R documentation

of '/Users/francescapetralia/Dropbox/iJRFNet' etc.

March 2, 2017

R topics documented:

`Derive_network` *Compute permutation-based FDR of importance scores and return estimated interactions.*

Description

This function computes permutation-based FDR of importance scores and returns gene-gene interactions.

Usage

```
Derive_network(out.iJRFNet,out.perm,TH)
```

Arguments

`out.iJRFNet`	Output from object of class `iJRFNet`.
`out.perm`	Output from object of class `Run_permutation`.
`TH`	Threshold for FDR.

Value

List of estimated interactions.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Xie, Y., Pan, W. and Khodursky, A.B., 2005. A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data. *Bioinformatics*, **21**(23), pp.4280-4288.

Examples

```
# --- Generate data sets
 nclasses=2               # number of data sets / classes
 n1<-n2<-20               # sample size for each data set
 p<-5                   # number of variables (genes/proteins)
 genes.name<-paste("G",seq(1,p),sep="")   # genes name

 data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
 data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

  # --- Run iJRFNet and obtain importance score of interactions
  out.iJRFNet<-iJRFNet(X=list(data1,data2),genes.name=genes.name,model="iJRF")

 # --- Obtain importance scores for M permuted data sets
  out.perm<-iJRFNet_permutation(X=list(data1,data2), ntree=1000,mtry=sqrt(5),
  genes.name=genes.name,M=5,model="iJRF")

  # --- Derive final networks
  final.net<-Derive_network(out.iJRFNet,out.perm,0.001)
```

`FinalScore_parallel` *Derive final importance scores for object of class iJRFNet_parallel.*

Description

This function returns importance score for each gene-gene (protein-protein) interaction.

Usage

```
FinalScore_parallel(importance, model, genes.name)
```

Arguments

importance	A matrix containing importance scores. When model iRafNet is implemented, importance is a two dimensional matrix (p x p) with p being the total number of genes/proteins. When either function iJRF or ptmJRF is implemented, importance is a three dimensional matrix of importance scores (p x p x C) with p being the total number of genes/proteins and C the number of classes.
model	Variable indicating which iJRFNet model will be implemented. Takes values in c("iJRF", "iRafNet", "ptmJRF")
genes.name	Vector containing genes name. The order needs to match the rows/columns of importance.

Value

A matrix with I rows and C + 2 columns where I is the total number of gene-gene (protein-protein) interactions and C is the total number of classes. The first two columns contain gene names for each interaction while the remaining columns contain importance scores for different classes. When model iRafNet is implemented, the number of classes is 1 and therefore only three columns will be returned.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2                # number of data sets / classes
n1<-n2<-20                # sample size for each data sets
p<-5                    # number of variables (genes/proteins)
genes.name<-paste("G",seq(1,p),sep="")    # genes/proteins name

data1<-matrix(rnorm(p*n1),p,n1)        # generate data1
data2<-matrix(rnorm(p*n2),p,n1)        # generate data2


 ##-------------------------------------------------------------------------##
 ## ---  Run iJRFNet

 ## --- Run multiple jobs in parallel and combine them

  out.new<-array(0,c(p,p,nclasses))
  n.var=0
  for (k in 1:3){
     out<-iJRFNet_parallel(X=list(data1,data2),genes.name=genes.name,
     model="iJRF",parallel=c(k,2))

     n.target<-dim(out$importance)[2]

     for (c in 1:nclasses) {
     out.new[,seq(n.var+1,n.var+n.target),c]<-out$importance[,,c];}
```

```
  n.var=n.var+n.target
 }

## --- Derive interactions
FinalScore_parallel(importance=out.new,model="iJRF",genes.name=genes.name)
```

`FinalScore_parallel_permutation` *Derive final importance scores of interactions for one permuted data set run in parallel.*

Description

This function returns final importance scores of interactions for one permuted data set run in parallel.

Usage

