# Package: difconet (via r-universe)

August 21, 2024

**Version** 1.0-4

**Date** 2017-05-26

**Title** Differential Coexpressed Networks

**Depends** R (>= 3.1.0), gplots

**Imports** stats, utils, stringr, data.table, mvtnorm, graphics, grDevices

**Description** Estimation of DIFferential COexpressed NETworks using diverse and user metrics. This package is basically used for three functions related to the estimation of differential coexpression. First, to estimate differential coexpression where the coexpression is estimated, by default, by Spearman correlation. For this, a metric to compare two correlation distributions is needed. The package includes 6 metrics. Some of them needs a threshold. A new metric can also be specified as a user function with specific parameters (see difconet.run). The significance is be estimated by permutations. Second, to generate datasets with controlled differential correlation data. This is done by either adding noise, or adding specific correlation structure. Third, to show the results of differential correlation analyses. Please see <http://bioinformatica.mty.itesm.mx/difconet> for further information.

**License** GPL (>= 2)

**URL** http://bioinformatica.mty.itesm.mx/difconet

**NeedsCompilation** no

**Author** Elpidio-Emmanuel Gonzalez-Valbuena [aut], Victor Trevino [aut, cre]

**Maintainer** Victor Trevino <vtrevino@itesm.mx>

**Date/Publication** 2017-05-31 06:24:37 UTC

**Repository** https://vtrevino.r-universe.dev

**RemoteUrl** https://github.com/cran/difconet

**RemoteRef** HEAD

**RemoteSha** a193181ae126c78c33de2c8dc843aa85bb8000c7

# Contents



---

`difconet.build.controlled.dataset`
*GENERATES A DATASET CONTROLLING FOR NOISE AND GENES CONNECTED IN NETWORKS*

---

## Description

This function takes a normal dataset and generate simulated tumor stages by adding progressive levels of noise. It may add artificial networks of genes connected at given correlations that can progressively increase or decrease their level of correlation.

## Usage

```
difconet.build.controlled.dataset(data,
    noise.genes = round(nrow(data)*0.1),
    noise.sigma = c(0.0, 0.1, 0.2),
    nonoise.sigma = c(0.0, 0.01, 0.01),
    netcov = matrix(c(
      0.90, 0.90, 0.75, 0.75, 0.60, 0.60, 0.45, 0.45, 0.30, 0.30,
      0.15, 0.15, 0.30, 0.30, 0.45, 0.45, 0.60, 0.60, 0.75, 0.75,
      0.95, 0.95, 0.80, 0.80, 0.65, 0.65, 0.50, 0.50, 0.35, 0.35,
      0.10, 0.10, 0.25, 0.25, 0.40, 0.40, 0.55, 0.55, 0.70, 0.70,
      1.00, 1.00, 0.85, 0.85, 0.70, 0.70, 0.55, 0.55, 0.40, 0.40,
      0.05, 0.05, 0.20, 0.20, 0.35, 0.35, 0.50, 0.50, 0.65, 0.65
      ), ncol=3),
    genes.nets = 10,
    corfunc=function(a,b) cor(a,b,method="spearman"),
    verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `data` | data.frame or matrix representing the normal dataset. Rows are genes and columns are samples. |
| `noise.genes` | the number of genes from data that will noised. |
| `noise.sigma` | Levels of gaussian noise to be added (at zero mean) expressed in a cumulative manner. |
| `nonoise.sigma` | Levels of gaussian noise to be added (at zero mean) for the rest of the genes. |

| | |
|---|---|
| netcov | numeric matrix of correlation levels for networks, rows represent networks and columns represent stages. |
| genes.nets | The number of genes in each generated network. |
| corfunc | Correlation method used. |
| verbose | Print progress. |

## Details

This function generates a simulated tumor progression dataset based on normal data. The progression is done by stages. The number of stages is given by the length of noise.sigma. Each stage will have the same dimensions than data (plus the networks). The stages will be N, T1, T2, and so on. The N is meant to be the data itself with no noise but for generality, the first element of noise.sigma specifies the level of noise for N (default to 0). The next values of noise.sigma will be used to generate T1, T2, and so on. Thus the returned data will be estimated by N=data+noise.sigma[1], T1=N+noise.sigma[2], T2=T1+noise.sigma[3], and so on. Note that noise.sigma will be added only to a specific number of rows given by noise.genes. The value returned is a list of the generated matrices. In top of that, the nonoise.sigma specify the level of noise added to those genes not selected to be noised. This is meant to be lower levels of noise than noise.sigma to avoid that data in stages is just a copy of previous data. This function also adds full connected networks of genes connected at netcov levels. The data added has mean=0 and sd=1. The number of rows represent the networks added. The columns represent the stages.

## Value

List of stages.

## Author(s)

Elpidio Gonzalez and Victor Trevino <vtrevino@itesm.mx>

## References

Gonzalez-Valbuena and Trevino 2017 Metrics to Estimate Differential Co-Expression Networks *Journal Pending* **volume** 00–10

## See Also

difconet.noise.inspection. difconet.run.

## Examples

