# Depicting gene co-expression networks underlying eQTLs

Nathalie Vialaneix, Laurence Liaubet, Magali San Cristobal

# Depicting gene co-expression networks underlying eQTLs

Nathalie Villa-Vialaneix, Laurence Liaubet and Magali SanCristobal

**Abstract** Deciphering the biological mechanisms underlying a list of genes whose expression is under partial genetic control (*i.e.*, having at least one eQTL) may not be as easy as for a list of differential genes. Indeed, no specific phenotype (*e.g.*, health or production phenotype) is linked to the list of transcripts under study. There is a need to find a coherent biological interpretation of a list of genes under (partial) genetic control. We propose a pipeline using appropriate statistical tools to build a co-expression network from the list of genes, then to finely depict the network structure. Graphical models are relevant, because they are based on partial correlations, closely linked with causal dependencies. Highly connected genes (hubs) and genes that are important for the global structure of the network (genes with high betweenness) are often biologically meaningful. Extracting modules of genes that are highly connected permits a significant enrichment in one biological function for each module, thus linking statistical results with biological significance. This approach has been previously used on a pig eQTL dataset [32] and was proven to be highly relevant. Throughout the chapter, we define statistical notions linked with network theory, and applied them on a reduced data set of genes with eQTL that were found in the pig species to illustrate the basics of network inference and mining.

Nathalie Villa-Vialaneix
MIAT, Universit de Toulouse, INRA, Castanet Tolosan, France, e-mail: nathalie.villa@toulouse.inra.fr

Laurence Liaubet and Magali SanCristobal
GenPhySE, Universit de Toulouse, INRA, INPT, INP-ENVT, Castanet Tolosan, France, e-mail: laurence.liaubet@toulouse.inra.fr, e-mail: magali.san-cristobal@toulouse.inra.fr

## 1 Introduction

In the search for genetic mechanisms underlying production or health phenotypes (terminal, say), GWAS studies have been intensively used, and have shown their limits. Classical tools in integrative biology aim at discovering links between terminal phenotypes and fine phenotypes (transcriptome, proteome, metabolome, ...), in a huge number. Integrating both approaches is possible: searching for a genetic basis of fine phenotypes (*e.g.*, eQTL, mQTL studies). The step further goes back to the terminal phenotypes with the precious and fine knowledge acquired with omics data. The focus of this chapter is linked to integrative biology and eQTL studies. The common pipeline for differential analysis is the use of linear models for testing differential expression at each gene, followed by a correction for multiple testing. This provides a list of genes whose expressions vary with a phenotype of interest. Then a functional analysis is performed: GO terms, KEGG pathways... Bibliographic mining is also interesting. The major limitation of this is the incomplete annotation encountered in livestock species: there may be only a part of transcripts that could not be given a gene name (*e.g.*, 78% in our pig transcripts have a gene name and about half have an associated function), mandatory for bibliographic mining.

eQTL studies provide genetic markers (the so-called eQTLs) that have a partial control on gene expression, and a list of genes whose expression is partially under genetic control (genes with eQTL). Upstream, there is some genetic control; genetic markers (the eQTLs) are often observed displayed in genomic clusters (*e.g.*, [20]). Downstream, a transcriptional control exists then a regulation of biological functions. Focusing on genes whose expression is genetically controlled (at least partially), we would like to address some questions. Do they also cluster? Is there a link between clusters of co-expression and biological functions?

The most appropriate tool to achieve this goal is networks. Given the strong loss of information with bibliographic networks (incomplete annotation), an alternative is co-expression networks. Indeed this statistical approach is based on all expression information, independent of the annotation. There exists various kinds of co-expression networks. We will see in the following that Graphical Gaussian Models (GGM, based on partial correlation) are very appropriate, in the sense that they are close to causative biological meaning.