```
FinalScore_parallel_permutation(importance, model, genes.name,
                                to.store=NULL)
```

Arguments

`importance`	A matrix containing importance scores. When model `iRafNet` is implemented, `importance` is a two dimensional matrix (`p` x `p`) with `p` being the total number of genes/proteins. When either function `iJRF` or `ptmJRF` is implemented, `importance` is a three dimensional matrix of importance scores (`p` x `p` x `C`) with `p` being the total number of genes/proteins and `C` the number of classes.
`model`	Variable indicating which iJRFNet model needs to be imlemented. Takes values in `c("iJRF", "iRafNet","ptmJRF")`
`genes.name`	Vector containing genes name. The order needs to match the rows/columns of `importance`.
`to.store`	Optional Integer. Total number of importance scores to be stored. When omitted, all importance scores will be returned. Note that to compute FDR and derive the final network via function `Derive_network` we do not need all `(p-p) x p / 2` importance scores where `p` is the total number of proteins/genes. A sufficiently large number would work. This number is usually chosen based on the number of nodes and is the maximum number of interactions that you would expect.

Value

A matrix with `to.store` rows and `C` columns where `to.store` is the numbere of top importance scores and `C` is the number of classes. When `to.store` is omitted, all importance scores will be returned.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes/proteins)
genes.name<-paste("G",seq(1,p),sep="")   # genes/proteins name

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

 ##-----------------------------------------------------------------------------##
 ## ---  Run iJRF

 # --- Run multiple jobs in parallel and combine them
  out.new<-array(0,c(p,p,nclasses))
  n.var=0
  for (k in 1:3){
     out<-iJRFNet_parallel_permutation(X=list(data1,data2),
     genes.name=genes.name, model="iJRF",parallel=c(k,2),seed=1)

     n.target<-dim(out$importance)[2]
     for (c in 1:nclasses) {
     out.new[,seq(n.var+1,n.var+n.target),c]<-out$importance[,,c];}
     n.var=n.var+n.target
   }

  # --- Derive interactions
  FinalScore_parallel_permutation(importance=out.new,model="iJRF",
                                  genes.name=genes.name)
```

`Hubgenes_barplot` *Degree plot of hubgenes.*

Description

This function returns a degree plot for the top hub-genes in a given network.

Usage

```
Hubgenes_barplot(net.final,net1=NULL,net2=NULL,genes.name,
                 name.net1=NULL,name.net2=NULL,num.hub=NULL)
```

Arguments

net.final	List object from function Derive_network. The cth element of the list contains estimated interaction for the cth class.
net1	Optional integer. The network whose hubgenes must be plot. This is basically the element of the list net.final that needs to be considered. If omitted, it is set to 1.
net2	Optional integer. This is a network used for comparison and is the element of the list net.final that needs to be considered for comparison. When net2 is not omitted, the plot shows the degree of hubgenes in net1 indicating for each of them how many edges are shared with net2 and how many edges are net1- and net2- specific.
genes.name	A vector containing gene names.
name.net1	Optional string containing the name of net1.
name.net2	Optional string containing the name of net2.
num.hub	Number of top hubgenes the function will plot.

Value

Degree plot of top num.hub hub-genes in net1. When net2 is not omitted, for each hub-gene in net1, the plot will show the number of connecting edges shared with net2 and the number of connecting edges that are net1 and net2 specific.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Xie, Y., Pan, W. and Khodursky, A.B., 2005. A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data. *Bioinformatics*, **21**(23), pp.4280-4288.

Examples

```
  # --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes/proteins)
genes.name<-paste("G",seq(1,p),sep="")   # genes/proteins name

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

 # --- Run iJRF and obtain importance score of interactions
 out.iJRFNet<-iJRFNet(X=list(data1,data2),genes.name=genes.name,
                      model="iJRF")

 # --- Obtain importance scores for M permuted data sets
 out.perm<-iJRFNet_permutation(X=list(data1,data2), ntree=1000,
             mtry=sqrt(5),genes.name=genes.name,M=5,model="iJRF")
```

```
# --- Derive final networks
net.final<-Derive_network(out.iJRFNet,out.perm,0.001)

# --- Degree plot
final.net<-Hubgenes_barplot(net.final,genes.name=genes.name)
```

`iJRF` *Joint Random Forest for the simultaneous estimation of multiple related networks*

Description

Algorithm for the simultaneous estimation of multiple related networks. Some of the functions utilized are a modified version of functions contained in the R package randomForest (A. Liaw and M. Wiener, 2002).

Usage

```
iJRF(X, W=NULL, ntree=NULL, mtry=NULL,genes.name)
```

Arguments

`X`	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_j` is a `(p x n_j)` matrix with rows corresponding to genes and columns to samples. Missing values are not allowed.
`W`	`(p x p)` Optional symmetric matrix containing sampling scores. When omitted, the standard JRF algorithm without weighted sampling scheme will be implemented. Element `(i,j)` contains interaction score `(i - j)`. Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene `i` interacting with gene `j`. Columns and rows of `W` must be in the same order as the columns of `X`. Sampling scores `W` are computed considering one prior data such as protein-protein interactions.
`ntree`	Numeric value: number of trees. If omitted, 1000 trees are considered.
`mtry`	Numeric value: number of predictors to be sampled at each node. If omitted, `mtry` is set to the square root of the number of variables.
`genes.name`	Vector containing genes name. The order needs to match the rows of `x_j`.

