Package ‘LncMod’

June 27, 2015

Type Package

Title Predicting Modulator and Functional/Survival Analysis

Version 1.1

Author Yongsheng Li,Zishan Wang,Juan Xu*,Xia Li*

Maintainer Zishan Wang <azureskyeric@gmail.com>

Description Predict modulators regulating the ability of effectors to regulate their targets and produce modulator-effector-target triplets followed by goterm functional enrichment and survival analysis.This is mainly applied to long non-coding RNAs (lncRNAs) as candidate modulators regulating the ability of transcription factors (TFs) to regulate their corresponding targets.

License GPL-2

LazyData TRUE

biocViews long non-coding RNA,lncRNA,modulator,triplet,function,survival

Depends R (>= 2.15.2),survival,pheatmap,parallel

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-27 13:24:19

R topics documented:

`LncMod-package` *LncMod*

Description

Modulator prediction and functional/survival analysis

`datatests` *Data for Examples*

Description

This object contains data for examples.

Format

A list with 18 variables:

m_app four lncRNA symbols of which,the first two,as a modulator,can affect the ability of effectors to regulate their corresponding targets,the third expression is not in initial expression profile,the last expression is in initial expression profile but not in filterd expression profile by IQR.

ET a dataframe representing TF-target regulations,all of which are gene symbols,in glioblastoma.It includes 20 rows standing for 20 pairs of regulation and 2 columns (the first is TF and the second is corresponding target).

M_exp a dataframe representing lncRNAs expression profile in glioblastoma.It includes 4 rows standing for 4 lncRNAs and 451 columns standing for 451 samples.Its rownames are lncRNA symbols.

E_exp a dataframe representing TFs expression profile in glioblastoma.It includes 12 rows standing for 12 TFs and 451 columns standing for 451 samples.Its rownames are TF gene symbols.

T_exp a dataframe representing targets expression profile in glioblastoma.It includes 22 rows standing for 22 targets and 451 columns standing for 451 samples.Its rownames are target gene symbols.

tri_bs 2 lncRNA-TF-target (modulator-effector-target) symbol triplets whose factors are not only in expression profile (*M_exp,E_exp,T_exp*),respectively,but also in the expression and survival profile (*exp_sur*).

tri_enrich 30 lncRNA-TF-target symbol triplets in glioblastoma.

background an character vector representing all genes expressed in glioblastoma.It includes 18275 gene symbols.

GOterms a list with 3 goterms and corresponding gene symbols.

GOterms_mark a dataframe with two columns(the first is GOterm name,the second is the corresponding cancerhallmark to which goterm belongs).

- **exp_sur** a dataframe representing expression and survival information.It includes 426 rows standing for samples (rownames are sample tags) and 6 columns(the first 4 columns are molecular names while the rest columns are the survival information).
- **train** a character vector representing 213 trainning sample tags(There is no significant difference in some irrelevant factors (eg:age,gender,et al) between trainning samples and testing samples).
- **test** a character vector representing 212 test sample tags.

Details

All epression data is from a study about glioblastoma.The survival information is from TCGA.

Source

Zhou Du,T. et al. *Integrative genomic analyses reveal clinically relevant long noncoding RNAs in human cancer* nature structural & molecular biology,**20**, 908-913 (2013).

`tri.app` *Modulators Prediction*

Description

This function predicts a modulator affecting the ability of a effector to regulate its targets based on expression profiles.

Usage

```
tri.app(ms, ET, M.exp, E.exp, T.exp, N = 0.25, method = "pearson",
       iqr.filter = c(log2(1.5), log2(1.5), log2(1.5)),
       cor.MvsET = c(0.3, 0.3), cor.EvsT.dif = 0.45, cor.EvsT.abs = 0.4,
       ET.fc.filter = log2(1.5) , ET.p.filter = 0.01 ,
       rand = 100,correction="BH",cores=1)
```

Arguments

`ms`	a character string (vector) specifying candidate modulator names to predict.
`ET`	a dataframe representing effector-target regulations in which factors are effector/target names.
`M.exp`	a numeric dataframe representing expression profile of candidate modulator whose rownames is the candidate modulator names.
`E.exp`	a numeric dataframe representing expression profile of effectors whose rownames are effector names and column number must be equal to *M_exp*.
`T.exp`	a numeric dataframe representing expression profile of targets whose rownames are target names and column number must be equal to *M_exp*.
`N`	a numeric (rangs from 0 to 0.5,default 0.25) specifying proportion,by which LOW/HIGH sample group are extracted from a sorted modulator expression samples.

`method` a character string (default "pearson") indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated.

`iqr.filter` a numeric vector of the form c(modulator_iqr,effector_iqr,target_iqr) specifying IQR threshold to filter expression profiles (default (log2(1.5), log2(1.5), log2(1.5))).

