WGCNA Demo

Shaurya Jauhari (Email: shauryajauhari@gzhmu.edu.cn)

2019-10-29

Installing the package and setting up the options.

```
install.packages("BiocManager",
                 repos='http://cran.us.r-project.org',
                 dependencies = TRUE)
```

```
## Installing package into '/Users/mei/Library/R/3.6/library'
## (as 'lib' is unspecified)
```

```
## Warning: dependency 'BiocStyle' is not available
```

```
##
##   There is a binary version available but the source version is
##   later:
##             binary source needs_compilation
## BiocManager 1.30.7 1.30.9             FALSE
```

```
## installing the source package 'BiocManager'
```

```
BiocManager::install("WGCNA")
```

```
## Bioconductor version 3.9 (BiocManager 1.30.9), R 3.6.0 (2019-04-26)
```

```
## Installing package(s) 'WGCNA'
```

```
## Package which is only available in source form, and may need
##   compilation of C/C++/Fortran: 'WGCNA'
```

```
## installing the source package 'WGCNA'
```

```
## Warning in install.packages(...): installation of package 'WGCNA' had non-
## zero exit status
```

```
## Old packages: 'digest', 'jpeg', 'JuliaCall', 'mgcv', 'plotmo', 'rlang'
```

```
install.packages("ggdendro",
                 repos='http://cran.us.r-project.org',
                 dependencies = TRUE)
```

```
## Installing package into '/Users/mei/Library/R/3.6/library'
## (as 'lib' is unspecified)
```

```
##
## The downloaded binary packages are in
##  /var/folders/hm/c3_fjypn62v5xh5b5ygv267m0000gn/T//Rtmp59rqut/downloaded_packages
```

```
## Setting options

options(stringsAsFactors = FALSE)

library(WGCNA)
```

```
## Loading required package: dynamicTreeCut
```

```
## Loading required package: fastcluster
```

```
##
## Attaching package: 'fastcluster'

## The following object is masked from 'package:stats':
##
##     hclust

##

##
## Attaching package: 'WGCNA'

## The following object is masked from 'package:stats':
##
##     cor
```

```
library(ggdendro)
library(ggplot2)
library(cowplot)
```

```
##
## ************************************************************

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

## ************************************************************
```