Value

A matrix with `I` rows and `C + 2` columns where `I` is the total number of gene-gene interactions and `C` is the number of classes. The first two columns contain gene names for each interaction while the remaining columns contain importance scores for different classes.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")   # genes name
W<-abs(matrix(rnorm(p*p),p,p))    # generate weights for relationships

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

# --- Standardize variables to mean 0 and variance 1
 data1 <- t(apply(data1, 1, function(x) { (x - mean(x)) / sd(x) } ))
 data2 <- t(apply(data2, 1, function(x) { (x - mean(x)) / sd(x) } ))

# --- Run JRF and obtain importance score of interactions for each class
 out<-iJRF(X=list(data1,data2),W=W,mtry=round(sqrt(p-1)),
          ntree=1000,genes.name=genes.name)
```

`iJRFNet` *Derive importance scores for function of class iJRFNet.*

Description

This function computes importance score for `M` permuted data sets. Sample labels of target genes are randomly permuted and JRF is implemented. Resulting importance scores can be used to derive an estimate of FDR.

Usage

```
iJRFNet(X, W=NULL, ntree=NULL, mtry=NULL, model=NULL, genes.name,
        ptm.name=NULL)
```

Arguments

`X`	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_j` is a `(p x n_j)` matrix with rows corresponding to genes and columns to samples. Rows need to be the same across objects, while samples can vary. Missing values are not allowed. If `model="ptmJRF"`, the first object of the list must contain the expression of post translational modification variables. Only in this case, the number of variables in the first object might differ from that of other objects. Rows of `X[[1]]` does not need to be ordered in a specific way.
`W`	`(p x p)` Optional symmetric matrix containing sampling scores. When omitted, the standard JRF algorithm without weighted sampling scheme will be implemented. Element `(i,j)` contains interaction score `(i - j)`. Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene `i` interacting with gene `j`. Columns and rows of `W` must be in the same order as the columns of `X`. Sampling scores `W` are computed considering one prior data such as protein-protein interactions.

ntree	Numeric value: number of trees. If omitted, 1000 trees are considered.
mtry	Numeric value: number of predictors to be sampled at each node. If omitted, mtry is set to the square root of the number of variables.
model	Variable indicating which iJRFNet model needs to be run. Takes values in c("iJRF", "iRafNet","ptmJRF")
genes.name	Vector containing genes name. The order needs to match the rows of x_j.
ptm.name	List of post translational modification variables in protein domain. This list must be ordered as rows of X[[1]].

Value

A three dimensional matrix (I x M x C) with I being the number of total interactions, M the number of permutations and C the number of classes. Element (i,j,k) corresponds to the importance score for interaction i, permuted data j and class k.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")   # genes name

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

##-----------------------------------------------------------------------------##
## ---  Run iJRFNet

# --- Obtain importance scores of gene-gene (protein-protein) interactions
 out<-iJRFNet(X=list(data1,data2),genes.name=genes.name,model="iJRF")

##-----------------------------------------------------------------------------##
## ---  Run iRafNet

W<-abs(matrix(rnorm(p*p),p,p))    # generate weights for interactions

# --- Obtain importance scores of gene-gene (protein-protein) interactions
out<-iJRFNet(X=list(data1),W=W,genes.name=genes.name,model="iRafNet")
```

`iJRFNet_parallel` *Derive importance scores for a subset of target genes for functions of class iJRFNet.*

Description

This function computes importance score in parallel for a subset of target genes.

Usage

```
iJRFNet_parallel(X, W=NULL, ntree=NULL, mtry=NULL, model=NULL,
                 genes.name, ptm.name=NULL,parallel)
```

Arguments

`X`	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_j` is a `(p x n_j)` matrix with rows corresponding to genes and columns to samples. Rows need to be the same across objects, while samples can vary. Missing values are not allowed. If `model="ptmJRF"`, the first object of the list must contain the expression of post translational modification variables. Only in this case, the number of variables in the first object might differ from that of other objects. Rows of `X[[1]]` does not need to be ordered in a specific way.
`W`	`(p x p)` Optional symmetric matrix containing sampling scores. When omitted, the standard JRF algorithm without weighted sampling scheme will be implemented. Element `(i,j)` contains interaction score `(i - j)`. Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene `i` interacting with gene `j`. Columns and rows of `W` must be in the same order as the columns of `X`. Sampling scores `W` are computed considering one prior data such as protein-protein interactions.
`ntree`	Numeric value: number of trees. If omitted, 1000 trees are considered.
`mtry`	Numeric value: number of predictors to be sampled at each node. If omitted, `mtry` is set to the square root of the number of variables.
`model`	Variable indicating which iJRFNet model needs to be run. Takes values in `c("iJRF", "iRafNet","ptmJRF")`
`genes.name`	Vector containing genes name. The order needs to match the rows of `x_j`.
`ptm.name`	List of post translational modification variables in protein domain. This list must be ordered as rows of `X[[1]]`. This is required only if function `ptmJRF` is implemented.
`parallel`	Vector containing two elements `c(num.job,num.targets)`. The first element is the job number that is implemented, target genes will be divided in `J` jobs each containing a specific number of target genes. The second element contains the number of target genes considered in each job.

