

# Package ‘dc3net’

March 28, 2017

**Type** Package

**Title** Inferring Condition-Specific Networks via Differential Network Inference

**Version** 1.2.0

**Date** 2017-03-27

**Author** Gokmen Altay

**Maintainer** Gokmen Altay <altaylabs@gmail.com>

**Description** Performs differential network analysis to infer disease specific gene networks.

**License** GPL (>= 3)

**Depends** R (>= 3.3.3), c3net, igraph, RedeR

**Repository** CRAN

**Repository/R-Forge/Project** dc3net

**NeedsCompilation** no

**Date/Publication** 2017-03-27 22:20:23

## R topics documented:

dc3net . . . . .	2
dc3netcom . . . . .	5
dc3netdif . . . . .	6
geneNames . . . . .	7
maxofMIM . . . . .	7
normalData . . . . .	8
orderList . . . . .	9
probSameFilt . . . . .	10
rankList . . . . .	11
tumorData . . . . .	12
uniqNet . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

dc3net

*Differential C3NET***Description**

The dc3net is an R package that infers direct physical interactions of differential gene networks from gene expression datasets of multiple conditions.

**Usage**

```
dc3net(dataT = c(), dataC = c(), probes = c(), genes = c(), method = "cutoff",
methodValue = 0, itNum = 1, rankDif = 100, percentDif = 0.6, rankdCom = 10,
percentCom = 0.85, probFiltered = FALSE, visualization = 1)
```

**Arguments**

dataT	Raw microarray gene expression data set or mutual information matrix, e.g. Tumor dataset or tumor mim.
dataC	Raw microarray gene expression data set or mutual information matrix, e.g. Normal dataset or normal mim.
probes	A vector of probe names. The size of the vector should be equal to the number of rows in your data sets.
genes	A vector of gene names. The size of the vector should be equal to the number of rows in your data sets.
method	Significance test for elimination of nonsignificant edges. The available options for method parameter are "cutoff", "justp", "rank", "holm", "hochberg", "hommel", "bonferroni", "BH" and "BY". The last six parameter, starting from holm are multiple testing correction methods. methodValue and itnum parameters are dependent to method parameter.
methodValue	If the method is "cutoff", methodValue can be zero or predefined cut-off value. Zero means that mean of upper triangle will be taken as cutoff. If the method is "justp", methodValue and itnum (iteration number) parameters are need to be adjusted. For the "justp" method, methodValue corresponds to alpha value, e.g. 0.05. If the method is "rank", methodValue is the rank number. Rank corresponds to the number of interactions that will be taken as significant starting from the highest MI value. If the selected method is one of the MTC method, then methodValue and itnum parameters are need to be adjusted as it were in "justp" method.
itNum	Iteration number for "justp" and MTC methods.
rankDif	Predefined rank cut-off parameter for inferring differential network.
percentDif	Predefined rate parameter for inferring differential network.
rankdCom	Predefined rank cut-off parameter for inferring common network.
percentCom	Predefined rate parameter for inferring common network.

probFiltered	If probes of the same gene are filtered or not. Setting this option FALSE eliminates the interactions between the probes of the same genes.
visualization	Enables plotting the inferred networks. The visualization parameter takes three values, "0" for disabling the plot, "1" for plotting the differential network and "2" for plotting the common network.

## Details

The required inputs of the package are two different gene expression data sets, probe names and gene names. Users can also use pre-computed test and control mutual information (adjacency) matrices as input. Otherwise, the algorithm takes the two data sets and generates the matrices itself. The data sets need to be normalized together (e.g. using RMA) before using in dc3net. If the input data sets are precomputed mutual information matrices, then the algorithm skip this preprocessing step. The MI matrices are square adjacency matrices where the MI value corresponds to the weight of interaction for each gene pair. The diagonals are set to zero to ignore self-interactions. The next step is computing row wise ranked versions of these MI matrices in descending order. Here, rank 1 corresponds to the highest mutual information value in a row of the matrix. This ranked matrices will be used in comparing and filtering the networks at the comparison step. Then C3NET is applied to the test and control MI matrices to infer gene networks of direct physical interactions of test and control datasets independently.

