomic Interactions in Cancer. Cell 166, 740–754 (2016).

See Also

CELLector.Build_Search_Space,

CELLector.CellLine.BEMs

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building the CELLector searching space
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  mutOnly = FALSE,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers)

### visualising the Searching space as collapsible tree
CELLector.visualiseSearchingSpace(searchSpace = CSS,CLdata = CELLlineData)
```

CELLector.visualiseSearchingSpace_sunBurst

Visualising the CELLector searching space as a sunburst

Description

This function visualise the CELLector searching space as an interactive sunburst.

Usage

```
CELLector.visualiseSearchingSpace_sunBurst(searchSpace)
```

Arguments

searchSpace	A CELLector searching space as returned by the CELLector.Build_Search_Space function

Author(s)

Hanna Najgebauer and Francesco Iorio

See Also

CELLector.Build_Search_Space,

Examples

```
data(CELLector.PrimTum.BEMs)
data(CELLector.Pathway_CFEs)
data(CELLector.CFEs.CNAid_mapping)
data(CELLector.CFEs.CNAid_decode)
data(CELLector.HCCancerDrivers)
data(CELLector.CellLine.BEMs)


tumours_BEM<-CELLector.PrimTum.BEMs$COREAD
CELLlineData<-CELLector.CellLine.BEMs$COREAD

### unicize the sample identifiers for the tumour data
tumours_BEM<-CELLector.unicizeSamples(tumours_BEM)

### building the CELLector searching space
CSS<-CELLector.Build_Search_Space(ctumours = t(tumours_BEM),
                                  verbose = FALSE,
                                  minGlobSupp = 0.05,
                                  cancerType = 'COREAD',
                                  pathway_CFEs = CELLector.Pathway_CFEs,
                                  cnaIdMap = CELLector.CFEs.CNAid_mapping,
                                  mutOnly = FALSE,
                                  cnaIdDecode = CELLector.CFEs.CNAid_decode,
                                  cdg = CELLector.HCCancerDrivers)

### visualising the Searching space as sunburst
CELLector.visualiseSearchingSpace_sunBurst(searchSpace = CSS)
```

Index