Value

List object containing:

`num.par`	Integer. Parallel batch implemented.

`model` Variable indicating which iJRFNet model needs to be run. Takes values in `c("iJRF", "iRafNet","ptmJRF")`

`importance` A matrix containing importance score. When function `iRafNet`, this is a two dimensional matrix (`p` x `num.targets`) with `num.targets` being the number of targets considered for this parallel batch and `p` the number of genes. When function `iJRF` or `ptmJRF` is implemented, this is a three dimensional matrix of importance scores (`p` x `num.targets` x `C`) with `num.targets` being the number of targets considered in each batch, `p` the total number of genes/proteins and `C` the number of classes.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")   # genes name

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

# --- Run moultiple jobs and combine them for each function

  # -- function iJRF
  out.new<-array(0,c(p,p,nclasses))
  n.var=0
  for (k in 1:3){
     out<-iJRFNet_parallel(X=list(data1,data2),genes.name=genes.name,
     model="iJRF",parallel=c(k,2))

     n.target<-dim(out$importance)[2]
     for (c in 1:nclasses) {
     out.new[,seq(n.var+1,n.var+n.target),c]<-out$importance[,,c];}
     n.var=n.var+n.target
   }

  # -- function iRafNet
  W<-abs(matrix(rnorm(p*p),p,p))    # generate weights for interactions
  for (k in 1:3){
     out<-iJRFNet_parallel(X=list(data1),W=W,genes.name=genes.name,
     model="iRafNet",parallel=c(k,2))
     print(dim(out$importance))
     if (k==1) out.new<-out$importance
     if (k >2) out.new<-cbind(out.new,out$importance)
  }

   # -- function ptmJRF
```

```
genes.name<-paste("G",seq(1,p),sep="")    # genes name
ptm.name<-c("G1","G2","G3","G3","G4","G5","G1")    # ptm name
p.ptm<-length(ptm.name)

data1<-matrix(rnorm(p.ptm*n2),p.ptm,n1)       # generate PTM data
data2<-matrix(rnorm(p*n1),p,n1)       # generate global proteomics data

out.new<-array(0,c(p,p,nclasses)) # -- p x p matrix of importance scores
n.var=0
for (k in 1:3){

  out<-iJRFNet_parallel(X=list(data1,data2),genes.name=genes.name,
                 ptm.name=ptm.name,model="ptmJRF",parallel=c(k,2))

  n.target<-dim(out$importance)[2]
  for (c in 1:nclasses) {
  out.new[,seq(n.var+1,n.var+n.target),c]<-out$importance[,,c];}
  n.var=n.var+n.target
}
```

`iJRFNet_parallel_permutation` *Derive importance scores for a subset of target genes for functions of class iJRFNet.*

Description

This function computes importance score in parallel for a subset of target genes based on one permuted data.

Usage

