QUBIC Tutorial

Yu Zhang, Juan Xie, *and* Qin Ma

Gene expression data is very important in experimental molecular biology (Brazma and Vilo 2000), especially for cancer study (Fehrmann et al. 2015). The large-scale microarray data and RNA-seq data provide good opportunity to do the gene co-expression analyses and identify co-expressed gene modules; and the effective and efficient algorithms are needed to implement such analysis. Substantial efforts have been made in this field, such as Cheng and Church (2000), Plaid (Lazzeroni et al. 2002), Bayesian Biclustering (BCC, Gu and Liu 2008), among them Cheng and Church and Plaid has the R package implementation. It is worth noting that our in-house biclustering algorithm, QUBIC (Li et al. 2009), is reviewed as one of the best programs in terms of their prediction performance on benchmark datasets. Most importantly, it is reviewed as the best one for large-scale real biological data (Eren et al. 2012).

Until now, QUBIC has been cited over 300 times (via Google Scholar) and its web server, QServer, was developed in 2012 to facilitate the users without comprehensive computational background (Zhou et al. 2012). In the past five years, the cost of RNA-sequencing decreased dramatically, and the amount of gene expression data keeps increasing. Upon requests from users and collaborators, we developed this R package of QUBIC to void submitting large data to a webserver.

The unique features of our R package (Zhang et al. 2017) include (1) updated and more stable back-end resource code (re-written by C++), which has better memory control and is more efficient than the one published in 2009. For an input dataset in *Arabidopsis*, with 25,698 genes and 208 samples, we observed more than 40% time saving; and (2) comprehensive functions and examples, including discretize function, heatmap drawing and network analysis.

How to cite

```r
citation("QUBIC")
```

```
Please cite the QUBIC package in your work, whenever you use it:

  Yu Zhang, Juan Xie, Jinyu Yang, Anne Fennell, Chi Zhang, Qin Ma; QUBIC: a
  bioconductor package for qualitative biclustering analysis of gene co-expression
  data. Bioinformatics, 2017; 33 (3): 450-452. doi: 10.1093/bioinformatics/btw635

A BibTeX entry for LaTeX users is

  @Article{,
    title = {QUBIC: a bioconductor package for qualitative biclustering analysis of gene
    co-expression data},
    author = {Yu Zhang and Juan Xie and Jinyu Yang and Anne Fennell and Chi Zhang and Qin Ma},
    journal = {Bioinformatics},
    year = {2017},
    volume = {33},
    number = {3},
    pages = {450--452},
    doi = {10.1093/bioinformatics/btw635},
  }
```

Other languages

If R is not your thing, there is also a C version of `QUBIC`.

Help

If you are having trouble with this R package, contact the maintainer, Yu Zhang.

Install and load

Stable version from BioConductor

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("QUBIC")
```

Or development version from GitHub

```
install.packages("devtools")
devtools::install_github("zy26/QUBIC")
```

Load `QUBIC`

```
library("QUBIC")
```

Functions

There are nine functions provided by QUBIC package.

- `qudiscretize()`creates a discrete matrix for a given gene expression matrix;
- `BCQU()`performs a qualitative biclustering for real matrix;
- `BCQUD()`performs a qualitative biclustering for discretized matrix;
- `quheatmap()`can draw heatmap for singe bicluster or overlapped biclusters;
- `qunetwork()` can automatically create co-expression networks based on the identified biclusters by QUBIC;
- `qunet2xml()`can convert the constructed co-expression networks into XGMML format for further network analysis in Cytoscape, Biomax and JNets;
- *query-based biclustering* allows users to input additional biological information to guide the biclustering progress;
- *bicluster expanding* expands existing biclusters under specified consistency level;

The following examples illustrate how these functions work.

Example of a random matrix with two diferent embedded biclusters

```
library(QUBIC)
set.seed(1)
# Create a random matrix
test <- matrix(rnorm(10000), 100, 100)
colnames(test) <- paste("cond", 1:100, sep = "_")
rownames(test) <- paste("gene", 1:100, sep = "_")

# Discretization
```

```r
matrix1 <- test[1:7, 1:4]
matrix1
```

```
##            cond_1      cond_2     cond_3      cond_4
## gene_1 -0.6264538 -0.62036668  0.4094018  0.89367370
## gene_2  0.1836433  0.04211587  1.6888733 -1.04729815
## gene_3 -0.8356286 -0.91092165  1.5865884  1.97133739
## gene_4  1.5952808  0.15802877 -0.3309078 -0.38363211
## gene_5  0.3295078 -0.65458464 -2.2852355  1.65414530
## gene_6 -0.8204684  1.76728727  2.4976616  1.51221269
## gene_7  0.4874291  0.71670748  0.6670662  0.08296573
```

```r
matrix2 <- qudiscretize(matrix1)
matrix2
```

```
##        cond_1 cond_2 cond_3 cond_4
## gene_1     -1      0      0      1
## gene_2      0      0      1     -1
## gene_3      0     -1      0      1
## gene_4      1      0      0     -1
## gene_5      0      0     -1      1
## gene_6     -1      0      1      0
## gene_7      0      1      0     -1
```

```r
# Fill bicluster blocks
t1 <- runif(10, 0.8, 1)
t2 <- runif(10, 0.8, 1) * (-1)
t3 <- runif(10, 0.8, 1) * sample(c(-1, 1), 10, replace = TRUE)
test[11:20, 11:20] <- t(rep(t1, 10) * rnorm(100, 3, 0.3))
test[31:40, 31:40] <- t(rep(t2, 10) * rnorm(100, 3, 0.3))
test[51:60, 51:60] <- t(rep(t3, 10) * rnorm(100, 3, 0.3))

# QUBIC
res <- qubiclust(test)
summary(res)
```

```
##
## An object of class QUBICBiclustResult
##
## call:
##  qubiclust(x = test)
##
## Number of Clusters found: 40
##
## Cluster sizes:
##                     BC 1 BC 2 BC 3 BC 4 BC 5 BC 6 BC 7 BC 8 BC 9 BC 10 BC 11 BC 12 BC 13 BC 14
## Number of Rows        10   10    9    9    3    3    3    5    3     2     2     2     2     2
## Number of Columns      9    9    8    7    5    5    5    3    4     6     6     6     6     6
##                     BC 15 BC 16 BC 17 BC 18 BC 19 BC 20 BC 21 BC 22 BC 23 BC 24 BC 25 BC 26 BC 27
## Number of Rows          2     2     2     2     2     3     3     2     2     2     2     2     2
## Number of Columns       6     6     6     6     6     4     4     5     5     5     5     5     5
##                     BC 28 BC 29 BC 30 BC 31 BC 32 BC 33 BC 34 BC 35 BC 36 BC 37 BC 38 BC 39 BC 40
## Number of Rows          2     2     2     2     2     2     2     2     2     2     2     2     2
## Number of Columns       5     5     5     5     5     5     5     5     5     5     5     5     5
```

