Using the DCGL Package

Bao-Hong Liu[1,3] and Hui Yu[2,3]

August 17, 2010

[1]School of Life Science and Technology, Tongji University. Shanghai 200092, P.R. China.
[2]Key Laboratory of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences. Shanghai 200031, P.R. China.
[3]Shanghai Center for Bioinformation Technology,Shanghai 200035, P.R. China.

bhliu@scbit.org and yuhui@scbit.org

Contents

1 Introduction

This document gives instructions on how to use the functions in the package *DCGL*. *DCGL* is a free R package assisting differential coexpression analysis (DCEA), specifically, for identifying Differentially Coexpressed Genes (DCGs) and Differentially Coexpressed Links (DCLs) from gene expression microarray data. In the package five DCEA methods are implemented, including two proposed by us (`DCp` and `DCe`) and three other ones (`WGCNA`,`LRC`,`ASC`). The major input of *DCGL* are the expression data from two contrastive conditions with rows representing genes and columns representing microarrays. The output of *DCGL* always include a talbe of the DCGs mined from the input expression data. Note that DCe outputs DCLs as an additional result.

We also provide functions `expressionBasedfilter` and `varianceBasedfilter` to filter genes in the expression dataset and `systematicLinkfilter`, `qLinkfilter` and `percentLinkfilter` to filter gene coexpression links in coexpression networks.

The *DCGL* package employs R library *igraph* which must be installed in advance.

2 Getting started

Prior to using *DCGL*, users should download the installation file of *DCGL* to their local computer, and install *DCGL* as a package of their R computing environment. For linux users, they should type "R CMD INSTALL DCGL_1.0.tar.gz" in the shell (suppose the installation file "DCGL_1.0.tar.gz" is in the current working directory); for windows users, they should go to the R menu "Packages" and click the "Install package(s) from local zip files" and then locate the local file "DCGL_1.0.zip". If the package is installed successfully, a file folder named "DCGL" would appear beneath the folder "library" in the R installation directory.

To load the *DCGL* package, type `library(DCGL)`.

3 Methods

DCGL provides facilities for gene filtering, link filtering and DCGs/DCLs identification.

3.1 Gene Filtering

If there are too many genes in the expression dataset, one can filter out some genes using the two functions described below. Before gene filtering, the data from the two conditions should be combined into one gene expression matrix.

3.1.1 Expression based gene filtering

`expressionBasedfilter`: Genes which have a Between-Experiment Mean Expression Signal (BEMES) lower than the median of BEMES's of all genes will be filtered out (Prieto and etal.,2008). That is, half genes in the dataset will be filtered out after this filtering step.

3.1.2 Variability based gene filtering

`varianceBasedfilter`: This is an approximate test of the hypothesis that gene has the same variance as the median variance (Simon and Lam,2006). The variance of the log-values for each gene is compared to the median of all the variances. The *quantity*

$$quantity = (n-1) * var_i / var_m$$

for each gene is compared to a percentile of the chi-square distribution with n-1 degrees of freedom (n is the number of columns of the input gene-expression matrix or the sample size), where var_i is the variance of the i_{th} gene and var_m is the median of these gene-specific variances. Those genes not significantly more variable than the median gene are filtered out.

3.2 Link filtering

For all DCEA methods but WGCNA, a link filtering step is necessary to build up two gene coexpression networks for the two contrastive conditions. One can imagine two gene coexpression networks having identical linking structures but different edge weights (co-expression values). The input to link filtering methods always include two separate gene expression matrices for the two conditions, and the output often takes the form of two gene-versus-gene coexpression matrices. In the coexpression matrices, retained links have non-zero values while discarded links are denoted with zero values.

Three stand-alone functions are implemented for link filtering, which are the systematic investigation of the relationship between the correlation threshold and the clustering coefficient (`systematicLinkfilter`), the q-value based link filtering (`qLinkfilter`), and the correlation-value fraction based link filtering (`percentLinkfilter`). However, these link filtering functions are seldomly called as independent functions; instead, they are wrapped in the DCEA functions `DCp`, `DCe`, `ASC`, `LRC`, and can be determined through setting the "method" and "cutoff" parameters.

The first method is feasible for small datasets (for example, with a gene total less than 500); the second one is suitable for medium datasets (for example, with a gene total less than 5000); and the last one is recommended for the largest datasets.