`cor.MvsET` a numeric vector of the form c(cor.MvsE,corMvsT) specifying threshold for correlation between modulator and effector/target (default (0.3, 0.3)).

`cor.EvsT.dif` a numeric (default 0.45) specifying threshold for difference between effector-target correlation in LOW and HIGH sample group.

`cor.EvsT.abs` a numeric (default 0.4) specifying threshold for effector-target correlation either in LOW sample group or HIGH.

`ET.fc.filter` a numeric specifying threshold for fold change of effector expression in LOW sample group versus HIGH (default log2(1.5)).

`ET.p.filter` a numeric specifying threshold for p value of effector expression in LOW sample group versus HIGH (default 0.01).

`rand` a numeric specifying the number of disturbance (default 100).

`correction` Correction method (default "BH") in one of `p.adjust.methods`.

`cores` The number of cores (default 1) to use, i.e. at most how many child processes will be run simultaneously. Must be exactly 1 on Windows (which uses the master process).

Details

Note:All the arguments without default value must be assigned.

This function running a time checked whether a modulator in a sets,one by one,can affect the ability of a effector sets to regulate their corresponding targets.Please go to *Kai Wang,M. et al. Genome-wide identification of post-translational modulators of transcription factor activity in human B cells. Nature biotechnology,**27**, 829-837 (2009)* for detailed information.

The running time and the memory required was increasing as the possible triplets increased.To speed-up the analysis,the function implemented parallel computations when runned on multi-core machines. It used mclapply function in the parallel package to make use of all the CPUs available on the system, with each core simultaneously performing part of the runs.If the possible triplets are big,please work on a big memory machine.

Value

A list containing following components:

- `triplets` predicted triplets and related information,a 7 columns dataframe as following:
  - `modulator effector target` represented modulator/effector/target names,respectively;
  - `R_low R_high` effector-target correlation in LOW/HIGH sample group,respectively;
  - `P_value` significance of the triplet;
  - `fdr` corrected P_value by the assigned method;
- `initialnot` names of modulators whose expression is not in initial expression profile (*M.exp*);
- `filterdnot` names of modulators whose expression is in initial expression profile but not in filterd profile by IQR;

Examples

```
##Different types of candidate modulators to predict
##Here we take four candidate modulators for example
##Two for modulator;one for initialnot;one for filterdnot(see value section in details)
tri.app(ms=datatests[["m_app"]],ET=datatests[["ET"]],M.exp=datatests[["M_exp"]],
        E.exp=datatests[["E_exp"]],T.exp=datatests[["T_exp"]])
```

tri.basic *Overview of Triplets' Expression*

Description

Plot to show effector/target expression in a triplet (or triplets).

Usage

```
tri.basic(tri, M.exp, E.exp, T.exp, index = 1, N = 0.25)
```

Arguments

tri	a dataframe (or matrix) representing triplets to plot in which the first column is modulator id,the second is effector id and the third column is target id.The id style must be consistent with rownames of *M.exp E.exp T.exp*,respectively.
M.exp	a numeric dataframe (or matrix) representing modulator expression profile whose rownames is modulator id.
E.exp	a numeric dataframe (or matrix) representing effector expression profile whose rownames is effector id.
T.exp	a numeric dataframe (or matrix) representing target expression profile whose rownames is target id.
index	a numeric vector (default 1) specifying rowindex of triplets to plot.
N	a numeric (rangs from 0 to 0.5,default 0.25) specifying proportion,by which LOW/HIGH sample group are extracted from a sorted modulator expression samples.

Details

Note:All the arguments without default value must be assigned.

For each triplet,the plot consists of 3 parts.

The left-top is a barplot showing expression of effector in LOW/HIGH sample group. The right-top is a scatter diagram with a linear fitted line using the function *lm*.The scatter diagram shows expression of target versus effector in LOW/HIGH sample group in which red is the HIGH while green is the LOW.The right-bottom is a barplot showing expression of target in LOW/HIGH sample group.

Value

A dataframe whose rownames represent rowindex of triplets and columns represent 6 kinds of value of the plot (P_effector_target, P_effector, P_target, R_low, R_high, P_low, P_high).

- `P_effector_target` the significance of linear fitted lines;
- `P_effector` the significance of difference between effector expression in LOW/HIGH sample group.
- `P_target` the significance of difference between target expression in LOW/HIGH sample group.
- `R_low` the effector-target correlation in LOW sample group;
- `R_high` the effector-target correlation in HIGH sample group;
- `P_low` the significance of R_low;
- `P_high` the significance of R_high;

Examples

```
#One triplet
tri.basic(tri=datatests[["tri_bs"]],M.exp=datatests[["M_exp"]],
          E.exp=datatests[["E_exp"]],T.exp=datatests[["T_exp"]])
#Two triplets (or more)
tri.basic(tri=datatests[["tri_bs"]],M.exp=datatests[["M_exp"]],
          E.exp=datatests[["E_exp"]],T.exp=datatests[["T_exp"]],
  index=c(1,2))
```

tri.enrich *Modulator Functional Enrichment*

Description

Targets of a modulator in the triplets is enriched to GOterms based on the hypergeometric distribution. It can also owe GOterms to disease hallmarks at the same time.