```r
# Show heatmap
```

```r
hmcols <- colorRampPalette(rev(c("#D73027", "#FC8D59", "#FEE090", "#FFFFBF",
    "#E0F3F8", "#91BFDB", "#4575B4")))(100)
# Specify colors

par(mar = c(4, 5, 3, 5) + 0.1)
quheatmap(test, res, number = c(1, 3), col = hmcols, showlabel = TRUE)
```

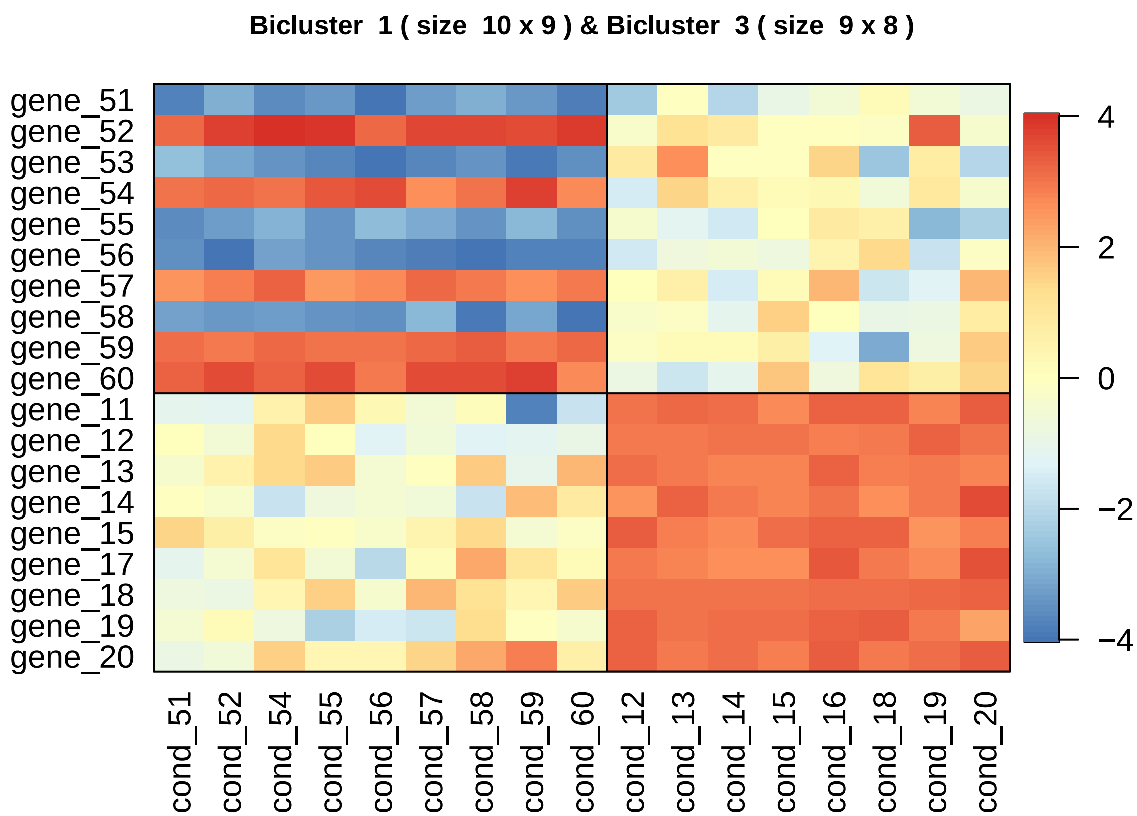

Figure 1: Heatmap for two overlapped biclusters in the simulated matrix

Example of *Saccharomyces cerevisiae*

The cached yeast expression matrix bundled for this vignette is derived from the original `BicatYeast` example distributed with the `biclust` package, and is retained here only to keep this historical QUBIC example reproducible after the runtime dependency on `biclust` was removed.

```r
library(QUBIC)
yeast_cache_path <- "data/yeast_expression_matrix.rds"

if (!file.exists(yeast_cache_path)) {
    stop("Missing cached yeast dataset bundled for the vignette: ", yeast_cache_path)
}

yeast_matrix <- readRDS(yeast_cache_path)

# Discretization
matrix1 <- yeast_matrix[1:7, 1:4]
matrix1
```

```
##            cold_green_6h cold_green_24h cold_roots_6h cold_roots_24h
## 249364_at    -0.2759300     -0.5108508    1.74476670     2.12442300
## 253423_at    -0.9405282      3.2669048   -0.37776557     0.06860917
## 250327_at     1.6419950      1.4484175    0.33474782    -0.15095752
## 247474_at     0.6903505      1.6705408    0.04386528    -0.45295456
## 252661_at     1.7493315      0.9260773    2.05519100     2.11200260
## 258239_at     0.6110116     -0.6083303    0.60419910     0.43582130
## 248910_at     1.4501406     -0.3107802    0.16640233     0.37186486
```

```
matrix2 <- qudiscretize(matrix1)
matrix2
```

```
##            cold_green_6h cold_green_24h cold_roots_6h cold_roots_24h
## 249364_at              0             -1             0              1
## 253423_at             -1              1             0              0
## 250327_at              1              0             0             -1
## 247474_at              0              1             0             -1
## 252661_at              0             -1             0              1
## 258239_at              1             -1             0              0
## 248910_at              1             -1             0              0
```