After inferring the network in a sparse manner, it is of high interest to mine its structure. Extracting interesting genes (e.g. highly connected, with high incidence on the global structure) can give clues for further biological hypotheses and future experiments. Extracting modules can lead to an enrichment in biological functions, making the link between statistical results and biological interpretation. The functional annotation of the modules, based on a limited number of genes (because of the poor annotation), can then give insights into possible biological functions for unnannotated genes ("guilt by association" approach, see [10] and [17] for a study which questions this approach).

In the article [32], the pipeline briefly described above highlighted key genes, and showed a strong enrichment of one biological function per module. Moreover, one module was linked with meat pH, a particularly interesting phenotype, since it

is related to meat production and quality. In this chapter we will present in detail the overall approach, explaining key aspects linked with network analysis, applying them on a subset of genes with eQTLs, extracted from the one studied in [32].

This chapter is organized as follows: Section 2 provides basic definitions and concepts for network studies. Section 3 deals with network inference and Section 4 with network mining. Finally Section 5 deals with biological interpretation of the results. Throughout this article, a small example study is performed using the free statistical software R: codes and datasets are available at `http://nathalievilla.org/bio_network`.

## 2 Basic definitions and concepts for graphs / networks

### *2.1 Networks*

A *network*, also frequently called a *graph*, is a mathematical object used to model *relationships* between *entities*. In its simplest form, it is composed of two sets $(V,E)$:

- the set $V = \{v_1,\ldots,v_p\}$ is a set of $p$ *nodes*, also called *vertices* that represent the entities;
- the set $E$ is a subset of the set of node pairs, $E \subset \big\{(v_i,v_j),\ i,j = 1,\ldots,p,\ i \neq j\big\}$: the nodes pairs in $E$ are called *edges* of the graph and model a given type of relationships between two entities.

In the following, nodes will be genes and edges will represent a relationship (*e.g.*, co-expression) between two genes. A network is often displayed as in Figure 1: the nodes are represented with circles and the edges with straight lines connecting two nodes.

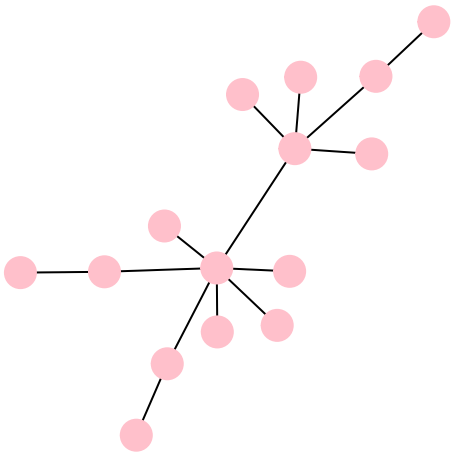

**Fig. 1** Example of the representation of a simple network with 15 nodes and 13 edges.

This lesson's scope is restricted to simple networks, *i.e.*, to undirected graphs (the edges do not have any direction), with no loop (there is no edge between a given node and itself) and simple edges (there is one edge at most between a pair of nodes). But networks can deal with many other types of real-life situations:

- *directed graphs* in which the edges have a direction, *i.e.*, the edge from the node $v_i$ to the node $v_j$ is not the same as the edge from the node $v_j$ to the node $v_i$. In this case, the edges are often called *arcs*;
- *weighted graphs* in which a (often positive) weight is associated to each edge;
- *graphs with multiple edges* in which a pair of nodes can be linked by several edges, that can eventually have different labels or weights to model different types of relationships;
- *labeled graphs* (or graph with node attributes) in which one or several labels are associated to each node, labels can be factors (*e.g.*, a gene function) or numeric values (*e.g.*, gene expression).