3.2.1 A systematic procedure for estimating a cutoff threshold of coexpression networks

`systematicLinkfilter`: This is a systematic procedure for inferring a cutoff threshold of coexpression networks directly from their topological properties (Elo and etal.,2007). The objective is to automatically select a threshold that preserves as many valid coexpression links as possible, while simultaneously controlling the number of false detections. The

procedure is based on comparing the observed clustering coefficient and its randomized counterpart as the number of connections is gradually decreased.

3.2.2 Filtering gene coexpression links according to the q-values of expression correlation values

`qLinkfilter`: For each of the two conditions, the coexpression values are associated with the corresponding p-values (student T-test of the zero nature of a PCC), and these p-values are sorted and transformed to q-values (or formally, false discovery rates). Gene links with q-values of coexpression values in either of two conditions lower than the cutoff (qth) are retained.

3.2.3 Filtering gene coexpression links according to the max expression correlation value

`percentLinkfilter`: Each gene link is associated with two correlation values (one out of condition A and the other out of condition B) and thus a list of 'maximum absolute values' for all correlation value pairs is decided. Then these 'maximum absolute values' are sorted in decreasing order. At last, a fraction of gene pairs with the highest max correlation values will be retained.

3.3 DCp for identifying DCGs

`DCp` necessitates a link filtering step before its main analysis, which can be specified through setting the "method" and "cutoff" parameters. After the link filtering, coexpression pairs with q-values/r-values of coexpression values in either of two conditions lower/higher than the cutoff (qth/rth) are retained.

DCp works on the filtered set of gene coexpression value pairs, where each pair is made up with two coexpression values of a gene pair calculated under two different conditions. The subset of coexpression value pairs associated with a particular gene can be written as two vectors X and Y (n is the length of the vector or the coexpression neighbors for a gene).

$$X = (x_{i1}, x_{i2}, ..., x_{in})$$

$$Y = (y_{i1}, y_{i2}, ..., y_{in})$$

Then a length normalized Euclidean distance is used for measuring differential coexpression (dC) of this gene.

$$dC_n(i) = \sqrt{\frac{(x_{i1} - y_{i1})^2 + (x_{i2} - y_{i2})^2 + ... + (x_{in} - y_{in})^2}{n}}$$

To evaluate whether a gene has significant dC, we perform a permutation test, in which we randomly permute the disease and normal conditions of the samples, calculate new PCCs, filter gene pairs based on the new PCCs, and calculate new dC statistics. The sample permutation is repeated (N) times, and a large number of permutation dC statistics form an empirical null distribution. The p-value for each gene can then be estimated.

3.4 DCe for identifying DCGs and DCLs

`DCe` is based on the 'Limit Fold Change' (LFC) model, a robust statistical method originally proposed for selecting DEGs from microarray data (Mutch and etal.,2002).

First, the correlation pairs are divided into three parts according to the pairing of signs of coexpression values and the multitude of coexpression values: pairs with same signs (N_1), pairs with different signs (N_2) and pairs with differently-signed high coexpression values (N_3). The first two parts are processed with the 'LFC' model separately to yield two subsets of DCLs (K_1,K_2), while the third part (N_3)adds to the set of DCLs directly. So a total of K=N_3+K_1+K_2 DCLs are determined from a total of N gene links. For a gene (g_i), the total number of links (n_i) and DCLs in particular(k_i) associated with it are counted, and the Binomial Probability model is used to estimate the significance of the gene being a DCG.

$$P(g_i) = \sum_{x=k_i}^{n_i} C_{n_i}^{x} (\frac{K}{N})^x (1 - \frac{K}{N})^{n_i - x}$$

3.5 WGCNA for identifying DCGs

`WGCNA` adopts the soft thresholding method to construct coexpression network of all possible gene pairs (Fuller and etal.,2007; van Nas and etal.,2009). The nodes of such a network correspond to genes, and edges between genes are tagged with values 'softly thresholded' from the Pearson correlation coefficients. By raising the absolute value of the Pearson correlation to a power

$$\beta \geq 1$$

the soft thresholding strategy emphasizes large correlations at the expense of low correlations. Specifically,

$$a_{ij} = |\frac{1 + cor(x_i, y_i)}{2}|^{\beta}$$

represents the edge weights of the coexpression networks (or transformed coexpression values). For a specific gene, there are two vectors of transformed coexpression values with a length of m-1 (m is the number of genes).

