1
Using the DiffCorr Package to Analyze and Visualize Differential Correlations in Biological Networks

Atsushi Fukushima and Kozo Nishida

1.1
Introduction

1.1.1
An Introduction to Omics and Systems Biology

In this century, a high-throughput technology is being harnessed in various applications to solve a diverse range of biological problems and to explore biological phenomena. Next-generation sequencers (NGS) can be used for measuring and monitoring thousands of small molecules simultaneously [1–4] and large genomic sequences can be acquired quickly and routinely. RNA sequencing with NGS (RNA-seq) measures nearly every transcript of cellular systems (i.e., transcriptome) [5–7]. The term *omics* refers to the comprehensive analysis of biological systems and approaches including genomics, transcriptomics, and metabolomics that have become a promising way to inspect complex network interactions in cellular systems. To understand the organizing principle of cellular functions at different levels, an integrative approach with large-scale omics data including genomics, transcriptomics, proteomics, and metabolomics, is required [8–10]. Although it means different things to different scientists, systems biology [11] is the study of the behavior of complex biological processes using integrated approaches and a collection of omics-based data sets, quantitative measurements of the behavior of interacting cellular components, and mathematical/computational models to predict and describe complex dynamic behaviors.

1.1.2
Correlation Networks in Omics and Systems Biology

Molecular interactions can be expressed simply as a network by measuring associations among molecules in omics data (e.g., see [12, 13]). Typical network analysis is based on transcriptome data sets obtained from microarray experiments and

Computational Network Analysis with R: Applications in Biology, Medicine and Chemistry, First Edition.
Edited by Matthias Dehmer, Yongtang Shi, and Frank Emmert-Streib.

RNA-seq. This is known as *gene co-expression analysis* (e.g., see reviews [14–17]). Correlation relationships are special cases of association that can be measured by correlation-based measures such as the Pearson correlation coefficient, r (Figure 1.1a), which can range from −1 to 1, where $r = 1$ represents a perfect positive linear relationship between gene expressions, while $r = -1$ indicates a perfect negative relationship. While $r = 0$ indicates no *linear* relationship between gene expressions, it does not mean that two gene expressions are statistically independent. Calculation of the Pearson correlation coefficient is not robust for outliers and assumes that the data are from a standard normal distribution. On the other hand, the Spearman rank correlation coefficient is more robust with respect to outliers; it measures a monotonic relationship between gene expressions. If the correlation between two gene expressions exceeds a threshold, these genes can be considered as *co-expressed*. Such associations can be described as "co-expression networks" or generally as "correlation networks," where nodes represent genes and links between nodes represent significant correlations that are above a given threshold. Typical co-expression network analysis is based on the correlation coefficient between preselected gene(s) and the rest of the genes in a data set; this is called a *guide-gene approach* [18]. Although a correlation does not always indicate a causal relationship, a network approach can provide clues about the regulatory mechanisms that underlie the biological processes, and it has been used to characterize genes involved in plant-specialized secondary metabolisms [14, 17, 19].

1.1.3 Network Modules and Differential Network Approaches

When assessing gene co-expression network data generated from a high-throughput microarray system, one can visualize a giant network component from a large number of interactions (e.g., see [20]). There are many approaches for summarizing such large-scale networks: graph clustering [21] has been used and differential co-expressions or *differential correlations* [22] have been identified by means of network analysis using omics data. In general, graph clustering such as Markov clustering [23] and DPClus [24] can be used for detecting co-expressed modules or clusters in a nonbiased manner. Graph clustering is an algorithm for efficiently extracting densely connected genes in co-expression networks. This approach has also provided insights into transcriptional organization in *Arabidopsis thaliana* (*Arabidopsis*) and *Oryza sativa* (rice) as well as *Solanum lycopersicum* (tomato) [25–29]. In addition to the mean levels of abundance [the identification of so-called "differentially expressed genes (DEGs)" between two samples] and the detection of clustered molecules with similar profile patterns, changes in the correlation patterns between molecules, referred to as *differential correlations*, are also informative [22, 30]. Differential network approaches can be performed by comparing two different networks, for example, normal and disease networks (Figure 1.1b). This type of differential network strategy [31] has been applied to animals and plants [19, 22, 30, 32]. Differential correlation

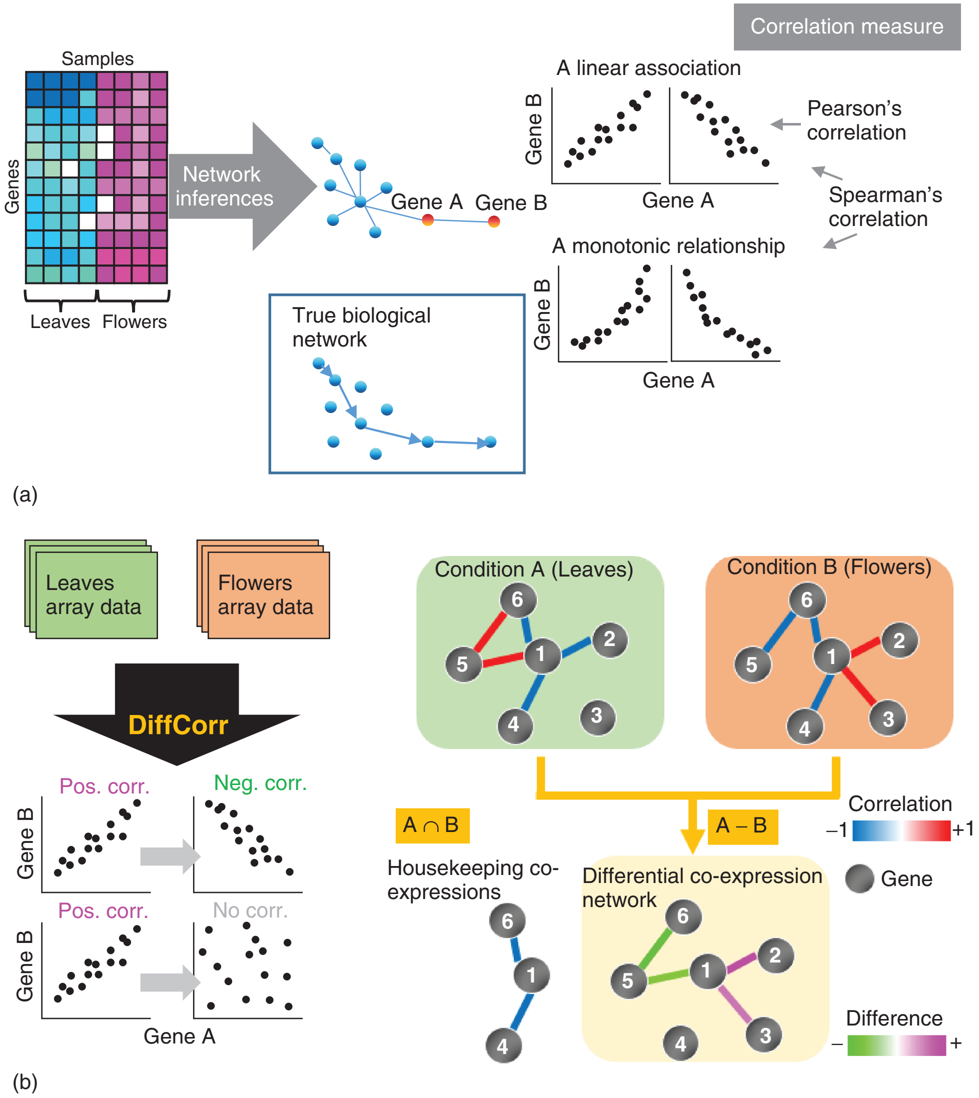


Figure 1.1 A gene–gene association measure and causal inferences in co-expression analysis. (a) Two kinds of major methods to measure the association between gene expressions. Although the Pearson correlation coefficient (PCC) is widely used in co-expression analysis in plant science, it can only be used to estimate a linear relationship between variables. A gene–gene association is not always a linear correlation. In general, information-theoretic measures can estimate a nonlinear relationship. Note that the Spearman correlation coefficient (SCC) can estimate a nonlinear relationship such as a monotonic function. (b) A concept of differential co-expression networks.

analysis in metabolomics has been used for dissecting complex metabolisms [33–35].

1.1.4 Aims of this Chapter

This chapter aims to (i) introduce the differential network concept in biological networks, (ii) demonstrate typical correlation network analysis using transcriptome and metabolome data sets, and (iii) highlight caveats in the correlation approach including the influence of the experimental setup used to generate correlation networks and the statistical approaches applied to assess these networks. We illustrate the utility of our DiffCorr package [36] by demonstrating biologically relevant, differentially correlated molecules in transcriptome co-expression and metabolite-to-metabolite correlation networks. The R code used in this chapter can be downloaded from the github repository: http://afukushima.github.io/diffcorrbook.

1.2 What is DiffCorr?

1.2.1 Background

There are a number of algorithms for detecting the differential correlation for large-scale omics data sets. Typical approaches for identifying differential correlations include topological overlap in a graph [37–40], extension of the traditional F-statistic [41], an additive model [42], Fisher's z-test [30, 36], an interaction score based on Renyi relative entropy [43], the Haar basis [32], the combination of the graphical Gaussian model and the posterior odds ratio [44], the liquid association concept [45, 46], a combination of robust correlations and hypothetical testing (called ROS-DET (RObust Switching mechanisms DETector)) [47], random resampling methods [48], graph-theoretic statistics [49], and an empirical Bayesian approach [50, 51]. Liu and coworkers implemented several of these methods to identify differential co-expressions in their R package DCGL [52, 53] (see also the review by Kayano *et al.* [54]). A tool to identify differential correlation patterns in omics data in an efficient and unbiased manner is needed. The simplest technique, based on Fisher's z-test of correlation coefficient to identify differential correlations, is not yet widely used and, to the best of our knowledge, is not implemented for omics data in the available R packages. We developed the DiffCorr package [36], a simple method for identifying pattern changes between two experimental conditions in correlation networks, which builds on a commonly used association measure, such as Pearson's correlation coefficient. DiffCorr calculates correlation matrices for each data set, identifies the first principal component-based “eigen-molecules” in the correlation networks, and tests differential correlations between the two groups based on Fisher’s z-test [36].

1.2.2 Methods

Fisher’s z-test was used to identify significant differences between two correlations based on its stringency test and its provision of conservative estimates of true differential correlations among molecules between two experimental conditions in the omics data [36]. To test whether the two correlation coefficients were significantly different, we first transformed the correlation coefficients for each of the two conditions, r_A and r_B, into Z_A and Z_B, respectively. The Fisher’s z-transformation of coefficient r_A is defined by $Z_\mathrm{A} = 1/2[\log(1 + r_\mathrm{A})/(1 - r_\mathrm{A})]$.