```
# QUBIC
x <- yeast_matrix
system.time(res <- qubiclust(x))
```

```
##    user  system elapsed
##   0.023   0.000   0.024
```

```
summary(res)
```

```
##
## An object of class QUBICBiclustResult
##
## call:
##  qubiclust(x = x)
##
## Number of Clusters found: 77
##
## Cluster sizes:
##                      BC 1 BC 2 BC 3 BC 4 BC 5 BC 6 BC 7 BC 8 BC 9 BC 10 BC 11 BC 12 BC 13 BC 14
## Number of Rows         72   53   37   80   56   98   47   26   45    35    29    34    42    33
## Number of Columns       4    5    7    3    4    2    4    7    4     5     6     5     4     5
##                      BC 15 BC 16 BC 17 BC 18 BC 19 BC 20 BC 21 BC 22 BC 23 BC 24 BC 25 BC 26 BC 27
## Number of Rows          41    41    32    53    39    38    74    29    47    23    27    33    22
## Number of Columns        4     4     5     3     4     4     2     5     3     6     5     4     6
##                      BC 28 BC 29 BC 30 BC 31 BC 32 BC 33 BC 34 BC 35 BC 36 BC 37 BC 38 BC 39 BC 40
## Number of Rows          41    40    19    51    20    32    16    31    23    30    44    44    41
## Number of Columns        3     3     6     2     5     3     6     3     4     3     2     2     2
##                      BC 41 BC 42 BC 43 BC 44 BC 45 BC 46 BC 47 BC 48 BC 49 BC 50 BC 51 BC 52 BC 53
## Number of Rows          26    13    39    39    19    25    25    25    25     9    18     8     8
## Number of Columns        3     6     2     2     4     3     3     3     3     8     4     9     9
##                      BC 54 BC 55 BC 56 BC 57 BC 58 BC 59 BC 60 BC 61 BC 62 BC 63 BC 64 BC 65 BC 66
## Number of Rows          36    12    23    32    16    31    20    14    28    11    18    18    17
## Number of Columns        2     6     3     2     4     2     3     4     2     5     3     3     3
##                      BC 67 BC 68 BC 69 BC 70 BC 71 BC 72 BC 73 BC 74 BC 75 BC 76 BC 77
## Number of Rows          24    24     9    14    14    20     9    12    17     8     3
## Number of Columns        2     2     5     3     3     2     4     3     2     3     5
```

We can draw heatmap for single bicluster.

```r
# Draw heatmap for the second bicluster identified in Saccharomyces cerevisiae data

library(RColorBrewer)
paleta <- colorRampPalette(rev(brewer.pal(11, "RdYlBu")))(11)
par(mar = c(5, 4, 3, 5) + 0.1, mgp = c(0, 1, 0), cex.lab = 1.1, cex.axis = 0.5,
    cex.main = 1.1)
quheatmap(x, res, number = 2, showlabel = TRUE, col = paleta)
```

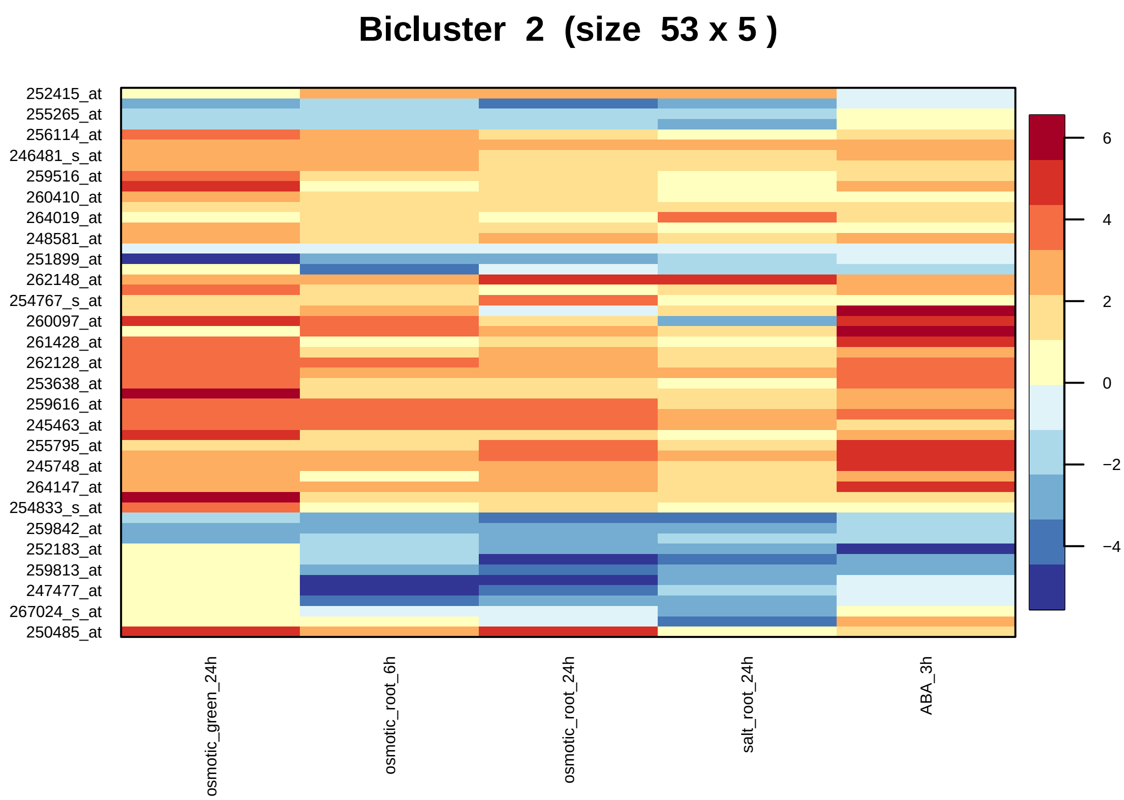

Figure 2: Heatmap for the second bicluster identified in the *Saccharomyces cerevisiae* data. The bicluster consists of 53 genes and 5 conditions

We can draw heatmap for overlapped biclusters.

```r
# Draw for the second and third biclusters identified in Saccharomyces cerevisiae data

par(mar = c(5, 5, 5, 5), cex.lab = 1.1, cex.axis = 0.5, cex.main = 1.1)
paleta <- colorRampPalette(rev(brewer.pal(11, "RdYlBu")))(11)
quheatmap(x, res, number = c(2, 3), showlabel = TRUE, col = paleta)
```

We can draw network for single bicluster.