## 2.2 Overview of standard issues for network analysis

This chapter will address two main issues posed by network analysis:

- The first one will be discussed in Section 3 and is called *network inference*: giving data (*i.e.*, variables observed for several subjects or objects), how to build a network whose edges represent the "direct links" between the variables? The nodes in the inferred network are the genes and the edges represent a strong "direct link" between the two gene expressions;
- The second issue comes when the network is already built or directly given: the practitioner then wants to understand the main characteristics of the network and to extract its most important nodes, groups, etc. This ensemble of methods, studied in Section 4, is called *network mining* and comprises (among other problems):
  - *network visualization*: when displaying a network, no *a priori* position is associated with its nodes and the network can thus be displayed in many different ways;
  - *node clustering*: an intuitive way to understand a network structure is to focus not on individual connections between nodes but on connections between densely connected groups of nodes. These groups are often called *clusters* or *communities* or *modules* and many works in the literature have focused on the problem of extracting these clusters.

## 2.3 eQTL data

Throughout this chapter, a subset of genes analyzed in [32] will be used to illustrate the basics of network inference and mining. The applications will be performed

using the free statistical software environment R (version 3.2.5). The packages used are:

- **huge** (version 1.2.7) for network inference;
- **igraph** (version 1.0.1) for creating network objects and for network mining.

The reader interested in this topic may also want to have a look at the "gRaphical Models in R" task view [1] where he/she will find further interesting packages.

To illustrate key steps, we propose the analysis of a small subset of data in [20, 32], which is a subset of 68 genes having at least one eQTL. This data will be refered to as "68-eqtl" throughout the chapter. This dataset can be downloaded at http://nathalievilla.org/doc/csv/subsetEQTL.csv. The data set consists of gene expressions for a "small" list of genes (transcripts). It is represented by the matrix $\mathbf{X}$:

$$n \text{ individuals } \underbrace{\left\{ \mathbf{X} = \begin{pmatrix} \cdot\,\cdot & \cdot & \cdot\,\cdot\,\cdot \\ \cdot\,\cdot & X_i^j & \cdot\,\cdot\,\cdot \\ \cdot\,\cdot & \cdot & \cdot\,\cdot\,\cdot \end{pmatrix} \right.}_{p \text{ variables (gene expressions)}},$$

where $X_i^j$ is the expression quantification of gene $j$ in individual $i$. Even restricting to a small subset of genes, having $n < p$ is the standard situation which, as discussed later, poses some problems for network inference. These data can be loaded using the following command line:

```
expression = read.csv("data/subsetEQTL.csv", row.names=1)
```

if the dataset provided at http://nathalievilla.org/doc/csv/subsetEQTL.csv is stored in subdirectory "data" of R working directory.

The boxplots of the $p = 68$ variables (genes) of the "68-eqtl" dataset are displayed in Figure 2 (left). The correlation matrix between the 68 genes is displayed in Figure 2 (right) showing that a potential structure has to be highlighted.

## 3 Network inference

The aim of this section is to choose an appropriate type of network, then to infer the network based on data (expression of the 68 genes). In short, "inferring a network" means building a graph for which:

- the nodes represent the $p$ genes;
- the edges represent a "direct" and "strong" relationship between two genes. This kind of relationships aims at tracking hierarchical influence and possible transcriptional or genetic regulations.

[1] https://cran.r-project.org/web/views/gR.html

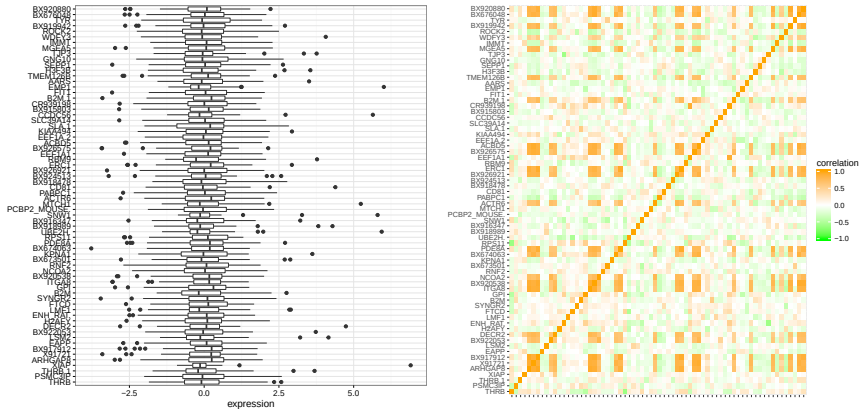


**Fig. 2** Left: boxplot of the gene expression distributions (68 genes). Right: heatmap of the correlation matrix between pairs of genes expression.