In the comparison step of DC3NET, there are four conditions that all must be validated at the same time for an edge to be included in test difnet. Suppose that we check the potential interaction geneA to geneB to be included in difnet or not. As we stated above, we have been computed row wise ranked versions of the MI matrices in descending order. So we know the rank of interaction geneA to geneB in control MI matrix. The first parameter of DC3NET, rankdif, is the predefined cutoff parameter that checks the interaction between geneA and geneB is one of the top ranked interactions in control MI matrix or not. If the rank of geneA and geneB in the ranked control MI matrix is greater than the predefined cutoff parameter, rankdif, then the first condition becomes valid for deciding it as a difnet interaction. rankdif parameter can be adjusted to any value between 1 and number of rows of control MI matrix. If user wants a stricter difnet, then rankdif parameter needs to be adjusted to a greater value. The second condition is the change in MI value of interaction from geneA to geneB in the control MI matrix. Here, algorithm uses MIDif value as the cutoff parameter. MIDif is defined as percentdif times the maximum MI value of the row of geneA in the control MI matrix. Default value for the percentdif parameter is 0.6. Depends on strictness of the differential network, user can increase or decrease the second cutoff parameter. The previous two conditions compared the interaction of geneA to geneB but we also need to compare the interaction of geneB to geneA. So the algorithm validates the first and second conditions also for the interaction of geneB to geneA. In this example, if four of the conditions are validated, then DC3NET infer this interaction as in test difnet and continue to perform same filtering process for all gene pairs in test network.

Lets now start to describe the way that the algorithm infers the common network. Common network can be inferred by looking for all the same interactions between test and control network. However, this is a very strict way of inferring common network. So alternatively, one may consider the ranks and MI value decreases in the other data set. More broadly, users may follow the manner of the difnet process described above but change the comparison parameter, rankdif, from greater to less and for the percentdif from less to greater. Additionally, at this time, we only look at one of the two conditions, rather than all the four conditions together, from geneA to geneB or geneB to geneA. In the package, rankdcom, and percentcom parameters correspond to rank difference and MIDif for common network. Finally, DC3NET infers test difnet and common network and save it to the

output folder of the package. If the visualization function is turned on, then the package plots the selected inferred network.

### Value

dc3net returns an environment `res` that contains the new inferred networks and mutual information matrices. Users can access to output "Differential Network" and "Common Network" by using `res$DifNet` and `res$CommonNet`, respectively. Users can also access to computed mutual information matrices of test and control data by using `res$mimT` and `res$mimC`. Thus, users can use this precomputed mutual information matrices on the next run of the package to save time.

Accessing Differential Network Table: `networks <- dc3net(tumorData, normalData, probes, geneNames) networks$DifNet`

### Author(s)

Gokmen Altay

### References

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowetz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296. Altay,G., Altay,N., Neal,D. (2013) Global assessment of network inference algorithms based on available literature of gene/protein interactions. Turkish Journal of Biology. 37:547-555. Castro MA, Wang X, Fletcher MN, Meyer KB and Markowetz F (2012). RedeR: R/Bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. Genome Biology, 13(4), pp. R29.

### See Also

[dc3netdif](#), [dc3netcom](#), [maxofMIM](#), [orderList](#), [probSameFilt](#), [rankList](#), [uniqNet](#)

### Examples

```
data(tumorData)

data(normalData)

data(geneNames)

probes <- rownames(tumorData)

networks <- dc3net(tumorData, normalData, probes, geneNames, visualization=0)

networks$DifNet ### Differential Network

networks$CommonNet ### Common Network
```

---

`dc3netcom`*Find Common Network*

---

**Description**

dc3netcom finds common network list.

**Usage**

```
dc3netcom(netTemp, vals, rankdif, percent)
```

**Arguments**

netTemp	Test network, e.g. tumor network.
vals	Statistics in the other (control) condition.
rankdif	Predefined rank cut-off parameter.
percent	Predefined rate parameter.

**Details**

dc3netcom finds differential network list.

**Value**

dc3dif returns common network data matrix. The data matrix includes differential network list with gene names, mim indexes, probe names, control indexes, control mutual information rates and control ranks.

**Author(s)**

Gokmen Altay

**References**

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowetz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

**See Also**

[dc3net](#)

---

`dc3netdif`*Find Differential Network*

---

**Description**

`dc3netdif` finds differential network list.