```r
# Construct the network for the second identified bicluster in Saccharomyces cerevisiae
net <- qunetwork(x, res, number = 2, group = 2, method = "spearman")
if (requireNamespace("qgraph", quietly = TRUE))
    qgraph::qgraph(net[[1]], groups = net[[2]], layout = "spring", minimum = 0.6,
        color = cbind(rainbow(length(net[[2]]) -  1), "gray"), edge.labels = FALSE)
```

We can also draw network for overlapped biclusters.

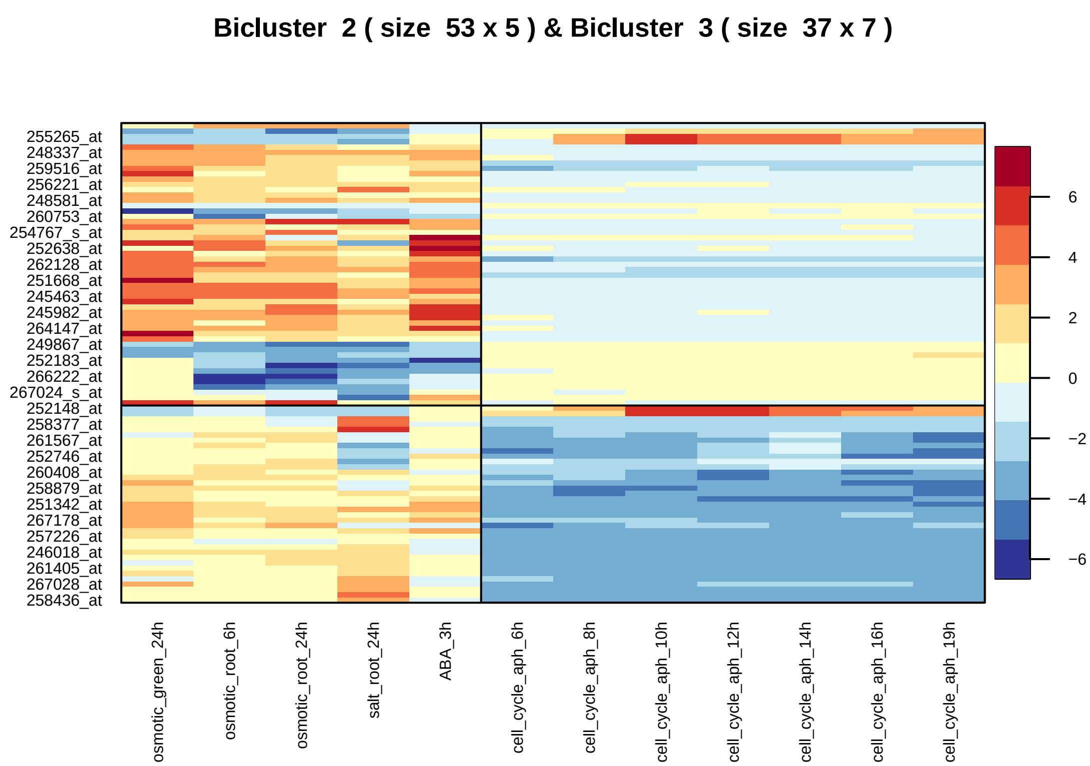

Figure 3: Heatmap for the second and third biclusters identified in the *Saccharomyces cerevisiae* data. Bicluster #2 (topleft) consists of 53 genes and 5 conditions, and bicluster #3 (bottom right) consists of 37 genes and 7 conditions.

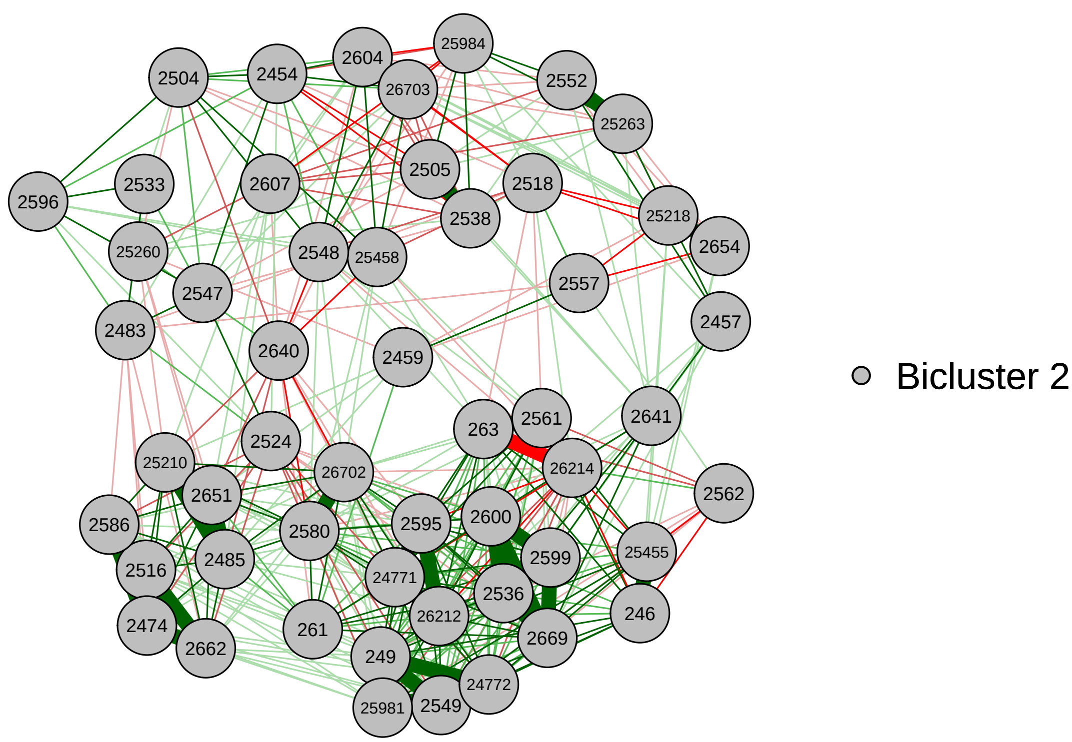

Figure 4: Network for the second bicluster identified in the *Saccharomyces cerevisiae* data.

