Package 'easier'

April 15, 2024

Title Estimate Systems Immune Response from RNA-seq data

Version 1.9.0

Description This package provides a workflow for the use of EaSIeR tool, developed to assess patients' likelihood
to respond to ICB therapies providing just the patients' RNA-
seq data as input. We integrate RNA-seq
data with different types of prior knowledge to extract quantitative descriptors of the tumor microenvironment from several points of view, including composition of the immune repertoire, and
activity of intra- and extra-cellular communications. Then, we use multi-
task machine learning trained
in TCGA data to identify how these descriptors can simultaneously predict several state-of-the-art hallmarks of anti-cancer immune response. In this way we derive cancer-
specific models and identify
cancer-specific systems biomarkers of immune response. These biomarkers have been experimentally validated
in the literature and the performance of EaSIeR predictions has been validated using independent datasets
form four different cancer types with patients treated with anti-PD1 or anti-PDL1 therapy.

License MIT + file LICENSE

Depends R (>= 4.2.0)

Imports progeny, easierData, dorothea (>= 1.6.0), decoupleR,
quantiseqr, ROCR, grDevices, stats, graphics, ggplot2, ggpubr,
DESeq2, utils, dplyr, tidyr, tibble, matrixStats, rlang,
BiocParallel, reshape2, rstatix, ggrepel, magrittr, coin

Suggests knitr, rmarkdown, BiocStyle, testthat, SummarizedExperiment,
viper

biocViews GeneExpression, Software, Transcription, SystemsBiology,
Pathways, GeneSetEnrichment, ImmunoOncology, Epigenetics,
Classification, BiomedicalInformatics, Regression,
ExperimentHubSoftware

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/easier

git_branch devel

git_last_commit 99b7351

git_last_commit_date 2023-10-24

Repository Bioconductor 3.19

Date/Publication 2024-04-15

Author Oscar Lapuente-Santana [aut, cre]
(<https://orcid.org/0000-0003-1995-8393>),
Federico Marini [aut] (<https://orcid.org/0000-0003-3252-7758>),
Arsenij Ustjanzew [aut] (<https://orcid.org/0000-0002-1014-4521>),
Francesca Finotello [aut] (<https://orcid.org/0000-0003-0712-4658>),
Federica Eduati [aut] (<https://orcid.org/0000-0002-7822-3867>)

Maintainer Oscar Lapuente-Santana <o.lapuente.santana@tue.nl>

Contents

assess_immune_response . . . 3
calc_z_score . . . 6
categorize_TMB . . . 7
compute_Ayers_expIS . . . 8
compute_CCpair_score . . . 8
compute_CC_pairs . . . 10
compute_cell_fractions . . . 11
compute_chemokines . . . 13
compute_CYT . . . 14
compute_Davoli_IS . . . 14
compute_IFNy . . . 15
compute_IMPRES_MSI . . . 16
compute_LR_pairs . . . 17
compute_pathway_activity . . . 18
compute_RIR . . . 19
compute_Roh_IS . . . 20
compute_scores_immune_response . . . 21
compute_Tcell_inflamed . . . 23
compute_TF_activity . . . 24
compute_TLS . . . 25
discretize . . . 25
easier-pkg . . . 27
explore_biomarkers . . . 27
get_OE_bulk . . . 30
get_semi_random_OE . . . 31
predict_immune_response . . . 33

predict_with_rmtlr 36
reannotate_genes 37
retrieve_easier_score 38
rmtlr_test 41

Index **43**

`assess_immune_response` *Assess easier score as predictor of patients' immune response*

Description

Evaluates the predictive performance of easier score as predictor of patients' immune response. This is done for each quantitative descriptor, an ensemble descriptor based on the average of the individual ones, and the gold standard scores. If provided, tumor mutational burden (TMB) is also used as predictor for comparison. Since both immune response and TMB are essential for effective immunotherapy response, an integrated score is calculated given two different approaches based on a applying either a weighted average or penalty to patients' easier score depending on their TMB category.

Usage

```
assess_immune_response(
  predictions_immune_response = NULL,
  patient_response = NULL,
  RNA_tpm = NULL,
  select_gold_standard = NULL,
  TMB_values = NULL,
  easier_with_TMB = "none",
  weight_penalty = NULL,
  verbose = TRUE
)
```

Arguments

`predictions_immune_response`
: list containing the predictions for each quantitative descriptor and for each task. This is the output from `predict_immune_response`.

`patient_response`
: character vector with two factors (Non-responders = NR, Responders = R).

`RNA_tpm`
: numeric matrix of patients' gene expression data as tpm values.

`select_gold_standard`
: character string with names of scores of immune response to be computed. Default scores are computed for: "CYT", "Roh_IS", "chemokines", "Davoli_IS", "IFNy", "Ayers_expIS", "Tcell_inflamed", "RIR", "TLS".

`TMB_values`
: numeric vector containing patients' tumor mutational burden (TMB) values.

`easier_with_TMB`
: character string indicating which approach should be used to integrate easier with TMB. If `TMB_values` provided, one of "weighted_average" (default) or "penalized_score". If `TMB_values` not provided, default is "none".

`weight_penalty`
: integer value from 0 to 1, which is used to define the weight or penalty for combining easier and TMB scores based on a weighted average or penalized score, in order to derive a score of patient's likelihood of immune response. The default value is 0.5.

`verbose`
: logical flag indicating whether to display messages about the process.

Value

When `patient_response` is provided, a roc curve plot and a bar plot that displays the average (across tasks) area under the ROC curve (AUC) values is returned. If `patient_response` is not provided, the easier score is represented as box plots (10 tasks) for each patient.

When `patient_response` is provided and `easier_with_TMB` = `weighted_average` or `easier_with_TMB` = `penalized_score`, an scatter plot that shows the AUC values of the integrated approach, easier score and TMB is returned. If in this case, `patient_response` is not provided, the integrated score is represented as a dot plot for each patient.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
cancer_type <- metadata(dataset_mariathasan)[["cancertype"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of TF activity (Garcia-Alonso et al., Genome Res, 2019)
tf_activities <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)

# Predict patients' immune response
predictions <- predict_immune_response(
  tfs = tf_activities,
  cancer_type = cancer_type,
  verbose = TRUE
)
```

```
# retrieve clinical response
patient_ICBresponse <- colData(dataset_mariathasan)[["BOR"]]
names(patient_ICBresponse) <- colData(dataset_mariathasan)[["pat_id"]]

# retrieve TMB
TMB <- colData(dataset_mariathasan)[["TMB"]]
names(TMB) <- colData(dataset_mariathasan)[["pat_id"]]

patient_ICBresponse <- patient_ICBresponse[names(patient_ICBresponse) %in% pat_subset]
TMB <- TMB[names(TMB) %in% pat_subset]

# Assess patient-specific likelihood of response to ICB therapy
output_eval_with_resp <- assess_immune_response(
  predictions_immune_response = predictions,
  patient_response = patient_ICBresponse,
  RNA_tpm = RNA_tpm,
  select_gold_standard = "IFNy",
  TMB_values = TMB,
  easier_with_TMB = "weighted_average",
)


RNA_counts <- assays(dataset_mariathasan)[["counts"]]
RNA_counts <- RNA_counts[, colnames(RNA_counts) %in% pat_subset]

# Computation of cell fractions  (Finotello et al., Genome Med, 2019)
cell_fractions <- compute_cell_fractions(RNA_tpm = RNA_tpm)