```
iJRFNet_parallel_permutation(X, W=NULL, ntree=NULL, mtry=NULL, model=NULL,
                      genes.name, ptm.name=NULL, parallel,seed)
```

Arguments

`X`	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_j` is a `(p x n_j)` matrix with rows corresponding to genes and columns to samples. Rows need to be the same across objects, while samples can vary. Missing values are not allowed. If `model="ptmJRF"`, the first object of the list must contain the expression of post translational modification variables. Only in this case, the number of variables in the first object might differ from that of other objects. Rows of `X[[1]]` does not need to be ordered in a specific way.
`W`	`(p x p)` Optional symmetric matrix containing sampling scores. When omitted, the standard JRF algorithm without weighted sampling scheme will be implemented. Element `(i,j)` contains interaction score `(i - j)`. Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene `i` interacting with gene `j`. Columns and rows of `W` must be in the same order as the columns of `X`. Sampling scores `W` are computed considering one prior data such as protein-protein interactions.

ntree	Numeric value: number of trees. If omitted, 1000 trees are considered.
mtry	Numeric value: number of predictors to be sampled at each node. If omitted, mtry is set to the square root of the number of variables.
model	Variable indicating which iJRFNet model needs to be run. Takes values in c("iJRF", "iRafNet","ptmJRF")
genes.name	Vector containing genes name. The order needs to match the rows of x_j.
ptm.name	List of post translational modification variables in protein domain. This list must be ordered as rows of X[[1]]. This is required only if function ptmJRF is implemented.
parallel	Vector containing two elements c(num.job,num.targets). The first element is the job number that is implemented, target genes will be divided in J jobs each containing a specific number of target genes. The second element contains the number of target genes considered in each job.
seed	Integer: Seed of permutation.

Value

List object containing:

num.par	Integer. Parallel batch implemented.
model	Variable indicating which iJRFNet model needs to be run. Takes values in c("iJRF", "iRafNet","ptmJRF")
importance	A matrix containing importance score. When function iRafNet, this is a two dimensional matrix (p x num.targets) with num.targets being the number of targets considered for this parallel batch and p the number of genes. When function iJRF or ptmJRF is implemented, this is a three dimensional matrix of importance scores (p x num.targets x C) with num.targets being the number of targets considered in each batch, p the total number of genes/proteins and C the number of classes.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")   # genes name

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

# --- Run multiple jobs and combine them for each function

 ##------------------------------------------------------------------------##
```

```
## ---  Run iJRF
 out.new<-array(0,c(p,p,nclasses))
 n.var=0
 for (k in 1:3){
    out<-iJRFNet_parallel_permutation(X=list(data1,data2),genes.name=genes.name,
    model="iJRF",parallel=c(k,2),seed=1)

    n.target<-dim(out$importance)[2]
    for (c in 1:nclasses) {
    out.new[,seq(n.var+1,n.var+n.target),c]<-out$importance[,,c];}
    n.var=n.var+n.target
  }

##-----------------------------------------------------------------------------##
## ---  Run iRafNet

 W<-abs(matrix(rnorm(p*p),p,p))    # generate weights for interactions
 for (k in 1:3){
    out<-iJRFNet_parallel_permutation(X=list(data1),W=W,
    genes.name=genes.name,model="iRafNet",parallel=c(k,2),seed=1)
    print(dim(out$importance))
    if (k==1) out.new<-out$importance
    if (k >2) out.new<-cbind(out.new,out$importance)
 }

  ##-----------------------------------------------------------------------------##
  ## ---  Run ptmJRF

  ## ---- Generate Data
  genes.name<-paste("G",seq(1,p),sep="")   # genes name
  ptm.name<-c("G1","G2","G3","G3","G4","G5","G1")   # ptm name
  p.ptm<-length(ptm.name)

  data1<-matrix(rnorm(p.ptm*n2),p.ptm,n1)       # generate PTM data
  data2<-matrix(rnorm(p*n1),p,n1)       # generate global proteomics data

  ## --- Run multiple jobs in parallel and combine them
  out.new<-array(0,c(p,p,nclasses)) # -- p x p matrix of importance scores
  n.var=0
  for (k in 1:3){

    out<-iJRFNet_parallel_permutation(X=list(data1,data2),genes.name=genes.name,
                  ptm.name=ptm.name,model="ptmJRF",parallel=c(k,2),seed=1)

    n.target<-dim(out$importance)[2]
    for (c in 1:nclasses) {
    out.new[,seq(n.var+1,n.var+n.target),c]<-out$importance[,,c];}
    n.var=n.var+n.target
  }
```

---

iJRFNet_permutation

Derive importance scores for M permuted data sets.

Description

This function computes importance score for M permuted data sets. Sample labels of target genes are randomly permuted and JRF is implemented. Resulting importance scores can be used to derive an estimate of FDR.

Usage

```
iJRFNet_permutation(X,W=NULL, ntree=NULL, mtry=NULL,
       genes.name=NULL, M=NULL, model, ptm.name=NULL,
       to.store=NULL)
```

Arguments

X	List object containing expression data for each class, X=list(x_1,x_2, ...) where x_j is a (p x n_j) matrix with rows corresponding to genes and columns to samples. Rows need to be the same across objects, while samples can vary. Missing values are not allowed. If model="ptmJRF", the first object of the list must contain the expression of post translational modification variables. Only in this case, the number of variables in the first object might differ from that of other objects. Rows of X[[1]] does not need to be ordered in a specific way.
W	(p x p) Optional symmetric matrix containing sampling scores. When omitted, the standard JRF algorithm without weighted sampling scheme will be implemented. Element (i,j) contains interaction score (i - j). Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene i interacting with gene j. Columns and rows of W must be in the same order as the columns of X. Sampling scores W are computed considering one prior data such as protein-protein interactions.
ntree	Numeric value: number of trees. If omitted, 1000 trees are considered.
mtry	Numeric value: number of predictors to be sampled at each node. If omitted, mtry is set to the square root of the number of variables.
genes.name	Vector containing genes name. The order needs to match the rows of x_j.
M	Integer: total number of permutations. If omitted, 100 permutations will be run.
model	Variable indicating which iJRFNet model needs to be run. Takes values in c("iJRF", "iRafNet","ptmJRF")
ptm.name	List of post translational modification variables in protein domain. This list must be ordered as rows of X[[1]].
to.store	Optional Integer. Total number of importance scores to be stored. When omitted, all importance scores will be stored. Note that to compute FDR we do not need all (p-p) x p / 2 importance scores where p is the total number of proteins/genes, a sufficiently large number would work. This number is usually chosen based on the number of nodes. Suggested value is p x 20.