Similarly, we transform coefficient r_B into Z_B. Differences between the two correlations can be tested using the equation

$$Z = \frac{\frac{1}{2}\log\frac{1+r_\mathrm{A}}{1-r_\mathrm{A}} - \frac{1}{2}\log\frac{1+r_\mathrm{B}}{1-r_\mathrm{B}}}{\sqrt{\frac{1}{n_\mathrm{A}-3}+\frac{1}{n_\mathrm{B}-3}}} \tag{1.1}$$

where n_A and n_B represent the sample size for each of the conditions for each biomolecule pair [29, 33, 34]. The Z value has an approximately Gaussian distribution under the null hypothesis that the population correlations are equal. Controlling the false discovery rate (FDR) described by Benjamini and Hochberg [55] is a stringent and practical method in multiple testing problems. However, while it assumes all tests to be independent, this is not the case for correlation tests. We, therefore, used the local FDR derived from the fdrtool package [56]. DiffCorr can explore differential correlations between two conditions in the context of postgenomics data types, namely transcriptomics and metabolomics. DiffCorr is simple to use in calculating differential correlations and is suitable for the first step toward inferring causal relationships and detecting biomarker candidates. The package can be downloaded from the CRAN repository: http://cran.r-project.org/web/packages/DiffCorr/.

1.2.3 Main Functions in DiffCorr

Here, we describe the features, functionalities, and structure of the DiffCorr package [36]. Functions in the DiffCorr package can be divided into three main categories: (i) module detection, constructing correlation networks, and calculating the eigen-molecules for each condition; (ii) visualization of eigen-molecule networks; and (iii) export of the results of testing based on Fisher’s z-test (Figure 1.2).

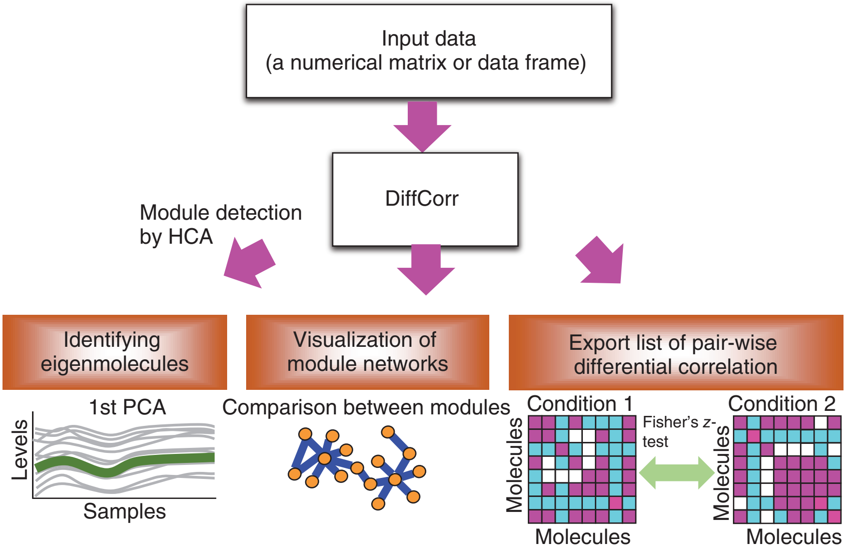


Figure 1.2 An overview of analysis steps and main functions in DiffCorr. An outline of the DiffCorr approach with the three main processes. HCA, hierarchical cluster analysis.

1) *get.eigen.molecule*: extracts conditional modules derived from hierarchical cluster analysis (HCA) using the cluster.molecule function. For the visualization of modules, get.eigen.molecule.graph also provides a graph object of eigengene [57] using the igraph package (http://igraph.org/).
2) *plot.DiffCorr.group*: draws module members for each condition. This function is based on the plot function using the igraph package (http://igraph.org/). This provides profile patterns of module members for each module.
3) *comp.2.cc.fdr*: exports a list of significantly differential correlations as a text file. This function uses the fdrtool package [56] to control the FDR. The resulting file contains molecule IDs (e.g., probe-set ID and metabolite name), conditional correlation coefficients, the *p*-values of the correlation test, the difference of the two correlations, the corresponding *p*-values, and the result of Fisher's *z*-test with control of the FDR. More detailed statistical descriptions for identifying differentially correlated molecules are in the next section.

1.2.4
Installing the DiffCorr Package

If the code is to be run while reading this chapter, the DiffCorr package must be installed from CRAN.

```
# If using Ubuntu, run "apt-get install libxml2-dev" first.
source("http://bioconductor.org/biocLite.R")
```

```
biocLite(c("pcaMethods", "multtest"))
install.packages("DiffCorr")
library(DiffCorr)
## Loading required package: pcaMethods
## Loading required package: Biobase
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: ’BiocGenerics’
##
## The following objects are masked from ’package:parallel’:
##
##     clusterApply, clusterApplyLB, clusterCall,
##     clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply,
##     parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
##
## The following object is masked from ’package:stats’:
##
##     xtabs
##
## The following objects are masked from ’package:base’:
##
##     anyDuplicated, append, as.data.frame, as.vector,
##     cbind, colnames, do.call, duplicated, eval, evalq,
##     Filter, Find, get,
##     intersect, is.unsorted, lapply, Map, mapply, match,
##     mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank,
##     rbind, Reduce, rep.int, rownames, sapply, setdiff,
##     sort,
##     table, tapply, union, unique, unlist, unsplit
##
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     ’browseVignettes()’. To cite Bioconductor, see
##     ’citation("Biobase")’, and for packages ’citation
##     ("pkgname")’.
##
##
## Attaching package: ’pcaMethods’
##
## The following object is masked from ’package:stats’:
```

```
##
##     loadings
##
## Loading required package: igraph
## Loading required package: fdrtool
## Loading required package: multtest
help(package="DiffCorr")
```

Please note R version 3.1.*. We use several Bioconductor [58] packages on the following pages. Some of them will not work if your R version is not consistent with the Bioconductor version. At the time of this writing (June 2015), Bioconductor release version (3.1) is not consistent with R release version (3.2).

To get started, install the following packages needed for this chapter.

```
biocLite("GEOquery")
biocLite("affy")
biocLite("genefilter")
biocLite("GOstats")
biocLite("ath1121501.db")

install.packages("spatstat")
install.packages("igraph")
```

1.3 Constructing Co-Expression (Correlation) Networks from Omics Data – Transcriptome Data set

In this section, we demonstrate the construction of co-expression networks using AtGenExpress development data sets [59]. AtGenExpress is a multinational project designed to quantify the transcriptome of the model plant *A. thaliana*; it contains a lot of Affymetrix ATH1 GeneChip (http://www.affymetrix.com/support/technical/datasheets/arab_datasheet.pdf). Our procedure described in this chapter has been applied not only to plants but also to bacteria and animals.

1.3.1 Downloading the Transcriptome Data set

We use data sets from leaf and flower samples from AtGenExpress development [59]. (NCBI Gene Expression Omnibus (GEO) [60] Accession: GSE5630 and GSE5632, respectively). For example, see the web site: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5632. To download the data sets, we accessed the NCBI GEO database via the GEOquery package [61]. NCBI GEO is a public repository for a wide range of high-throughput data such as transcriptome data sets [60]. It includes microarray-based experiments measuring mRNA, genomic DNA, and protein abundance, as well as nonarray techniques such as NGS data, serial analysis of gene expression (SAGE), and mass spectrometry proteomic data. The GEOquery package has a function *getGEOSuppFiles* to retrieve supplemental files to be attached to GEO Series (GSE), GEO platforms (GPL), and GEO samples (GSM). This function "knows" how to get these files based on the GEO accession. We can obtain the data sets as a raw CEL file and unpack them in the current directory or the current folder.

```
library("GEOquery")
## Setting options('download.file.method.GEOquery'='auto')
## AtGenExpress: Developmental series (flowers and pollen)
 ## Note that the data size is 143.9 Mb.
data <- getGEOSuppFiles("GSE5632")
untar("GSE5632/GSE5632_RAW.tar", exdir="GSE5632")

  ## AtGenExpress: Developmental series (leaves)
  ## Note that the data size is 127.5 Mb.
   data <- getGEOSuppFiles("GSE5630")
   untar("GSE5630/GSE5630_RAW.tar", exdir="GSE5630")
```

1.3.2 Data Filtering

Before calculation of the correlation relationships, all CEL files must be normalized to adjust technical variations between the arrays. Here, we use Robust Multichip Average (RMA) normalization via the affy package [62]. For more information, see Bolstad *et al.* [63].

```
library(affy)
## target files
tgt <- list.files("./GSE5630", pattern="*.CEL.gz",
full.names=TRUE)

## RMA normalization
eset.GSE5630 <- justRMA(filenames=tgt)
##
eset.GSE5630
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 22810 features, 60 samples
##   element names: exprs, se.exprs
## protocolData
##   sampleNames: GSM131495.CEL.gz GSM131496.CEL.gz …
##     GSM131554.CEL.gz (60 total)
##   varLabels: ScanDate
##   varMetadata: labelDescription
## phenoData
##   sampleNames: GSM131495.CEL.gz GSM131496.CEL.gz ...
##     GSM131554.CEL.gz (60 total)
##   varLabels: sample
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: ath1121501
dim(eset.GSE5630)
## Features  Samples
##    22810       60
tgt2 <- list.files("./GSE5632", pattern="*.CEL.gz",
full.names=TRUE)
## RMA normalization
eset.GSE5632 <- justRMA(filenames=tgt2)
eset.GSE5632
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 22810 features, 66 samples
##   element names: exprs, se.exprs
## protocolData
##   sampleNames: GSM131576.CEL.gz GSM131577.CEL.gz ...
##     GSM131641.CEL.gz (66 total)
##   varLabels: ScanDate
##   varMetadata: labelDescription
## phenoData
##   sampleNames: GSM131576.CEL.gz GSM131577.CEL.gz ...
##     GSM131641.CEL.gz (66 total)
##   varLabels: sample
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: ath1121501
dim(eset.GSE5632)
## Features  Samples
##    22810       66
```

Utilization of AFFX spike-in control probes to monitor sample throughout Affymetrix GeneChip. We discard all control probes. Data for all probe sets with the prefix “s_at” or “x_at” were also omitted as they may recognize transcripts from different genes, or cross-hybridization.

```
## filtering probesets with "AFFX", "s_at", and "x_at"
rmv <- c(grep("AFFX", rownames(eset.GSE5632)),
         grep("s_at", rownames(eset.GSE5632)),
         grep("x_at", rownames(eset.GSE5632))
         )
## The probe designs are the same between GSE5630 and
  GSE5632; 'rmv' can be re-used for GSE5630.
eset.GSE5632 <- eset.GSE5632[-rmv,]
dim(eset.GSE5632)
## Features  Samples
##    21685       66
eset.GSE5630 <- eset.GSE5630[-rmv, ]
dim(eset.GSE5630)
## Features  Samples
##    21685       60
```