Importing data files from female and male liver tissues from mice, and exploring them.

```
mydataf <- read.csv("../FemaleLiver-Data/LiverFemale3600.csv", header = TRUE)
colnames(mydataf)
```

```
##   [1] "substanceBXH"   "gene_symbol"    "LocusLinkID"    "ProteomeID"
##   [5] "cytogeneticLoc" "CHROMOSOME"     "StartPosition"  "EndPosition"
##   [9] "F2_2"           "F2_3"           "F2_14"          "F2_15"
##  [13] "F2_19"          "F2_20"          "F2_23"          "F2_24"
##  [17] "F2_26"          "F2_37"          "F2_42"          "F2_43"
##  [21] "F2_45"          "F2_46"          "F2_47"          "F2_48"
##  [25] "F2_51"          "F2_52"          "F2_54"          "F2_63"
##  [29] "F2_65"          "F2_66"          "F2_68"          "F2_69"
##  [33] "F2_70"          "F2_71"          "F2_72"          "F2_78"
##  [37] "F2_79"          "F2_80"          "F2_81"          "F2_83"
##  [41] "F2_86"          "F2_87"          "F2_88"          "F2_89"
##  [45] "F2_107"         "F2_108"         "F2_109"         "F2_110"
##  [49] "F2_111"         "F2_112"         "F2_117"         "F2_119"
##  [53] "F2_125"         "F2_126"         "F2_127"         "F2_141"
##  [57] "F2_142"         "F2_143"         "F2_144"         "F2_145"
##  [61] "F2_154"         "F2_155"         "F2_156"         "F2_157"
##  [65] "F2_162"         "F2_163"         "F2_164"         "F2_165"
##  [69] "F2_166"         "F2_167"         "F2_169"         "F2_180"
##  [73] "F2_181"         "F2_182"         "F2_187"         "F2_188"
##  [77] "F2_189"         "F2_190"         "F2_191"         "F2_192"
##  [81] "F2_194"         "F2_195"         "F2_200"         "F2_201"
##  [85] "F2_212"         "F2_213"         "F2_214"         "F2_215"
##  [89] "F2_221"         "F2_222"         "F2_223"         "F2_224"
```

```
##  [93] "F2_225"         "F2_226"         "F2_227"         "F2_228"
##  [97] "F2_241"         "F2_242"         "F2_243"         "F2_244"
## [101] "F2_245"         "F2_247"         "F2_248"         "F2_261"
## [105] "F2_263"         "F2_264"         "F2_270"         "F2_271"
## [109] "F2_272"         "F2_278"         "F2_287"         "F2_288"
## [113] "F2_289"         "F2_290"         "F2_291"         "F2_296"
## [117] "F2_298"         "F2_299"         "F2_300"         "F2_302"
## [121] "F2_303"         "F2_304"         "F2_305"         "F2_306"
## [125] "F2_307"         "F2_308"         "F2_309"         "F2_310"
## [129] "F2_311"         "F2_312"         "F2_320"         "F2_321"
## [133] "F2_323"         "F2_324"         "F2_325"         "F2_326"
## [137] "F2_327"         "F2_328"         "F2_329"         "F2_330"
## [141] "F2_332"         "F2_355"         "F2_357"
```

```
head(mydataf)
```

```
##   substanceBXH   gene_symbol LocusLinkID ProteomeID cytogeneticLoc
## 1  MMT00000044 1700007N18Rik       69339     286025              0
## 2  MMT00000046         Mast2       17776     157466              0
## 3  MMT00000051       Ankrd32      105377     321939              0
## 4  MMT00000076             0      383154          0              0
## 5  MMT00000080          Ldb2       16826     157383              0
## 6  MMT00000102          Rdhs      216453          0     10_70.0_cM
##   CHROMOSOME StartPosition EndPosition     F2_2    F2_3     F2_14    F2_15
## 1         16      50911260    50912491 -0.01810  0.0642  6.44e-05 -0.05800
## 2          4     115215318   115372404 -0.07730 -0.0297  1.12e-01 -0.05890
## 3         13      74940309    74982847 -0.02260  0.0617 -1.29e-01  0.08710
## 4         16      49345114    49477048 -0.00924 -0.1450  2.87e-02 -0.04390
## 5          5      43546124    43613704 -0.04870  0.0582 -4.83e-02 -0.03710
## 6         10       1337265     1347607  0.17600 -0.1890 -6.50e-02 -0.00846
##      F2_19       F2_20    F2_23    F2_24   F2_26    F2_37        F2_42
## 1  0.04830 -0.15197410 -0.00129 -0.23600 -0.0307 -0.02610  0.073705890
## 2  0.04430 -0.09380000  0.09340  0.02690 -0.1330  0.07570 -0.009193803
## 3 -0.11500 -0.06502607  0.00249 -0.10200  0.1420 -0.10200  0.064289290
## 4  0.00425 -0.23610000 -0.06900  0.01440  0.0363 -0.01820  0.477874600
## 5  0.02510  0.08504274  0.04450  0.00167 -0.0680  0.00567 -0.075348680
## 6 -0.00574 -0.01807182 -0.12500 -0.06820  0.1250  0.00998 -0.037366600
##     F2_43    F2_45   F2_46    F2_47   F2_48   F2_51   F2_52   F2_54
## 1 -0.0466 -0.00673 -0.0193  0.09040  0.0290  0.0356 -0.0388 -0.0360
## 2 -0.0075  0.01700  0.0722 -0.08390  0.0273 -0.0784 -0.0178  0.1120
## 3  0.0169 -0.01590 -0.1430 -0.00492 -0.0735  0.0657 -0.0197 -0.1290
## 4  0.1440  0.11100  0.0113  0.11900  0.0225  0.0932  0.1430  0.2640
## 5 -0.0673 -0.04720  0.0701 -0.08790 -0.0180 -0.1290 -0.0469 -0.0352
## 6 -0.0402 -0.02190  0.0269  0.13300  0.0732  0.1070 -0.0362 -0.0696
##      F2_63     F2_65   F2_66    F2_68    F2_69    F2_70    F2_71   F2_72
## 1 -0.05600  0.009840 -0.0261  0.00856 -0.01180 -0.03350 -0.08310 -0.0471
## 2  0.12300  0.051700  0.0731  0.08670  0.05710  0.00693 -0.00606 -0.0390
## 3 -0.14300 -0.061600  0.0419 -0.29000 -0.10800 -0.09950 -0.00315  0.0975
## 4 -0.09280 -0.000635 -0.0126  0.06910  0.02260 -0.08630 -0.22900  0.0178
## 5 -0.00166  0.058700 -0.0206 -0.13000  0.00392  0.05450 -0.11200  0.1070
## 6 -0.19400 -0.117000 -0.0400  0.06890  0.04320 -0.00338 -0.05270 -0.0416
##      F2_78        F2_79   F2_80   F2_81   F2_83   F2_86    F2_87    F2_88
## 1 -0.02820  0.047264410  0.0296  0.0114  0.0498 -0.0249 -0.00264 -0.02050
## 2  0.01870  0.008471275 -0.0687 -0.0114 -0.0262 -0.0215 -0.09580 -0.01930
## 3  0.01030 -0.134271000  0.1010  0.0521 -0.0607 -0.0285  0.02560 -0.01350
```

```