**Usage**

```
dc3netdif(netTemp, vals, rankdif, percent)
```

**Arguments**

<code>netTemp</code>	Test network, e.g. tumor network.
<code>vals</code>	Statistics in the other (control) condition.
<code>rankdif</code>	Predefined rank cut-off parameter.
<code>percent</code>	Predefined rate parameter.

**Details**

`dc3netdif` finds differential network list.

**Value**

`dc3netdif` returns differential network data matrix. The data matrix includes differential network list with gene names, mim indexes, probe names, control indexes, control mutual information rates and control ranks.

**Author(s)**

Gokmen Altay

**References**

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowetz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

**See Also**

[dc3net](#)

---

geneNames

*Gene Names*

---

### Description

A vector of gene names. The size of the vector should be equal to the number of rows in your data sets.

### Usage

```
data("geneNames")
```

### Format

A data frame with 0 observations on the following 2 variables.

x a character vector

y a numeric vector

### Source

<http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi>

### References

Singh,D., Febbo,P.G., Ross,K., et al. (2002) Gene expression correlates of clinical prostatecancer behavior. Cancer cell. 1:203-9.

### Examples

```
data(geneNames)
```

---

maxofMIM

*Find Max Partner For Each Gene*

---

### Description

maxofMIM takes the mutual information matrix as input and find max partner gene with highest mutual information value for each gene.

### Usage

```
maxofMIM(mim)
```

**Arguments**

mim                    A symmetric square mutual information matrix, where the elements (i,j) correspond to the mutual information  $I(i, j)$  or  $I(j, i)$  between variables i and j.

**Details**

maxofMIM scans each row of the mutual information matrix and find the column with the highest mutual information value for each row. Thus, it finds max partner gene with highest mutual information value for each gene.

**Value**

maxofMIM returns a data matrix that rows are probe/gene names and columns are mutual information values, row numbers of each gene and row numbers of partner gene with highest mutual information value, respectively.

**Author(s)**

Gokmen Altay

**References**

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowitz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

**See Also**

[dc3net](#), [orderList](#),

---

normalData

*Example Prostate Cancer Normal Gene Expression Dataset*

---

**Description**

Raw microarray gene expression data set. This data is randomly selected 500 gene subset of the original prostate cancer data set taken from Broad Institute.

**Usage**

```
data("normalData")
```

**Details**

The example data set is also given to provide a sample data set so that one can take it as reference while formatting the new data set as entry to dc3net. Data sets must be complete and no empty or non-numerical character allowed.



**Source**

This data was taken from <http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi>

**References**

Singh,D., Febbo,P.G., Ross,K., et al. (2002) Gene expression correlates of clinical prostatecancer behavior. Cancer cell. 1:203-9.

**Examples**

```
data(normalData)
```

---

orderList	<i>Order The List of Interactions</i>
-----------	---------------------------------------

---

**Description**

orderList takes the data matrix generated by maxofMIM function and sort it according to mutual information values in increasing or decreasing order.

**Usage**

```
orderList(mylist, colindx=1, decreasing=TRUE)
```

**Arguments**

mylist	The data matrix generated by maxofMIM function that rows are probe/gene names and columns are mutual information values, row numbers of each gene and row numbers of partner gene with highest mutual information value, respectively.
colindx	The number of column with mutual information values
decreasing	The order of sort. If the argument is set to FALSE, then the data matrix will be sort in increasing order.

**Details**

orderList takes the data matrix generated by maxofMIM function and sort it according to the column number stated in colindx argument. The direction of order is set by decreasing argument.

**Value**

orderList returns a sorted data matrix that rows are probe/gene names and columns are mutual information values, row numbers of each gene and row numbers of partner gene with highest mutual information value, respectively.

**Author(s)**

Gokmen Altay

**References**

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowitz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

**See Also**

[dc3net](#), [maxofMIM](#),

---

probSameFilt

*Filter Probes of The Same Gene*

---

**Description**

probSameFilt takes the mutual information matrix as input and set the mutual information values of same genes to zero. Thus, it eliminates the interactions between the probes of the same gene.