# Computation of pathway scores (Holland et al., BBAGRM, 2019;
# Schubert et al., Nat Commun, 2018)
pathway_activities <- compute_pathway_activity(
  RNA_counts = RNA_counts,
  remove_sig_genes_immune_response = TRUE
)

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)

# Computation of cell-cell interaction scores
ccpair_scores <- compute_CC_pairs(
  lrpairs = lrpair_weights,
  cancer_type = "pancan"
)

# Predict patients' immune response
predictions <- predict_immune_response(
  pathways = pathway_activities,
  immunecells = cell_fractions,
  tfs = tf_activities,
```

```
  lrpairs = lrpair_weights,
  ccpairs = ccpair_scores,
  cancer_type = cancer_type,
  verbose = TRUE
)

# Assess patient-specific likelihood of response to ICB therapy
output_eval_with_resp <- assess_immune_response(
  predictions_immune_response = predictions,
  patient_response = patient_ICBresponse,
  RNA_tpm = RNA_tpm,
  TMB_values = TMB,
  easier_with_TMB = "weighted_average",
)
```

`calc_z_score` *Perform matrix Z-score normalization*

Description

Applies z-score normalization on a numeric matrix per column. Z-score values are calculated based on the input matrix. If mean and standard deviation values are provided, these are used instead.

Usage

```
calc_z_score(X, mean, sd)
```

Arguments

`X`	numeric matrix.
`mean`	numeric vector with mean values.
`sd`	numeric vector with standard deviation values.

Value

A numeric matrix with values as z-scores.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
```

```
# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# apply z-score normalization
tpm_zscore <- calc_z_score(t(RNA_tpm))
```

`categorize_TMB` *Define tumor mutational burden (TMB) categories.*

Description

Encodes tumor mutational burden (TMB) from numerical into categorical variable.

Usage

```
categorize_TMB(TMB, thresholds = NULL)
```

Arguments

`TMB`	numeric vector with tumor mutational burden values.
`thresholds`	numeric vector to specify thresholds to be used. Default thresholds are low (<100), moderate (100-400) and high TMB (>400).

Value

A numeric vector assigning each sample a class from 1 (low TMB) to 3 (high TMB).

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")
dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()

TMB <- colData(dataset_mariathasan)[["TMB"]]
names(TMB) <- colData(dataset_mariathasan)[["pat_id"]]

# Convert TMB continous values into categories
TMB_cat <- categorize_TMB(TMB = TMB)
```

compute_Ayers_expIS *Compute Expanded Immune signature (Ayers_expIS) score*

Description

Calculates Ayers_expIS score as the average expression of its signature genes, as defined in Ayers et al., J. Clin. Invest, 2017.

Usage

```
compute_Ayers_expIS(matches, RNA_tpm)
```

Arguments

matches	numeric vector indicating the index of signature genes in RNA_tpm.
RNA_tpm	numeric matrix with rows=genes and columns=samples.

Value

A numeric matrix with rows=samples and columns=Expanded Immune signature score.

References

Ayers, M., Lunceford, J., Nebozhyn, M., Murphy, E., Loboda, A., Kaufman, D.R., Albright, A., Cheng, J.D., Kang, S.P., Shankaran, V., et al. (2017). IFN-y-related mRN A profile predicts clinical response to PD-1 blockade. J. Clin. Invest. 127, 2930–2940. https://doi.org/10.1172/JCI91190.

compute_CCpair_score *Compute cell-cell pair score*

Description

Derives a score for each cell-cell pair feature.

Usage

```
compute_CCpair_score(
  celltype1,
  celltype2,
  intercell_network,
  lrpairs_binary,
  lr_frequency,
  compute_log = TRUE
)
```

Arguments

celltype1	string character with first cell type involved in the interaction.
celltype2	string character with second cell type involved in the interaction.
intercell_network	matrix with data on cell types interaction network. This is available from easierData package through `easierData::get_intercell_networks()`.
lrpairs_binary	binary vector displaying LR pairs with non-zero frequency.
lr_frequency	numeric vector with LR pairs frequency across the whole TCGA database. This is available from easierData package through `easierData::get_lr_frequency_TCGA()`.
compute_log	boolean variable indicating whether the log of the weighted score should be returned.

Value

A numeric vector with weighted scores.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)

# remove ligand receptor pairs that are always NA
na_lrpairs <- apply(lrpair_weights, 2, function(x) {
  all(is.na(x))
})
lrpair_weights <- lrpair_weights[, na_lrpairs == FALSE]

# binarize the data: set a threshold to 10 TPM,
# only pairs where both ligand and receptor have TPM > 10 are kept
lrpairs_binary <- ifelse(lrpair_weights > log2(10 + 1), 1, 0)
```

```
# keep only the LR.pairs for which I have (non-zero) frequencies in the TCGA
lr_frequency <- suppressMessages(easierData::get_lr_frequency_TCGA())
lrpairs_binary <- lrpairs_binary[, colnames(lrpairs_binary) %in% names(lr_frequency)]

# cancer type specific network
intercell_networks <- suppressMessages(easierData::get_intercell_networks())
intercell_network_pancan <- intercell_networks[["pancan"]]
celltypes <- unique(c(
  as.character(intercell_network_pancan$cell1),
  as.character(intercell_network_pancan$cell2)
))
celltype1 <- celltypes[1]
celltype2 <- celltypes[1]

# compute the CC score for each patient
CCpair_score <- compute_CCpair_score(celltype1, celltype2,
  intercell_network_pancan,
  lrpairs_binary, lr_frequency,
  compute_log = TRUE
)
```

`compute_CC_pairs` *Compute cell-cell interactions scores using computed ligand-receptor weights*

Description

Infers scores of cell-cell interactions in the tumor microenvironment (Lapuente-Santana et al., Patterns, 2021) using the ligand-receptor weights obtained from `compute_LR_pairs` as input.

Usage

```
compute_CC_pairs(lrpairs = NULL, cancer_type = "pancan", verbose = TRUE)
```

Arguments

`lrpairs`	output of the compute_LR_pairs function. A matrix of log2(TPM +1) weights with samples in rows and ligand-receptor pairs in columns. This is the output from `compute_LR_pairs`.
`cancer_type`	string detailing the cancer type whose cell-cell interaction network will be used. By default, a pan-cancer network is selected whose network represents the union of all ligand-receptor pairs present across the 18 cancer types studied in Lapuente-Santana et al., Patterns, 2021.
`verbose`	logical value indicating whether to display informative messages about the process.

Value

A matrix of scores with samples in rows and cell-cell pairs in columns.