Value

A three dimensional matrix (I x M x C) with I being the number of total interactions, M the number of permutations and C the number of classes. Element (i,j,k) corresponds to the importance score for interaction i, permuted data j and class k.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2                # number of data sets / classes
n1<-n2<-20                # sample size for each data sets
p<-5                    # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")    # genes name

data1<-matrix(rnorm(p*n1),p,n1)        # generate data1
data2<-matrix(rnorm(p*n2),p,n1)        # generate data2

# --- Obtain importance scores for M permuted data sets
 out<-iJRFNet_permutation(X=list(data1,data2), ntree=1000,
 mtry=sqrt(5), genes.name=genes.name, M=5, model="iJRF")
```

`iJRF_permutation` *Joint Random Forest for the simultaneous estimation of multiple related networks*

Description

Derive importance score for models of class iJRF based on one permuted data set.

Usage

```
iJRF_permutation(X, W=NULL, ntree=NULL, mtry=NULL,
               genes.name, seed, to.store=NULL)
```

Arguments

`X`	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_j` is a `(p x n_j)` matrix with rows corresponding to genes and columns to samples. Missing values are not allowed.
`W`	`(p x p)` Optional symmetric matrix containing sampling scores. When omitted, the standard JRF algorithm without weighted sampling scheme will be implemented. Element `(i,j)` contains interaction score `(i - j)`. Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene `i` interacting with gene `j`. Columns and rows of `W` must be in the same order as the columns of `X`. Sampling scores `W` are computed considering one prior data such as protein-protein interactions.
`ntree`	Numeric value: number of trees. If omitted, `ntree` is set to `1000`
`mtry`	Numeric value: number of predictors to be sampled at each node. If omitted, `mtry` is set to the square root of the number of variables.

genes.name	Vector containing genes name. The order needs to match the rows of x_j.
seed	Integer. Seed of permutation
to.store	Optional Integer. Total number of importance scores to be stored. When omitted, all importance scores will be stored. Note that to compute FDR we do not need all (p-p) x p / 2 importance scores where p is the total number of proteins/genes, a sufficiently large number would work. This number is usually chosen based on the number of nodes. Suggested value is p x 20.

Value

A matrix with I rows and C + 2 columns where I is the total number of gene-gene interactions and C is the number of classes. The first two columns contain gene names for each interaction while the remaining columns contain importance scores for different classes.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2                # number of data sets / classes
n1<-n2<-20                # sample size for each data sets
p<-5                    # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")    # genes name
W<-abs(matrix(rnorm(p*p),p,p))     # generate weights for relationships

data1<-matrix(rnorm(p*n1),p,n1)        # generate data1
data2<-matrix(rnorm(p*n2),p,n1)        # generate data2

# --- Run JRF and obtain importance score of interactions for each class
 out<-iJRF_permutation(X=list(data1,data2),W=W,
                       genes.name=genes.name,seed=1)
```

---

iRafNet *Integrative random forest for co-expression network inference*

Description

This function fits iRafNet, a flexible unified integrative algorithm that allows information from prior data, such as protein-protein interactions and gene knock-down, to be jointly considered for gene regulatory network inference. This function takes as input only one set of sampling scores, computed considering one prior data such as protein-protein interactions or gene expression from knock-out experiments. Note that some of the functions utilized are a modified version of functions contained in the R package randomForest (A. Liaw and M. Wiener, 2002).

Usage

```
iRafNet(X, W, ntree=NULL, mtry=NULL,genes.name)
```

Arguments

X	(n x p) Matrix containing expression levels for n samples and p genes.
W	(p x p) Symmatrix matrix containing iRafNet sampling scores. Element (i,j) contains score for interaction (i - j). Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene i interacting with gene j. Columns and rows of W must be in the same order as the columns of X. Sampling scores W are computed considering one prior data such as protein-protein interactions.
ntree	Numeric value: number of trees. If omitted, 1000 trees are considered.
mtry	Numeric value: number of predictors to be sampled at each node. If omitted, mtry is set to the square root of the number of variables.
genes.name	Vector containing genes name. The order needs to match the rows of x_j.