The main advantage of using networks over raw data is that such a model focuses on "strong" links and is thus more robust. Also, inference can be combined/compared with/to bibliographic networks to incorporate prior knowledge into the model but, unlike bibliographic networks, networks inferred from one of the model presented below can handle even unknown (*i.e.*, not annotated) genes into the analysis.

Even if alternative approaches exist, a common way to infer a network from gene expression data is to use the steps described in Figure 3:

1. First, the user calculates pairwise similarities (correlations, partial correlations, information based similarities such as the mutual information ...) between pairs of genes;
2. second, the smallest (or less significant) similarities are thresholded (using a simple threshold chosen by a given heuristic or a test or sparse approaches with penalization while calculating the similarities or other more sophisticated methods);
3. lastly, the network is built from the nonzero similarities, putting an edge between two genes with a nonzero similarity (which thus correspond to the highest values, in a given sense that depends on the thresholding method, of the similarity).

This approach leads to produce *undirected* networks. Additionaly, the edges of the network can be weighted by the strength of the relationship (*i.e.*, the absolute value of the similarity) and signed by the sign of the relation (*i.e.*, if the similarity is positive or negative). This approach is used in [19] to integrate DE genes and eQTL genes in a single co-expression network related to obesity in pigs.

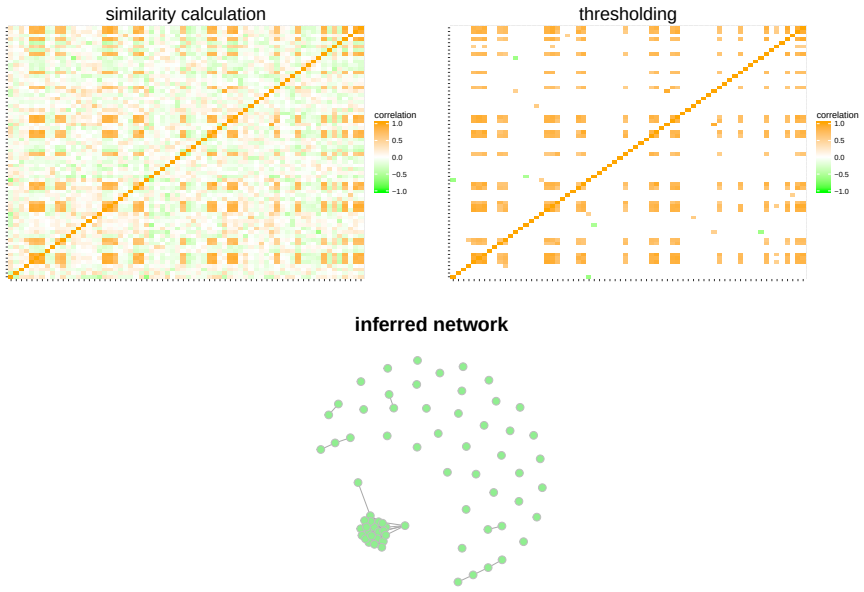


**Fig. 3** Main steps in network inference.

## *3.1 Limits of the Pearson correlation*

A simple, naive approach to infer a network from gene expression data is to calculate pairwise correlations between gene expressions and then to simply threshold the smallest ones, possibly, using a test of significance. This approach is sometimes called *relevance network* [6, 7]. The R package **huge**[2] can be used to infer networks in such a way. However, if easy to interpret, this approach may lead to strongly misunderstanding the regulation relationships between genes. To better understand the problem posed by using direct correlations in network inference, we will discuss the simple situation described in Figure 4. In this model, a single gene, denoted by

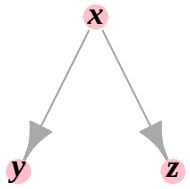


**Fig. 4** Small model showing the limit of the correlation coefficient to track regulation links: when two genes $y$ and $z$ are regulated by a common gene $x$, the correlation coefficient between the expression of $y$ and the expression of $z$ is strong as a consequence.