References

Oscar Lapuente-Santana, Maisa van Genderen, Peter A. J. Hilbers, Francesca Finotello, and Federica Eduati. 2021. Interpretable Systems Biomarkers Predict Response to Immune-Checkpoint Inhibitors. Patterns, 100293. https://doi.org/10.1016/j.patter.2021.100293.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)

# Computation of cell-cell interaction scores
ccpair_scores <- compute_CC_pairs(
  lrpairs = lrpair_weights,
  cancer_type = "pancan"
)
```

---

compute_cell_fractions

Compute immune cell fractions from gene expression using quanTIseq

---

Description

Estimates cell fractions from TPM bulk gene expression using quanTIseq method from Finotello et al., Genome Med, 2019.

Usage

```
compute_cell_fractions(RNA_tpm = NULL, verbose = TRUE)
```

Arguments

RNA_tpm	data.frame containing TPM values with HGNC symbols in rows and samples in columns.
verbose	logical value indicating whether to display messages about the number of immune cell signature genes found in the gene expression data provided.

Value

A numeric matrix of normalized enrichment scores with samples in rows and cell types in columns.

References

Finotello F, Mayer C, Plattner C, Laschober G, Rieder D, Hackl H, Krogsdam A, Loncova Z, Posch W, Wilflingseder D, Sopper S, Ijsselsteijn M, Brouwer TP, Johnson D, Xu Y, Wang Y, Sanders ME, Estrada MV, Ericsson-Gonzalez P, Charoentong P, Balko J, de Miranda NFDCC, Trajanoski Z. Molecular and pharmacological modulators of the tumor immune contexture revealed by deconvolution of RNA-seq data. Genome Medicine, 2019. 11(1):34. https://doi.org/10.1186/s13073-019-0638-6

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Some genes are causing issues due to approved symbols matching more than one gene
genes_info <- easier:::reannotate_genes(cur_genes = rownames(RNA_tpm))
## Remove non-approved symbols
non_na <- !is.na(genes_info$new_names)
RNA_tpm <- RNA_tpm[non_na, ]
genes_info <- genes_info[non_na, ]## Remove entries that are withdrawn
RNA_tpm <- RNA_tpm[-which(genes_info$new_names == "entry withdrawn"), ]
genes_info <- genes_info[-which(genes_info$new_names == "entry withdrawn"), ]
## Identify duplicated new genes
```

```
newnames_dup <- unique(genes_info$new_names[duplicated(genes_info$new_names)])
newnames_dup_ind <- do.call(c, lapply(newnames_dup, function(X) which(genes_info$new_names == X)))
newnames_dup <- genes_info$new_names[newnames_dup_ind]
## Retrieve data for duplicated genes
tmp <- RNA_tpm[genes_info$old_names[genes_info$new_names %in% newnames_dup],]
## Remove data for duplicated genes
RNA_tpm <- RNA_tpm[-which(rownames(RNA_tpm) %in% rownames(tmp)),]
## Aggregate data of duplicated genes
dup_genes <- genes_info$new_names[which(genes_info$new_names %in% newnames_dup)]
names(dup_genes) <- rownames(tmp)
if (anyDuplicated(newnames_dup)){
tmp2 <- stats::aggregate(tmp, by = list(dup_genes), FUN = "mean")
rownames(tmp2) <- tmp2$Group.1
tmp2$Group.1 <- NULL
}
# Put data together
RNA_tpm <- rbind(RNA_tpm, tmp2)

# Computation of cell fractions (Finotello et al., Genome Med, 2019)
cell_fractions <- compute_cell_fractions(RNA_tpm = RNA_tpm)
```

`compute_chemokines` *Compute chemokine signature (chemokines) score*

Description

Calculates chemokines score as the PC1 score that results from applying PCA to the expression of its signature genes, defined in Messina et al., Sci. Rep., 2012.

Usage

```
compute_chemokines(matches, RNA_tpm)
```

Arguments

`matches`	numeric vector indicating the index of signature genes in `RNA_tpm`.
`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Value

A numeric matrix with samples in rows and chemokines score in a column.

References

Messina, J.L., Fenstermacher, D.A., Eschrich, S., Qu, X., Berglund, A.E., Lloyd, M.C., Schell, M.J., Sondak, V.K., Weber, J.S., and Mule, J.J. (2012). 12-Chemokine gene signature identifies lymph node-like structures in melanoma: potential for patient selection for immunotherapy? Sci. Rep. 2, 765. https://doi.org/10.1038/srep00765.

compute_CYT *Compute cytolytic activity (CYT) score*

Description

Calculates the CYT score using the geometric mean of its signature genes, as defined in Rooney et al., Cell, 2015.

Usage

```
compute_CYT(matches, RNA_tpm)
```

Arguments

matches	numeric vector indicating the index of signature genes in RNA_tpm.
RNA_tpm	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Value

A numeric matrix with samples in rows and CTY score in a column.

References

Rooney, M.S., Shukla, S.A., Wu, C.J., Getz, G., and Hacohen, N. (2015). Molecular and genetic properties of tumors associated with local immune cytolytic activity. Cell 160, 48–61. https://doi.org/10.1016/j.cell.2014.12.0

compute_Davoli_IS *Compute Davoli immune signature (Davoli_IS) score*

Description

Calculates Davoli_IS score as the average of the expression of its signature genes after applying rank normalization, as defined in Davoli et al., Science, 2017.

Usage

```
compute_Davoli_IS(matches, RNA_tpm)
```

Arguments

matches	numeric vector indicating the index of signature genes in RNA_tpm.
RNA_tpm	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Value

A numeric matrix with samples in rows and Davoli_IS score in a column.

References

Davoli, T., Uno, H., Wooten, E.C., and Elledge, S.J. (2017). Tumor aneuploidy correlates with markers of immune evasion and with reduced response to immunotherapy. Science 355. https://doi.org/10.1126/science.aaf83

`compute_IFNy` *Compute IFNy signature (IFNy) score*

Description

Calculates IFNy signature score as the average expression of its signature genes, as defined in Ayers et al., J. Clin. Invest, 2017.

Usage

```
compute_IFNy(matches, RNA_tpm)
```

Arguments

`matches`	numeric vector indicating the index of signature genes in `RNA_tpm`.
`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Value

A numeric matrix with samples in rows and IFNy score in a column.

References

Ayers, M., Lunceford, J., Nebozhyn, M., Murphy, E., Loboda, A., Kaufman, D.R., Albright, A., Cheng, J.D., Kang, S.P., Shankaran, V., et al. (2017). IFN-y-related mRN A profile predicts clinical response to PD-1 blockade. J. Clin. Invest. 127, 2930–2940. https://doi.org/10.1172/JCI91190.

`compute_IMPRES_MSI` *Compute Immuno-Predictive Score (IMPRES) and Micro Satellite Instability (MSI) status score*

Description

Calculates IMPRES score by logical comparison of checkpoint gene pairs expression, as defined in Auslander et al., Nat. Med., 2018.

Usage

```
compute_IMPRES_MSI(sig, len, match_F_1, match_F_2, RNA_tpm)
```

Arguments

`sig`	can be either 'IMPRES' or 'MSI'.
`len`	the length of gene_1 vector.
`match_F_1`	numeric vector indicating the index of signature genes defined in 'gene_1' in `RNA_tpm`.
`match_F_2`	numeric vector indicating the index of signature genes defined in 'gene_2' in `RNA_tpm`.
`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Details

Calculates MSI status score by logical comparison of MSI-related gene pairs, as defined in Fu et al., BMC Genomics, 2019.

Value

A numeric matrix with samples in rows and IMPRES score in a column.

References

Auslander,N., Zhang,G., Lee,J.S., Frederick, D.T., Miao, B., Moll,T.,Tian, T., Wei,Z., Madan, S., Sullivan, R.J., et al. (2018). Robust prediction of response to immune checkpoint blockade therapy in metastatic melanoma. Nat. Med. 24, 1545–1549. https://doi.org/10.1038/s41591-018-0157-9.