**Usage**

```
probSameFilt(mim, genes)
```

**Arguments**

mim	A symmetric square mutual information matrix, where the elements $(i,j)$ correspond to the mutual information $I(i,j)$ or $I(j,i)$ between variables $i$ and $j$ .
genes	A vector of gene names

**Details**

The use of probSameFilt is very important when working on probe level. If this function is not enabled, then the interactions between probes of the same gene, which should not be taken into account, will be included in differential or common network computation. This may result in difnet or commonnet tables which includes same gene interactions e.g. GPR50-GPR50.

**Value**

probSameFilt returns mutual information matrix that the interaction scores between probes of the same gene is updated to 0.

**Author(s)**

Gokmen Altay

## References

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowitz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

## See Also

[dc3net](#),

## Examples

```
data(tumorData)

data(geneNames)

tumorData <- copula(tumorData)

tumorData <- makemim(tumorData)

probSameFilt(tumorData, geneNames)
```

---

rankList

*Find Rank List*

---

## Description

rankList finds the rank of test list in the mutual information matrix of control to see any rank difference using mutual information matrix indexes.

## Usage

```
rankList(topTest, mimCont)
```

## Arguments

topTest	A data matrix that has columns 4 and 5 have mim indexes
mimCont	A mutual information matrix of control dataset

## Details

rankList takes mim indexes from 4th and 5th columns of the data matrix and finds the rank of test list in the mutual information matrix of control to see any rank difference.

## Value

rankList returns a data matrix that has ranks of test list in the control MI matrix

**Author(s)**

Gokmen Altay

**References**

Altay, G., Asim, M., Markowetz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

**See Also**

[dc3net](#), [maxofMIM](#),

---

tumorData

*Example Prostate Cancer Tumor Gene Expression Dataset*

---

**Description**

Raw microarray gene expression data set. This data is randomly selected 500 gene subset of the original prostate cancer data set taken from Broad Institute.

**Usage**

```
data("tumorData")
```

**Details**

The example data set is also given to provide a sample data set so that one can take it as reference while formatting the new data set as entry to dc3net. Data sets must be complete and no empty or non-numerical character allowed.

**Source**

This data was taken from <http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi>

**References**

Singh,D., Febbo,P.G., Ross,K., et al. (2002) Gene expression correlates of clinical prostatecancer behavior. Cancer cell. 1:203-9.

**Examples**

```
data(tumorData)
```

---

`unIQNet`*Unique Network*

---

**Description**

`unIQNet` makes the differential or common network list unique.

**Usage**

```
unIQNet(tops = c())
```

**Arguments**

`tops`                    A data matrix that contains list of interactions.

**Details**

`unIQNet` removes the redundant interactions in the generated differential or common network. For ex. it is possible that the generated network can include both geneA-to-geneB and geneB-to-geneA interactions. `unIQNet` eliminates one of this interaction so the list of final network is unique.

**Value**

`unIQNet` returns a unique data matrix that includes network list with gene names, mim indexes, probe names, control indexes, control mutual information rates and control ranks.

**Author(s)**

Gokmen Altay

**References**

Altay, G. et. al. "Genome-wide differential gene network analysis software and an application in LnCap prostate cancer", bioRxiv, 2017. Altay, G., Asim, M., Markowitz, F., Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics. 12: 296.

**See Also**

[dc3net](#), [maxofMIM](#), [orderList](#), [rankList](#), [dc3netdif](#), [dc3netcom](#)

# Index

## \*Topic **datasets**

geneNames, [7](#)  
normalData, [8](#)  
tumorData, [12](#)

## \*Topic **misc**

dc3net, [2](#)  
dc3netcom, [5](#)  
dc3netdif, [6](#)  
maxofMIM, [7](#)  
orderList, [9](#)  
probSameFilt, [10](#)  
rankList, [11](#)  
uniqNet, [13](#)

dc3net, [2](#), [5](#), [6](#), [8](#), [10–13](#)

dc3netcom, [4](#), [5](#), [13](#)

dc3netdif, [4](#), [6](#), [13](#)

geneNames, [7](#)

maxofMIM, [4](#), [7](#), [10](#), [12](#), [13](#)

normalData, [8](#)

orderList, [4](#), [8](#), [9](#), [13](#)

probSameFilt, [4](#), [10](#)

rankList, [4](#), [11](#), [13](#)

tumorData, [12](#)

uniqNet, [4](#), [13](#)