## 4  0.00166  0.064096960  0.0103 -0.0258 -0.0837  0.1880  0.03310 -0.00652
## 5  0.01190  0.008985630 -0.1030 -0.1400 -0.0282 -0.1090  0.02070 -0.01370
## 6 -0.03040  0.025920240  0.0697  0.1150  0.0953  0.0127  0.05490  0.00311
##     F2_89  F2_107  F2_108   F2_109  F2_110  F2_111  F2_112   F2_117
## 1  0.0826 -0.0421  0.0663  0.03620  0.0808 -0.0404  0.0877  0.07240
## 2 -0.1140  0.0815  0.0285  0.00299 -0.0407 -0.0657  0.0643 -0.00022
## 3  0.0796  0.0553 -0.0380  0.12900 -0.0361  0.0441 -0.1640 -0.01420
## 4  0.1550  0.0458  0.0752  0.12200 -0.0104  0.0914 -0.0355  0.06520
## 5 -0.0288 -0.1220  0.1270 -0.09390  0.1200 -0.0850  0.1400  0.00867
## 6  0.0955 -0.1520 -0.0670 -0.00599 -0.0438  0.0634  0.1380 -0.04010
##    F2_119   F2_125   F2_126   F2_127   F2_141  F2_142   F2_143   F2_144
## 1 -0.0210  0.04540 -0.03220 -0.00654  0.03490 -0.0315 -0.02170  0.00370
## 2 -0.0877  0.00167  0.00321 -0.01260 -0.04530 -0.0579  0.05920  0.00239
## 3 -0.0279  0.00677  0.07360  0.01750  0.10900 -0.0216 -0.01250  0.05460
## 4  0.1280  0.05940  0.01630  0.00292  0.00714 -0.0565  0.10200  0.03480
## 5  0.1440  0.08710 -0.03360  0.17300  0.08270  0.0594 -0.00317 -0.06750
## 6  0.1310 -0.12600  0.00484 -0.00256 -0.06800  0.0941 -0.04220  0.12000
##    F2_145      F2_154    F2_155  F2_156   F2_157   F2_162  F2_163  F2_164
## 1  0.0322 -0.02150730 -0.000958 -0.0850  0.00462  0.03990  0.0716 -0.0923
## 2 -0.0383  0.02457782 -0.030300 -0.1260 -0.06670 -0.00637 -0.0161 -0.2340
## 3  0.0403 -0.01674888  0.059900  0.0311 -0.05190  0.01890  0.0207  0.0929
## 4  0.0245  0.06776892  0.016500 -0.0382  0.02120  0.06690  0.0512 -0.2450
## 5  0.0495  0.13520570  0.016500  0.0832  0.04350  0.19300  0.0586 -0.0768
## 6  0.1080 -0.05128296 -0.005590  0.0136  0.09910  0.06770 -0.0520  0.1550
##     F2_165  F2_166  F2_167   F2_169   F2_180      F2_181  F2_182   F2_187
## 1  0.10900  0.0102  0.0337  0.00911  0.03210  0.03144772  0.0543  0.01120
## 2 -0.09610 -0.1290 -0.0109 -0.11300 -0.00677 -0.16704700 -0.0239  0.00304
## 3  0.00917  0.0874 -0.1260 -0.00949 -0.09900  0.02700180 -0.0570 -0.05160
## 4  1.23000 -0.0402 -0.0635  0.06880  0.03790 -0.02058180  0.0227  0.04180
## 5  0.04600  0.0484  0.2810  0.07210 -0.00630  0.37074790  0.0618  0.10800
## 6  0.07890  0.0336  0.0648  0.14400  0.02770  0.09297908  0.0601  0.02960
##     F2_188  F2_189   F2_190  F2_191  F2_192  F2_194   F2_195  F2_200
## 1  0.01060  0.1130 -0.03960 -0.0504  0.0877 -0.0563 -0.00557 -0.0484
## 2 -0.03580 -0.1330 -0.01830 -0.0623 -0.0648 -0.0652  0.05020 -0.0912
## 3 -0.04970  0.1660  0.05000  0.0498  0.0431 -0.0224 -0.10700  0.0715
## 4  0.01010  0.2170  0.00206 -0.0155  0.6550  0.2820 -0.01310 -0.0387
## 5  0.12100  0.0237  0.02960  0.1130  0.0839  0.1050  0.15500  0.0823
## 6  0.00198  0.0251  0.00059 -0.0282  0.0429  0.0697  0.04930  0.0414
##    F2_201      F2_212  F2_213  F2_214   F2_215  F2_221   F2_222  F2_223
## 1 -0.0273 -0.10816380 -0.0183 -0.0132 -0.00432 -0.6630  0.01440  0.0310
## 2 -0.0180  0.05682362 -0.0238  0.0721  0.03910  0.1070  0.00923 -0.0397
## 3  0.0432 -0.13217820  0.0205 -0.0411  0.07670 -0.0783 -0.06860 -0.0254
## 4 -0.0667 -0.32395020 -0.0245  0.0865  0.06470 -2.0000  0.00874  0.0847
## 5  0.1140  0.03542023 -0.2020  0.0822  0.04260  0.1030 -0.10100  0.1630
## 6 -0.0708 -0.10881230  0.0359 -0.0678 -0.11000 -0.1420  0.08430 -0.0610
##     F2_224   F2_225   F2_226  F2_227  F2_228  F2_241  F2_242   F2_243
## 1  0.00818 -0.00892 -0.08710  0.0129  0.0937  0.0313  0.0821  0.00621
## 2 -0.06400  0.06300 -0.00152  0.0555  0.0947 -0.0387  0.0592 -0.00636
## 3 -0.05680 -0.13300 -0.07560 -0.0557 -0.0890 -0.1460 -0.0739 -0.01120
## 4 -0.09720  0.00746 -0.55200  0.0415  0.0733  0.0815  0.1100  0.21400
## 5  0.07410 -0.01640  0.08700 -0.0557 -0.1910  0.0219  0.0913  0.01120
## 6  0.08760 -0.03960  0.10200  0.0190 -0.1190  0.0687 -0.0525 -0.00716
##    F2_244   F2_245    F2_247       F2_248   F2_261  F2_263   F2_264
## 1  0.0307 -0.13700  0.075300 -0.096881950 -0.01670 -0.0928 -0.00957
```

```
## 2  0.0614  0.02850 -0.000633  0.001598228 -0.00267 -0.0198  0.16300
## 3 -0.0528  0.05050  0.027700 -0.067933370 -0.02220 -0.0684 -0.04930
## 4  0.0135 -0.13500 -0.003100  0.072318780  0.01030 -0.3150  0.08420
## 5  0.1190  0.00383  0.041700 -0.038618510  0.11800  0.0123  0.03700
## 6 -0.1460 -0.14500  0.029400  0.035281240 -0.05660  0.0917 -0.08080
##    F2_270   F2_271  F2_272   F2_278  F2_287      F2_288    F2_289  F2_290
## 1  0.0287 -0.01300 -0.0292 -0.03810 -0.0488  0.17361240 -0.097900  0.0383
## 2 -0.1310 -0.04260 -0.0514  0.07260 -0.0481 -0.16211430 -0.123000 -0.1370
## 3  0.0328  0.00537 -0.0259 -0.14400  0.0170  0.25924220 -0.041400 -0.0229
## 4  0.0351       NA  0.0730  0.00914  0.0556  0.18311140  0.051700  0.1780
## 5 -0.0142  0.00563 -0.0504 -0.05970 -0.0871  0.20897910 -0.000188 -0.0328
## 6  0.0362  0.00790 -0.0246 -0.07330  0.0125 -0.04778892  0.082500  0.1360
##     F2_291      F2_296  F2_298   F2_299    F2_300  F2_302  F2_303  F2_304
## 1  0.01850 -0.08937784  0.0230 -0.06250 -0.000142  0.0344  0.0358 -0.0139
## 2 -0.05720 -0.07416870 -0.0688 -0.06540 -0.102000 -0.0780 -0.0820 -0.1830
## 3 -0.00664 -0.05915232 -0.0134  0.09740  0.015500 -0.0934  0.1780  0.0842
## 4  0.05250 -0.21653720 -0.2210 -0.00266  0.545000  0.0127  0.0273 -0.0928
## 5 -0.16600 -0.07897525  0.1410 -0.12900  0.090600 -0.1330 -0.2120 -0.0797
## 6  0.04620  0.03811979 -0.0346  0.04690 -0.034800  0.0110  0.0323  0.1660
##    F2_305      F2_306   F2_307   F2_308  F2_309   F2_310  F2_311  F2_312
## 1  0.0134 -0.03145069  0.02780 -0.01190 -0.0744  0.00197 -0.0151 -0.0721
## 2 -0.0270 -0.09822316 -0.07890 -0.05480 -0.1320 -0.11000 -0.1130 -0.0805
## 3  0.0870  0.15520470  0.03410 -0.06830  0.0555 -0.04060  0.0835  0.0514
## 4  0.0469  0.10038160 -2.00000  0.05240  0.1260  0.07280  0.0600 -0.0455
## 5 -0.0191 -0.11958500  0.00294 -0.10600 -0.0518 -0.13200  0.0494  0.0221
## 6 -0.0866  0.05385017  0.09570 -0.00949  0.1120  0.20800  0.0872 -0.0555
##    F2_320  F2_321  F2_323    F2_324  F2_325  F2_326  F2_327   F2_328
## 1 -0.0118  0.0200  0.0222  0.047700 -0.0488  0.0168 -0.0309  0.02740
## 2 -0.1200  0.0101 -0.1610 -0.049200 -0.0350 -0.0738 -0.1730 -0.07380
## 3  0.0713 -0.1130  0.0466  0.000612  0.1210  0.0996  0.1090  0.02730
## 4 -0.0464  0.0667 -0.1850 -0.270000  0.0803  0.0424  0.1610  0.05120
## 5  0.0272 -0.0938  0.1020  0.113000 -0.0859 -0.1340  0.0639  0.00731
## 6  0.0748 -0.1420  0.0590 -0.080000 -0.1200  0.1230  0.1870  0.05410
##    F2_329  F2_330  F2_332  F2_355    F2_357
## 1 -0.0310  0.0660 -0.0199 -0.0146  0.065000
## 2 -0.2010 -0.0820 -0.0939  0.0192 -0.049900
## 3  0.1200 -0.0629 -0.0395  0.1090  0.000253
## 4  0.2410  0.3890  0.0251 -0.0348  0.114000
## 5  0.1240 -0.0212  0.0870  0.0512  0.024300
## 6  0.0699  0.0708  0.1450 -0.0399  0.037500
```

```
## LocusLinkID and ProteomeID are annotations from the said databases
## http://www.ncbi.nlm.nih.gov/LocusLink/
```