1.3.3 Calculation of the Correlation and Visualization of Correlation Networks

For large-scale data matrices, computation of the correlation coefficient is very time-consuming and memory-filling. The following filter steps significantly reduce the number of targets for further statistical analyses via the genefilter package [64]. We use a filter function for the expression level and the coefficient of variation. The ratio of the standard deviation and the mean of a gene's expression values across all samples must be higher than a given threshold.

```
library(genefilter)
##
## Attaching package: 'genefilter'
##
## The following object is masked from 'package:base':
##
##     anyNA
## RMA returns normalized expression levels in log2 scale.
## Before applying the filter the values must be un-logged.

## GSE5632
e.mat <- 2^exprs(eset.GSE5632)
## filter: keep genes with cv between .5 and 10,
## and where 20% of samples had exprs. > 100
ffun <- filterfun(pOverA(0.2, 100), cv(0.5, 10))
filtered <- genefilter(e.mat,ffun)
# apply filter, and put expression back on log scale
eset.GSE5632.sub <- log2(e.mat[filtered, ])
dim(eset.GSE5632.sub)
## [1] 4262   66
## GSE5630
e.mat <- 2^exprs(eset.GSE5630)
```

```
ffun <- filterfun(pOverA(0.2, 100), cv(0.5, 10))
filtered <- genefilter(e.mat,ffun)
# apply filter, and put expression back on log scale
eset.GSE5630.sub <- log2(e.mat[filtered, ])
dim(eset.GSE5630.sub)
## [1] 1905   60
```

Next, we identify common probe sets between the two data sets.

```
#### common probesets between GSE5632 and GSE5630
comm <- intersect(rownames(eset.GSE5632.sub),
  rownames(eset.GSE5630.sub))
head(comm)
## [1] "244977_at" "245005_at" "245035_at" "245041_at"
  "245052_at" "245088_at"
length(comm)
## [1] 1224
eset.GSE5632.sub <- eset.GSE5632.sub[comm, ]  ## flowers
eset.GSE5630.sub <- eset.GSE5630.sub[comm, ]  ## leaves
dim(eset.GSE5630.sub)
## [1] 1224   60
dim(eset.GSE5632.sub)
## [1] 1224   66
```

We can obtain the correlation matrices for each data set by Spearman’s rank-order correlation, as in

```
## corr
GSE5632.cor <- cor(t(eset.GSE5632.sub), method="spearman")
GSE5630.cor <- cor(t(eset.GSE5630.sub), method="spearman")
```

Visualization on a pseudo-color heatmap is performed as follows (Figure 1.3).

```
library(spatstat)
##
## spatstat 1.41-1       (nickname: ’Ides of March’)
## For an introduction to spatstat, type ’beginner’
##
## Note: spatstat version 1.41-1 is out of date by more
  than 3 months; we recommend upgrading to the latest
  version.
##
## Attaching package: ’spatstat’
##
## The following object is masked from ’package:genefilter’:
##
```

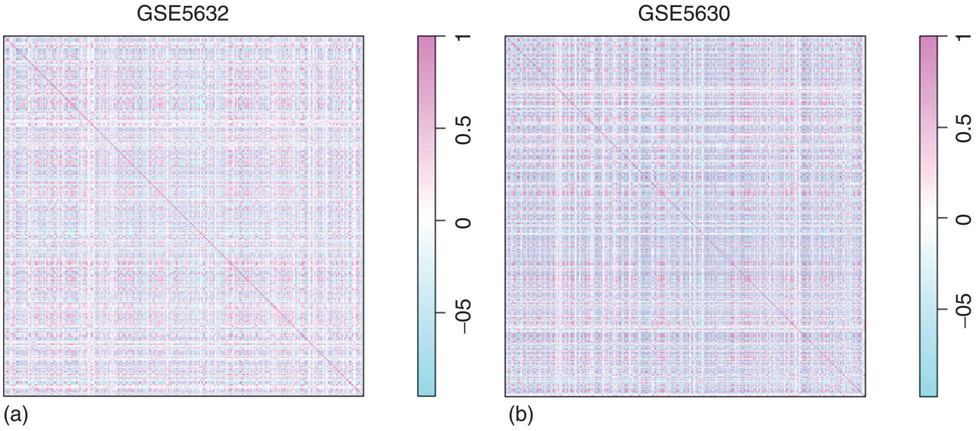
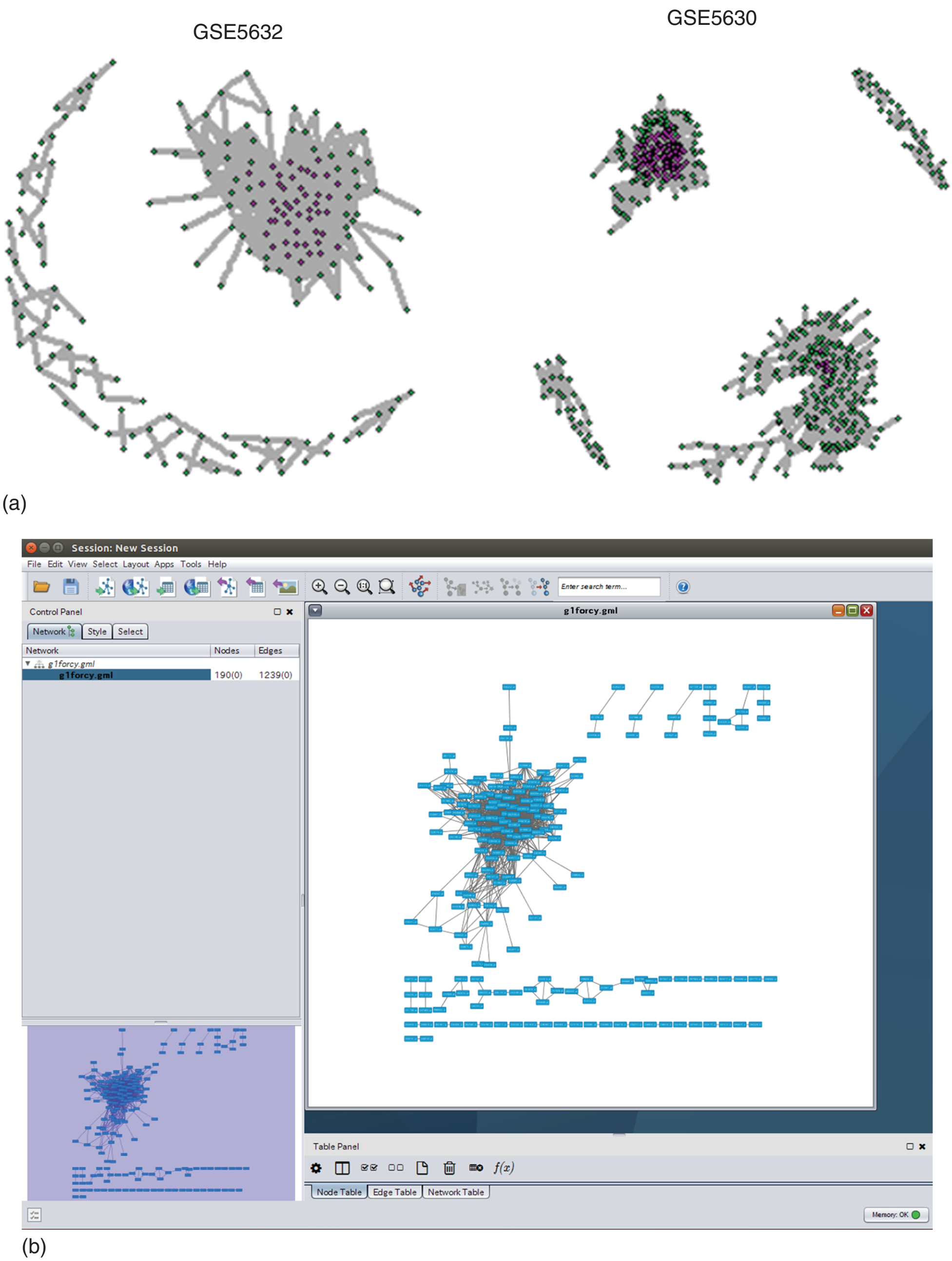


Figure 1.3 Heatmaps of the correlation matrices. Heatmaps of the gene expression correlation matrices. Horizontal and vertical show the probe set identifiers in each experiment. Pink = positive correlation, blue = negative correlation between the two probe sets.

```
##     area
##
## The following object is masked from 'package:affy':
##
##     intensity
##
## The following objects are masked from 'package:igraph':
##
##     diameter, edges, vertices
##
## The following object is masked from 'package:pcaMethods':
##
##     leverage
par(mfrow=c(1,2))
plot(im(GSE5632.cor[nrow(GSE5632.cor):1,]),
   col=cm.colors(256), main="GSE5632")
plot(im(GSE5630.cor[nrow(GSE5630.cor):1,]),
   col=cm.colors(256), main="GSE5630")
```

Construction of the co-expression networks can be started via the igraph package (http://igraph.org/) and they can be visualized (Figure 1.4a). The threshold value, $r_s \geq 0.95$, is set, as in

```
library(igraph)
## co-expression networks with GSE5632
# SCC >= 0.95
```



Network	Nodes	Edges
g1forcy.gml		
g1forcy.gml	190(0)	1239(0)

Figure 1.4 Correlation network visualization with the igraph package and Cytoscape. (a) Correlation networks with the igraph package. Nodes are the probe sets, and the edges mean that there are correlation coefficients over 0.95 between the connected nodes. We colored the nodes that are in the degree over 20 (magenta) and those that are not (green). (b) Correlation networks with Cytoscape [65]. Cytoscape has functionality to change the layout of the network interactively. Here, we applied yFiles [66] "Organic" layout to the network.

```
g1 <- graph.adjacency(GSE5632.cor, weighted=TRUE,
  mode="lower")
g1 <- delete.edges(g1, E(g1)[ weight < 0.95 ])
g1 <- igraph::simplify(g1, remove.multiple = TRUE, re-
move.loops = TRUE)
g1 <- delete.vertices(g1, which(igraph::degree(g1)<1))
plot(g1, vertex.size=3, edge.width=3, vertex.color=ifelse
  (igraph::degree(g1)>20,"Magenta","Green"),
  vertex.label="", layout=layout.kamada.kawai)

## co-expression networks with GSE5630
# SCC >= 0.95


g2 <- graph.adjacency(GSE5630.cor, weighted=TRUE,
  mode="lower")
g2 <- delete.edges(g2, E(g2)[ weight < 0.95 ])
g2 <- igraph::simplify(g2, remove.multiple = TRUE,
  remove.loops = TRUE)
g2 <- delete.vertices(g2, which(igraph::degree(g2)<1))
plot(g2, vertex.size=3, edge.width=3, vertex.color=ifelse
  (igraph::degree(g2)>20,"Magenta","Green"),
  vertex.label="", layout=layout.kamada.kawai)
```

The current *plot* function in the igraph package (http://igraph.org/) generates a static image and lacks interactivity. To explore the co-expression network in detail (e.g., zooming, panning, and viewing the weights by clicking), we put aside the R console for now and use Cytoscape [65]. Cytoscape is an open source software for visualizing networks and integrating the networks with any type of attribute data. By using Cytoscape, you can interactively explore the network and change the visual style (e.g., edge color and width) corresponding to the attribute data (e.g., edge weight). The igraph package can export igraph object to several types of graph formats. Here, we export igraph object as GML (Graph Modeling Language) and import GML to Cytoscape.

```
write.graph(g1, "g1forcy.gml", format="gml")
write.graph(g2, "g2forcy.gml", format="gml")
```

To import this GML, click the "Import Network From File" toolbar button in Cytoscape. You can easily change the network layout; here, we applied the yFiles [66] "Organic" layout to these two networks (Figure 1.4b).