Fu, Y., Qi, L., Guo, W., Jin, L., Song, K., You, T., Zhang, S., Gu, Y., Zhao, W., and Guo, Z. (2019). A qualitative transcriptional signature for predicting microsatellite instability status of right-sided Colon Cancer. BMC Genomics 20, 769.

compute_LR_pairs *Compute ligand-receptor pair weights from TPM bulk gene expression*

Description

Quantifies ligand-receptor interactions in the tumor microenvironment from TPM bulk gene expression (Lapuente-Santana et al., Patterns, 2021) by using prior knowledge coming from ligand-receptor pair annotations from the database of Ramilowski (Ramilowski et al., Nat Commun, 2015). Each ligand-receptor weight is defined as the minimum of the log2(TPM+1) expression of the ligand and the receptor.

Usage

```
compute_LR_pairs(RNA_tpm = NULL, cancer_type = "pancan", verbose = TRUE)
```

Arguments

RNA_tpm	A data.frame containing TPM values with HGNC symbols in rows and samples in columns.
cancer_type	A string detailing the cancer type whose ligand-receptor pairs network will be used. A pan-cancer network is selected by default, whose network represents the union of all ligand-receptor pairs present across the 18 cancer types studied in Lapuente-Santana et al., Patterns, 2021.
verbose	A logical value indicating whether to display messages about the number of ligand-receptor genes found in the gene expression data provided.

Value

A matrix of weights with samples in rows and ligand-receptor pairs in columns.

References

Oscar Lapuente-Santana, Maisa van Genderen, Peter A. J. Hilbers, Francesca Finotello, and Federica Eduati. 2021. Interpretable Systems Biomarkers Predict Response to Immune-Checkpoint Inhibitors. Patterns, 100293. https://doi.org/10.1016/j.patter.2021.100293.

Ramilowski, J., Goldberg, T., Harshbarger, J. et al. A draft network of ligand–receptor-mediated multicellular signalling in human. Nat Commun 6, 7866 (2015). https://doi.org/10.1038/ncomms8866

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")
```

```
dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)
lrpair_weights[1:5, 1:5]
```

`compute_pathway_activity` *Compute pathway activity from gene expression using PROGENy*

Description

Infers pathway activity from counts bulk gene expression using PROGENy method from Holland et al., BBAGRM, 2019 and Schubert et al., Nat Commun, 2018.

Usage

```
compute_pathway_activity(
  RNA_counts = NULL,
  remove_sig_genes_immune_response = TRUE,
  verbose = TRUE
)
```

Arguments

`RNA_counts`	data.frame containing raw counts values with HGNC gene symbols as row names and samples identifiers as column names.
`remove_sig_genes_immune_response`	logical value indicating whether to remove signature genes involved in the derivation of hallmarks of immune response. This list is available from easierData package through `easierData::get_cor_scores_genes()`.
`verbose`	logical value indicating whether to display messages about the number of pathway signature genes found in the gene expression data provided.

Value

A matrix of activity scores with samples in rows and pathways in columns.

References

Schubert M, Klinger B, Klunemann M, Sieber A, Uhlitz F, Sauer S, Garnett MJ, Bluthgen N, Saez-Rodriguez J. "Perturbation-response genes reveal signaling footprints in cancer gene expression." Nature Communications: 10.1038/s41467-017-02391-6

Holland CH, Szalai B, Saez-Rodriguez J. "Transfer of regulatory knowledge from human to mouse for functional genomics analysis." Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms. 2019. DOI: 10.1016/j.bbagrm.2019.194431.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_counts <- assays(dataset_mariathasan)[["counts"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_counts <- RNA_counts[, colnames(RNA_counts) %in% pat_subset]

# Computation of pathway activity
# (Holland et al., BBAGRM, 2019; Schubert et al., Nat Commun, 2018)
pathway_activity <- compute_pathway_activity(
  RNA_counts = RNA_counts,
  remove_sig_genes_immune_response = TRUE
)
```

---

compute_RIR *Compute three scores from the immune resistance program: resF_down (RIR), resF_up, resF (resF_up - resF_down)*

Description

Calculates RIR score by combining a set of gene signatures associated with upregulation and downregulation of T cell exclusion, post-treatment and functional resistance. We used the original approach defined in Jerby-Arnon et al., Cell, 2018.

Usage

```
compute_RIR(RNA_tpm, RIR_program)
```

Arguments

RNA_tpm	data.frame containing TPM values with HGNC symbols in rows and samples in columns.
RIR_program	list with gene signatures included in the immune resistance program from Jerby-Arnon et al., 2018.

Details

The gene signatures were provided by original work: https://github.com/livnatje/ImmuneResistance

Value

A numeric matrix with samples in rows and three RIR scores as columns: "resF_up" (upregulated score), "resF_down" (downregulated score) and "resF" (upregulated score - downregulated score).

References

Jerby-Arnon, L., Shah, P., Cuoco, M.S., Rodman, C., Su, M.-J., Melms, J.C., Leeson, R., Kanodia, A., Mei, S., Lin, J.-R., et al. (2018). A Cancer Cell Program Promotes T Cell Exclusion and Resistance to Checkpoint Blockade. Cell 175, 984–997.e24. https://doi.org/10.1016/j.cell.2018.09.006.

---

compute_Roh_IS *Compute Roh immune score (Roh_IS)*

Description

Calculates Roh_IS score as the geometric-mean of its signature genes, defined in Roh et al., Sci. Transl. Med., 2017.

Usage

```
compute_Roh_IS(matches, RNA_tpm)
```

Arguments

matches	numeric vector indicating the index of signature genes in RNA_tpm.
RNA_tpm	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Value

A numeric matrix with samples in rows and Roh_IS score in a column.

References

Roh, W., Chen, P.-L., Reuben, A., Spencer, C.N., Prieto, P.A., Miller, J.P., Gopalakrishnan, V., Wang, F., Cooper, Z.A., Reddy, S.M., et al. (2017). Integrated molecular analysis of tumor biopsies on sequential CTLA-4 and PD-1 blockade reveals markers of response and resistance. Sci. Transl. Med. 9. https://doi.org/10.1126/scitranslmed.aah3560.

`compute_scores_immune_response`

Compute published scores of immune response

Description

Calculates the transcriptomics-based scores of hallmarks of anti-cancer immune response.

Usage

```
compute_scores_immune_response(
  RNA_tpm = NULL,
 selected_scores = c("CYT", "Roh_IS", "chemokines", "Davoli_IS", "IFNy", "Ayers_expIS",
    "Tcell_inflamed", "RIR", "TLS"),
  verbose = TRUE
)
```

Arguments

`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.
`selected_scores`	character string with names of scores of immune response to be computed. Default scores are computed for: "CYT", "Roh_IS", "chemokines", "Davoli_IS", "IFNy", "Ayers_expIS", "Tcell_inflamed", "RIR", "TLS".
`verbose`	logical variable indicating whether to display informative messages.

Value

A numeric matrix with samples in rows and published scores (gold standards) in columns.

References

Rooney, Michael S., Sachet A. Shukla, Catherine J. Wu, Gad Getz, and Nir Hacohen. 2015. "Molecular and Genetic Properties of Tumors Associated with Local Immune Cytolytic Activity." Cell 160 (1): 48–61. https://doi.org/10.1016/j.cell.2014.12.033.

Cabrita, Rita, Martin Lauss, Adriana Sanna, Marco Donia, Mathilde Skaarup Larsen, Shamik Mitra, Iva Johansson, et al. 2020. "Tertiary Lymphoid Structures Improve Immunotherapy and Survival in Melanoma." Nature 577 (7791):561–65. https://doi.org/10.1038/s41586-019-1914-8.