Moving on, we extract expression data from the master dataframe. Recall that the rows represent genes and the columns represent different samples (mice) in the original data. WGCNA requires that genes be given in columns.

```
exprdata = as.data.frame(t(mydataf[, -c(1:8)]))
names(exprdata) = mydataf$gene_symbol
rownames(exprdata) = names(mydataf)[-c(1:8)]


## Let us consider a subset of data for this demonstration. We'll use first 500 features.
```

```
exprdata <- exprdata[,1:500]
```

```
gsg = goodSamplesGenes(exprdata, verbose = 3)
```

```
##  Flagging genes and samples with too many missing values...
##   ..step 1
```

```
gsg$allOK
```

```
## [1] TRUE
```

Scale Free Topology

A scale free network topology is the one where all nodes's degree distribution is in abidance to power law ,i.e. **P(k) ~ k^- beta)**. If any nodes have to be added to this connected network, the degrees are accordingly adjusted.

```
trial_powers <- c(c(1:10), seq(from=12, to=20, by=2))
sft_thresh <- pickSoftThreshold (exprdata,
                                 dataIsExpr = TRUE,
                                 powerVector = trial_powers,
                                 corFnc = cor,
                                 corOptions = list(use = 'p'),
                                 networkType = "unsigned")
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
##    Power SFT.R.sq   slope truncated.R.sq mean.k. median.k. max.k.
## 1      1 0.000592 -0.0489       0.283000 100.000  1.03e+02 161.00
## 2      2 0.273000 -0.8680       0.809000  33.800  3.34e+01  75.60
## 3      3 0.484000 -1.4100       0.942000  14.800  1.36e+01  42.40
## 4      4 0.626000 -1.6100       0.955000   7.550  6.30e+00  26.20
## 5      5 0.752000 -1.6600       0.941000   4.320  3.39e+00  17.20
## 6      6 0.859000 -1.4900       0.877000   2.690  2.03e+00  12.00
## 7      7 0.232000 -3.0500       0.044400   1.800  1.19e+00  10.60
## 8      8 0.256000 -3.0400       0.059600   1.270  7.33e-01   9.62
## 9      9 0.219000 -2.6300       0.000234   0.934  4.68e-01   8.88
## 10    10 0.205000 -3.2000       0.002140   0.715  3.09e-01   8.29
## 11    12 0.144000 -2.4600       0.028900   0.459  1.43e-01   7.35
## 12    14 0.150000 -2.3200       0.021300   0.323  6.62e-02   6.62
## 13    16 0.152000 -1.6800       0.026400   0.242  3.18e-02   6.02
## 14    18 0.159000 -1.6300       0.027800   0.190  1.60e-02   5.52
## 15    20 0.872000 -1.2500       0.911000   0.155  8.18e-03   5.09
```

```
cat("The best estimate for use is", sft_thresh$powerEstimate, ".")
```

```
## The best estimate for use is 6 .
```

```
print(sft_thresh$fitIndices)
```

```
##    Power    SFT.R.sq       slope truncated.R.sq     mean.k.    median.k.
## 1      1 0.000591956 -0.04892954    0.282584079 100.3934169 1.032604e+02
## 2      2 0.272783992 -0.86849966    0.808785217  33.8373469 3.341476e+01
## 3      3 0.483755802 -1.41438909    0.941534027  14.7534022 1.362163e+01
## 4      4 0.626177009 -1.61216723    0.954882010   7.5478368 6.299250e+00
## 5      5 0.751600155 -1.66289137    0.941321384   4.3206844 3.386659e+00
## 6      6 0.858673907 -1.49250031    0.877183575   2.6934134 2.028749e+00
```