$$a_1 = (a_{11}, a_{21}, ..., a_{m-1,1})$$

$$a_2 = (a_{12}, a_{22}, ..., a_{m-1,2})$$

For the i_{th} gene, $k_1(i)$ and $k_2(i)$ denote the whole-network connectivity in networks 1 and 2 (sum of the a/b values), respectively.

$$k_1(i) = a_{11} + a_{21} + ... + a_{m-1,1}$$

$$k_2(i) = a_{12} + a_{22} + ... + a_{m-1,2}$$

To achieve a more fair comparison of connectivities, the authors in the original paper divide each gene connectivity with the maximum connectivity, i.e.,

$$K_1(i) = \frac{k_1(i)}{max(k_1)}$$

and

$$K_2(i) = \frac{k_2(i)}{max(k_2)}$$

Finally a measure of differential connectivity is

$$WGCNA_i = |K_1(i) - K_2(i)|$$

Described above is the original version of the WGCNA method. We also include another variant of the WGCNA method by defining the ultimate differential coexpression measure using the length-normalized Euclidean distance:

$$WGCNA_i = \sqrt{\frac{(a_{11} - a_{12})^2 + (a_{21} - a_{22})^2 + ... + (a_{m-1,1} - a_{m-1,2})^2}{m-1}}$$

The higher the WGCNA score is, the more likely a gene is a DCG.

3.6 ASC for identifying DCGs

`ASC` is abbreviated from 'Average Specific Connection' (Choi and etal.,2005). The ASC method employs the 'hard thresholding' strategy to construct the coexpression networks. I.e., coexpression values lower than the cut-off are converted to zero, while those higher are converted to one. ASC pays special attention to the 'specific connections', or links existing in only one coexpression network. The specific connections for a gene in networks 1 and 2 total (SC_{i1} and SC_{i2}), respectively.

$$ASC_i = \frac{SC_{i1} + SC_{i2}}{2}$$

The higher the ASC score is, the more likely a gene is a DCG.

3.7 LRC for identifying DCGs

`LRC` is abbreviated from 'Log Ratio Connections' (Reverter and etal.,2005). The LRC method also employs the 'hard thresholding' strategy to construct the coexpression networks. With degrees of a same gene counted in the two networks ($degree_{i1}$ and $degree_{i2}$), we have

$$LRC_i = |\log_{10} \frac{degree_{i2}}{degree_{i1}}|$$

The higher the LRC score is, the more likely a gene is a DCG.

4 Dataset

DCGL includes three simulated datasets, each having a total of 1000 genes. These expression data were simulated by software SynTReN (Bulcke and etal.,2006) using networks originated from E.coli regulatory network. To simulate datasets, the underlying networks for condition one were fixed, but they had 10% links perturbed for condition two. Specifically, those links were eliminated to naught (dataset A), toggled to the opposite (dataset B), or eliminated and toggled half-and-half (dataset C).

5 Example

The following examples are based on the test dataset C (dataC).

5.1 link filtering

```
> library(DCGL)
> data(dataC)
> dataC[1:3, 1:3]

          Sample1   Sample2   Sample3
EG10006 0.4533798 0.3461948 0.1350374
EG10007 0.6430745 0.9126034 0.6241110
EG10008 0.6570491 0.9214012 0.5319754
```

The first ten samples (Sample1 to Sample10) belong to one condition and the remaining ten samples belong to another condition. So we firstly divide dataC to two parts corresponding to condition A (exprs.1) and condition B (exprs.2) respectively.

```
> exprs.1 <- dataC[, 1:10]
> exprs.2 <- dataC[, 11:20]
```