$x$, strongly regulates the expression of two other genes, $y$ and $z$. This situation is well

[2] http://cran.r-project.org/web/packages/huge

illustrated using the simple mathematical model

$$X \sim \mathscr{U}[0,1], \qquad Y \sim 2X+1+\varepsilon_1 \text{ and } Z \sim -2X+2+\varepsilon_2$$

in which $\mathscr{U}[0,1]$ is the uniform distribution in $[0,1]$ and $\varepsilon_1$ and $\varepsilon_2$ are independent and centered Gaussian random variables independent of $X$ with a standard deviation equal to 0.1. A quick simulation with R gives the following results:

```
x = rnorm(100)
y = 2*x+1+rnorm(100,0,0.1)
cor(x,y)

  ## [1] 0.9988261

z = -2*x+1+rnorm(100,0,0.1)
cor(x,z)

  ## [1] -0.998756

cor(y,z)

  ## [1] -0.9980506
```

Hence, even though there is not a direct (regulation) link between $z$ and $y$, these two variables are highly correlated (the correlation coefficient is larger than 0.99) as a result of their common regulation by $x$.

## *3.2 Partial correlation and Gaussian Graphical Model (GGM)*

This result is unwanted and using a *partial correlation* can deal with such strong indirect correlation coefficients. The partial correlation between $y$ and $z$ is the correlation between the expression of $y$ and $z$, *knowing the expression of* $x$. In the above example, it is equal to the correlation between the residuals of the linear models:

$$Y = \beta_1 X + \varepsilon_1 \text{ and } Z = \beta_2 X + \varepsilon_2$$

and in our case, it is equal to

```
cor(lm(z~x)$residuals,lm(y~x)$residuals)

  ## [1] -0.1933699
```

which is much smaller than the direct correlation, while the other two partial correlations remain large:

```
cor(lm(x~y)$residuals,lm(z~y)$residuals)

  ## [1] -0.6208908

cor(lm(x~z)$residuals,lm(y~z)$residuals)

  ## [1] 0.6481373
```

When using partial correlation, the *conditional dependency graph* is thus estimated. Under a Gaussian model (see [11] for further explanations), in which the gene expressions $X = (X^j)_{j=1,\dots,p}$ are supposed to be distributed as centered Gaussian random variables with covariance matrix $\Sigma$, this graph is defined as follows:

$$v_j \longleftrightarrow v_{j'} (\text{genes } j \text{ and } j' \text{ are linked}) \Leftrightarrow \mathbb{C}\text{or}\left(X^j, X^{j'} | (X^k)_{k\neq j,j'}\right) \neq 0$$

in which the last quantity is called *partial correlation*, $\pi_{jj'}$. In this framework, $\mathbf{S} = \Sigma^{-1}$ is called the *concentration matrix* and is related to the partial correlation $\pi_{jj'}$ between $X^j$ and $X^{j'}$ by the following relation:

$$\pi_{jj'} = -\frac{\mathbf{S}_{jj'}}{\sqrt{\mathbf{S}_{jj}\mathbf{S}_{j'j'}}}. \tag{1}$$

This equation indicates that non zero partial correlations (i.e., edges in the conditional dependency graph) are also non zero entries of the concentration matrix $\mathbf{S}$.