```
net <- qunetwork(x, res, number = c(2, 3), group = c(2, 3), method = "spearman")
if (requireNamespace("qgraph", quietly = TRUE))
    qgraph::qgraph(net[[1]], groups = net[[2]], layout = "spring", minimum = 0.6,
        legend.cex = 0.5, color = c("red", "blue", "gold", "gray"), edge.labels = FALSE)
```

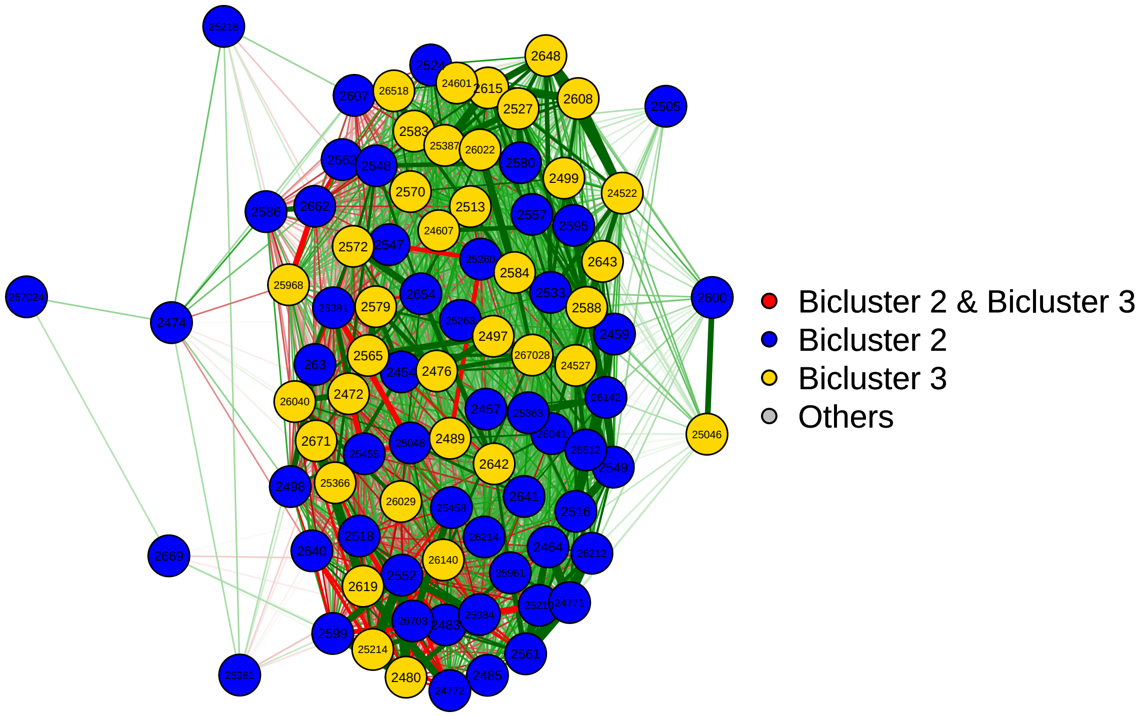

Figure 5: Network for the second and third biclusters identified in the *Saccharomyces cerevisiae* data.

```
# Output overlapping heatmap XML, could be used in other software such
# as Cytoscape, Biomax or JNets
sink('tempnetworkresult.gr')
qunet2xml(net, minimum = 0.6, color = cbind(rainbow(length(net[[2]]) - 1), "gray"))
sink()
# We can use Cytoscape, Biomax or JNets open file named 'tempnetworkresult.gr'
```

Example of *Escherichia coli* data

The *Escherichia coli* data consists of 4,297 genes and 466 conditions.

```
library(QUBIC)
library(QUBICdata)
data("ecoli", package = "QUBICdata")

# Discretization
matrix1 <- ecoli[1:7, 1:4]
matrix1
```

```
##       dinI_U_N0025 dinP_U_N0025 lexA_U_N0025 lon_U_N0025
## b4634     9.077693     9.225537     9.138900    9.114353
## b3241     7.122300     7.195453     7.051193    7.124200
## b3240     7.184417     7.336610     7.283377    7.188263
```

```
## b3243     7.902090     7.963167     7.847747    7.943650
## b3242     6.801900     6.843213     6.795007    6.889897
## b2836     9.114207     9.133303     9.167487    9.189480
## b0885     9.057120     8.918723     8.985483    9.002663
```

```r
matrix2 <- qudiscretize(matrix1)
matrix2
```

```
##       dinI_U_N0025 dinP_U_N0025 lexA_U_N0025 lon_U_N0025
## b4634           -1            1            0           0
## b3241            0            1           -1           0
## b3240           -1            1            0           0
## b3243            0            1           -1           0
## b3242            0            0           -1           1
## b2836           -1            0            0           1
## b0885            1           -1            0           0
```

```r
# QUBIC
res <- qubiclust(ecoli, r = 1, q = 0.06, c = 0.95, o = 100,
    f = 0.25, k = max(ncol(ecoli)%/%20, 2))
system.time(res <- qubiclust(ecoli, r = 1, q = 0.06, c = 0.95,
    o = 100, f = 0.25, k = max(ncol(ecoli)%/%20, 2)))
```

```
##    user  system elapsed
##   3.048   0.024   3.076
```

```r
summary(res)
```

```
##
## An object of class QUBICBiclustResult
##
## call:
##  qubiclust(x = ecoli, r = 1, q = 0.06, c = 0.95, o = 100, f = 0.25,
##     k = max(ncol(ecoli)%/%20, 2))
##
## Number of Clusters found: 19
##
## Cluster sizes:
##                    BC 1 BC 2 BC 3 BC 4 BC 5 BC 6 BC 7 BC 8 BC 9 BC 10 BC 11 BC 12 BC 13 BC 14
## Number of Rows      437  121   51  108  103   65   41   26   27    20    25    23    17    18
## Number of Columns    29   45   94   44   38   38   31   33   31    27    19    20    23    21
##                    BC 15 BC 16 BC 17 BC 18 BC 19
## Number of Rows        14    15    13     5     6
## Number of Columns     26    20    22    32    23
```

```r
# Draw heatmap for the 5th bicluster identified in Escherichia coli data

library(RColorBrewer)
paleta <- colorRampPalette(rev(brewer.pal(11, "RdYlBu")))(11)
par(mar = c(5, 4, 3, 5) + 0.1, mgp = c(0, 1, 0), cex.lab = 1.1, cex.axis = 0.5,
    cex.main = 1.1)
quheatmap(ecoli, res, number = 5, showlabel = TRUE, col = paleta)
```

