Package 'CELLector'

April 12, 2018

Type Package

Title Genomics guided selection of cancer cell lines

Version 0.2.0

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

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 function

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,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 (spcified 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(ctumours,
                             cancerType,
                             minlen = 1,
                             verbose = TRUE,
                             mutOnly = FALSE,
                             cnaOnly = FALSE,
                             minGlobSupp = 0.01,
                             FeatureToExclude = NULL,
                             pathway_CFEs = NULL,
                             pathwayFocused = NULL,
                             subCohortDefinition = NULL,
                             NegativeDefinition = FALSE,
                             cnaIdMap,
                             cnaIdDecode,
                             cdg)
```

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 (spcified 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)

`minGlobSupp` Minimal size of the outpputted subtypes, as ratio of the number patients included in the whole cohort.

`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.

`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

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

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)

### 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 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) 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 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.

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.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 igh 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.

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.

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)
haed(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).

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.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

```
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.

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.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,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)
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.

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.

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 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) 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 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.

References

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

See Also

`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.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.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)))
```

Index