### 3.3 Estimating the conditional dependency graph with Graphical LASSO

The empirical estimator $\widehat{\Sigma}$ of $\Sigma$ is calculated from the $n \times p$-matrix of gene expression $\mathbf{X}$ generated from the Gaussian distribution $\mathcal{N}(0, \Sigma)$,

$$\widehat{\Sigma}_{jj'} := \frac{1}{n}\sum_i \left(\mathbf{X}_i^j - \overline{X}^j\right)^2 \text{ with } \overline{X}^j = \frac{1}{n}\sum_i \mathbf{X}_i^j,$$

calculated from the observations $\mathbf{X}$. A major issue when using $\Sigma^{-1}$ for estimating $\mathbf{S}$ is that the empirical estimator $\widehat{\Sigma}$ is ill-conditioned because it is calculated with only a few number $n$ of observations: the sample size $n$ is usually much lower than the number of variables $p$. Hence, $\widehat{\Sigma}^{-1}$ is a poor estimate of $\mathbf{S}$ and must not be used as it is.

Several attempts to deal with such a problem have been proposed. The seminal work [29, 30] uses shrinkage, *i.e.*, $\mathbf{S}$ is estimated by $\widehat{\mathbf{S}} = \left(\widehat{\Sigma} + \lambda \mathbb{I}\right)^{-1}$ (for a given small $\lambda \in \mathbb{R}^+$). Then, the obtained partial correlations are thresholded either by choosing a given thresholding value or a given number of edges or by using a test

statistics presented in [29], which is itself based on a Bayesian model. This method is implemented in the R package **GeneNet**[3].

The previous method is a two-step method which first estimates the partial correlations and then selects the most significant ones. An alternative method is to simultaneously estimate and select the partial correlations using a sparse penalty. It is known under the name *Graphical LASSO* (or *GLasso*). Under a GGM framework, partial correlation is also related to the estimation of the following linear models:

$$X^j = \sum_{k \neq j} \beta_k^j X^k + \varepsilon_j \tag{2}$$

by the relation

$$\beta_k^j = -\frac{\mathbf{S}_{jk}}{\mathbf{S}_{jj}}$$

which, combined with Equation (1) shows again that non zero entries of the linear model coefficients correspond exactly to non zero partial correlations.

Hence, several authors [23, 15] have proposed to integrate a sparse penalty in the estimation of (2) by ordinary least squares (OLS):

$$\forall\, j = 1, \ldots, p, \qquad \arg\min_{\beta^j} \left[ \sum_{i=1}^{n} \left( \mathbf{X}_i^j - \sum_{k \neq j} \beta_k^j \mathbf{X}_i^k \right)^2 + \lambda \|\beta^j\|_{L^1} \right] \tag{3}$$

where $\|\beta^j\|_{L^1} = \sum_{k \neq j} |\beta_k^j|$ is the $L_1$-*norm* of $\beta^j \in \mathbb{R}^{p-1}$ which is added to the OLS minimization problem in order to force only a restricted number of non zero entries in $\beta^j$. $\lambda$ is a regularization parameter that controls the sparseness of $\beta^j$ (the larger $\lambda$, the fewer the number of non zero entries in $\beta^j$). It is generally varied during the learning process and the most adequate value is selected. This method is implemented in the R package **huge**.

Finally, several approaches have been proposed to deal with the choice of a proper $\lambda$: [21] proposes the StARS approach which is based on a stability criterion while [22] and [14] propose approaches based on a modification of the BIC criterion. All these methods are implemented in the R package **huge**.

### *3.4 Application*

Using the "68-eqtl" data, a network can be inferred using the method described in [23] with the R package **huge**. The package is loaded with

```
library(huge)
```

The concentration matrix is estimated for several values of $\lambda$ with:

[3] https://cran.r-project.org/web/packages/GeneNet

```
glassoRes = huge(as.matrix(expression), nlambda=100,
                 method="glasso")
```

The option `nlambda` is used to set the number of regularization parameter values $\lambda$ used for the estimation. The result is a list of estimated concentration matrices (one for each value of $\lambda$, whose sparsity decreases when $\lambda$ decreases), stored in `glassoRes$icov`. These matrices are (almost) all sparse, which means that most of their entries are equal to zero (the matrices obtained with small $\lambda$ contains much fewer zeros than the ones with larger $\lambda$).