McClanahan, Mark Ayers AND Jared Lunceford AND Michael Nebozhyn AND Erin Murphy AND Andrey Loboda AND David R. Kaufman AND Andrew Albright AND Jonathan D. Cheng AND S. Peter Kang AND Veena Shankaran AND Sarina A. Piha-Paul AND Jennifer Yearley AND Tanguy Y. Seiwert AND Antoni Ribas AND Terrill K. 2017. "IFN-y–Related mRNA Profile Predicts Clinical Response to PD-1 Blockade." The Journal of Clinical Investigation 127 (8): 2930–40. https://doi.org/10.1172/JCI91190.

Roh, Whijae, Pei-Ling Chen, Alexandre Reuben, Christine N. Spencer, Peter A. Prieto, John P. Miller, Vancheswaran Gopalakrishnan, et al. 2017. "Integrated Molecular Analysis of Tumor Biopsies on Sequential CTLA-4 and PD-1 Blockade Reveals Markers of Response and Resistance." Science Translational Medicine 9 (379). https://doi.org/10.1126/scitranslmed.aah3560.

Davoli, Teresa, Hajime Uno, Eric C. Wooten, and Stephen J. Elledge. 2017. "Tumor Aneuploidy Correlates with Markers of Immune Evasion and with Reduced Response to Immunotherapy." Science 355 (6322). https://doi.org/10.1126/science.aaf8399.

Messina, Jane L., David A. Fenstermacher, Steven Eschrich, Xiaotao Qu, Anders E. Berglund, Mark C. Lloyd, Michael J. Schell, Vernon K. Sondak, Jeffrey S. Weber, and James J. Mule. 2012. "12-Chemokine Gene Signature Identifies Lymph Node-Like Structures in Melanoma: Potential for Patient Selection for Immunotherapy?" Scientific Reports 2 (1): 765. https://doi.org/10.1038/srep00765.

Auslander, Noam, Gao Zhang, Joo Sang Lee, Dennie T. Frederick, Benchun Miao, Tabea Moll, Tian Tian, et al. 2018. "Robust Prediction of Response to Immune Checkpoint Blockade Therapy in Metastatic Melanoma." Nature Medicine 24(10): 1545–49. https://doi.org/10.1038/s41591-018-0157-9.

Fu, Yelin, Lishuang Qi, Wenbing Guo, Liangliang Jin, Kai Song, Tianyi You, Shuobo Zhang, Yunyan Gu, Wenyuan Zha, and Zheng Guo. 2019. "A Qualitative Transcriptional Signature for Predicting Microsatellite Instability Status of Right-Sided Colon Cancer." BMC Genomics 20 (1): 769. https://doi.org/10.1186/s12864-019-6129-8.

Jerby-Arnon, Livnat, Parin Shah, Michael S. Cuoco, Christopher Rodman, Mei-Ju Su, Johannes C. Melms, Rachel Leeso, et al. 2018. "A Cancer Cell Program Promotes t Cell Exclusion and Resistance to Checkpoint Blockade." Cell 175 (4): 984–997.e24. https://doi.org/10.1016/j.cell.2018.09.006.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
cancer_type <- metadata(dataset_mariathasan)$cancertype

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]
```

```
# Computation of different hallmarks of anti-cancer immune responses
hallmarks_of_immune_response <- c(
  "CYT", "Roh_IS", "chemokines", "Davoli_IS", "IFNy"
)

scores_immune_response <- compute_scores_immune_response(
  RNA_tpm = RNA_tpm,
  selected_scores = hallmarks_of_immune_response
)
```

`compute_Tcell_inflamed` *Compute T cell-inflamed signature (Tcell_inflamed) score*

Description

Calculates Tcell_inflamed score using a weighted sum of housekeeping normalized expression of its signature genes, as defined in Cristescu et al., Science, 2018.

Usage

```
compute_Tcell_inflamed(housekeeping, predictors, weights, RNA_tpm)
```

Arguments

`housekeeping`	numeric vector indicating the index of houskeeping genes in `RNA_tpm`.
`predictors`	numeric vector indicating the index of predictor genes in `RNA_tpm`.
`weights`	numeric vector containing the weights.
`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Details

Weights were available at Table S2B from Cristescu R, et al. Pan-tumor genomic biomarkers for PD-1 checkpoint blockade-based immunotherapy. Science. (2018) 362:eaar3593. doi: 10.1126/science.aar3593.

Value

A numeric matrix with samples in rows and Tcell_inflamed score in a column.

References

Ayers, M., Lunceford, J., Nebozhyn, M., Murphy, E., Loboda, A., Kaufman, D.R., Albright, A., Cheng, J.D., Kang, S.P., Shankaran, V., et al. (2017). IFN-y-related mRNA profile predicts clinical response to PD-1 blockade. J. Clin. Invest. 127, 2930–2940. https://doi.org/10.1172/JCI91190.

`compute_TF_activity` *Compute transcription factor activity from gene expression using DoRothEA*

Description

Infers transcription factor (TF) activity from TPM bulk gene expression using DoRothEA method from Garcia-Alonso et al., Genome Res, 2019.

Usage

```
compute_TF_activity(RNA_tpm = NULL, verbose = TRUE)
```

Arguments

`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.
`verbose`	logical value indicating whether to display messages about the number of regulated genes found in the gene expression data provided.

Value

A numeric matrix of activity scores with samples in rows and TFs in columns.

References

Garcia-Alonso L, Holland CH, Ibrahim MM, Turei D, Saez-Rodriguez J. "Benchmark and integration of resources for the estimation of human transcription factor activities." Genome Research. 2019. DOI: 10.1101/gr.240663.118.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]
```

```
# Computation of TF activity (Garcia-Alonso et al., Genome Res, 2019)
tf_activity <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)
```

`compute_TLS` *Computation of tertiary lymphoid structures signature (TLS) score*

Description

Calculates TLS score using the geometric-mean of the expression of its signature genes, as defined in Cabrita et al., Nature, 2020.

Usage

```
compute_TLS(matches, RNA_tpm)
```

Arguments

`matches`	numeric vector indicating the index of signature genes in `RNA_tpm`.
`RNA_tpm`	data.frame containing TPM values with HGNC symbols in rows and samples in columns.

Value

A numeric matrix with samples in rows and TLS score in a column.

References

Cabrita, R., Lauss, M., Sanna, A., Donia, M., Skaarup Larsen, M., Mitra, S., Johansson, I., Phung, B., Harbst, K., Vallon-Christersson, J., et al. (2020). Tertiary lymphoid structures improve immunotherapy and survival in melanoma. Nature 577, 561–565.

`discretize` *Converts a continuous variable into categorical*

Description

It is used to bin continuous gene expression values from a given gene signature into categories.

Usage

```
discretize(v, n_cat)
```

Arguments

v	numeric vector with gene mean expression across samples.
n_cat	number of categories to bin continuous values, here gene expression values.

Details

The source code was provided by original work: https://github.com/livnatje/ImmuneResistance

Value

A numeric vector providing an integer value (e.g. category) for each gene.