Value

Importance score for each regulatory relationship. The first column contains gene name of regulators, the second column contains gene name of targets, and third column contains corresponding importance scores.

References

Petralia, F., Wang, P., Yang, J., Tu, Z. (2015) Integrative random forest for gene regulatory network inference, *Bioinformatics*, **31**, i197-i205.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
n<-20                  # sample size
p<-5                   # number of genes
genes.name<-paste("G",seq(1,p),sep="")   # genes name
data<-matrix(rnorm(p*n),p,n)      # generate expression matrix
W<-abs(matrix(rnorm(p*p),p,p))    # generate weights for interactions

# --- Run iRafNet and obtain importance score of interactions
out<-iRafNet(X=list(data),W=W,genes.name=genes.name)
```

---

iRafNet_permutation

Integrative random forest for co-expression network inference

---

Description

This function fits iRafNet, a flexible unified integrative algorithm that allows information from prior data, such as protein-protein interactions and gene knock-down, to be jointly considered for gene regulatory network inference. This function takes as input only one set of sampling scores, computed considering one prior data such as protein-protein interactions or gene expression from knock-out experiments. Note that some of the functions utilized are a modified version of functions contained in the R package randomForest (A. Liaw and M. Wiener, 2002).

Usage

```
iRafNet_permutation(X, W, ntree=NULL, mtry=NULL, genes.name,
                    seed, to.store=NULL)
```

Arguments

X	(n x p) Matrix containing expression levels for n samples and p genes.
W	(p x p) Symmatrix matrix containing iRafNet sampling scores. Element (i,j) contains score for interaction (i - j). Scores must be non-negative. Larger value of sampling score corresponds to higher likelihood of gene i interacting with gene j. Columns and rows of W must be in the same order as the columns of X. Sampling scores W are computed considering one prior data such as protein-protein interactions.
ntree	Numeric value: number of trees. If omitted, ntree is set to 1000.
mtry	Numeric value: number of predictors to be sampled at each node. If omitted, mtry is set to the square root of the number of variables.
genes.name	Vector containing genes name. The order needs to match the rows of x_j.
seed	Integer: Seed of permutation.
to.store	Optional Integer. Total number of importance scores to be stored. When omitted, all importance scores will be stored. Note that to compute FDR we do not need all (p-p) x p / 2 importance scores where p is the total number of proteins/genes, a sufficiently large number would work. This number is usually chosen based on the number of nodes. Suggested value is p x 20.

Value

Importance score for each regulatory relationship. The first column contains gene name of regulators, the second column contains gene name of targets, and third column contains corresponding importance scores.

References

Petralia, F., Wang, P., Yang, J., Tu, Z. (2015) Integrative random forest for gene regulatory network inference, *Bioinformatics*, **31**, i197-i205.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
n<-20                  # sample size
p<-5                   # number of genes
```

```
genes.name<-paste("G",seq(1,p),sep="")   # genes name
data<-matrix(rnorm(p*n),p,n)      # generate expression matrix
W<-abs(matrix(rnorm(p*p),p,p))    # generate weights for interactions

# --- Run iRafNet and obtain importance score of interactions
out<-iRafNet_permutation(X=list(data),W=W,genes.name=genes.name,seed=1)
```

`Plot_Modules` *Derive Network Modules.*

Description

This function returns the list of modules.

Usage

```
Plot_Modules(net.final,genes.name)
```

Arguments

`net.final`	Network to plot.
`genes.name`	A vector containing gene names.

Value

Return list of modules.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Xie, Y., Pan, W. and Khodursky, A.B., 2005. A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data. *Bioinformatics*, **21**(23), pp.4280-4288.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-40                   # number of variables (genes/proteins)
genes.name<-paste("G",seq(1,p),sep="")   # genes/proteins name

data1<-matrix(rnorm(p*n1),p,n1)       # generate data1
data2<-matrix(rnorm(p*n2),p,n1)       # generate data2

 # --- Run iJRF and obtain importance score of interactions
 out.iJRFNet<-iJRFNet(X=list(data1,data2),genes.name=genes.name,
```

```
                        model="iJRF")

# --- Degree plot
final.net<-Plot_Modules(out.iJRFNet[sample(dim(out.iJRFNet)[1],200),c(1,2)]
             ,genes.name=genes.name)
```

---

ptmJRF — *Joint Random Forest for the simultaneous estimation of interaction networks based on gene expression data, global proteomics data and post translational modification (PTM) data.*

Description

Algorithm for the simultaneous estimation of multiple related networks. Some of the functions utilized are a modified version of functions contained in the R package randomForest (A. Liaw and M. Wiener, 2002).

Usage