The number of genes in dataC is moderate (1000), so the most time-consuming link filtration function (`qLinkfilter`) can be run. Through link filtering, correlation pairs with q values lower than qth(0.25) in either condition are kept whereas the other pairs are set to 0. Links$rth contain the two correlation thresholds for both conditions; Links$cor.filtered$cor.filtered.1 and Links$cor.filtered$cor.filtered.2 keep the filtered correlation matrices for condition A and B.

```
> Links <- qLinkfilter(exprs.1, exprs.2, 0.25)
> Links$rth

$rth.1
[1] 0.5864986

$rth.2
[1] 0.6609703

> Links$cor.filtered$cor.filtered.1[1:5, 1:5]

          EG10006   EG10007   EG10008   EG10012   EG10020
EG10006 0.0000000 0.0000000 0.0000000 0.0000000 0.6280721
EG10007 0.0000000 0.0000000 0.9779899 0.0000000 0.0000000
EG10008 0.0000000 0.9779899 0.0000000 0.3272847 0.5899156
EG10012 0.0000000 0.0000000 0.3272847 0.0000000 0.0000000
EG10020 0.6280721 0.0000000 0.5899156 0.0000000 0.0000000

> Links$cor.filtered$cor.filtered.2[1:5, 1:5]
```

```
          EG10006   EG10007     EG10008   EG10012     EG10020
EG10006 0.0000000 0.0000000  0.00000000  0.000000  0.16323178
EG10007 0.0000000 0.0000000  0.68937477  0.000000  0.00000000
EG10008 0.0000000 0.6893748  0.00000000 -0.733558 -0.05414101
EG10012 0.0000000 0.0000000 -0.73355799  0.000000  0.00000000
EG10020 0.1632318 0.0000000 -0.05414101  0.000000  0.00000000
```

With a sufficiently small dataset, one can try the systematic investigation of the relationship between correlation threshold and clustering coefficient. Here we demonstrate it on the top 100 genes of dataC as an example.

```
> exprs <- dataC[1:100, ]
> C_r <- systematicLinkfilter(exprs)
```

The function outputs a table of 'correlation threshold' vs. 'clustering coefficient'.

```
> plot(C_r[, 1], C_r[, 2], xlab = "correlation threshold", ylab = "C-C0")
> lines(C_r[, 1], C_r[, 2])
```

A curve of 'correlation threshold' vs. 'clustering coefficient' is plotted, which may assist the users to determine correlation threshold (Figure 1). Further details are to be found in the original paper (Elo and etal.,2007).

5.2 DCp for identifying DCGs

For the 'Differential Coexpression Profile' (DCp) method, there are three choices for the wrapped link-filtering algorithms, which can be specified by setting the parameter 'method' as 'qth','rth', or 'percent'. The parameter 'cutoff' also needs to be specified, which may have different explanation in light of different link-filtering 'method'.

The following shows the usage of the 'qth' version of DCp method.

```
> Result <- DCp(exprs.1, exprs.2, method = "qth", cutoff = 0.25,
+     N = 0)
> Result[1:10, ]
```

```
               dC length
EG10006 0.5907748    394
EG10007 0.5089185    414
EG10008 0.6662489    382
EG10012 0.7727753    178
EG10020 0.6651723    608
EG10022 0.8111587     21
EG10023 0.7727377    105
EG10024 0.7094249    226
EG10025 0.5901216    196
EG10026 0.9735308    205
```

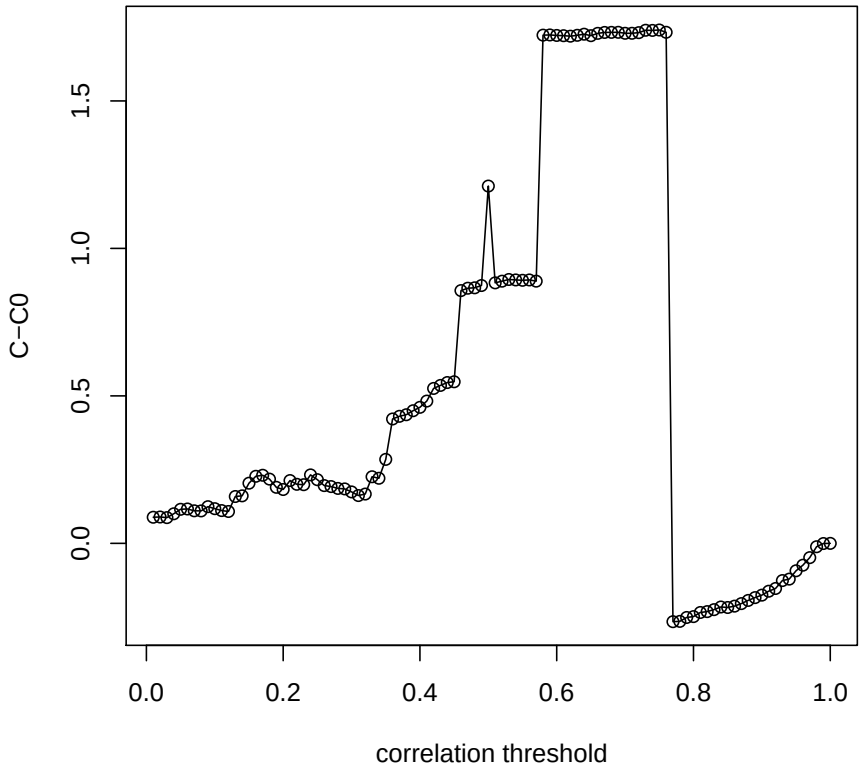

Figure 1: Behavior of the clustering coefficient at different correlation thresholds.