```
## Not run: difconet.noise.inspection(normaldata, tumordata, sigma=0:15/10)
```

difconet.noise.inspection *PLOT ESTIMATED CORRELATION DISTRIBUTION AFTER ADDING NOISE*

## Description

Plots the estimated correlation distribution of a normal dataset after adding different levels of gaussian noise. It is used to estimate the level of noise needed to be added to a normal dataset to match the correlation distribution of a tumor dataset. This assumes that the correlation distribution of the tumor dataset is sharper around zero.

## Usage

```
difconet.noise.inspection(ndata, tdata, sigma=c(0.5, 0.75, 1.25), maxgenes=5000,
  corfunc=function(a,b) cor(a,b,method="spearman"))
```

## Arguments

| | |
|---|---|
| ndata | The normal dataset. Rows are genes and columns are samples. |
| tdata | The tumor dataset. Rows are genes and columns are samples. Rows of tumor and normal datasets should be the same. |
| sigma | Levels of gaussian noise to be added (at zero mean). |
| maxgenes | Number of genes used to estimate the correlation distribution. If the number of rows in normal/tumor datasets are larger than maxgenes, maxgenes random genes are used for the estimation. |
| corfunc | Correlation method used. |

## Details

Plots the estimated density of correlation distributions of normal, tumor, and normal after adding sigma levels of noise.

## Value

Nothing.

## Author(s)

Elpidio Gonzalez and Victor Trevino <vtrevino@itesm.mx>

## References

Gonzalez-Valbuena and Trevino 2017 Metrics to Estimate Differential Co-Expression Networks *Journal Pending* **volume** 00–10

## See Also

difconet.build.controlled.dataset. difconet.run.

## Examples

```
## Not run: difconet.noise.inspection(normaldata, tumordata, sigma=0:15/10)
```

---

difconet.plot.gene.correlations

*PLOTS THE CORRELATIONS OF A SPECIFIC GENE*

---

## Description

Draw scatter plots of the correlations of a specific gene.

## Usage

```
difconet.plot.gene.correlations(dObj, gene,
  stages=1:length(dObj$stages.data), type=c("density","scatter")[1],
  main=rownames(dObj$stages.data[[1]])[gene],
  legends=TRUE, plot=TRUE, ... )
```

## Arguments

| | |
|---|---|
| dObj | The difconet object. |
| gene | Numeric or character. The gene index/rowname whose correlations will be drawn. |
| stages | Numeric or character. The stages to be included. If **type="scatter"** and more than two stages, a call to pairs is used instead of plot. |
| type | Character. The type of plot **density** or **scatter**. |
| main | Character. The main title passed to plot. |
| legends | Logical. Specifies whether the legends are drawn when **type="density"**. |
| plot | Logical. Specifies whether the plots are actually drawn (to get the correlations). |
| ... | Further parameters passed to plot/pairs. |

## Details

Run the whole process of estimation differences in correlations for a given dataset. The estimations are done for all **metric** values, all **cutoff** values across all **comparisons**.

## Value

The correlations of the gene across stages (invisible).

## Author(s)

Elpidio Gonzalez and Victor Trevino <vtrevino@itesm.mx>

## References

Gonzalez-Valbuena and Trevino 2017 Metrics to Estimate Differential Co-Expression Networks *Journal Pending* **volume** 00–10

## See Also

difconet.run.

## Examples

