Iterative network guided cMapping and validation

Supplementary Material and Methods - Supplementary Code: GDSC_BASAL_EXP_PREPROCESSING

This document describes functions, scripts and data objects used in the software enclosed to the paper entitled *A semi-supervised approach for refining transcriptional signatures of drug response and repositioning predictions*, by Francesco Iorio et al, submitted as research paper to PLoS ONE.

April 28, 2014

`EL_statistic`	*Basal expression level statistics generation for a single gene*

Description

This function normalises the pre-processed expression signal of given gene across multiple samples, by estimating the density function of its expression first, then computing expression level scores as described in the supplementary methods of our manuscript.

Usage

```
EL_statistic(expression_pattern, ret.pvals = FALSE)
```

Arguments

`expression_pattern`
A numerical vector containing the pre-processed basal expression profiles of the a gene across m samples

`ret.pvals` A bolean parameter specifying whether cumulative probabilities should be returned for all the samples

Value

A numerical vector containing the basal expression level statistics for the input gene or a list containing this vector and the vector of cumulative probabilities for all the samples

Author(s)

Francesco Iorio (iorio@ebi.ac.uk)
Copyright (c) 2014 - 2019, EMBL - European Bioinformatics Institute
Distributed under the GPLv3 License
See accompanying file LICENSE.txt or copy at http://www.gnu.org/licenses/pl-3.0.html
Paper website: http://www.ebi.ac.uk/~iorio/PLoS_ONE_Submission

Examples

```
## loading the pre-processd basal expression dataset of the GDSC [1] cell lines
load('DATA/GDSC_basalEXP.ro')

## visualising histograms of the expression level of "CYFIP2" across all the cell lines
hist(basalEXP["CYFIP2",],main="CYFIP2")

## computing expression level statistics for the selected 20 genes across all the cell lines
elevels<-EL_statistic(basalEXP["CYFIP2",])

hist(elevels,main="CYFIP2")
```

EL_statistics *Basal expression level statistics generation for multiple genes*

Description

This function normalises the pre-processed expression signal of multiple genes across multiple samples, by estimating the density function of their expression first, then computing expression level scores as described in the supplementary methods of our manuscript.

Usage

```
EL_statistics(expression_data, show_progress = TRUE)
```

Arguments

`expression_data`
An n x m double matrix, containing the pre-processed basal expression profiles of the n genes across m samples with row names corresponding to gene symbols and column names correspond to sample identifiers

`show_progress` A boolean parameter specifying if a progress bar should be visualised (default = TRUE)

Value

An n x m double matrix, containing the expression level statistics of the n genes across m samples in the input matrix, with row names corresponding to gene symbols and column names correspond to sample identifiers.

Author(s)

Francesco Iorio (iorio@ebi.ac.uk)

Paper website: http://www.ebi.ac.uk/~iorio/PLoS_ONE_Submission

References

[1] Garnett,M.J. et al. (2012) Systematic identification of genomic markers of drug sensitivity in cancer cells. Nature, 483, 570-575.

Examples

```
## loading the pre-processd basal expression dataset of the GDSC [1] cell lines
load('DATA/GDSC_basalEXP.ro')

## visualising histograms of the expression level of 15 genes across all the cell lines
par(mfrow=c(3,5))
for (i in 1:15){
  hist(basalEXP[i,],main=rownames(basalEXP)[i])
}

## computing expression level statistics for the selected 20 genes across all the cell lines
elevels<-EL_statistics(basalEXP[1:15,])

## visualising histograms of the expression level statistics of the selected 15 genes across all the cell l
par(mfrow=c(3,5))
for (i in 1:15){
  hist(elevels[i,],main=rownames(elevels)[i])
}
```

`basalRanked_lists` *Computing genome-wide ranked lists of genes from their basal expression level statistics*

Description

This Function turns the genome wide basal expression level statistics into ranked list of genes sorted according to these values

Usage

```
basalRanked_lists(medNorm_basalExp)
```

Arguments

`medNorm_basalExp`
: An n x m double matrix containing the basal expression level statistics of n genes across m samples. Rownames should contains gene symbols and column names should contain sample identifiers.

Value

An n x m dataframe containing, for each sample, the genes sorted according to their basal expression level (in decreasing order).

Author(s)

Francesco Iorio (iorio@ebi.ac.uk)

Paper website: http://www.ebi.ac.uk/~iorio/PLoS_ONE_Submission

Examples

```
## loading the pre-processd basal expression dataset of the GDSC [1] cell lines
load('DATA/GDSC_basalEXP.ro')

## computing expression level statistics for all the genes (this may take a while)
elevels<-EL_statistics(basalEXP)

## computing ranked lists from expression level statistics for all the genes
rankedLists<-basalRanked_lists(elevels)

## visualising genes in the top 10 positions across the first 5 samples
print(rankedLists[1:10,1:5])
```

Index