'Result' is a table of DCGs, in which the first column gives the dC value for every gene, the second column gives the length of its 'Differential Coexpression Profiles' (or, 'degree' in the coexpression networks).

At this demonstration we set N=0, specifying that no permutation is incurred. If N>0, the permutation step will be repeated N times for estimating p-values. Correspondingly, 'Result' will be extended to four columns of 'dC' value, 'profile length', 'p value' and 'FWER'.

5.3 DCe for identifying DCGs and DCLs

For the method DCe, you can also choose one of three link filtering options. The following is exemplified with the 'qth' option.

```
> Results <- DCe(exprs.1, exprs.2, method = "qth", cutoff = 0.25,
+     nbins = 20, p = 0.1, figname = c("LFC.s.jpeg", "LFC.d.jpeg"))
> Results$DCGs[1:10, ]

        All.links DC.links DCL_same DCL_diff DCL_switch             p
EG13414       588      338       37       62        239 3.145296e-139
EG10170       595      331       21       60        250 1.218457e-130
EG10624       583      327       16       67        244 1.711025e-130
EG10350       603      333       20       60        253 3.792867e-130
EG13736       594      330       31       59        240 5.082115e-130
EG10398       602      331       31       56        244 1.250686e-128
EG10441       573      319       35       52        232 4.239250e-126
EG11540       579      318       15       62        241 1.802350e-123
EG30070       599      323       27       64        232 2.415852e-122
EG12541       598      318       22       51        245 3.060324e-118
                     q
EG13414 3.145296e-136
EG10170 1.218457e-127
EG10624 1.711025e-127
EG10350 3.792867e-127
EG13736 5.082115e-127
EG10398 1.250686e-125
EG10441 4.239250e-123
EG11540 1.802350e-120
EG30070 2.415852e-119
EG12541 3.060324e-115

> Results$DCL.same[1:10, ]

    Gene.1  Gene.2       cor.1        cor.2
1  EG10024 EG10012  0.81617035  0.045106980
2  EG10026 EG10023  0.05169217  0.868128384
3  EG10026 EG10025 -0.79035351 -0.022894525
4  EG10052 EG10008  0.60671292  0.011411682
```

```
5  EG10055 EG10008  0.64599310  0.009903623
6  EG10082 EG10054  0.04237940  0.730497994
7  EG10082 EG10059  0.03397799  0.717081246
8  EG10082 EG10072  0.64274138  0.033586636
9  EG10095 EG10044  0.06950081  0.868045498
10 EG10109 EG10007  0.62771284  0.006943263
```

```
> Results$DCL.diff[1:10, ]
```

```
    Gene.1  Gene.2       cor.1       cor.2
1  EG10049 EG10026 -0.89472063  0.41455106
2  EG10061 EG10026 -0.87645970  0.58490389
3  EG10079 EG10078  0.97457478 -0.40227097
4  EG10110 EG10109 -0.29427998  0.96955919
5  EG10111 EG10026  0.89654124 -0.60581126
6  EG10112 EG10061  0.50386949 -0.96183454
7  EG10112 EG10111 -0.38284091  0.95820855
8  EG10149 EG10044  0.03810221 -0.93670079
9  EG10152 EG10133  0.92601219 -0.34645884
10 EG10153 EG10012  0.96124730 -0.02015280
```

```
> Results$DCL.switched[1:10, ]
```

```
      Gene.1  Gene.2      cor.1      cor.2
822  EG10719 EG10714  0.9992682 -0.9964497
820  EG10716 EG10714  0.8348981 -0.9991135
815  EG10714 EG10490  0.9990988 -0.9903306
339  EG10466 EG10350  0.9987030 -0.9861219
125  EG10350 EG10349  0.9986616 -0.9922408
2479 EG11886 EG10047  0.9629320 -0.9985700
1843 EG11401 EG10714  0.9985022 -0.9933157
467  EG10539 EG10500 -0.9864642  0.9984770
4908 EG13965 EG10624 -0.9172892  0.9984616
2158 EG11565 EG10170 -0.9186945  0.9980134
```

'Result' is a list with four components, one for DCGs and another three for different types of DCLs. The DCGs table include seven columns - all links, DCLs, same signed DCLs, differently signed DCLs, switched links, the p value, and the FWER value.