```
ptmJRF(X, ntree=NULL, mtry=NULL,genes.name,ptm.name)
```

Arguments

X	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_1` is a `(F x n_j)` matrix with rows corresponding to post translational modification sites and columns to samples, while `x_j` for `j > 1` is a `(p x n_j)` matrix with rows corresponding to proteins and columns to samples. For `x_2, x_3, ...` Rows need to be the same corresponding to the same proteins, while samples can vary. Missing values are not allowed. Rows of object `x_1` does not need to be ordered in a specific way.
ntree	Numeric value: number of trees.
mtry	Numeric value: number of predictors to be sampled at each node.
genes.name	Vector containing genes name. The order needs to match the rows of `x_j`.
ptm.name	List of post translational modification variables in protein domain. This list must be ordered as rows of `X[[1]]`.

Value

A matrix with `I` rows and `C + 2` columns where `I` is the total number of gene-gene interactions and `C` is the number of classes. The first two columns contain gene names for each interaction while the remaining columns contain importance scores for different classes.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")   # genes name
ptm.name<-c("G1","G2","G3","G3","G4","G5","G1")   # ptm name
p.ptm<-length(ptm.name)

data1<-matrix(rnorm(p.ptm*n2),p.ptm,n1)       # generate PTM data
data2<-matrix(rnorm(p*n1),p,n1)       # generate global proteomics data

# --- Run JRF and obtain importance score of interactions for each class
 out<-ptmJRF(X=list(data1,data2),genes.name=genes.name,ptm.name=ptm.name)
```

`ptmJRF_permutation` *Joint Random Forest for the simultaneous estimation of interaction networks based on gene expression data, global proteomics data and post translational modification (PTM) data.*

Description

Algorithm for the simultaneous estimation of multiple related networks. Some of the functions utilized are a modified version of functions contained in the R package randomForest (A. Liaw and M. Wiener, 2002).

Usage

```
ptmJRF_permutation(X, ntree=NULL, mtry=NULL, genes.name,
                   ptm.name, seed, to.store=NULL)
```

Arguments

`X`	List object containing expression data for each class, `X=list(x_1,x_2, ... )` where `x_1` is a `(F x n_j)` matrix with rows corresponding to post translational modification sites and columns to samples, while `x_j` for `j > 1` is a `(p x n_j)` matrix with rows corresponding to proteins and columns to samples. For `x_2, x_3, ...` Rows need to be the same corresponding to the same proteins, while samples can vary. Missing values are not allowed. Rows of object `x_1` does not need to be ordered in a specific way.
`ntree`	Numeric value: number of trees.
`mtry`	Numeric value: number of predictors to be sampled at each node.
`genes.name`	Vector containing genes name. The order needs to match the rows of `x_j`.
`ptm.name`	List of post translational modification variables in protein domain. This list must be ordered as rows of `X[[1]]`.
`seed`	Integer. Permutation seed.

`to.store` Optional Integer. Total number of importance scores to be stored. When omitted, all importance scores will be stored. Note that to compute FDR we do not need all `(p-p) x p / 2` importance scores where `p` is the total number of proteins/genes, a sufficiently large number would work. This number is usually chosen based on the number of nodes. Suggested value is `p x 20`.

Value

A matrix with `I` rows and `C + 2` columns where `I` is the total number of gene-gene interactions and `C` is the number of classes. The first two columns contain gene names for each interaction while the remaining columns contain importance scores for different classes.

`to.store` Optional Integer. Total number of importance scores to be stored. When omitted, all importance scores will be stored. Note that to compute FDR we do not need all `(p-p) x p / 2` importance scores where `p` is the total number of proteins/genes, a sufficiently large number would work. This number is usually chosen based on the number of nodes. Suggested value is `p x 20`.

References

Petralia, F., Song, W.M., Tu, Z. and Wang, P. (2016). New method for joint network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. *Journal of proteome research*, **15**(3), pp.743-754.

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

Examples

```
# --- Generate data sets
nclasses=2               # number of data sets / classes
n1<-n2<-20               # sample size for each data sets
p<-5                   # number of variables (genes)
genes.name<-paste("G",seq(1,p),sep="")   # genes name
ptm.name<-c("G1","G2","G3","G3","G4","G5","G1")   # ptm name
p.ptm<-length(ptm.name)

data1<-matrix(rnorm(p.ptm*n2),p.ptm,n1)       # generate PTM data
data2<-matrix(rnorm(p*n1),p,n1)       # generate global proteomics

# --- Run JRF and obtain importance score of interactions
 out<-ptmJRF(X=list(data1,data2),genes.name=genes.name,
         ptm.name=ptm.name)
```