1.3.4 Graph Clustering

Various graph clustering algorithms including Markov clustering [23] and DPClus [24] were applied in *Arabidopsis* and rice microarray data sets to find co-expression modules, clusters consisting of densely connected co-expressed genes [25–29]. Graph clustering algorithms include hierarchical clustering, density-based and local searches, and other optimization-based clustering [21]. Such network-module-based approaches are now widely used in attempts to predict new genes involved in biological processes [17, 67]. Other network-based approaches have been applied to annotate unknown genes [68], to explore possible genes involved in carbon/nitrogen-responsive machineries [69], and to prioritize candidate genes for a wide variety of traits [70]. We use a Fast Greedy modularity optimization algorithm [71] for finding gene co-expression modules. The igraph package implements this algorithm as a *fastgreedy.community* function. The algorithm runs in essentially linear time, $O(n \log^2 n)$, on a network with n vertices and reduces computation time.

```
g1.fc <- fastgreedy.community(g1)
sizes(g1.fc)
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## 41 43 23  5  4  3  3  3  3  2  2  2  4  5  2  2  2  2  2  2  2  2  3  2  2
## 26 27 28 29 30 31 32 33 34
##  3  3  2  4  2  2  2  4  2
g2.fc <- fastgreedy.community(g2)
sizes(g2.fc)
## Community sizes
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
## 195  63 112  12  16  15  11   5   5   2   2   2   2   5   2   2   2   2
##  19  20  21  22  23  24  25  26  27  28
##   2   2   3   2   3   2   2   2   2   2
```

We can access each module member easily as in

```
## accessing module 1 in GSE5632
mod1 <- membership(g1.fc)[membership(g1.fc)==1]
## extracting probeset names in module 1
mod1.p <- names(mod1)
## accessing module 2 in GSE5632
mod2 <- membership(g1.fc)[membership(g1.fc)==2]
mod2.p <- names(mod2)
## accessing module 3 in GSE5632
mod3 <- membership(g1.fc)[membership(g1.fc)==3]
mod3.p <- names(mod3)
```

We detected 34 modules (or communities) in the co-expression networks with GSE5632 (flower samples) and 28 modules in the co-expression networks with GSE5630 (leaf samples). We focus on subnetworks in the top three clusters of the graph clustering results. To assess cluster fidelity, Gene Ontology (GO) term enrichment analyses were performed.

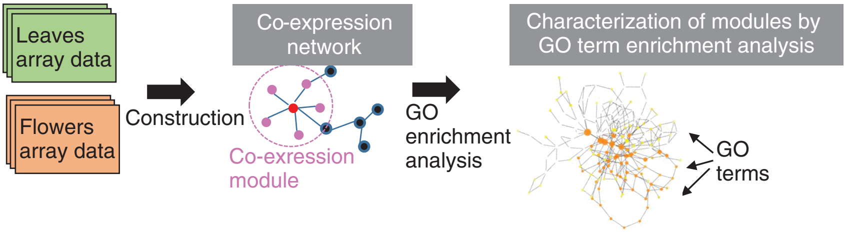


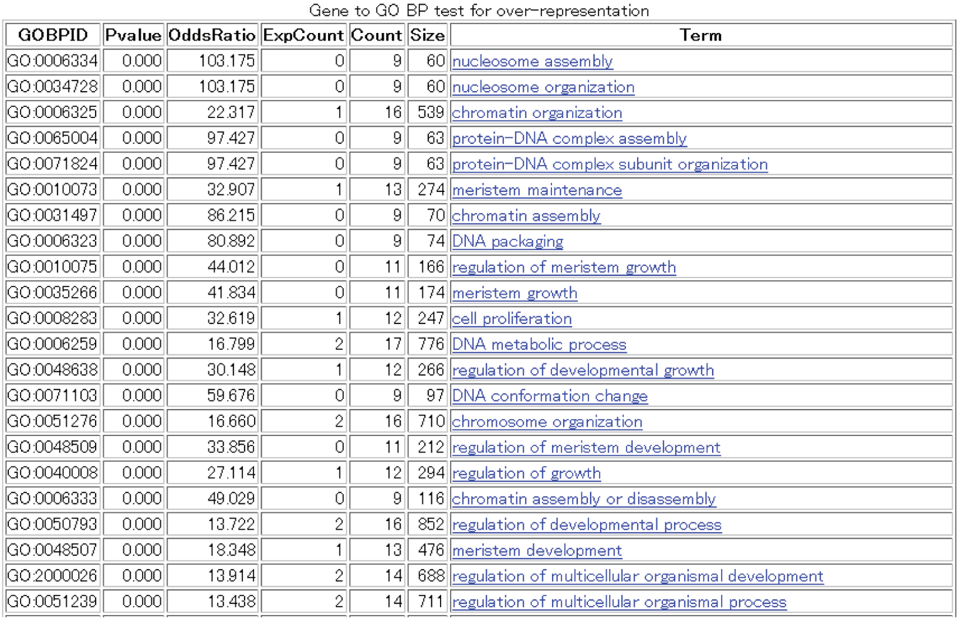
Figure 1.5 Workflow for constructing a co-expression network from microarray data and for evaluating detected network modules by Gene Ontology (GO) term enrichment analysis.

1.3.5 Gene Ontology Enrichment Analysis

Enrichment analysis can be combined with pathway analysis to evaluate whether a particular molecular group is significantly over- or underrepresented. Examples are gene set enrichment analysis [72] and other functional enrichment analyses using GO and biochemical pathways (for comprehensive reviews, see [73] or [74]). Here, we use the GOstats package [75] to perform GO term enrichment analysis of the detected co-expression modules (Figure 1.5). GOstats provides an easy-to-use set of functions for such enrichment analysis for GO terms.

```
library(GOstats)
## Loading required package: Category
## Loading required package: stats4
## Loading required package: Matrix
## Loading required package: AnnotationDbi
## Loading required package: GenomeInfoDb
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
##
## The following object is masked from 'package:igraph':
##
##     compare
##
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
##
## The following object is masked from 'package:Matrix':
##
##     expand
##
```

```
## The following objects are masked from ’package:spatstat’:
##
##     reflect, shift
##
## The following object is masked from ’package:igraph’:
##
##     simplify
##
## Loading required package: GO.db
## Loading required package: DBI
##
## Loading required package: graph
##
## Attaching package: ’graph’
##
## The following object is masked from ’package:spatstat’:
##
##     edges
##
## The following objects are masked from ’package:igraph’:
##
##     degree, edges
##
##
## Attaching package: ’GOstats’
##
## The following object is masked from
  ’package:AnnotationDbi’:
##
##     makeGOGraph
library(GO.db)
library(ath1121501.db)
## Loading required package: org.At.tair.db
ls("package:ath1121501.db")
##  [1] "ath1121501"             "ath1121501.db"
##  [3] "ath1121501_dbconn"      "ath1121501_dbfile"
##  [5] "ath1121501_dbInfo"      "ath1121501_dbschema"
##  [7] "ath1121501ACCNUM"       "ath1121501ARACYC"
##  [9] "ath1121501ARACYCENZYME" "ath1121501CHR"
## [11] "ath1121501CHRLENGTHS"   "ath1121501CHRLOC"
## [13] "ath1121501CHRLOCEND"    "ath1121501ENZYME"
## [15] "ath1121501ENZYME2PROBE" "ath1121501GENENAME"
## [17] "ath1121501GO"           "ath1121501GO2ALLPROBES"
## [19] "ath1121501GO2PROBE"     "ath1121501MAPCOUNTS"
## [21] "ath1121501ORGANISM"     "ath1121501ORGPKG"
## [23] "ath1121501PATH"          "ath1121501PATH2PROBE"
## [25] "ath1121501PMID"          "ath1121501PMID2PROBE"
## [27] "ath1121501SYMBOL"
?ath1121501ACCNUM
## starting httpd help server ... done
## gene universe
x <- ath1121501ACCNUM
mapped.probes <- mappedkeys(x)
length(mapped.probes)
## [1] 20335
geneUniv <- AnnotationDbi::as.list(x[mapped.probes])

## target probes
mod1.p.gene <- unique(unlist(AnnotationDbi::as.list
  (x[mod1.p])))
mod2.p.gene <- unique(unlist(AnnotationDbi::as.list
  (x[mod2.p])))
mod3.p.gene <- unique(unlist(AnnotationDbi::as.list
  (x[mod3.p])))

## mod1
hgCutoff <- 0.0001
params <- new("GOHyperGParams",
              geneIds=mod1.p.gene,
              universeGeneIds=geneUniv,
              annotation="ath1121501",
              ontology="BP",
              pvalueCutoff=hgCutoff,
              conditional=FALSE,
              testDirection="over")
## Warning in makeValidParams(.Object): converting univ
  from list to atomic
## vector via unlist
## Warning in makeValidParams(.Object): removing
  duplicate IDs in
## universeGeneIds
hgOver <- hyperGTest(params)

df <- summary(hgOver)
names(df)
## [1] "GOBPID"     "Pvalue"     "OddsRatio" "ExpCount"
  "Count"      "Size"
## [7] "Term"
pvalues(hgOver)[1:3]
##   GO:0006334   GO:0034728   GO:0006325
## 5.932371e-15 5.932371e-15 8.885765e-15
## reporting the results by GO term enrichment analysis
htmlReport(hgOver, file="res_mod1.html")
## enriched gene with "nucleosome assembly" terms in mod1

## mod2
params <- new("GOHyperGParams",
              geneIds=mod2.p.gene,
              universeGeneIds=geneUniv,
              annotation="ath1121501",
              ontology="BP",
              pvalueCutoff=hgCutoff,
              conditional=FALSE,
              testDirection="over")
## Warning in makeValidParams(.Object): converting univ
  from list to atomic
## vector via unlist
## Warning in makeValidParams(.Object): removing duplicate
IDs in
## universeGeneIds
hgOver <- hyperGTest(params)
## reporting the results by GO term enrichment analysis
htmlReport(hgOver, file="res_mod2.html")
## enriched gene with "cell proliferation" terms in mod2

## mod3
params <- new("GOHyperGParams",
              geneIds=mod3.p.gene,
              universeGeneIds=geneUniv,
              annotation="ath1121501",
              ontology="BP",
              pvalueCutoff=hgCutoff,
              conditional=FALSE,
              testDirection="over")
## Warning in makeValidParams(.Object): converting univ
  from list to atomic
## vector via unlist
## Warning in makeValidParams(.Object):
  removing duplicate IDs in
## universeGeneIds
hgOver <- hyperGTest(params)
## reporting the results by GO term enrichment analysis
htmlReport(hgOver, file="res_mod3.html")
## enriched gene with "RNA methylation" terms in mod3
```