5.4 Narrowing down preferential DCLs

DCe is recommended the optimal method for DCL identification because this method can select DCLs more accurately. In reality, biologists usually welcome a smaller set of DCLs than a larger one. So narrowing down preferential DCLs is necessary. This can be achieved by setting higher coexpression value cut-offs (qth or rth) or stricter outlier fractions (p) of the LFC model.

For switched links:

```
> Results$DCL.switched[1:10, ]
```

```
      Gene.1  Gene.2      cor.1      cor.2
822  EG10719 EG10714  0.9992682 -0.9964497
820  EG10716 EG10714  0.8348981 -0.9991135
815  EG10714 EG10490  0.9990988 -0.9903306
339  EG10466 EG10350  0.9987030 -0.9861219
125  EG10350 EG10349  0.9986616 -0.9922408
2479 EG11886 EG10047  0.9629320 -0.9985700
1843 EG11401 EG10714  0.9985022 -0.9933157
467  EG10539 EG10500 -0.9864642  0.9984770
4908 EG13965 EG10624 -0.9172892  0.9984616
2158 EG11565 EG10170 -0.9186945  0.9980134
```

The DCLs can be selected by sorting max absolute correlation of a link. Because it was proved that the DCLs with highest max absolute correlation were more accurate.

For same signed and differently signed DCLs, one effective method is to lower the outlier fractions (p) of the LFC model. For example, when the p=0.1, the result is:

```
> Results <- DCe(exprs.1, exprs.2, method = "qth", cutoff = 0.25,
+     nbins = 20, p = 0.1, figname = c("LFC.s.jpeg", "LFC.d.jpeg"))
> Results$DCGs[1:10, ]
```

```
        All.links DC.links DCL_same DCL_diff DCL_switch             p
EG13414       588      338       37       62        239 3.145296e-139
EG10170       595      331       21       60        250 1.218457e-130
EG10624       583      327       16       67        244 1.711025e-130
EG10350       603      333       20       60        253 3.792867e-130
EG13736       594      330       31       59        240 5.082115e-130
EG10398       602      331       31       56        244 1.250686e-128
EG10441       573      319       35       52        232 4.239250e-126
EG11540       579      318       15       62        241 1.802350e-123
EG30070       599      323       27       64        232 2.415852e-122
EG12541       598      318       22       51        245 3.060324e-118
                    q
EG13414 3.145296e-136
EG10170 1.218457e-127
EG10624 1.711025e-127
EG10350 3.792867e-127
EG13736 5.082115e-127
EG10398 1.250686e-125
EG10441 4.239250e-123
EG11540 1.802350e-120
EG30070 2.415852e-119
EG12541 3.060324e-115
```

when the p=0.01, the result is:

```
> Results <- DCe(exprs.1, exprs.2, method = "qth", cutoff = 0.25,
+     nbins = 20, p = 0.01, figname = c("LFC.s.jpeg", "LFC.d.jpeg"))
> Results$DCGs[1:10, ]

        All.links DC.links DCL_same DCL_diff DCL_switch             p
EG10170       595      260        3        7        250 6.244839e-171
EG13414       588      258        6       13        239 3.495622e-170
EG10350       603      259        1        5        253 5.643703e-168
EG11031       591      256        2        5        249 3.610319e-167
EG10398       602      257        3       10        244 6.904445e-166
EG10441       573      251        7       12        232 2.115086e-165
EG10624       583      252        3        5        244 2.601132e-164
EG13736       594      254        2       12        240 3.459804e-164
EG12541       598      253        1        7        245 3.635239e-162
EG11540       579      249        0        8        241 1.005535e-161
                    q
EG10170 6.244839e-168
EG13414 3.495622e-167
EG10350 5.643703e-165
EG11031 3.610319e-164
EG10398 6.904445e-163
EG10441 2.115086e-162
EG10624 2.601132e-161
EG13736 3.459804e-161
EG12541 3.635239e-159
EG11540 1.005535e-158
```

We can see that the numbers of same and different signed DCLs decreased.