```r
library(RColorBrewer)
paleta <- colorRampPalette(rev(brewer.pal(11, "RdYlBu")))(11)
par(mar = c(5, 4, 3, 5), cex.lab = 1.1, cex.axis = 0.5, cex.main = 1.1)
quheatmap(ecoli, res, number = c(4, 8), showlabel = TRUE, col = paleta)
```

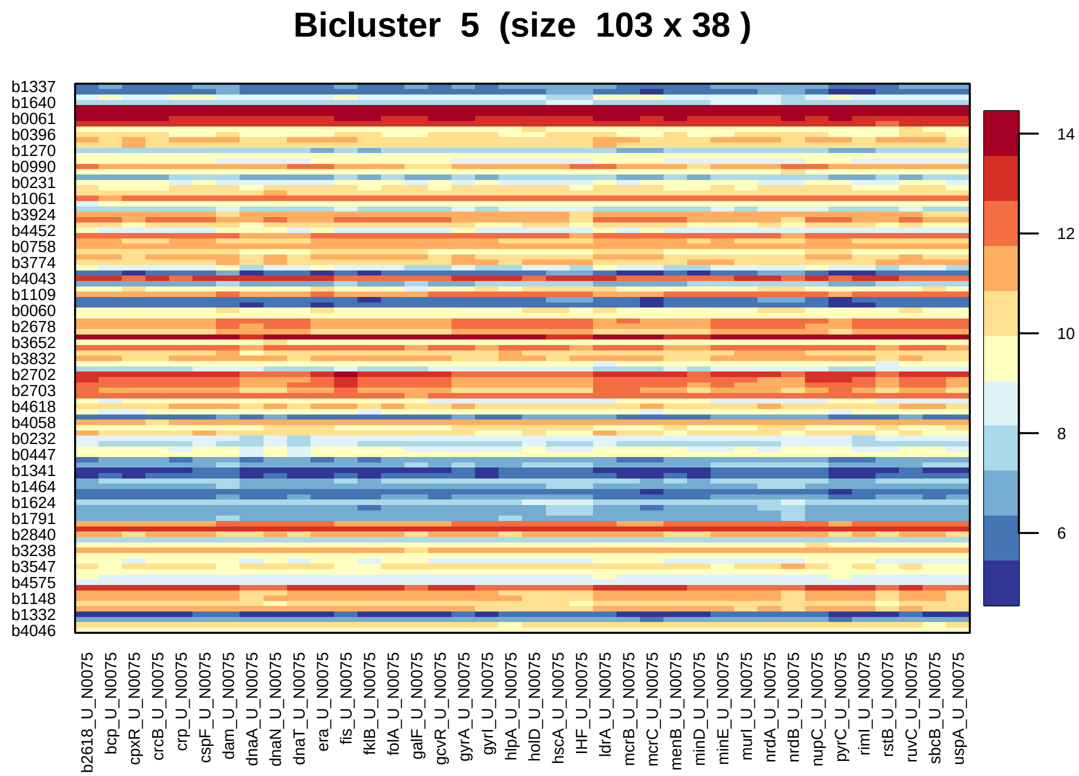

Figure 6: Heatmap for the fifth bicluster identified in the *Escherichia coli* data. The bicluster consists of 103 genes and 38 conditions

```
# construct the network for the 5th identified bicluster in Escherichia coli data
net <- qunetwork(ecoli, res, number = 5, group = 5, method = "spearman")
if (requireNamespace("qgraph", quietly = TRUE))
    qgraph::qgraph(net[[1]], groups = net[[2]], layout = "spring", minimum = 0.6,
        color = cbind(rainbow(length(net[[2]]) - 1), "gray"), edge.labels = FALSE)
```

```
# construct the network for the 4th and 8th identified bicluster in Escherichia coli data
net <- qunetwork(ecoli, res, number = c(4, 8), group = c(4, 8), method = "spearman")
if (requireNamespace("qgraph", quietly = TRUE))
    qgraph::qgraph(net[[1]], groups = net[[2]], legend.cex = 0.5, layout = "spring",
        minimum = 0.6, color = c("red", "blue", "gold", "gray"), edge.labels = FALSE)
```

Query-based biclustering

We can conduct a query-based biclustering by adding the weight parameter. In this example, the instance file "511145.protein.links.v10.txt" (Szklarczyk et al. 2014) was downloaded from string (http://string-db.org/download/protein.links.v10/511145.protein.links.v10.txt.gz) and decompressed and saved in working directory.