```
xdata <- matrix(rnorm(1000), ncol=100)
xpredictor <- sample(c("A","B","C","D"),100,replace=TRUE)
dObj <- difconet.run(xdata, xpredictor, metric = 4, num_perms = 10,
  comparisons = list(c("A","D"), c("A","B"), c("B","D")),
  perm_mode = "columns")

#Top highest metric in first comparison but showing correlations in only 3 stages
difconet.plot.gene.correlations(dObj, order(dObj$combstats[[1]][,"M4.dist"],
  decreasing=TRUE)[1], type="s", stages=1:3)
#Bottom lowest metric in second comparison showing all stages
difconet.plot.gene.correlations(dObj, order(dObj$combstats[[2]][,"M4.dist"],
  decreasing=TRUE)[1], type="d")
#Another specific gene (3), showing densities of correlations
difconet.plot.gene.correlations(dObj, 3, type="d")
```

---

# difconet.plot.histograms.heatmap2

*PLOT A HEATMAP REPRESENTATION OF THE DISTRIBUTION OF CORRELATIONS OF MANY GENES*

---

## Description

Draw a heatmap whose rows are genes and columns are segments of the histogram of the distribution of correlations per gene. The height/density of the histogram is shown in colors.

## Usage

```
difconet.plot.histograms.heatmap2(dObj,
  genes=1:10,
  stages=1:length(dObj$stages.data),
  qprobs=c(0,.50,.975,.995), ...)
```

## Arguments

| | |
|---|---|
| dObj | The difconet object. |
| genes | Numeric or character. The gene indexes/rownames included. |
| stages | Numeric or character. The stages to be included. |
| qprobs | The quantiles used to draw the heatmap. Should be 4 points. Each has specific color codes. |
| ... | Further parameters passed to plot/pairs. |

## Details

A heatmap is draw representing the distribution of correlations of several genes across stages.

## Value

Nothing.

## Author(s)

Elpidio Gonzalez and Victor Trevino <vtrevino@itesm.mx>

## References

Gonzalez-Valbuena and Trevino 2017 Metrics to Estimate Differential Co-Expression Networks *Journal Pending* **volume** 00–10

## See Also

difconet.run.

## Examples

```
xdata <- matrix(rnorm(1000), ncol=100)
xpredictor <- sample(c("A","B","C","D"),100,replace=TRUE)
dObj <- difconet.run(xdata, xpredictor, metric = 4, num_perms = 10,
  comparisons = list(c("A","D"), c("A","B"), c("B","D")),
  perm_mode = "columns")

  #Top highest metric in first comparison but showing correlations in only 3 stages
  difconet.plot.gene.correlations(dObj, order(dObj$combstats[[1]][,"M4.dist"],
    decreasing=TRUE)[1], type="s", stages=1:3)
  #Bottom lowest metric in second comparison showing all stages
  difconet.plot.gene.correlations(dObj, order(dObj$combstats[[2]][,"M4.dist"],
    decreasing=TRUE)[1], type="d")
  #Another specific gene (1), showing densities of correlations
  difconet.plot.gene.correlations(dObj, 1, type="d")
```

difconet.run *RUNS A DIFCONET ANALYSIS*

## Description

Estimates the DIFferential COrrelation NETworks analysis from a given dataset.

## Usage