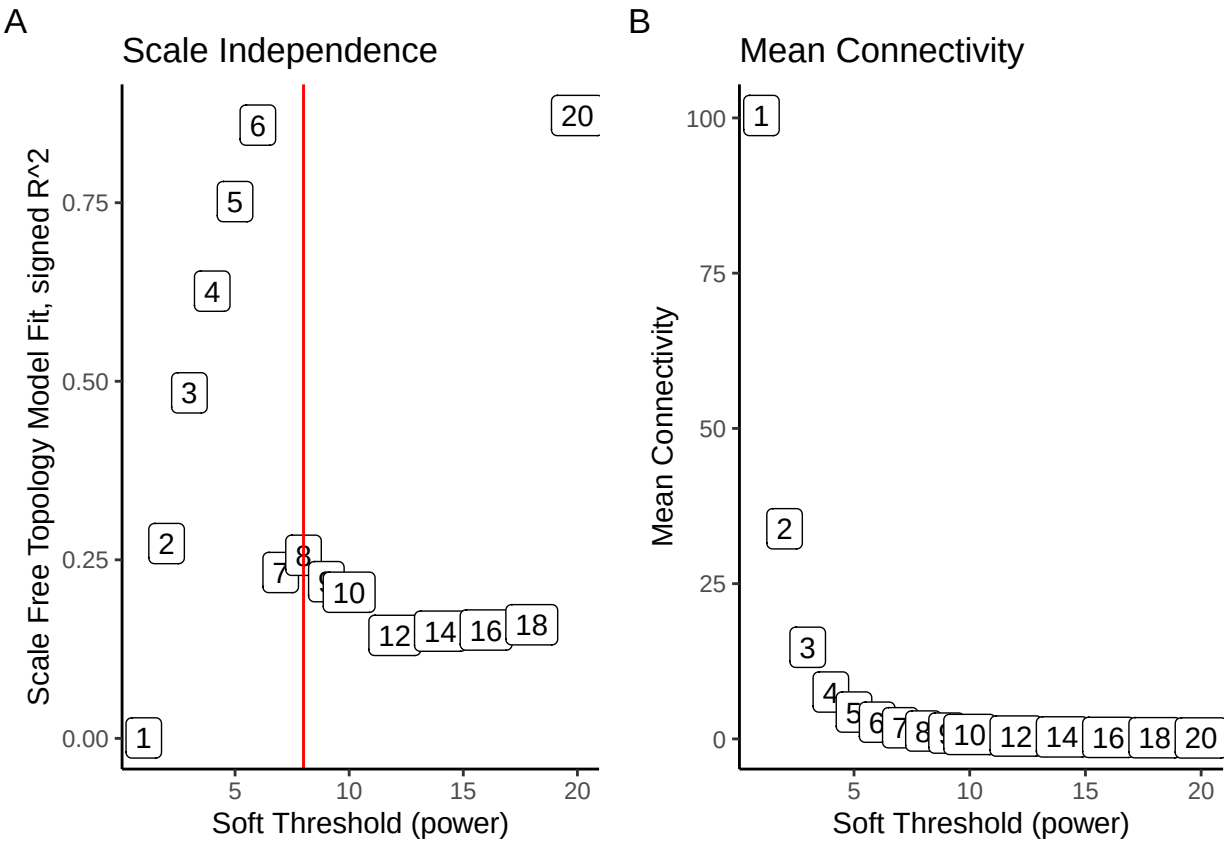
```
## 7      7 0.232232581 -3.05120613    0.044428656   1.7965152 1.186614e+00
## 8      8 0.256337946 -3.04292255    0.059632912   1.2661169 7.334995e-01
## 9      9 0.219405670 -2.63191579    0.000233961   0.9338283 4.684453e-01
## 10    10 0.204590608 -3.20415305    0.002141256   0.7153326 3.091732e-01
## 11    12 0.144177374 -2.46273973    0.028862575   0.4593058 1.427019e-01
## 12    14 0.150084643 -2.32214905    0.021330651   0.3228727 6.622747e-02
## 13    16 0.151669038 -1.67666388    0.026396129   0.2421974 3.178078e-02
## 14    18 0.159074112 -1.63426374    0.027829208   0.1904629 1.595709e-02
## 15    20 0.872145990 -1.24994141    0.910750187   0.1550880 8.177421e-03
##         max.k.
## 1  160.589081
## 2   75.613728
## 3   42.390932
## 4   26.155630
## 5   17.211457
## 6   12.023663
## 7   10.570718
## 8    9.616364
## 9    8.884697
## 10   8.288647
## 11   7.348466
## 12   6.617885
## 13   6.021118
## 14   5.518548
## 15   5.086358
```

It has been determined that *beta=6 is the best exponent for unsigned networks, and beta=12 is the optimal one for signed networks.*

Typically, the degrees per node will "steadily" decrease with the increase in the number of nodes of the network cluster. This is another way of understanding the notion of scale-free topology.

```
# Scale-free topology fit index as a function of the soft-thresholding power

p1 <- ggplot(data = as.data.frame(sft_thresh),
       aes (x = sft_thresh$fitIndices[,1],
     y = (-sign(sft_thresh$fitIndices[,3]) * sft_thresh$fitIndices[,2]),
     label = trial_powers)) +
  geom_label()+

  # the sign expression returns the vector of signs of estimated slopes, multiplied by the correlation

     labs( x="Soft Threshold (power)",
     y="Scale Free Topology Model Fit, signed R^2", title = "Scale Independence", tag = "A")+
  theme_classic()+
  geom_vline(xintercept = 8, color="red") # Red line corresponds to using an R^2 cut-off


# Mean connectivity as a function of the soft-thresholding power

p2 <- ggplot(data = as.data.frame(sft_thresh),
       aes (x = sft_thresh$fitIndices[,1],
     y = sft_thresh$fitIndices[,5],
     label = trial_powers)) +
  geom_label()+
```

```
  # the sign expression returns the vector of signs of estimated slopes, multiplied by the correlation

     labs( x="Soft Threshold (power)",
     y="Mean Connectivity", title = "Mean Connectivity", tag = "B")+
  theme_classic()


plot_grid(p1,p2,ncol=2)
```



Topological Overlap Measure

The next steps is to caliberate the **Topological Overlap Measure (TOM)**. It is premised over the notion of shared relationships between two clusters.

```
## Now we generate adjacency values and TOM similarity matrices on the basis of selected softpower.

softPower = 7;

#calclute the adjacency matrix
adj= adjacency(exprdata,
               type = "unsigned",
               power = softPower);

#turn adjacency matrix into topological overlap to minimize the effects of noise and spurious associati
TOM = TOMsimilarityFromExpr(exprdata,
                            networkType = "unsigned",
                            TOMType = "unsigned",
```

```
                              power = softPower);
```

```
## TOM calculation: adjacency..
## ..will not use multithreading.
##  Fraction of slow calculations: 0.437876
## ..connectivity..
## ..matrix multiplication (system BLAS)..
## ..normalization..
## ..done.
```

```
colnames(TOM) = rownames(TOM) = names(exprdata) # same rows as columns, therefore adjacency.
dissTOM=1-TOM # "distance/ dissimilarity = 1 - similarity"
```