```
# Here is an example to download and extract the weight
library(igraph)
url <- "http://string-db.org/download/protein.links.v10/511145.protein.links.v10.txt.gz"
tmp <- tempfile()
download.file(url, tmp)
graph = read.graph(gzfile(tmp), format = "ncol")
unlink(tmp)
```

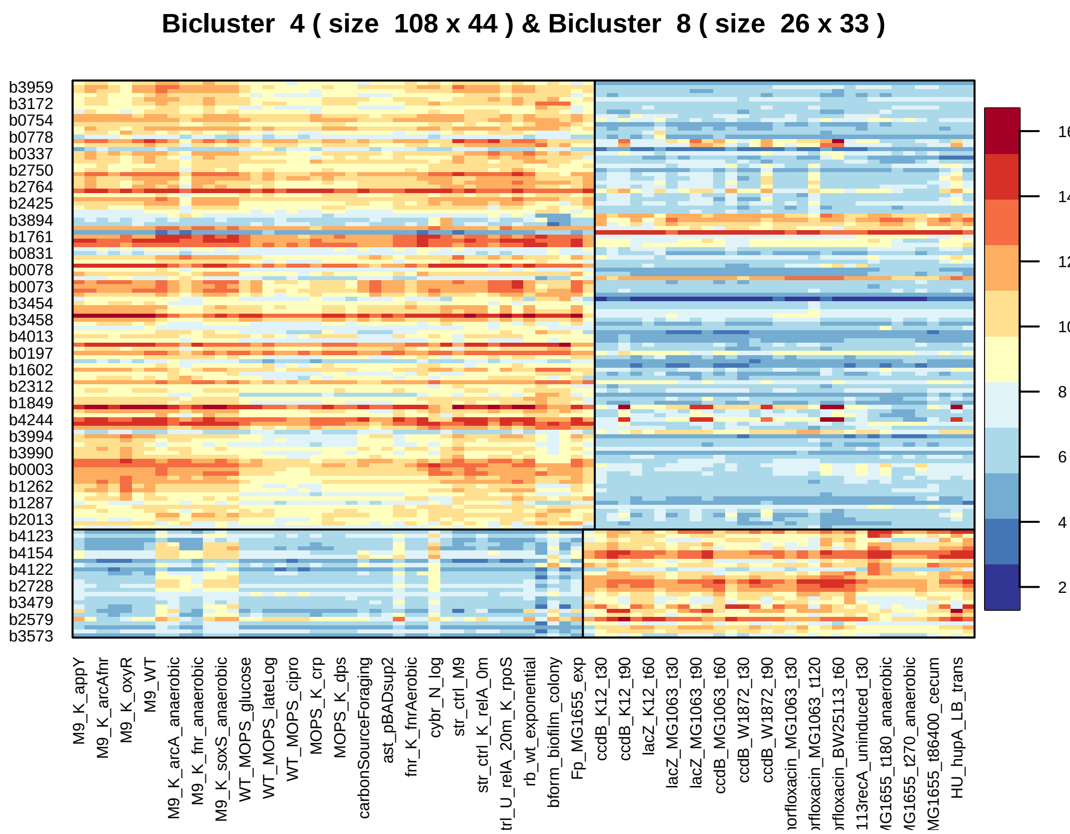

Figure 7: Heatmap for the fourth and eighth biclusters identified in the *Escherichia coli* data. Bicluster #4 (topleft) consists of 108 genes and 44 conditions, and bicluster #8 (bottom right) consists of 26 genes and 33 conditions

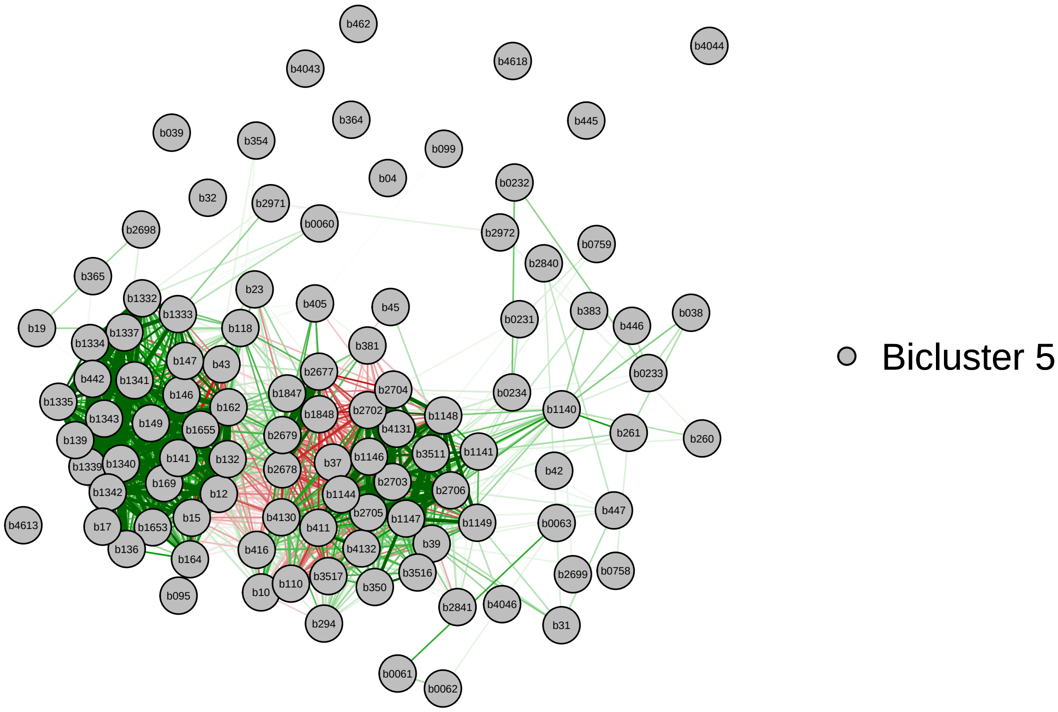

Figure 8: Network for the fifth bicluster identified in the *Escherichia coli* data.

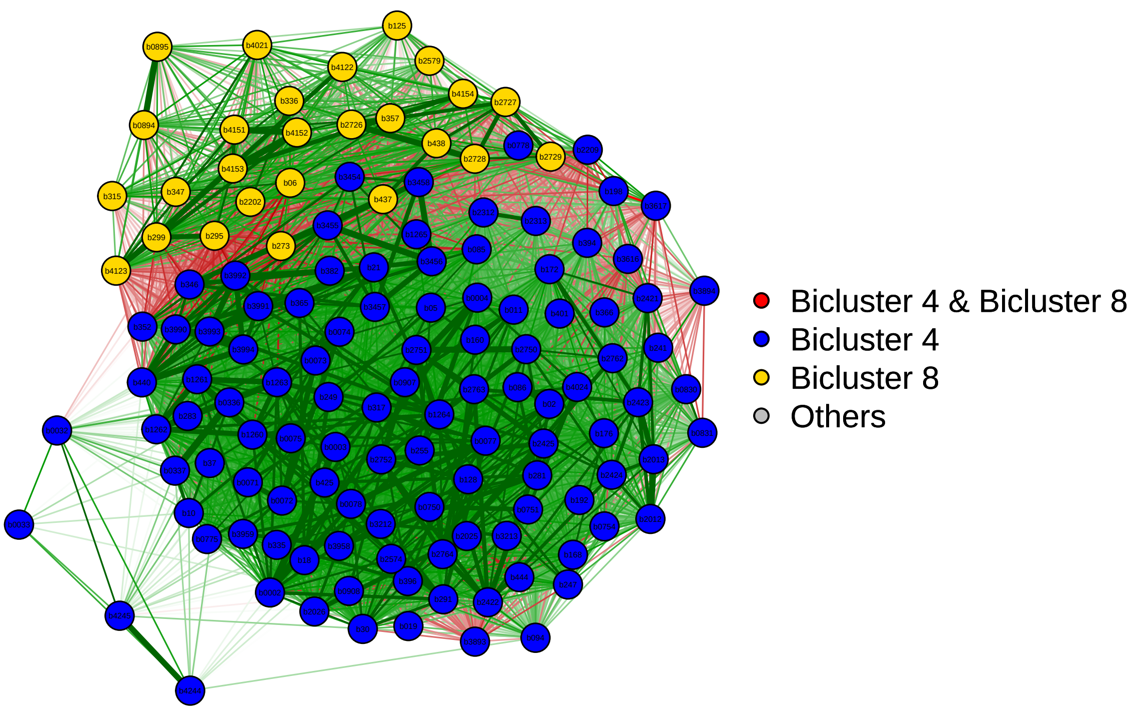

Figure 9: Network for the fourth and eighth biclusters identified in the *Escherichia coli* data.