To select one of the 100 concentration matrices, the function `huge.select` implements several model selection methods. Among them, the 'StARS' method chooses the largest $\lambda$ so that the obtained concentration matrix is replicable with random sub-sampling. More precisely, many random subsamples are generated and a criterion is computed to assess the stability of any given edges in the inference obtained from all subsamples. The most sparse graph which is still stable according to these criteria is the one chosen by the method. This approach can be used with:

```
glassoFinal = huge.select(glassoRes, criterion="stars")
```

which results in an object that contains the optimal value of lambda, `glassoFinal$opt.lambda` (here equals to 0.3551), the optimal $68 \times 68$-concentration matrix in `glassoFinal$opt.icov` and the optimal sparse adjacency matrix of the inferred network in `glassoFinal$refit`. The result of the selection is summarized in Figure 5, which is produced by the following command line:

```
plot(glassoFinal)
```

Finally, a network R object can be obtained for further studies using the R package **igraph**. More precisely, the function `graph_from_adjacency_matrix` can be used on the sparse adjacency matrix `glassoFinal$refit` and the function `simplify` is used to remove multiple edges and loops.

```
glassoNet = graph_from_adjacency_matrix(glassoFinal$refit,
                                        mode="max")
glassoNet = simplify(glassoNet)
glassoNet
```

```
## IGRAPH U--- 68 232 --
## + edges:
##  [1]  1--18  1--27  1--31  1--40  1--41  2--17  4-- 8  4--11  4--62  5-- 6
## [11]  5-- 7  5--11  5--19  5--20  5--21  5--26  5--39  5--40  5--43  5--44
## [21]  5--52  5--56  5--63  5--64  5--65  5--67  5--68  6-- 7  6--10  6--11
## [31]  6--19  6--20  6--25  6--26  6--39  6--40  6--43  6--44  6--56  6--61
## [41]  6--67  6--68  7--10  7--11  7--19  7--20  7--21  7--26  7--34  7--35
## [51]  7--39  7--40  7--43  7--44  7--46  7--52  7--56  7--61  7--63  7--65
```

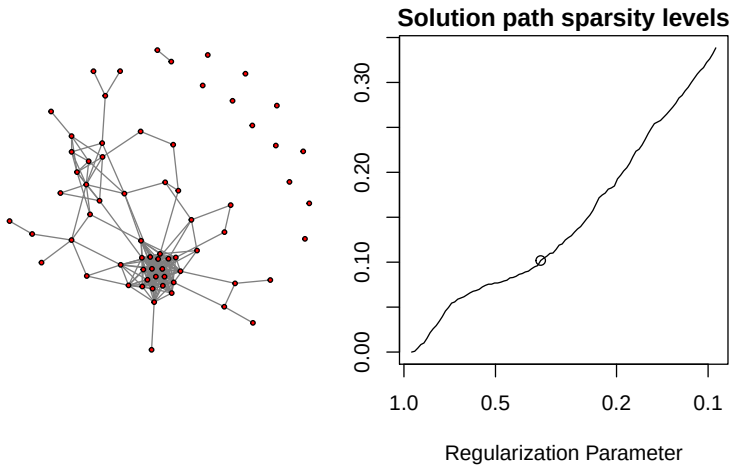


**Fig. 5** Summary of the result of the 'StARS' selection method. Left: Selected network. Right: Solution sparsity (% of inferred edges over the number of pairs of nodes in the graph) versus $\lambda$. The chosen $\lambda$ is emphasized with a dot on the curve.

```
## [61]  7--67  7--68  9--29 10--11 10--21 10--25 10--34 10--39 10--43 10--44
## [71] 10--49 10--61 10--67 10--68 11--19 11--20 11--21 11--25 11--34 11--35
## [81] 11--39 11--40 11--43 11--44 11--67 11--68 12--28 12--46 12--64 13--18
## + ... omitted several edges
```

This graph (an `igraph` object) contains $p$ =68 nodes and 232 edges.