Module Detection

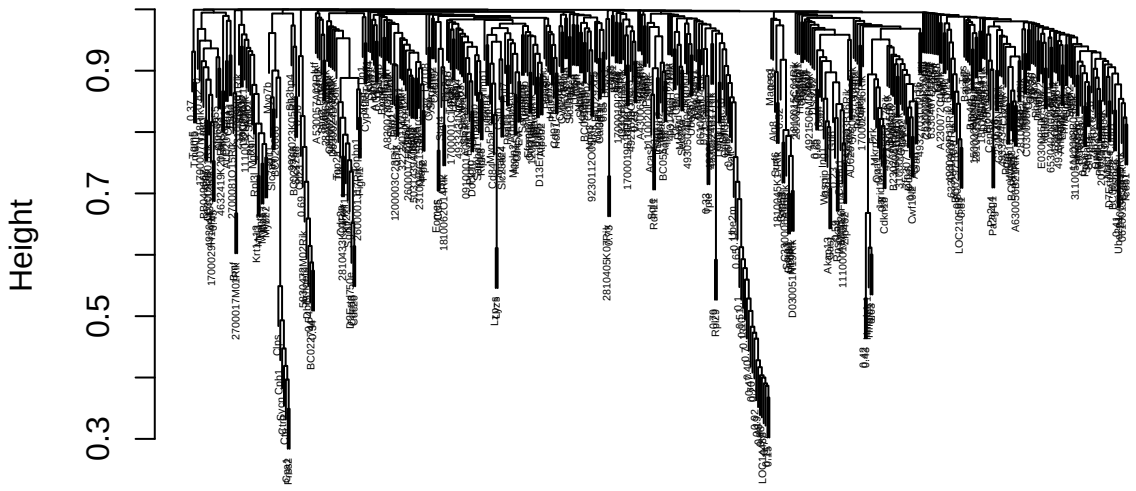
```
#hierarchical clustering of the genes based on the TOM dissimilarity measure

library(flashClust)
```

```
##
## Attaching package: 'flashClust'
```

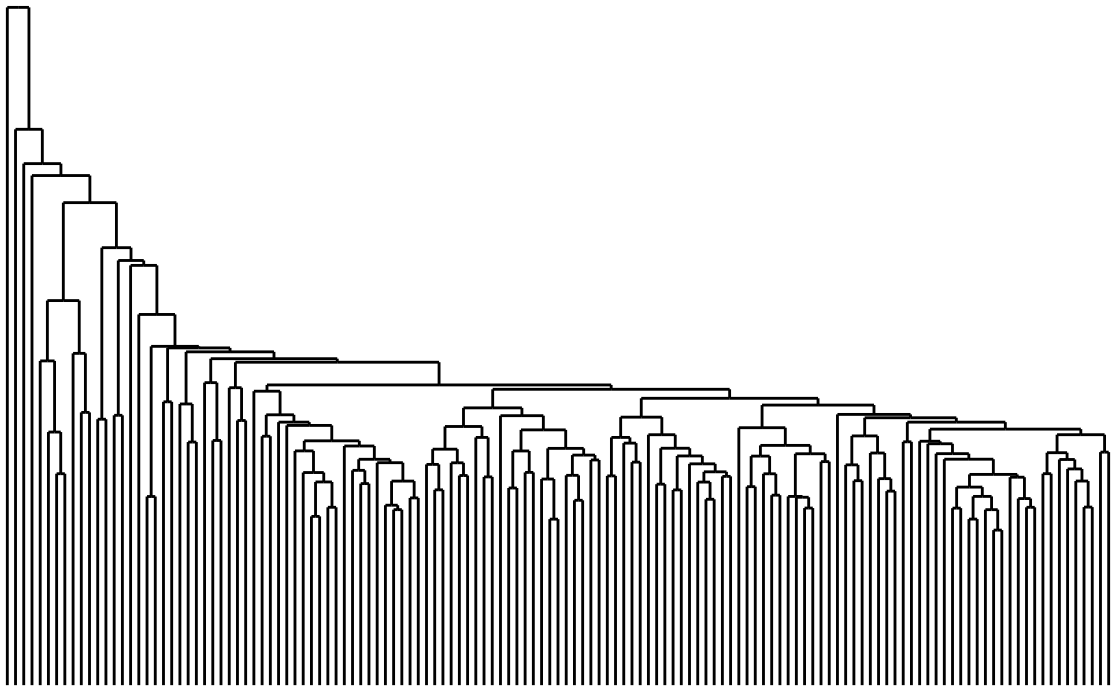
```
## The following object is masked from 'package:fastcluster':
##
##     hclust
```

```
## The following object is masked from 'package:stats':
##
##     hclust
```

```
gene_tree = flashClust(as.dist(dissTOM),
                       method="average")

#plot the resulting clustering tree (dendrogram)
plot(gene_tree,
     xlab="",
     sub="",
     cex=0.3)
```




```
sample_tree <- as.dendrogram(hclust(dist(exprdata), method = "average"))

dplot <- ggdendrogram(data= sample_tree, rotate = FALSE)+
  theme_dendro()+
  ggtitle("Sample clustering to detect outliers")+
  theme(plot.title = element_text(hjust = 0.5))+
  xlab("Samples")+
  geom_label()

print(dplot)
```



```
## Set the minimum module size

min_mod_size = 20

## Module identification using dynamic tree cut

dynamic_mods = cutreeDynamic(dendro = gene_tree,
                             method="tree",
                             minClusterSize = min_mod_size)

dynamic_mods1 = cutreeDynamic(dendro = gene_tree,
                              distM = dissTOM,
                              method="hybrid",
                              deepSplit = 2,
                              pamRespectsDendro = FALSE,
                              minClusterSize = min_mod_size)
```

```
##  ..cutHeight not given, setting it to 0.996  ===>  99% of the (truncated) height range in dendro.
##  ..done.
```

```
## we could highlight module labels and the size(number of nodes) of each module. Label 0 is reserved f

table(dynamic_mods)
```

```
## dynamic_mods
##   0   1   2   3   4   5   6   7   8
## 203  57  53  40  37  31  29  25  25
```

```
table(dynamic_mods1)
```

```
## dynamic_mods1
##   0   1   2   3   4   5
## 168 145  72  45  44  26
```