```
difconet.run(data, predictor, metric=c(1,2,3,4,5,6), cutoff=0.3, blocs=5000,
  num_perms=10, comparisons="all", perm_mode="columns", use_all_perm = TRUE,
  save_perm=FALSE, speedup=0, verbose=TRUE, metricfunc=NULL,
  corfunc=function(a,b) cor(a,b,method="spearman") )
```

## Arguments

| | |
|---|---|
| data | data.frame or matrix represent the dataset. Genes in rows, samples in columns. |
| predictor | Factor or numeric vector representing the classes of each column in data. The correlations will be estimated for each class separately. |
| metric | The metrics needed to be calculated. Valid values are 1 to 6 and 8. 1 to 6 are already implemented and shown in details. 8 specifies a user-defined metric specified in metricfunc. |
| cutoff | Cut off values used for metric 1 and/or 3. |
| blocs | Number of rows per block. Because of memory issues, the correlations are estimated by blocks of genes. This value represent the size of the block. Larger values requires more memory if needed. Lower values requiere more cycles and therefore it is slower but makes it computable depending on database size and memory. |
| num_perms | Number of permutations. |
| comparisons | Character or list. If character, it could be "all" to specify all possible combinations of classes. If set to "seq", classes are taken in order and comparisons are done by first versus second, second versus third, and so on. If this is a list containing vectors of two elements, the estimations are done for the specific comparisons included (numeric or character). |
| perm_mode | Character. It determines the how the permutated data is generated. It can be permutated by "columns", permutated by "rows" (all classes/stages), or permutated by rows within each class separately using "rows.class", or "all" in which all data is shuffled. |
| use_all_perm | Logical. If TRUE, it uses all permutated data to estimate the p-value, otherwise it uses only the same row permutations to estimate the p-value (it requires a lot more permutations). |
| save_perm | Logical. If TRUE, it save all permutated data. It may require more memory. |

| | |
|---|---|
| speedup | Numeric. Determines whether the calculation will be sped up. This is experimental. The value specify which metric will be used to speed up. This is done by modeling the dependency of the metric and p-value using 1 percent of the rows. |
| verbose | Logical. Determines if printing progress information. |
| metricfunc | Function. Specify the function to be used if a metric==8 is included. The function should receive dObj, a, and b which correspond to the difconet object and the a and b vectors of correlations needed to estimate the value of the metric. It is assumed a distance-like measure (non-negative) and values close to 0 means no difference whereas larger values represent more dissimilar correlations. |
| corfunc | Function. Specify the function that estimates the correlations, similar to the cor function. The default uses cor and spearman coefficients. |

## Details

Run the whole process of estimation differences in correlations for a given dataset. The estimations are done for all **metric** values, all **cutoff** values across all **comparisons**.

## Value

A difconet object represented as a list. The items are the followings:

| | |
|---|---|
| stage | Vector. A copy of **predictor** (classes). |
| labels | Vector. The levels or values of the different classes. |
| comparisons | The specified **comparisons** parameter. |
| num_perms | The specified number of permutations **num_perms** parameter. |
| perm_mode | The specified number of permutations **perm_mode** parameter. |
| use_all_perm | The specified number of permutations **use_all_perm** parameter. |
| speedup | The specified **speedup** parameter. |
| verbose | The specified **verbose** parameter. |
| metricfunc | The specified **metricfunc** parameter. |
| combinations | A data.frame of the combinations that were compared. |
| stages.data | A list of datasets. This is only the original **data** split by classes. |
| combstats | A list of all comparisons made. Each element contains a matrix whose rows represent the genes and columns represent the results of all metrics (metric.dist : metric value, metric.p : p-value, metric.q : q-value, metric.expr.p : p-value of differential expression for comparison purposes, metric.expr.q : q-value of differential expression.) |
| combdens | A list of the densities of the metric for observed data and permutations. This can be used to compare the estimated metric statistics. |
| permutations | List. If **save_perm==TRUE**, it saves all permutated data. |

## Author(s)

Elpidio Gonzalez and Victor Trevino <vtrevino@itesm.mx>

## References

Gonzalez-Valbuena and Trevino 2017 Metrics to Estimate Differential Co-Expression Networks *Journal Pending* **volume** 00–10

## See Also

difconet.build.controlled.dataset.

## Examples

```
xdata <- matrix(rnorm(1000), ncol=100)
xpredictor <- sample(c("A","B","C","D"),100,replace=TRUE)
dObj <- difconet.run(xdata, xpredictor, metric = 4, num_perms = 10,
  comparisons = list(c("A","D"), c("A","B"), c("B","D")),
  perm_mode = "columns")

## Not run:
  #xpredictor contains A, B, C, and D.
  #xdata contains the data matrix
  dObj <- difconet.run(xdata, xpredictor,
  metric = c(1,2,4),
  cutoff = 0.6,
  blocs = 7000,
  num_perms = 10,
  comparisons = list(c("A","D"), c("A","B"), c("B","D")),
  perm_mode = "columns")

## End(Not run)
```

# Index