References

Jerby-Arnon, L., Shah, P., Cuoco, M.S., Rodman, C., Su, M.-J., Melms, J.C., Leeson, R., Kanodia, A., Mei, S., Lin, J.-R., et al. (2018). A Cancer Cell Program Promotes T Cell Exclusion and Resistance to Checkpoint Blockade. Cell 175, 984–997.e24. https://doi.org/10.1016/j.cell.2018.09.006.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Log2 transformation:
log2_RNA_tpm <- log2(RNA_tpm + 1)

# Prepare input data
r <- list()
r$tpm <- log2_RNA_tpm

# Gene signature of immune resistance program
score_signature_genes <- suppressMessages(easierData::get_scores_signature_genes())
RIR_gene_signature <- score_signature_genes$RIR

# Compute gene average expression across samples
r$genes_dist <- r$genes_mean <- rowMeans(r$tpm)
```

```
# Bin genes into 50 expression bins according to their average
r$genes_dist_q <- discretize(r$genes_dist, n_cat = 50)
```

`easier-pkg` *easier: predicting immune response by using quantitative descriptors of the tumor microenvironment extracted from RNA-seq data.*

Description

This package streamlines the assessment of patients' likelihood of immune response using EaSIeR approach.

References

Lapuente-Santana, Oscar, Maisa van Genderen, Peter A. J. Hilbers, Francesca Finotello, and Federica Eduati. 2021. "Interpretable Systems Biomarkers Predict Response to Immune-Checkpoint Inhibitors." Patterns, 100293. https://doi.org/10.1016/j.patter.2021.100293.

`explore_biomarkers` *Explore biomarkers of immune response*

Description

Provides a good overview of the computed features (biomarkers) including the corresponding weights from the trained model. If `patient_response` is provided, this function shows statistically significant biomarkers between responders (R) and non-responders (NR) patients.

Usage

```
explore_biomarkers(
  pathways = NULL,
  immunecells = NULL,
  tfs = NULL,
  lrpairs = NULL,
  ccpairs = NULL,
  cancer_type,
  patient_label = NULL,
  verbose = TRUE
)
```

Arguments

pathways	numeric matrix with pathways activity (rows = samples; columns = pathways). This is the output from compute_pathway_activity.
immunecells	numeric matrix with immune cell quantification (rows = samples; columns = cell types). This is the output from compute_cell_fractions.
tfs	numeric matrix with transcription factors activity (rows = samples; columns = transcription factors). This is the output from compute_TF_activity.
lrpairs	numeric matrix with ligand-receptor weights (rows = samples; columns = ligand-receptor pairs). This is the output from compute_LR_pairs.
ccpairs	numeric matrix with cell-cell scores (rows = samples; columns = cell-cell pairs). This is the output from compute_CC_pairs.
cancer_type	character string indicating which cancer-specific model should be used to compute the predictions. This should be available from the cancer-specific models. The following cancer types have a corresponding model available: "BLCA", "BRCA", "CESC", "CRC", "GBM", "HNSC", "KIRC", "KIRP", "LIHC", "LUAD", "LUSC", "NSCLC", "OV", "PAAD", "PRAD", "SKCM", "STAD", "THCA" and "UCEC".
patient_label	character vector with two factor levels, e.g. NR (Non-responders) vs R (Responders), pre- vs on- treatment.
verbose	logical flag indicating whether to display messages about the process.

Value

- A combined plot for each type of quantitative descriptors, showing the original distribution of the features and the importance of these features for the trained models #'
- Volcano plot displaying relevant biomarkers differentiating responders vs non-responders patients.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
cancer_type <- metadata(dataset_mariathasan)[["cancertype"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]
```

```
# Computation of TF activity
tf_activity <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)

# retrieve clinical response
patient_ICBresponse <- colData(dataset_mariathasan)[["BOR"]]
names(patient_ICBresponse) <- colData(dataset_mariathasan)[["pat_id"]]
patient_ICBresponse <- patient_ICBresponse[names(patient_ICBresponse) %in% pat_subset]

# Investigate possible biomarkers
output_biomarkers <- explore_biomarkers(
  tfs = tf_activity,
  cancer_type = cancer_type,
  patient_label = patient_ICBresponse
)



RNA_counts <- assays(dataset_mariathasan)[["counts"]]
RNA_counts <- RNA_counts[, colnames(RNA_counts) %in% pat_subset]

# Computation of cell fractions
cell_fractions <- compute_cell_fractions(RNA_tpm = RNA_tpm)

# Computation of pathway scores
pathway_activity <- compute_pathway_activity(
  RNA_counts = RNA_counts,
  remove_sig_genes_immune_response = TRUE
)

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)

# Computation of cell-cell interaction scores
ccpair_scores <- compute_CC_pairs(
  lrpairs = lrpair_weights,
  cancer_type = "pancan"
)

# Investigate possible biomarkers
output_biomarkers <- explore_biomarkers(
  pathways = pathway_activity,
  immunecells = cell_fractions,
  lrpairs = lrpair_weights,
  tfs = tf_activity,
  ccpairs = ccpair_scores,
  cancer_type = cancer_type,
  patient_label = patient_ICBresponse
```

```
)
```

`get_OE_bulk` *Compute overall expression (OE) of the immune resistance program used in the computation of repressed immune resistance signature (RIR) score.*

Description

This function calculates the overall expression of the immune resistance program which is based on a set of gene signatures associated with T cell exclusion, post-treatment and functional resistance.

Usage

```
get_OE_bulk(
  r,
  gene_sign = NULL,
  num_rounds = 1000,
  full_flag = FALSE,
  verbose = TRUE
)
```

Arguments

`r`	list containing a numeric matrix with bulk RNA-Seq data (tpm values) and a character string with the available gene names.
`gene_sign`	list containing different character strings associated with subsets of the resistance program.
`num_rounds`	integer value related to the number of random gene signatures samples to be computed for normalization. Original work indicates that 1000 random signatures were sufficient to yield an estimate of the expected value.
`full_flag`	logical flag indicating whether to return also random scores.
`verbose`	logical flag indicating whether to display messages about the process.

Details

The source code was provided by original work: https://github.com/livnatje/ImmuneResistance

Value

A numeric matrix with computed scores for each sample and subset of signatures included in the immune resistance program (rows = samples; columns = gene signatures)

References

Jerby-Arnon, L., Shah, P., Cuoco, M.S., Rodman, C., Su, M.-J., Melms, J.C., Leeson, R., Kanodia, A., Mei, S., Lin, J.-R., et al. (2018). A Cancer Cell Program Promotes T Cell Exclusion and Resistance to Checkpoint Blockade. Cell 175, 984–997.e24. https://doi.org/10.1016/j.cell.2018.09.006.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Log2 transformation:
log2_RNA_tpm <- log2(RNA_tpm + 1)

# Prepare input data
r <- list()
r$tpm <- log2_RNA_tpm
r$genes <- rownames(log2_RNA_tpm)

# Gene signature of immune resistance program
score_signature_genes <- suppressMessages(easierData::get_scores_signature_genes())
RIR_gene_signature <- score_signature_genes$RIR

# Apply function to calculate OE:
res_scores <- get_OE_bulk(r, gene_sign = RIR_gene_signature, verbose = TRUE)
```

`get_semi_random_OE` *Compute random scores of the immune resistance program used in the computation of repressed immune resistance signature (RIR) score.*

Description

Calculates random scores to yield a robust estimate of the immune resistance program values. This is used by get_OE_bulk function.

Usage