It is always better to have a enhanced, color-coded visualization of modules for precise interpretation.

```
## we can conveniently assign colors to these labels. "Grey" is assigned to label 0.

dynamic_mod_colors = labels2colors(dynamic_mods)
table(dynamic_mod_colors)
```

```
## dynamic_mod_colors
##     black      blue     brown     green      grey      pink       red
##        25        53        40        31       203        25        29
## turquoise    yellow
##        57        37
```

```
dynamic_mod_colors1 = labels2colors(dynamic_mods1)
table(dynamic_mod_colors1)
```

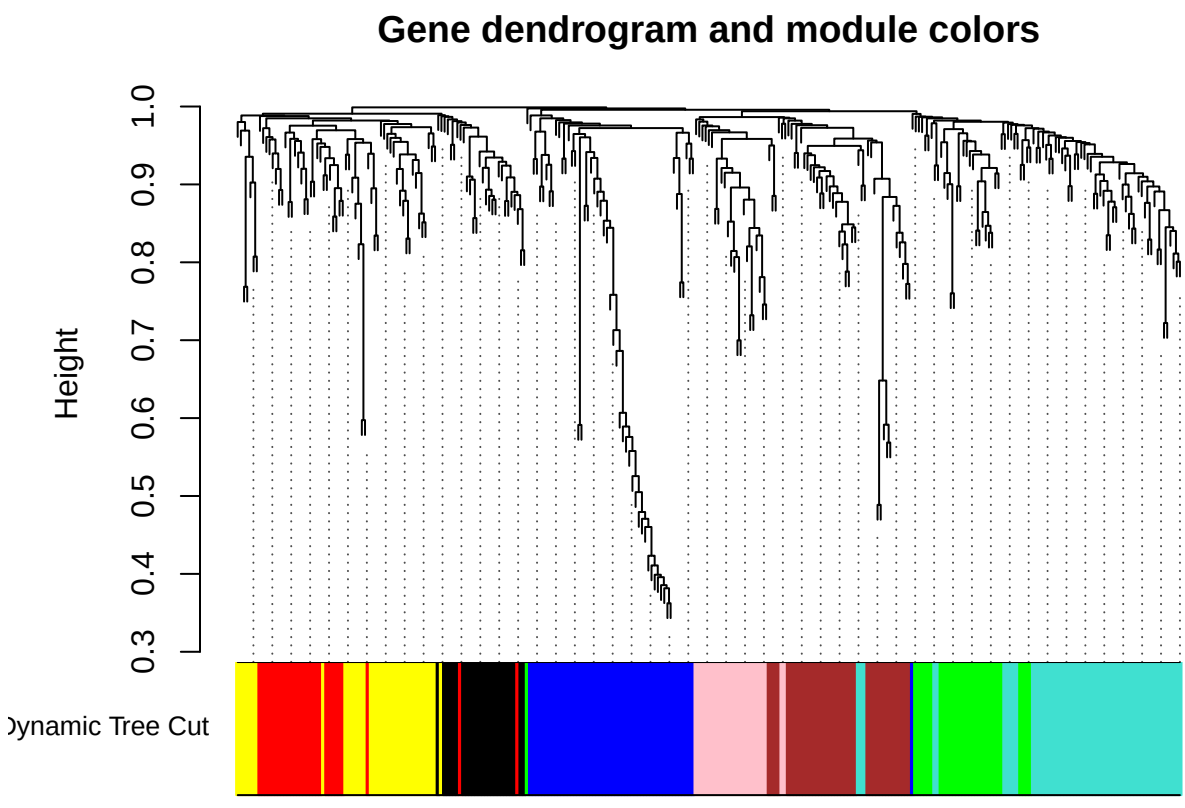
```
## dynamic_mod_colors1
##      blue     brown     green      grey turquoise    yellow
##        72        45        26       168       145        44
```

We can also choose to discard the independent nodes (since they do not belong to any network) and just map the coeherent modules.

```
## discard the unassigned genes, and focus on the rest

rest_genes= (dynamic_mod_colors != "grey")
dissTOM1 = 1 - TOMsimilarityFromExpr(exprdata[,rest_genes],
                                     power = softPower)
```

```
## TOM calculation: adjacency..
## ..will not use multithreading.
##  Fraction of slow calculations: 0.431795
## ..connectivity..
## ..matrix multiplication (system BLAS)..
## ..normalization..
## ..done.
```

```
## Let's plot the tree now.

colnames(dissTOM1) = rownames(dissTOM1) = names(exprdata[rest_genes])
hier_tree = flashClust(as.dist(dissTOM1), method="average" )
plotDendroAndColors(hier_tree,
                    dynamic_mod_colors[rest_genes],
                    "Dynamic Tree Cut",
                    dendroLabels = FALSE,
                    hang = 0.03,
                    addGuide = TRUE,
                    guideHang = 0.05,
                    main = "Gene dendrogram and module colors")
```




```
## set the diagonal of the dissimilarity to NA
```