5.5 WGCNA, ASC and LRC for identifying DCGs

Users can also identify DCGs using the method of `WGCNA`,`ASC` and `LRC`.

The method WGCNA does not need the link filtering step; but it needs a new parameter 'power' which is necessary in the 'soft thresholding' (or transformation) of the original correlation values. There are two variants for the WGCNA method: the first one, a combination of the two original papers (Fuller and etal.,2007; van Nas and etal.,2009), and the second one with integration from the 'DCp' method. Here we use power=12 as an example.

```
> Results <- WGCNA(exprs.1, exprs.2, power = 12, variant = "WGCNA")
> Results[1:10]

    EG10006     EG10007     EG10008     EG10012     EG10020     EG10022
0.130436661 0.002313100 0.070444785 0.003444917 0.021990313 0.012220093
    EG10023     EG10024     EG10025     EG10026
0.009754864 0.058267889 0.139482501 0.003448272
```

The other two methods 'ASC' and 'LRC' also requires setting the link-filtering 'method', but they exert a hard thresholding strategy and basically compares the numbers of connected edges.

```
> Results <- ASC(exprs.1, exprs.2, method = "qth", cutoff = 0.25)
> Results[1:10]

EG10006 EG10007 EG10008 EG10012 EG10020 EG10022 EG10023 EG10024 EG10025
  193.0   162.5   157.0    75.0   172.5    10.5    52.0   111.0    92.5
EG10026
   97.0

> Results <- LRC(exprs.1, exprs.2, method = "qth", cutoff = 0.25)
> Results[1:10]

    EG10006     EG10007     EG10008     EG10012     EG10020     EG10022
0.867762025 0.077915080 0.454258372 0.050654764 0.187339802 0.041392685
    EG10023     EG10024     EG10025     EG10026
0.115393419 0.106268337 0.004196115 0.414973348
```

References

[Prieto and etal.,2008] Prieto, C., Risueno, A., Fontanillo, C. and De las Rivas, J. (2008) Human gene coexpression landscape: confident network derived from tissue transcriptomic profiles. *PLoS One*,3, e3911.

[Simon and Lam,2006] Simon, R. and Lam, A. (2006) BRB Array Tools Users Guide. Technical Reports. *Biometric Research Branch, National Cancer Institute*

`http://linus.nci.nih.gov/~brb/download_full_new.html`

[Elo and etal.,2007] Elo, L. L., Jarvenpaa, H., Oresic, M., Lahesmaa, R. and Aittokallio, T. (2007) Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process. *Bioinformatics*, 23, 2096-2103.

[Mutch and etal.,2002] Mutch, D. M.,Berger, A.,Mansourian, R.,Rytz, A.,Roberts, M. A. (2002) The limit fold change model: a practical approach for selecting differentially expressed genes from microarray data. *BMC Bioinformatics*, 3, 17.

[Fuller and etal.,2007] Fuller, T.F., Ghazalpour, A., Aten, J.E., Drake, T.A., Lusis, A.J. and Horvath, S.(2007) Weighted gene coexpression network analysis strategies applied to mouse weight. *Mamm Genome*,18, 463-472.

[van Nas and etal.,2009] van Nas, A., Guhathakurta, D., Wang, S.S., Yehya, N., Horvath, S., Zhang, B., Ingram-Drake, L., Chaudhuri, G., Schadt, E.E., Drake, T.A., Arnold, A.P. and Lusis, A.J. (2009) Elucidating the role of gonadal hormones in sexually dimorphic gene coexpression networks. *Endocrinology*,150, 1235-1249.

[Choi and etal.,2005] Choi, J.K., Yu, U., Yoo, O.J. and Kim, S. (2005) Differential coexpression analysis using microarray data and its application to human cancer. *Bioinformatics*, 21, 4348-4355.

[Reverter and etal.,2005] Reverter, A., Ingham, A., Lehnert, S.A., Tan, S.H., Wang, Y., Ratnakumar, A. and Dalrymple, B.P. (2006) Simultaneous identification of differential gene expression and connectivity in inflammation, adipogenesis and cancer. *Bioinformatics*, 22, 2396-2404.

[Bulcke and etal.,2006] Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B. and Marchal, K. (2006) SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7, 43.