```
get_semi_random_OE(
  r,
  genes_dist_q,
  b_sign,
  num_rounds = 1000,
  full_flag = FALSE,
  random_seed = 1234
)
```

Arguments

r	list containing a numeric matrix with bulk RNA-Seq data (tpm values) and a character string with the available gene names.
genes_dist_q	factor variable obtained as output from the function discretize. Original work binned genes into 50 expression bins according their average gene expression across samples.
b_sign	logical vector representing whether signature genes were found in bulk tpm matrix.
num_rounds	integer value related to the number of random gene signatures samples to be computed for normalization. Original work indicates that 1000 random signatures were sufficient to yield an estimate of the expected value.
full_flag	logical flag indicating whether to return also random scores.
random_seed	integer value to set a seed for the selection of random genes used to generate a random score.

Details

The source code was provided by original work: https://github.com/livnatje/ImmuneResistance

Value

A numeric vector containing the estimated random score for each sample.

References

Jerby-Arnon, L., Shah, P., Cuoco, M.S., Rodman, C., Su, M.-J., Melms, J.C., Leeson, R., Kanodia, A., Mei, S., Lin, J.-R., et al. (2018). A Cancer Cell Program Promotes T Cell Exclusion and Resistance to Checkpoint Blockade. Cell 175, 984–997.e24. https://doi.org/10.1016/j.cell.2018.09.006.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")
```

```
dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Log2 transformation:
log2_RNA_tpm <- log2(RNA_tpm + 1)

# Prepare input data
r <- list()
r$tpm <- log2_RNA_tpm
r$genes <- rownames(log2_RNA_tpm)

# Gene signature of immune resistance program
score_signature_genes <- suppressMessages(easierData::get_scores_signature_genes())
RIR_gene_signature <- score_signature_genes$RIR

# Compute gene average expression across samples
r$genes_dist <- r$genes_mean <- rowMeans(r$tpm)

# Center gene expression matrix
r$zscores <- sweep(r$tpm, 1, r$genes_mean, FUN = "-")

# Bin genes into 50 expression bins according to their average
r$genes_dist_q <- discretize(r$genes_dist, n.cat = 50)

# Match genes from exc.down signature with genes from expression matrix
b_sign <- is.element(r$genes, RIR_gene_signature[["exc.down"]])

# Compute random score:
rand_scores <- get_semi_random_OE(r, r$genes_dist_q, b_sign)
```

`predict_immune_response`

Compute predicted immune response

Description

Calculates predictions of patients' immune response using the quantitative descriptors data as input features and the optimized model parameters derived from the trained models. These models are available from easierData package through `easierData::get_opt_models()`.

Usage

```
predict_immune_response(
  pathways = NULL,
  immunecells = NULL,
  tfs = NULL,
  lrpairs = NULL,
  ccpairs = NULL,
  cancer_type,
  verbose = TRUE
)
```

Arguments

pathways	numeric matrix with pathways activity (rows = samples; columns = pathways).
immunecells	numeric matrix with immune cell quantification (rows = samples; columns = cell types).
tfs	numeric matrix with transcription factors activity (rows = samples; columns = transcription factors).
lrpairs	numeric matrix with ligand-receptor weights (rows = samples; columns = ligand-receptor pairs).
ccpairs	numeric matrix with cell-cell scores (rows = samples; columns = cell-cell pairs).
cancer_type	character string indicating which cancer-specific model should be used to compute the predictions. This should be available from the cancer-specific models. The following cancer types have a corresponding model available: "BLCA", "BRCA", "CESC", "CRC", "GBM", "HNSC", "KIRC", "KIRP", "LIHC", "LUAD", "LUSC", "NSCLC", "OV", "PAAD", "PRAD", "SKCM", "STAD", "THCA" and "UCEC".
verbose	logical flag indicating whether to display messages about the process.

Value

A list containing the predictions for each quantitative descriptor and for each task. Given that the model training was repeated 100 times with randomized-cross validation, a set of 100 predictions is returned.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
cancer_type <- metadata(dataset_mariathasan)[["cancertype"]]
```

```
# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of TF activity (Garcia-Alonso et al., Genome Res, 2019)
tf_activity <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)

# Predict patients' immune response
predictions_immune_response <- predict_immune_response(
  tfs = tf_activity,
  cancer_type = cancer_type
)



RNA_counts <- assays(dataset_mariathasan)[["counts"]]
RNA_counts <- RNA_counts[, colnames(RNA_counts) %in% pat_subset]

# Computation of cell fractions (Finotello et al., Genome Med, 2019)
cell_fractions <- compute_cell_fractions(RNA_tpm = RNA_tpm)

# Computation of pathway scores (Holland et al., BBAGRM, 2019;
# Schubert et al., Nat Commun, 2018)
pathway_activity <- compute_pathway_activity(
  RNA_counts = RNA_counts,
  remove_sig_genes_immune_response = TRUE
)

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)

# Computation of cell-cell interaction scores
ccpair_scores <- compute_CC_pairs(
  lrpairs = lrpair_weights,
  cancer_type = "pancan"
)

# Predict patients' immune response
predictions_immune_response <- predict_immune_response(
  pathways = pathway_activity,
  immunecells = cell_fractions,
  tfs = tf_activity,
  lrpairs = lrpair_weights,
  ccpairs = ccpair_scores,
  cancer_type = cancer_type
```

```
)
```

`predict_with_rmtlr` *Predict single-view immune response*

Description

Obtains predictions of immune response for individual quantitative descriptors by using a cancer-specific model learned with Regularized Multi-Task Linear Regression algorithm (RMTLR).

Usage

```
predict_with_rmtlr(
  view_name,
  view_info,
  view_data,
  opt_model_cancer_view_spec,
  opt_xtrain_stats_cancer_view_spec,
  verbose = TRUE
)
```

Arguments

`view_name`	character string containing the name of the input view.
`view_info`	character string informing about the family of the input data.
`view_data`	list containing the data for each input view.
`opt_model_cancer_view_spec`	cancer-view-specific model feature parameters learned during training. These are available from easierData package through `easierData::get_opt_models()`.
`opt_xtrain_stats_cancer_view_spec`	cancer-view-specific features mean and standard deviation of the training set. These are available from easierData package through `easierData::get_opt_xtrain_stats()`.
`verbose`	logical flag indicating whether to display messages about the process.

Value

A list of predictions, one for each task, in a matrix format (rows = samples; columns = [runs).

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")
```

```
dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
cancer_type <- metadata(dataset_mariathasan)[["cancertype"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of TF activity (Garcia-Alonso et al., Genome Res, 2019)
tf_activities <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)

view_name <- "tfs"
view_info <- c(tfs = "gaussian")
view_data <- list(tfs = as.data.frame(tf_activities))

# Retrieve internal data
opt_models <- suppressMessages(easierData::get_opt_models())
opt_xtrain_stats <- suppressMessages(easierData::get_opt_xtrain_stats())

opt_model_cancer_view_spec <- lapply(view_name, function(X) {
  return(opt_models[[cancer_type]][[X]])
})
names(opt_model_cancer_view_spec) <- view_name
opt_xtrain_stats_cancer_view_spec <- lapply(view_name, function(X) {
  return(opt_xtrain_stats[[cancer_type]][[X]])
})
names(opt_xtrain_stats_cancer_view_spec) <- view_name

# Predict using rmtlr
prediction_view <- predict_with_rmtlr(
  view_name = view_name,
  view_info = view_info,
  view_data = view_data,
  opt_model_cancer_view_spec = opt_model_cancer_view_spec,
  opt_xtrain_stats_cancer_view_spec = opt_xtrain_stats_cancer_view_spec
)
```

---

reannotate_genes *Gene re-annotation using HGNC symbols*

Description

Performs gene re-annotation using curated data from the HGNC.