```r
ecoli.weight <- get.adjacency(graph, attr = "weight")
```

```r
library(QUBIC)
library(QUBICdata)
data("ecoli", package = "QUBICdata")
data("ecoli.weight", package = "QUBICdata")
res0 <- qubiclust(ecoli, verbose = FALSE)
res0
```

```
##
## An object of class QUBICBiclustResult
##
## call:
##  qubiclust(x = ecoli, verbose = FALSE)
##
## Number of Clusters found: 100
##
## First  5  Cluster sizes:
##                      BC 1 BC 2 BC 3 BC 4 BC 5
## Number of Rows:       437  519  178  180  169
## Number of Columns:     29   22   52   50   53
```

```r
res4 <- qubiclust(ecoli, weight = ecoli.weight, verbose = FALSE)
res4
```

```
##
## An object of class QUBICBiclustResult
##
## call:
##  qubiclust(x = ecoli, verbose = FALSE, weight = ecoli.weight)
##
## Number of Clusters found: 100
##
## First  5  Cluster sizes:
##                      BC 1 BC 2 BC 3 BC 4 BC 5
## Number of Rows:       437  519  178  180  169
## Number of Columns:     29   22   52   50   53
```

Bicluster-expanding

we can expand existing biclustering results to recruite more genes according to certain consistency level:

```r
res5 <- qubiclust(x = ecoli, seedbicluster = res, f = 0.25, verbose = FALSE)
summary(res5)
```

```
##
## An object of class QUBICBiclustResult
##
## call:
##  qubiclust(x = ecoli, f = 0.25, verbose = FALSE, seedbicluster = res)
##
## Number of Clusters found: 19
##
## Cluster sizes:
##                     BC 1 BC 2 BC 3 BC 4 BC 5 BC 6 BC 7 BC 8 BC 9 BC 10 BC 11 BC 12 BC 13 BC 14
## Number of Rows       593  151   51  110  117   68   84   27   43    20    36    30    17    19
```

```
## Number of Columns   29   45   94   44   38   38   31   33   31    27    19    20    23    21
##                  BC 15 BC 16 BC 17 BC 18 BC 19
## Number of Rows       14    16    16     5     6
## Number of Columns    26    20    22    32    23
```

Session Information

```
sessionInfo()
```

```
## R version 4.6.0 Patched (2026-05-01 r89994)
## Platform: aarch64-apple-darwin23
## Running under: macOS Tahoe 26.3.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.6/Resources/lib/libRlapack.dylib;  LAPACK
    version 3.12.1
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] QUBICdata_1.41.0   RColorBrewer_1.1-3 QUBIC_1.41.0
##
## loaded via a namespace (and not attached):
##  [1] dotCall64_1.2      gtable_0.3.6       spam_2.11-3        xfun_0.57          ggplot2_4.0.3
##  [6] htmlwidgets_1.6.4 psych_2.6.5        lattice_0.22-9     quadprog_1.5-8     vctrs_0.7.3
## [11] tools_4.6.0        generics_0.1.4     stats4_4.6.0       parallel_4.6.0     tibble_3.3.1
## [16] cluster_2.1.8.2    pkgconfig_2.0.3    qgraph_1.9.8       Matrix_1.7-5       data.table_1.18.4
## [21] checkmate_2.3.4    S7_0.2.2           lifecycle_1.0.5    compiler_4.6.0     farver_2.1.2
## [26] stringr_1.6.0      fields_17.3        mnormt_2.1.2       tinytex_0.59       codetools_0.2-20
## [31] glasso_1.11        htmltools_0.5.9    maps_3.4.3         fdrtool_1.2.18     yaml_2.3.12
## [36] htmlTable_2.5.0    Formula_1.2-5      pillar_1.11.1      Hmisc_5.2-5        abind_1.4-8
## [41] rpart_4.1.27       nlme_3.1-169       lavaan_0.6-21      gtools_3.9.5       tidyselect_1.2.1
## [46] digest_0.6.39      stringi_1.8.7      dplyr_1.2.1        reshape2_1.4.5     fastmap_1.2.0
## [51] grid_4.6.0         colorspace_2.1-2   cli_3.6.6          magrittr_2.0.5     base64enc_0.1-6
## [56] dichromat_2.0-0.1 pbivnorm_0.6.0     foreign_0.8-91     corpcor_1.6.10     scales_1.4.0
## [61] backports_1.5.1    rmarkdown_2.31     jpeg_0.1-11        igraph_2.3.1       otel_0.2.0
## [66] nnet_7.3-20        gridExtra_2.3      png_0.1-9          pbapply_1.7-4      evaluate_1.0.5
## [71] knitr_1.51         viridisLite_0.4.3 rlang_1.2.0        Rcpp_1.1.1-1.1     glue_1.8.1
## [76] rstudioapi_0.18.0 R6_2.6.1           plyr_1.8.9
```

References

Brazma, Alvis, and Jaak Vilo. 2000. "Gene Expression Data Analysis." *FEBS Letters* 480 (1): 17–24.

Cheng, Yizong, and George M Church. 2000. "Biclustering of Expression Data." In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, edited by Philip Bourne, Michael Gribskov, Russ Altman, et al., vol. 8. AAAI Press, Menlo Park, CA.

Eren, Kemal, Mehmet Deveci, Onur Küçüktunç, and Ümit V. Çatalyürek. 2012. "A Comparative Analysis of Biclustering Algorithms for Gene Expression Data." *Briefings in Bioinformatics*, ahead of print. https://doi.org/10.1093/bib/bbs032.

Fehrmann, Rudolf SN, Juha M Karjalainen, Małgorzata Krajewska, et al. 2015. "Gene Expression Analysis Identifies Global Gene Dosage Sensitivity in Cancer." *Nature Genetics* 47 (2): 115–25.

Gu, Jiajun, and Jun S Liu. 2008. "Bayesian Biclustering of Gene Expression Data." *BMC Genomics* 9 (Suppl 1): S4.

Lazzeroni, Laura, Art Owen, et al. 2002. "Plaid Models for Gene Expression Data." *Statistica Sinica* 12 (1): 61–86.

Li, Guojun, Qin Ma, Haibao Tang, Andrew H Paterson, and Ying Xu. 2009. "QUBIC: A Qualitative Biclustering Algorithm for Analyses of Gene Expression Data." *Nucleic Acids Research* 37 (15): e101.

Szklarczyk, Damian, Andrea Franceschini, Stefan Wyder, et al. 2014. "STRING V10: Protein–Protein Interaction Networks, Integrated over the Tree of Life." *Nucleic Acids Research* 43: D447–52. https://doi.org/10.1093/nar/gku1003.

Zhang, Yu, Juan Xie, Jinyu Yang, Anne Fennell, Chi Zhang, and Qin Ma. 2017. "QUBIC: A Bioconductor Package for Qualitative Biclustering Analysis of Gene Co- Expression Data." *Bioinformatics* 33 (3): 450–52. https://doi.org/10.1093/bioinformatics/btw635.

Zhou, Fengfeng, Qin Ma, Guojun Li, and Ying Xu. 2012. "QServer: A Biclustering Server for Prediction and Assessment of Co-Expressed Gene Clusters." *PLoS ONE* 7 (3): e32660. https://doi.org/10.1371/journal.pone.0032660.