Gene to GO BP test for over-representation

GOBPID	Pvalue	OddsRatio	ExpCount	Count	Size	Term
GO:0006334	0.000	103.175	0	9	60	nucleosome assembly
GO:0034728	0.000	103.175	0	9	60	nucleosome organization
GO:0006325	0.000	22.317	1	16	539	chromatin organization
GO:0065004	0.000	97.427	0	9	63	protein-DNA complex assembly
GO:0071824	0.000	97.427	0	9	63	protein-DNA complex subunit organization
GO:0010073	0.000	32.907	1	13	274	meristem maintenance
GO:0031497	0.000	86.215	0	9	70	chromatin assembly
GO:0006323	0.000	80.892	0	9	74	DNA packaging
GO:0010075	0.000	44.012	0	11	166	regulation of meristem growth
GO:0035266	0.000	41.834	0	11	174	meristem growth
GO:0008283	0.000	32.619	1	12	247	cell proliferation
GO:0006259	0.000	16.799	2	17	776	DNA metabolic process
GO:0048638	0.000	30.148	1	12	266	regulation of developmental growth
GO:0071103	0.000	59.676	0	9	97	DNA conformation change
GO:0051276	0.000	16.660	2	16	710	chromosome organization
GO:0048509	0.000	33.856	0	11	212	regulation of meristem development
GO:0040008	0.000	27.114	1	12	294	regulation of growth
GO:0006333	0.000	49.029	0	9	116	chromatin assembly or disassembly
GO:0050793	0.000	13.722	2	16	852	regulation of developmental process
GO:0048507	0.000	18.348	1	13	476	meristem development
GO:2000026	0.000	13.914	2	14	688	regulation of multicellular organismal development
GO:0051239	0.000	13.438	2	14	711	regulation of multicellular organismal process

Figure 1.6 HTML report of Gene Ontology (GO) enrichment analysis. Results of network Module 1 by GO enrichment analysis (filename: res_mod1.html). GO biological process ontology terms are listed in order of predominance in the cluster module.

Please see the resultant HTML files by using a web browser. The predominant function in the biological process within the three modules was assessed. Module# 1 using flower samples (GSE5632) was involved in "nucleosome assembly" within the "Biological Process" domain. Modules 2 and 3 were related to "cell proliferation" and "RNA methylation," respectively (Figure 1.6).

1.4 Differential Correlation Analysis by DiffCorr Package

1.4.1 Calculation of Differential Co-Expression between Organs in Arabidopsis

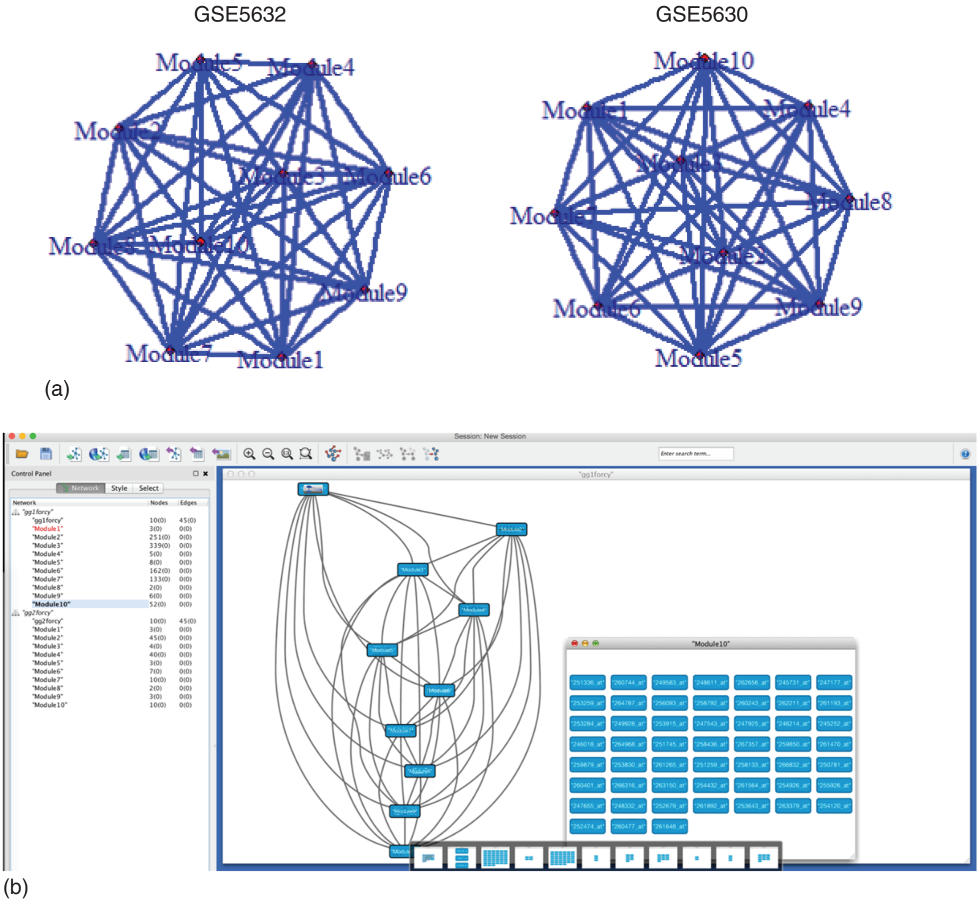
We calculate differential co-expression between leaf and flower samples in AtGenExpress development [59]. To test whether two correlated modules in co-expression networks are significantly different, we first calculate the eigen-molecule or "eigengene" [57] in the network as a representative correlation pattern within each module. The eigen-molecule is based on the first principal component (PC) of a data matrix of a module extracted from HCA using the *hclust* function in R. The *get.eigen.molecule* function uses the pcaMethods package [76] to perform principal component analysis (PCA) and returns the top 10

PCs (default). Using these eigen-molecule modules, we can also test differential correlations between modules in addition to pairwise differential correlations between molecules (Figure 1.7a).

```
## Clusters on each subset
dim(eset.GSE5632.sub)
## [1] 1224   66
dim(eset.GSE5630.sub)
## [1] 1224   60
data <- cbind(eset.GSE5632.sub, eset.GSE5630.sub)
hc.flowers <- cluster.molecule(data[, 1:66],
  method="pearson", linkage="average")  ## 66
  flowers samples
hc.leaves <- cluster.molecule(data[, 67:126],
  method="pearson", linkage="average")  ## 60
  leaves samples

## Cut the tree at a correlation of 0.6 using cutree
  function
#library(dynamicTreeCut)
g1 <- cutree(hc.flowers, h=0.4)
g2 <- cutree(hc.leaves, h=0.4)
```

```
table(g1[table(g1)!=1])
##    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##    2 185 242   4   6 121 104   2   4  39   6  12   5   3   8   2   2   4
##   19  20  21  22  23  24  25  28  29  30  31  32  33  34  35  36  37  38
##    7   1   7   2  10   9   4   1   5   1   8   1   1   1   2   4   3   1
##   39  40  41  42  43  44  45  46  47  48  50  51  52  53  54  55  57  58
##    3   3   9   2   3   2   2   3   1   1   2   3  11   3   1   3   2   1
##   59  60  61  62  63  64  65  66  67  68  69  70  71  73  75  76  77
##    1   1   2   1   1   2   1   1   1   1   1   1   1   1   1   1   2
table(g1)
## g1
##    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##    3 251 339   5   8 162 133   2   6  52   7  14   6   6  11   6   2   6
##   19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##    7   1  11   2  16  10   8   2   2   1   7   1   9   3   1   3   2   5
##   37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##    3   2   4   7  17   2   4   3   3   4   1   2   2   2   5  15   4   1
##   55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##    4   1   2   2   1   1   2   1   1   2   2   1   1   1   1   2   1   1
##   73  74  75  76  77
##    1   1   1   1   2
table(g2)
## g2
##    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##    3   1 279  10 271 392  13   1   4  13   3  45   4  40   3   7  10   2
##   19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   3  10   7   9   1   9   1   8   6   3   2   4   4   4   2   3   3   4
##  37  38  39  40  41  42  43  44  45  46  47  48  49
##   2   6   2   2   1   6   1   7   3   4   3   1   2

##
res1 <- get.eigen.molecule(data, groups=g1,
  whichgroups=c(1:10), methods="svd", n=2)
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
res2 <- get.eigen.molecule(data, groups=g2,
  whichgroups=c(11:20), methods="svd", n=2)
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
####################################
## Visualizing module networks
####################################
gg1 <- get.eigen.molecule.graph(res1)
plot(gg1, layout=layout.fruchterman.reingold(gg1))

write.modules(g1, res1, outfile="module1_list.txt")

gg2 <- get.eigen.molecule.graph(res2)
plot(gg2, layout=layout.fruchterman.reingold(gg2))

write.modules(g2, res2, outfile="module2_list.txt")
```

R *plot* function still lacks interactivity here. However, you might want to see the nodes in the modules in the same network view. Here, we also use Cytoscape [65] to visualize the module network with nested network file format (NNF). For more details about NNF, please refer to the Cytoscape user manual (http://



Network	Nodes	Edges
"gg1forcy"		
"gg1forcy"	10(0)	45(0)
"Module1"	3(0)	0(0)
"Module2"	251(0)	0(0)
"Module3"	339(0)	0(0)
"Module4"	5(0)	0(0)
"Module5"	8(0)	0(0)
"Module6"	162(0)	0(0)
"Module7"	133(0)	0(0)
"Module8"	2(0)	0(0)
"Module9"	6(0)	0(0)
"Module10"	52(0)	0(0)
"gg2forcy"		
"gg2forcy"	10(0)	45(0)
"Module1"	3(0)	0(0)
"Module2"	45(0)	0(0)
"Module3"	4(0)	0(0)
"Module4"	40(0)	0(0)
"Module5"	3(0)	0(0)
"Module6"	7(0)	0(0)
"Module7"	10(0)	0(0)
"Module8"	2(0)	0(0)
"Module9"	3(0)	0(0)
"Module10"	10(0)	0(0)