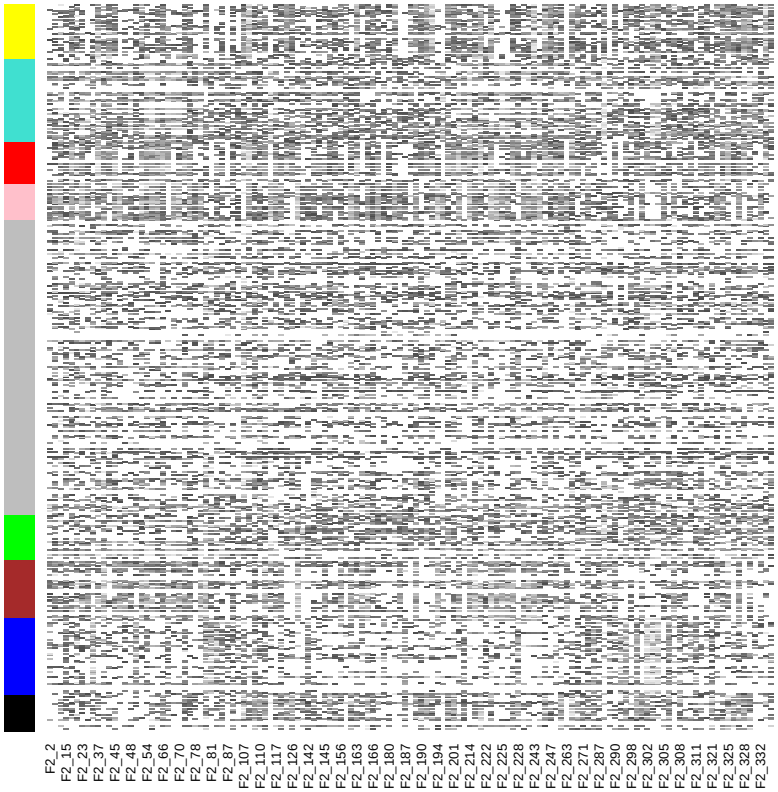
```
diag(dissTOM1) = NA;

## Visualize the TOM plot. Raise the dissimilarity matrix to the power of 4 to bring out the module str

sizeGrWindow(7,7)
TOMplot(dissTOM1, hier_tree, as.character(dynamic_mod_colors[rest_genes]))
```

Let us extract the modules now. The modules could also be envisioned as communities or colonies of like entities.

```
module_colors= setdiff(unique(dynamic_mod_colors), "grey")
for (color in module_colors){
    module = names(exprdata)[which(dynamic_mod_colors == color)]
    write.table(module,
                paste0("module_",color, ".txt"),
                sep="\t",
                row.names=FALSE,
                col.names=FALSE,
                quote=FALSE)
}
```

Let us now visualize the expression patterns of these genes, as they're clustered. There ought to be distinct pattern cues inside and outside of a cluster.

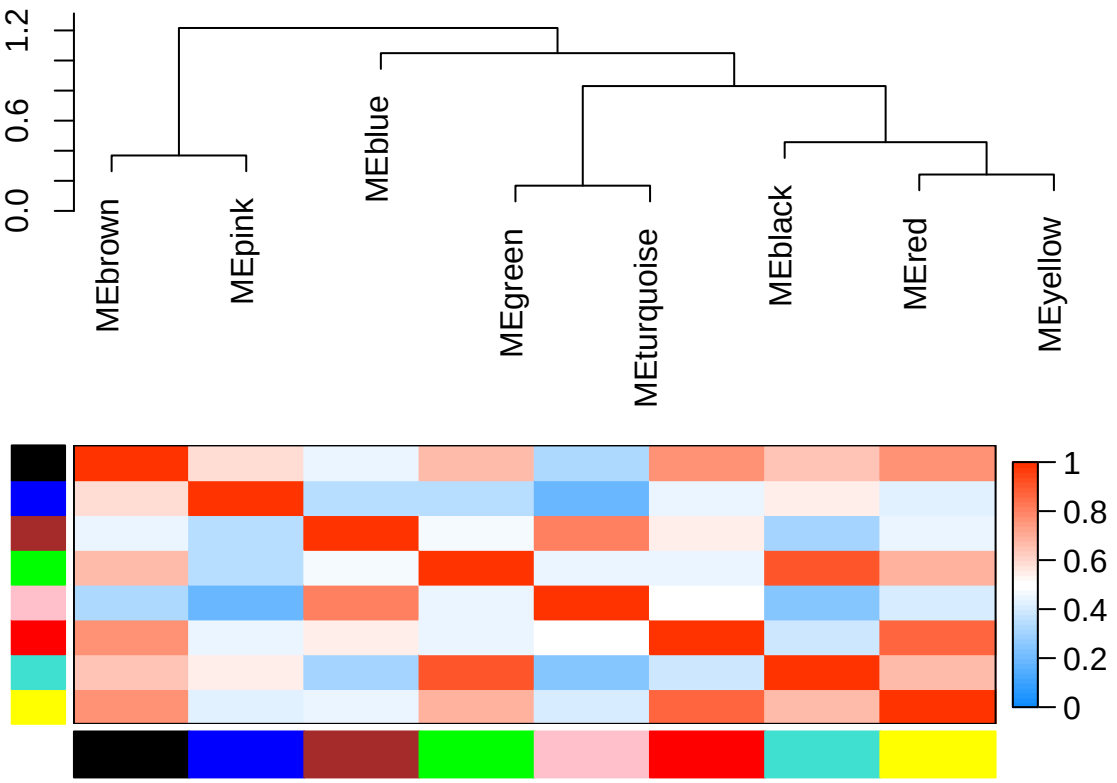
```
module.order <- unlist(tapply(1:ncol(exprdata),
                              as.factor(dynamic_mod_colors),
                              I))
m <- t(t(exprdata[,module.order])/apply(exprdata[,module.order],2,max))

heatmap (t(m),
         zlim = c(0,1),
         col = gray.colors(100),
         Rowv = NA,
         Colv = NA,
         labRow = NA,
         scale = "none",
         RowSideColors = dynamic_mod_colors[module.order])
```



We can now look at the module gene listings and try to interpret their functions .. for instance using [] (http://amigo.geneontology.org/rte)

Quantify module similarity by eigengene correlation. Eigengenes: Module representatives, putatively (by definition) the rank 1 vectors that span the module expression.

```
MEList <-  moduleEigengenes(exprdata, colors = dynamic_mod_colors)
MEs <- MEList$eigengenes
plotEigengeneNetworks(MEs, "",
                      marDendro = c(0,4,1,2),
                      marHeatmap = c(3,4,1,2))
```


0.0
0.6
1.2
MEbrown
MEpink
MEblue
MEgreen
MEturquoise
MEblack
MEred
MEyellow
1
0.8
0.6
0.4
0.2
0