Usage

```
reannotate_genes(cur_genes)
```

Arguments

cur_genes	character string containing gene HGNC symbols to be consider for re-annotation.

Details

Source code adapted from quanTIseq helper function mapGenes from quantiseqr package.

Value

A data.frame with the old gene HGNC symbol and the new corresponding gene HGNC symbol.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Select some genes to check possible updated gene names
genes_to_check <- rownames(RNA_tpm)[400:450]
genes_info <- reannotate_genes(cur_genes = genes_to_check)
```

---

retrieve_easier_score *Retrieve easier scores of immune response*

Description

Calculates easier score and if applicable, both weighted average and penalized score based on the combination of easier score and TMB.

Usage

```
retrieve_easier_score(
  predictions_immune_response = NULL,
  TMB_values = NULL,
  easier_with_TMB = c("weighted_average", "penalized_score"),
  weight_penalty,
  verbose = TRUE
)
```

Arguments

predictions_immune_response	list containing the predictions for each quantitative descriptor and for each task. This is the output from predict_immune_response.
TMB_values	numeric vector containing patients' tumor mutational burden (TMB) values.
easier_with_TMB	character string indicating which approach should be used to integrate easier with TMB: "weighted_average" (default) and "penalized_score".
weight_penalty	integer value from 0 to 1, which is used to define the weight or penalty for combining easier and TMB scores based on a weighted average or penalized score, in order to derive a score of patient's likelihood of immune response. The default value is 0.5.
verbose	logical flag indicating whether to display messages about the process.

Value

A data.frame with samples in rows and easier scores in columns.

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]
cancer_type <- metadata(dataset_mariathasan)[["cancertype"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of TF activity (Garcia-Alonso et al., Genome Res, 2019)
```

```
tf_activities <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)

# Predict patients' immune response
predictions <- predict_immune_response(
  tfs = tf_activities,
  cancer_type = cancer_type,
  verbose = TRUE
)

# retrieve clinical response
patient_ICBresponse <- colData(dataset_mariathasan)[["BOR"]]
names(patient_ICBresponse) <- colData(dataset_mariathasan)[["pat_id"]]

# retrieve TMB
TMB <- colData(dataset_mariathasan)[["TMB"]]
names(TMB) <- colData(dataset_mariathasan)[["pat_id"]]

patient_ICBresponse <- patient_ICBresponse[names(patient_ICBresponse) %in% pat_subset]
TMB <- TMB[names(TMB) %in% pat_subset]

easier_derived_scores <- retrieve_easier_score(
  predictions_immune_response = predictions,
  TMB_values = TMB,
  easier_with_TMB = c("weighted_average", "penalized_score"),
  weight_penalty = 0.5
)


RNA_counts <- assays(dataset_mariathasan)[["counts"]]
RNA_counts <- RNA_counts[, colnames(RNA_counts) %in% pat_subset]

# Computation of cell fractions  (Finotello et al., Genome Med, 2019)
cell_fractions <- compute_cell_fractions(RNA_tpm = RNA_tpm)

# Computation of pathway scores (Holland et al., BBAGRM, 2019;
# Schubert et al., Nat Commun, 2018)
pathway_activities <- compute_pathway_activity(
  RNA_counts = RNA_counts,
  remove_sig_genes_immune_response = TRUE
)

# Computation of ligand-receptor pair weights
lrpair_weights <- compute_LR_pairs(
  RNA_tpm = RNA_tpm,
  cancer_type = "pancan"
)

# Computation of cell-cell interaction scores
ccpair_scores <- compute_CC_pairs(
  lrpairs = lrpair_weights,
  cancer_type = "pancan"
```

```
)

# Predict patients' immune response
predictions <- predict_immune_response(
  pathways = pathway_activities,
  immunecells = cell_fractions,
  tfs = tf_activities,
  lrpairs = lrpair_weights,
  ccpairs = ccpair_scores,
  cancer_type = cancer_type,
  verbose = TRUE
)

easier_derived_scores <- retrieve_easier_score(
  predictions_immune_response = predictions,
  TMB_values = TMB,
  easier_with_TMB = c("weighted_average", "penalized_score"),
  weight_penalty = 0.5
)
```

`rmtlr_test` *Regularized Multi-Task Linear Regression (RMTLR) model predictions*

Description

Computes the predictions as a matrix multiplication using both the features input data and the features estimated weights.

Usage

```
rmtlr_test(x_test, coef_matrix)
```

Arguments

`x_test`	numeric matrix containing features values (rows = samples; columns = features).
`coef_matrix`	numeric matrix containing the parameters values derived from model training (rows = features; columns = tasks).

Value

Numeric matrix of predicted values (rows = samples; columns = tasks).

Examples

```
# using a SummarizedExperiment object
library(SummarizedExperiment)
# Using example exemplary dataset (Mariathasan et al., Nature, 2018)
# from easierData. Original processed data is available from
# IMvigor210CoreBiologies package.
library("easierData")

dataset_mariathasan <- easierData::get_Mariathasan2018_PDL1_treatment()
RNA_tpm <- assays(dataset_mariathasan)[["tpm"]]

# Select a subset of patients to reduce vignette building time.
pat_subset <- c(
  "SAM76a431ba6ce1", "SAMd3bd67996035", "SAMd3601288319e",
  "SAMba1a34b5a060", "SAM18a4dabbc557"
)
RNA_tpm <- RNA_tpm[, colnames(RNA_tpm) %in% pat_subset]

# Computation of TF activity (Garcia-Alonso et al., Genome Res, 2019)
tf_activities <- compute_TF_activity(
  RNA_tpm = RNA_tpm
)

# Parameters values should be defined as a matrix
# with features as rows and tasks as columns
estimated_parameters <- matrix(rnorm(n = (ncol(tf_activities) + 1) * 10),
  nrow = ncol(tf_activities) + 1, ncol = 10
)
rownames(estimated_parameters) <- c("(Intercept)", colnames(tf_activities))
colnames(estimated_parameters) <- c(
  "CYT", "Ock_IS", "Roh_IS", "chemokines",
  "Davoli_IS", "IFNy", "Ayers_expIS", "Tcell_inflamed", "RIR", "TLS"
)

# Compute predictions using parameters values
pred_test <- rmtlr_test(
  x_test = tf_activities,
  coef_matrix = estimated_parameters
)
```

Index

assess_immune_response, 3

calc_z_score, 6
categorize_TMB, 7
compute_Ayers_expIS, 8
compute_CC_pairs, 10
compute_CCpair_score, 8
compute_cell_fractions, 11
compute_chemokines, 13
compute_CYT, 14
compute_Davoli_IS, 14
compute_IFNy, 15
compute_IMPRES_MSI, 16
compute_LR_pairs, 17
compute_pathway_activity, 18
compute_RIR, 19
compute_Roh_IS, 20
compute_scores_immune_response, 21
compute_Tcell_inflamed, 23
compute_TF_activity, 24
compute_TLS, 25

discretize, 25

easier-pkg, 27
explore_biomarkers, 27

get_OE_bulk, 30
get_semi_random_OE, 31

predict_immune_response, 33
predict_with_rmtlr, 36

reannotate_genes, 37
retrieve_easier_score, 38
rmtlr_test, 41