"Module10"

"251336_at"	"260744_at"	"249583_at"	"248611_at"	"262656_at"	"245731_at"	"247177_at"
"253259_at"	"264787_at"	"256093_at"	"258792_at"	"260243_at"	"262211_at"	"261193_at"
"253284_at"	"249928_at"	"253915_at"	"247543_at"	"247925_at"	"246214_at"	"245252_at"
"246018_at"	"264968_at"	"251745_at"	"258436_at"	"267357_at"	"259850_at"	"261470_at"
"259879_at"	"253830_at"	"261265_at"	"251259_at"	"258133_at"	"266832_at"	"250781_at"
"260401_at"	"266316_at"	"263150_at"	"254432_at"	"261564_at"	"254926_at"	"255926_at"
"247655_at"	"248332_at"	"252679_at"	"261892_at"	"253643_at"	"263379_at"	"254120_at"
"252474_at"	"260477_at"	"261648_at"				

Figure 1.7 Module network visualization with the DiffCorr package and Cytoscape. (a) Differentially co-expressed module networks with the DiffCorr package. Nodes are the probe set modules; the edges mean that there is a significant difference in co-expression between two nodes. (b) Differentially co-expressed module networks with Cytoscape. To explore panel (a) interactively, we imported panel (a) to Cytoscape and applied yFiles “Hierarchic” layout to the network.

manual.cytoscape.org/en/latest/Supported_Network_File_Formats.html#nnf). You can see the nodes in the modules and change the layout when you import the NNF to Cytoscape (Figure 1.7b).

```
write.graph(gg1, "tmp1.ncol", format="ncol")
write.graph(gg2, "tmp2.ncol", format="ncol")
tmp1 <- read.table("tmp1.ncol")
tmp2 <- read.table("tmp2.ncol")
tmp1$V3 <- "pp"
tmp2$V3 <- "pp"
```

```
tmp1$V4 <- "gg1forcy"
tmp2$V4 <- "gg2forcy"
tmp1 <- tmp1[, c("V4", "V1", "V3", "V2")]
tmp2 <- tmp2[, c("V4", "V1", "V3", "V2")]
write.table(tmp1, file="gg1forcy.nnf", row.names=FALSE,
  col.names=FALSE)
write.table(tmp2, file="gg2forcy.nnf", row.names=FALSE,
  col.names=FALSE)
module1_list <- read.table("module1_list.txt", skip=1)
module2_list <- read.table("module2_list.txt", skip=1)
module1_list$V1 <- sub("^", "Module", module1_list$V1)
module2_list$V1 <- sub("^", "Module", module2_list$V1 -
10)
write.table(module1_list, file="gg1forcy.nnf", append=TRUE,
  row.names=FALSE, col.names=FALSE)
write.table(module2_list, file="gg2forcy.nnf", append=TRUE,
  row.names=FALSE, col.names=FALSE)
```

You can inspect groups of interest graphically. For example, we look at groups 21 and 24.

```
plotDiffCorrGroup(data, g1, g2, 21, 24, 1:66, 67:126,
                  scale.center=TRUE, scale.scale=TRUE,
                  ylim=c(-5,5))
```

The genes were grouped according to their expression patterns in each subtype (flower or leaf samples) using the cluster.molecule function. We used (1 – correlation coefficient) as a distance measure (the cutoff value was a coefficient of 0.6) based on the *cutree* function. We then visualized the module network using the *get.eigen.molecule* and *get.eigen.molecule.graph* functions (Figure 1.8).

The *comp.2.cc.fdr* function provides the resulting pairwise differential co-expressions from a data set.

```
## Export the results (FDR < 0.05)
comp.2.cc.fdr(output.file="Transcript_DiffCorr_res.txt",
  data[,1:66], data[,67:126], threshold=0.05)
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and
  eta0
## Step 3... compute p-values and estimate empirical
  PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting

##
```

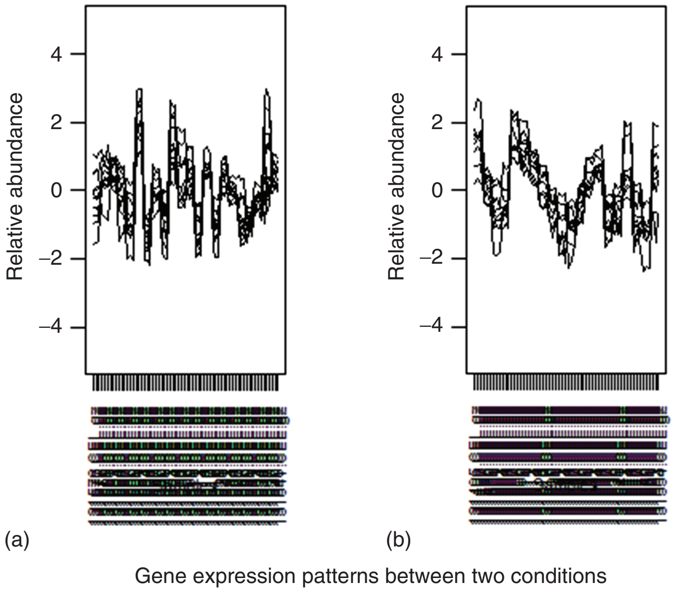


Figure 1.8 An example of gene expression patterns between two conditions.

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and
  eta0
## Step 3... compute p-values and estimate empirical
  PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting

##
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and
  eta0
## Step 3... compute p-values and estimate empirical
  PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting
```

Regarding local fdr methods, see the original paper [56].

1.4.2 Exploring the Metabolome Data of Flavonoid-Deficient Arabidopsis

Flavonoid-deficient and wild-type *Arabidopsis* has been investigated using gas chromatography coupled with mass spectrometry (GC–MS)-based metabolite profiling [33, 77]. The mutant lacks gene encoding chalcone synthase (CHS, EC 2.3.1.74), a key enzyme of the flavonoid biosynthesis pathway. The CHS mutant *transparent testa 4* (*tt4*) cannot synthesize any flavonoids, plant secondary metabolites that function as protectants against ultraviolet B (UV-B) irradiation. This data set from Kusano *et al.* [77] consists of the metabolite profiles of 37 aerial part samples, including two genotypes: 17 Columbia-0 wild-type (Col-0) and 20 *tt4* plants. The data also contain a wide-range of primary metabolites including amino acids, organic acids, fatty acids, sugars, and sugar alcohols. The metabolome data set is available in the DiffCorr package, as in

```
data(AraMetLeaves)
dim(AraMetLeaves)
## [1] 59 50
```

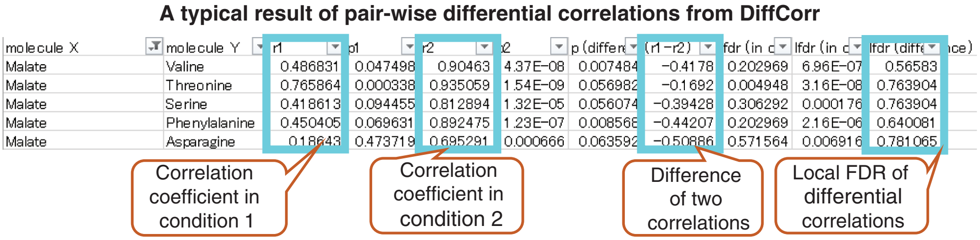
The data matrix, AraMetLeaves, contains 59 metabolites (rows) and 50 observations (columns). For a comparison with data from the aerial parts [77], we selected 59 commonly detected metabolites in both data sets using MetMask (http://metmask.sourceforge.net) [78]. Note that another genotype, called *mto1* (*methionine overaccumulation 1*), exists in the data matrix. For more details, see also the help page of AraMetLeaves.

```
colnames(AraMetLeaves)
##  [1] "Col0.1"  "Col0.2"  "Col0.3"  "Col0.4"  "Col0.5"  "Col0.6"  "Col0.7"
##  [8] "Col0.8"  "Col0.9"  "Col0.10" "Col0.11" "Col0.12" "Col0.13" "Col0.14"
## [15] "Col0.15" "Col0.16" "Col0.17" "tt4.1"   "tt4.2"   "tt4.3"   "tt4.4"
## [22] "tt4.5"   "tt4.6"   "tt4.7"   "tt4.8"   "tt4.9"   "tt4.10"  "tt4.11"
## [29] "tt4.12"  "tt4.13"  "tt4.14"  "tt4.15"  "tt4.16"  "tt4.17"  "tt4.18"
## [36] "tt4.19"  "tt4.20"  "mto1.1"  "mto1.2"  "mto1.3"  "mto1.4"  "mto1.5"
## [43] "mto1.6"  "mto1.7"  "mto1.8"  "mto1.9"  "mto1.10" "mto1.11" "mto1.12"
## [50] "mto1.13"
?AraMetLeaves
```

The differential correlation between *tt4* and Col-0 can be obtained as follows:

```
comp.2.cc.fdr(output.file="Met_DiffCorr_res.txt",
              log10(AraMetLeaves[,1:17]),
  ## Col-0 (17 samples)
              log10(AraMetLeaves[,18:37]),
  ## tt4 (20 samples)
              method="pearson",
              threshold=1.0)
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and
  eta0
## Step 3... compute p-values and estimate empirical
  PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting

##
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and
•   eta0
## Step 3... compute p-values and estimate empirical
•   PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting

##
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and
•   eta0
## Step 3... compute p-values and estimate empirical
•   PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting
```

As shown in the result, ASCII file "Met_DiffCorr_res.txt," the DiffCorr package detected significant differential correlations between sinapate and aromatic metabolites in *tt4* and wild-type plants (Figure 1.9). As reported previously [77], aromatic metabolites in the shikimate pathway, namely sinapate, phenylalanine (Phe), and tyrosine (Tyr), were significantly correlated in *tt4* but not in wild-type plants. This implies a linkage with the role of sinapoyl-malate against UV-B irradiation in the flavonoid-less *tt4* mutant (Figure 1.10). We showed that *Arabidopsis* attempts to compensate for a deficiency in either flavonoid or sinapoyl-malate production by over-accumulating alternative protectants [79]. These results suggest that DiffCorr can be applied to not only transcriptomic data, but also to other postgenomics data types including metabolomic data.


A typical result of pair-wise differential correlations from DiffCorr
Correlation coefficient in condition 1
Correlation coefficient in condition 2
Difference of two correlations
Local FDR of differential correlations

molecule X	molecule Y	r1	p1	r2	p2	p (differe	(r1−r2)	fdr (in c	lfdr (in c	lfdr (diffe nce)
Malate	Valine	0.486831	0.047498	0.90463	4.37E−08	0.007484	−0.4178	0.202969	6.96E−07	0.56583
Malate	Threonine	0.765864	0.000338	0.935059	1.54E−09	0.056982	−0.1692	0.004948	3.16E−08	0.763904
Malate	Serine	0.418613	0.094455	0.812894	1.32E−05	0.056074	−0.39428	0.306292	0.000176	0.763904
Malate	Phenylalanine	0.450405	0.069631	0.892475	1.23E−07	0.008568	−0.44207	0.202969	2.16E−06	0.640081
Malate	Asparagine	0.18643	0.473719	0.695291	0.000666	0.063592	−0.50886	0.571564	0.006916	0.781065

Figure 1.9 A typical result of pairwise differential correlations from the DiffCorr package.

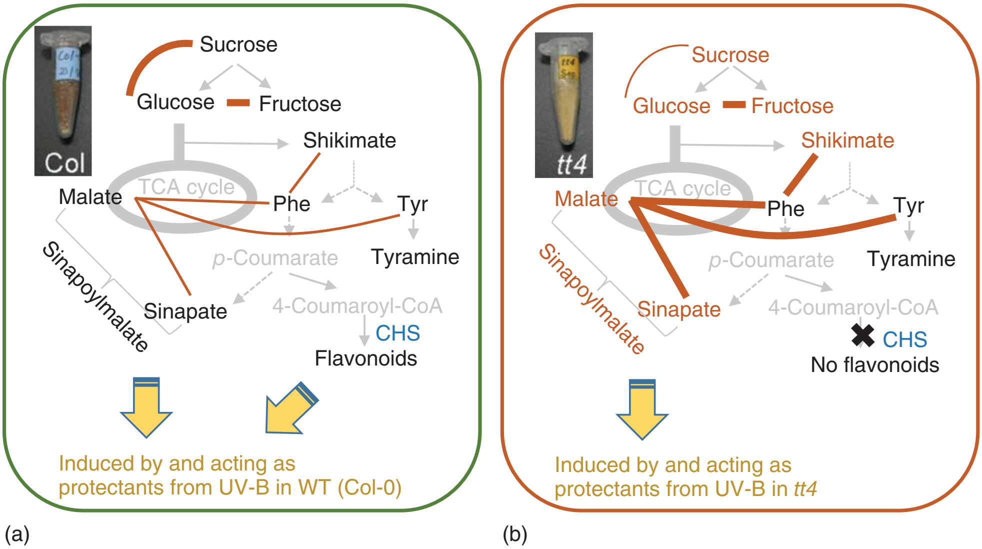


Figure 1.10 Interconnections in the pathways of central metabolism and aromatic amino acids for WT (a) and *tt4* plants (b). The metabolites whose levels changed less than 10% ($p < 0.05$) in *tt4*/WT are indicated with orange characters in (b); the metabolites with black characters in (b) exhibited no significant changes; the metabolites with gray characters were undetectable. The gray arrows indicate the metabolic pathways. The curved lines show correlations between metabolite pairs. The thickness of the edges between the metabolites represents the significance of correlations ($r_{\mathrm{Met}} > 0.88$). Although the sinapoylmalate level in *tt4* did not increase, correlations of malate with aromatic compounds were intensified in the *tt4* mutant, indicating a possible adaptive response to UV stress by the flavonoid-deficient *tt4* mutant by reconfiguration of the networks in *tt4*. Inset photo images show that *tt4* plants exhibit yellow seed color due to the nonaccumulation of proanthocyanidins in the seed coat. Abbreviations: CHS, chalcone synthase; *p*-coumarate, 4-hydroxycinnamic acid; Phe, phenylalanine; and Tyr, tyrosine.

1.4.3 Avoiding Pitfalls in (Differential) Correlation Analysis

The methods described here may allow researchers to gain a deeper understanding of condition-associated changes in molecular expression patterns (e.g., a gene co-expression) beyond the differential expression seen under two conditions. In transcriptome analysis using microarrays and RNA-seq, typical experiments are performed with a small number, for example, three biological replicates. In general, correlation analysis including differential correlation requires large sample sizes (e.g., larger than 10). The significance level (e.g., *p*- or local fdr values) of the correlation should be calculated to remove unreliable correlations because Pearson correlation outliers can have a strong effect on the estimation. Scatter plots must be used to visualize the overall pattern of correlations and each scatter plot must be inspected carefully. The correlation value must NEVER be used to show causation, because a correlation does not guarantee causation. Although it is a nontrivial task to identify causal regulatory systems from correlation patterns, the changes in correlation patterns provide a clue for important aspects of cellular regulations, indicating changes in regulatory systems across different physiological states.

1.5 Conclusion

With these example cases, we have described the power of the R package DiffCorr [36] to estimate differential networks in postgenomic data. This package affords users a simple and effective framework to detect differential correlations between two conditions in omics data. It is based on Fisher's z-test and makes it simple to calculate differential correlations. In this chapter, the concept of differential correlation approaches was introduced. We have described the background of our and some related works. We also highlighted the potential pitfalls in the correlation approach. The differential network approach is useful for the first step toward inferring causal relationships and for detecting biomarker candidates. DiffCorr based on the concept of "differential network biology" [22, 31] is suitable not only for transcriptomic and metabolomic data, but also for proteomic data, genome-wide association studies, and integrated omics data [8, 80]. In the near future, mining differential correlation patterns may be more significant biologically.

Acknowledgments

This work was partly supported by a Grant-in-Aid for Young Scientists (B; grant no. 26850024) from the Ministry of Education, Culture, Sports, Science, and Technology of Japan. We thank U. Petralia for editorial assistance.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Lister, R., Gregory, B.D., and Ecker, J.R. (2009) Next is now: new technologies for sequencing of genomes, transcriptomes, and beyond. *Curr. Opin. Plant Biol.*, **12** (2), 107–118.
2. Werner, T. (2010) Next generation sequencing in functional genomics. *Briefings Bioinf.*, **11** (5), 499–511.
3. Bolger, M.E., Weisshaar, B., Scholz, U., Stein, N., Usadel, B., and Mayer,

K.F. (2014) Plant genome sequencing - applications for crop improvement. *Curr. Opin. Biotechnol.*, **26**, 31–37.

4. Solomon, K.V., Haitjema, C.H., Thompson, D.A., and O'Malley, M.A. (2014) Extracting data from the muck: deriving biological insight from complex microbial communities and non-model organisms with next generation sequencing. *Curr. Opin. Biotechnol.*, **28**, 103–110.
5. Wang, Z., Gerstein, M., and Snyder, M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10** (1), 57–63.
6. Levin, J.Z., Yassour, M., Adiconis, X., Nusbaum, C., Thompson, D.A., Friedman, N., Gnirke, A., and Regev, A. (2010) Comprehensive comparative analysis of strand-specific RNA sequencing methods. *Nat. Methods*, **7** (9), 709–715.
7. Garber, M., Grabherr, M.G., Guttman, M., and Trapnell, C. (2011) Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat. Methods*, **8** (6), 469–477.
8. Fukushima, A., Kusano, M., Redestig, H., Arita, M., and Saito, K. (2009) Integrated omics approaches in plant systems biology. *Curr. Opin. Chem. Biol.*, **13** (5-6), 532–538.
9. Stitt, M. (2013) Systems-integration of plant metabolism: means, motive and opportunity. *Curr. Opin. Plant Biol.*, **16** (3), 381–388.
10. Sweetlove, L.J., Obata, T., and Fernie, A.R. (2013) Systems analysis of metabolic phenotypes: what have we learnt? *Trends Plant Sci.*, 2014 Apr; **19** (4), 222–30. doi: 10.1016/j.tplants.2013.09.005.
11. Kitano, H. (2002) Systems biology: a brief overview. *Science*, **295** (5560), 1662–1664.
12. Toubiana, D., Fernie, A.R., Nikoloski, Z., and Fait, A. (2013) Network analysis: tackling complex data to study plant metabolism. *Trends Biotechnol.*, **31** (1), 29–36.
13. Fukushima, A. and Kusano, M. (2014) A network perspective on nitrogen metabolism from model to crop plants using integrated 'omics' approaches. *J. Exp. Bot.*, **65** (19), 5619–5630.
14. Saito, K., Hirai, M.Y., and Yonekura-Sakakibara, K. (2008) Decoding genes with coexpression networks and metabolomics – 'majority report by precogs'. *Trends Plant Sci.*, **13** (1), 36–43.
15. Usadel, B., Obayashi, T., Mutwil, M., Giorgi, F.M., Bassel, G.W., Tanimoto, M., Chow, A., Steinhauser, D., Persson, S., and Provart, N.J. (2009) Co-expression tools for plant biology: opportunities for hypothesis generation and caveats. *Plant Cell Environ.*, **32** (12), 1633–1651.
16. Tohge, T. and Fernie, A.R. (2012) Co-expression and co-responses: within and beyond transcription. *Front. Plant Sci.*, **3**, 248.
17. Yonekura-Sakakibara, K., Fukushima, A., and Saito, K. (2013) Transcriptome data modeling for targeted plant metabolic engineering. *Curr. Opin. Biotechnol.*, **24** (2), 285–290.
18. Aoki, K., Ogata, Y., and Shibata, D. (2007) Approaches for extracting practical information from gene co-expression networks in plant biology. *Plant Cell Physiol.*, **48** (3), 381–390.
19. Higashi, Y. and Saito, K. (2013) Network analysis for gene discovery in plant-specialized metabolism. *Plant Cell Environ.*, **36** (9), 1597–1606.
20. Arabidopsis Interactome Mapping Consortium (2011) Evidence for network evolution in an *Arabidopsis* interactome map. *Science*, **333** (6042), 601–607.
21. Li, X., Wu, M., Kwoh, C.K., and Ng, S.K. (2010) Computational approaches for detecting protein complexes from protein interaction networks: a survey. *BMC Genomics*, **11** (Suppl. 1), S3.
22. de la Fuente, A. (2010) From 'differential expression' to 'differential networking' – identification of dysfunctional regulatory networks in diseases. *Trends Genet.*, **26** (7), 326–333.
23. Enright, A.J., Van Dongen, S., and Ouzounis, C.A. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30** (7), 1575–1584.
24. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., and Kanaya, S. (2006) Development and implementation of an algorithm for detection of protein

complexes in large interaction networks. *BMC Bioinf.*, **7**, 207.