Usage

```
tri.enrich(tri, GOterms, background, inter.thr = 2,
          GOterms.mark = NULL,correction="BH")
```

Arguments

`tri`	a dataframe (or matrix) representing the triplets used to enrich.The first column is modulator;the second column is effetor;the third column is target.
`GOterms`	a list whose variable is a GOterm name and the content is genes annotated on the GOterm.
`background`	a vector containing a gene set in which GOterm annotated genes must be.Its id style must be consistent with the id format in *GOterms*.

`inter.thr` a numeric (default 2) representing min number of intersection between a modulator's targets and a GOterms genes.

`GOterms.mark` a dataframe (or matrix;default NULL) with 2 columns in which the first represent GOterm sets to be enriched while the second represent hallmark to which the GOterm belongs;

`correction` correction method (default "BH") in one of `p.adjust.methods`.

Details

Note:All the arguments without default value must be assigned.

If *background* is NULL,then targets of a modulator is enriched to the GOterms genes passed in; If *background* is not NULL,then targets of a modulator is enriched to the GOterms genes filterd by the background.

If *GOterms.mark* is NULL,it only do GOterms enrichment; If *GOterms.mark* is not NULL,it also owe GOterms to disease marks.

Value

If *GOterms.mark* is NULL,it is a 6 column dataframe as following:

- `modulator` the modulator name;
- `GOterm` the GOterm name;
- `mtarnum` the target number of a modulator;
- `GOtarnum` the gene number of a GOterm;
- `internum` the number of intersected factor between a GOterm genes and a modulator targets;
- `P_value` the significance of enrichment;
- `fdr` corrected P_value by the assigned method;

If *GOterms.mark* is not NULL,it added a seventh column(named "mark" representing the disease mark) besides six columns above.

See Also

`phyper`

Examples

```
#Functional enrichment without disease hallmarks
tri.enrich(tri=datatests[["tri_enrich"]],GOterms=datatests[["GOterms"]],
           background=datatests[["background"]])
#Funtional enrichment with disease hallmarks
tri.enrich(tri=datatests[["tri_enrich"]],GOterms=datatests[["GOterms"]],
           background=datatests[["background"]],
   GOterms.mark=datatests[["GOterms_mark"]])
```

tri.surv *Survival Analysis of Triplets*

Description

Generate plots describing expression and survival comparison for train/test sample groups of a triplet (or triplets).

Usage

```
tri.surv(tri, exp.sur, train, test, index = 1)
```

Arguments

`tri`	a character string dataframe (or matrix) specifying triplets in which the first column is modulator,the second is effector,the third is target.
`exp.sur`	a dataframe specifying expression and survival information.Its rownames are sample names.Its colnames are factor names in triplets and survival tag (see example data in details).
`train`	a character string vector specifying train sample names.
`test`	a character string vector specifying test sample names.
`index`	a numeric vector (default 1) reprsenting rowindex of triplets analyed.

Details

Note:All the arguments without default value must be assigned.

For the output,a triplet corresponds 6 plots,3 for train samples and 3 for test samples.For train sameples,one plot is to show expression of triplet,another is to show risk scores,the other is comparison of survivorship curve between highrisk and lowrisk sameples.

Value

A dataframe whose rows represent different triplets while columns represent 15 kinds of information on the triplet. The columns are:

- `modulator` the modulator name;
- `effector` the effector name;
- `target` the target name;
- `coef_modulator` the *coxph* coefficient of modulator;
- `p_modulator` the significance of *coef_lnc*;
- `coef_effector` the *coxph* coefficient of effector;
- `p_effector` the significance of *coef_effector*;
- `coef_target` the *coxph* coefficient of target;
- `p_target` the significance of *coef_target*;

- N_train1 the sample number with low risk score in train sameples;
- N_train2 the sample number with high risk score in train sameples;
- dif_train the significance of survival difference between low/high risk sameples in train samples;
- N_test1 the sample number with low risk score in test sameples;
- N_test2 the sample number with high risk score in test sameples;
- dif_test the significance of survival difference between low/high risk sameples in test samples;

See Also

coxph,Surv,survdiff

Examples

```
#a triplet
tri.surv(tri=datatests[["tri_bs"]],exp.sur=datatests[["exp_sur"]],
         train=datatests[["train"]],test=datatests[["test"]])
#two triplets(or more)
tri.surv(tri=datatests[["tri_bs"]],exp.sur=datatests[["exp_sur"]],
         train=datatests[["train"]],test=datatests[["test"]],
 index=c(1,2))
```

Index