25. Fukushima, A., Kanaya, S., and Arita, M. (2009) Characterizing gene coexpression modules in *Oryza sativa* based on a graph-clustering approach. *Plant Biotechnol.*, **26**, 485–493.
26. Ma, S., Gong, Q., and Bohnert, H.J. (2007) An Arabidopsis gene network based on the graphical Gaussian model. *Genome Res.*, **17** (11), 1614–1625.
27. Mentzen, W.I. and Wurtele, E.S. (2008) Regulon organization of *Arabidopsis*. *BMC Plant Biol.*, **8**, 99.
28. Mao, L., Van Hemert, J.L., Dash, S., and Dickerson, J.A. (2009) *Arabidopsis* gene co-expression network and its functional modules. *BMC Bioinf.*, **10**, 346.
29. Fukushima, A., Nishizawa, T., Hayakumo, M., Hikosaka, S., Saito, K., Goto, E., and Kusano, M. (2012) Exploring tomato gene functions based on coexpression modules using graph clustering and differential coexpression approaches. *Plant Physiol.*, **158** (4), 1487–1502.
30. 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** (24), 4348–4355.
31. Ideker, T. and Krogan, N.J. (2012) Differential network biology. *Mol. Syst. Biol.*, **8**, 565.
32. Gillis, J. and Pavlidis, P. (2009) A methodology for the analysis of differential coexpression across the human lifespan. *BMC Bioinf.*, **10**, 306.
33. Fukushima, A., Kusano, M., Redestig, H., Arita, M., and Saito, K. (2011) Metabolomic correlation-network modules in *Arabidopsis* based on a graph-clustering approach. *BMC Syst. Biol.*, **5**, 1.
34. Morgenthal, K., Weckwerth, W., and Steuer, R. (2006) Metabolomic networks in plants: transitions from pattern recognition to biological interpretation. *Biosystems*, **83** (2-3), 108–117.
35. Weckwerth, W., Loureiro, M.E., Wenzel, K., and Fiehn, O. (2004) Differential metabolic networks unravel the effects of silent plant phenotypes. *Proc. Natl. Acad. Sci. U.S.A.*, **101** (20), 7809–7814.
36. Fukushima, A. (2013) DiffCorr: an R package to analyze and visualize differential correlations in biological networks. *Gene*, **518** (1), 209–214.
37. Tesson, B.M., Breitling, R., and Jansen, R.C. (2010) DiffCoEx: a simple and sensitive method to find differentially coexpressed gene modules. *BMC Bioinf.*, **11**, 497.
38. Altay, G., Asim, M., Markowetz, F., and Neal, D.E. (2011) Differential C3NET reveals disease networks of direct physical interactions. *BMC Bioinf.*, **12**, 296.
39. Ray, M. and Zhang, W. (2010) Analysis of Alzheimer's disease severity across brain regions by topological analysis of gene co-expression networks. *BMC Syst. Biol.*, **4**, 136.
40. Yu, H., Liu, B.H., Ye, Z.Q., Li, C., Li, Y.X., and Li, Y.Y. (2011) Link-based quantitative methods to identify differentially coexpressed genes and gene pairs. *BMC Bioinf.*, **12**, 315.
41. Lai, Y., Wu, B., Chen, L., and Zhao, H. (2004) A statistical method for identifying differential gene-gene co-expression patterns. *Bioinformatics*, **20** (17), 3146–3155.
42. Kostka, D. and Spang, R. (2004) Finding disease specific alterations in the co-expression of genes. *Bioinformatics*, **20** (Suppl. 1), i194–i199.
43. Cho, S.B., Kim, J., and Kim, J.H. (2009) Identifying set-wise differential co-expression in gene expression microarray data. *BMC Bioinf.*, **10**, 109.
44. Chu, J.H., Lazarus, R., Carey, V.J., and Raby, B.A. (2011) Quantifying differential gene connectivity between disease states for objective identification of disease-relevant genes. *BMC Syst. Biol.*, **5**, 89.
45. Li, K.C. (2002) Genome-wide coexpression dynamics: theory and application. *Proc. Natl. Acad. Sci. U.S.A.*, **99** (26), 16875–16880.
46. Valcarcel, B., Wurtz, P., Seich al Basatena, N.K., Tukiainen, T., Kangas, A.J., Soininen, P., Jarvelin, M.R., Ala-Korpela, M., Ebbels, T.M., and de Iorio, M. (2011) A differential network approach to exploring differences between biological states: an application to prediabetes. *PLoS One*, **6** (9), e24702.

47. Kayano, M., Takigawa, I., Shiga, M., Tsuda, K., and Mamitsuka, H. (2011) ROS-DET: robust detector of switching mechanisms in gene expression. *Nucleic Acids Res.*, **39** (11), e74.

48. Watson, M. (2006) CoXpress: differential co-expression in gene expression data. *BMC Bioinf.*, **7**, 509.

49. Odibat, O. and Reddy, C.K. (2012) Ranking differential hubs in gene co-expression networks. *J. Bioinform. Comput. Biol.*, **10** (1), 1240002.

50. Dawson, J.A. and Kendziorski, C. (2012) An empirical Bayesian approach for identifying differential coexpression in high-throughput experiments. *Biometrics*, **68** (2), 455–465.

51. Dawson, J.A., Ye, S., and Kendziorski, C. (2012) R/EBcoexpress: an empirical Bayesian framework for discovering differential co-expression. *Bioinformatics*, **28** (14), 1939–1940.

52. Liu, B.H., Yu, H., Tu, K., Li, C., Li, Y.X., and Li, Y.Y. (2010) DCGL: an R package for identifying differentially coexpressed genes and links from gene expression microarray data. *Bioinformatics*, **26** (20), 2637–2638.

53. Yang, J., Yu, H., Liu, B.H., Zhao, Z., Liu, L., Ma, L.X., Li, Y.X., and Li, Y.Y. (2013) DCGL v2.0: an R package for unveiling differential regulation from differential co-expression. *PLoS One*, **8** (11), e79729.

54. Kayano, M., Shiga, M., and Mamitsuka, H. (2014) Detecting differentially coexpressed genes from labeled expression data: a brief review. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **11** (1), 154–167.

55. Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B*, **57** (1), 289–300.

56. Strimmer, K. (2008) fdrtool: a versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics*, **24** (12), 1461–1462.

57. Langfelder, P. and Horvath, S. (2007) Eigengene networks for studying the relationships between co-expression modules. *BMC Syst. Biol.*, **1**, 54.

58. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J. *et al* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, **5** (10), R80.

59. Schmid, M., Davison, T.S., Henz, S.R., Pape, U.J., Demar, M., Vingron, M., Scholkopf, B., Weigel, D., and Lohmann, J.U. (2005) A gene expression map of *Arabidopsis thaliana* development. *Nat. Genet.*, **37** (5), 501–506.

60. Barrett, T., Wilhite, S.E., Ledoux, P., Evangelista, C., Kim, I.F., Tomashevsky, M., Marshall, K.A., Phillippy, K.H., Sherman, P.M., Holko, M. *et al* (2013) NCBI GEO: archive for functional genomics data sets – update. *Nucleic Acids Res.*, **41** (Database issue), D991–D995.

61. Davis, S. and Meltzer, P.S. (2007) GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. *Bioinformatics*, **23** (14), 1846–1847.

62. Gautier, L., Cope, L., and Bolstad, B.M. (2004) Irizarry RA: affy – analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, **20** (3), 307–315.

63. Bolstad, B.M., Irizarry, R.A., Astrand, M., and Speed, T.P. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19** (2), 185–193.

64. Gentleman, R., Carey, V., Huber, W., and Hahne, F. (2015) genefilter: methods for filtering genes from high-throughput experiments. R package version 1.48.1.

65. Saito, R., Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.L., Lotia, S., Pico, A.R., Bader, G.D., and Ideker, T. (2012) A travel guide to Cytoscape plugins. *Nat. Methods*, **9** (11), 1069–1076.

66. Wiese, R., Eiglsperger, M., and Kaufmann, M. (2001) yFiles: visualization and automatic layout of graphs, in *Proceedings of the 9th International Symposium on Graph Drawing (GD 2001)*, Springer-Verlag.

67. Kusano, M. and Fukushima, A. (2013) Current challenges and future potential of tomato breeding using omics approaches. *Breed. Sci.*, **63** (1), 31–41.

68. Horan, K., Jang, C., Bailey-Serres, J., Mittler, R., Shelton, C., Harper, J.F., Zhu, J.K., Cushman, J.C., Gollery, M., and Girke, T. (2008) Annotating genes of known and unknown function by large-scale coexpression analysis. *Plant Physiol.*, **147** (1), 41–57.

69. Gutierrez, R.A., Lejay, L.V., Dean, A., Chiaromonte, F., Shasha, D.E., and Coruzzi, G.M. (2007) Qualitative network models and genome-wide expression data define carbon/nitrogen-responsive molecular machines in *Arabidopsis*. *Genome Biol.*, **8** (1), R7.

70. Lee, I., Ambaru, B., Thakkar, P., Marcotte, E.M., and Rhee, S.Y. (2010) Rational association of genes with traits using a genome-scale gene network for *Arabidopsis thaliana*. *Nat. Biotechnol.*, **28** (2), 149–156.

71. Clauset, A., Newman, M.E., and Moore, C. (2004) Finding community structure in very large networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, **70** (6, Pt. 2), 066111.

72. Hung, J.H., Yang, T.H., Hu, Z., Weng, Z., and DeLisi, C. (2012) Gene set enrichment analysis: performance evaluation and usage guidelines. *Briefings Bioinf.*, **13** (3), 281–291.

73. Chagoyen, M. and Pazos, F. (2013) Tools for the functional interpretation of metabolomic experiments. *Briefings Bioinf.*, **14** (6), 737–744.

74. Khatri, P., Sirota, M., and Butte, A.J. (2012) Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Comput. Biol.*, **8** (2), e1002375.

75. Falcon, S. and Gentleman, R. (2007) Using GOstats to test gene lists for GO term association. *Bioinformatics*, **23** (2), 257–258.

76. Stacklies, W., Redestig, H., Scholz, M., Walther, D., and Selbig, J. (2007) pcaMethods – a bioconductor package providing PCA methods for incomplete data. *Bioinformatics*, **23** (9), 1164–1167.

77. Kusano, M., Fukushima, A., Arita, M., Jonsson, P., Moritz, T., Kobayashi, M., Hayashi, N., Tohge, T., and Saito, K. (2007) Unbiased characterization of genotype-dependent metabolic regulations by metabolomic approach in *Arabidopsis thaliana*. *BMC Syst. Biol.*, **1**, 53.

78. Redestig, H., Kusano, M., Fukushima, A., Matsuda, F., Saito, K., and Arita, M. (2010) Consolidating metabolite identifiers to enable contextual and multi-platform metabolomics data analysis. *BMC Bioinf.*, **11**, 214.

79. Kusano, M., Tohge, T., Fukushima, A., Kobayashi, M., Hayashi, N., Otsuki, H., Kondou, Y., Goto, H., Kawashima, M., Matsuda, F. *et al* (2011) Metabolomics reveals comprehensive reprogramming involving two independent metabolic responses of *Arabidopsis* to UV-B light. *Plant J.*, **67** (2), 354–369.

80. Kim, T.Y., Kim, H.U., and Lee, S.Y. (2010) Data integration and analysis of biological networks. *Curr. Opin. Biotechnol.*, **21** (1), 78–84.