Gene names (included in the column names of the expression matrix) can be attached to the nodes as an attribute called "name" which is then easily used when displaying the network or selecting nodes. This setting is performed with the function `V`:

```
V(glassoNet)$name = colnames(expression)
```

As shown in Figure 5, the inferred network is composed of several groups of nodes that are not connected with each other. These groups are called the *connected components of the graph*. Using **igraph**, they can be extracted with the function `components`:

```
glassoComp = components(glassoNet)
head(glassoComp$membership)

  ##    THRB PSMC3IP  THRB.1    XIAP ARHGAP8  X91721
  ##       1       1       2       1       1       1

glassoComp$csize

  ##  [1] 55  1  2  1  1  1  1  1  1  1  1  1  1

glassoComp$no

  ## [1] 13
```

The inferred network has `glassoComp$no`=13 connected components, most of them composed of only one node. The largest connected component has `glassoComp$csize`=55 nodes. The number of the connected component of a given gene in the gene network is given in `glassoComp$membership` and the connected components can thus be obtained with the function `induced_subgraph`:

```
glassoSubNet = induced_subgraph(glassoNet,
  glassoComp$membership==which.max(glassoComp$csize))
```

Finally, the largest connected component of the inferred network, which contains 55 nodes and 231 edges, will be named "55-eqtl network" in the sequel. This network is the one that will be studied further in the next section which is devoted to network mining. This graph can be exported into an external format, such as the widely used "graphml" format, with the function `write_graph`

```
write_graph(glassoSubNet, file="results/lcc.graphml",
            format="graphml")
```

The obtained file can then be imported in most softwares dedicated to graph mining for exploratory purposes. More information about the possible formats for graph exportation is available with

```
help(write_graph)
```

# 4 Network mining

In this section, a graph $\mathscr{G} = (V,E)$ is supposed to be given, where $V = \{v_1,\dots,v_p\}$ is the set of nodes and $E$ is the set of edges. Mining a network is the process in which the user extracts information about the most important nodes or about groups of nodes that are densely connected.

## *4.1 Network visualization*

Visualization tools are used to display the graph in a meaningful and aesthetic way. Standard approaches in this area use *force directed placement* (FDP) algorithms (see [16], among others). The principle of these algorithms can be illustrated by an analogy to the following physical mechanism which:

- attaches attractive forces to the edges of the graph (similar to springs) in order to force connected nodes to be represented close to each other;
- attaches repulsive forces between all pairs of nodes (similar to electric forces) to force nodes to be displayed separately.

The algorithm performs iteratively from an (usually random) initial position of the nodes until stabilization. The R package **igraph** (see [8]) implements several layouts and even several FDP based layouts for static representation of the network.

Using **igraph**, the network inferred in Section 3 can be displayed using the functions `layout.fruchterman.reingold` (for calculating the layout with the FDP method of [16]) and `plot.igraph` (for displaying it on a graphical device). The result of the function `layout.fruchterman.reingold` is a matrix with 2 columns and 55 rows that contains the positions of the nodes. It can be attached to the `igraph` object as a graph attribute named "layout" to be used when passed to the function `plot`. Several characteristics of the graph representation, that are related to nodes and edges (colours, shapes, labels...), can be defined in the `plot.igraph` options.

```
glassoSubNet$layout =
  layout.fruchterman.reingold(glassoSubNet)
plot(glassoSubNet, vertex.size=0,
     vertex.label.color="black",
     vertex.label.cex=0.8)
```

More information on the `plot.igraph` options are provided in the help:

```
help(igraph.plotting)
```

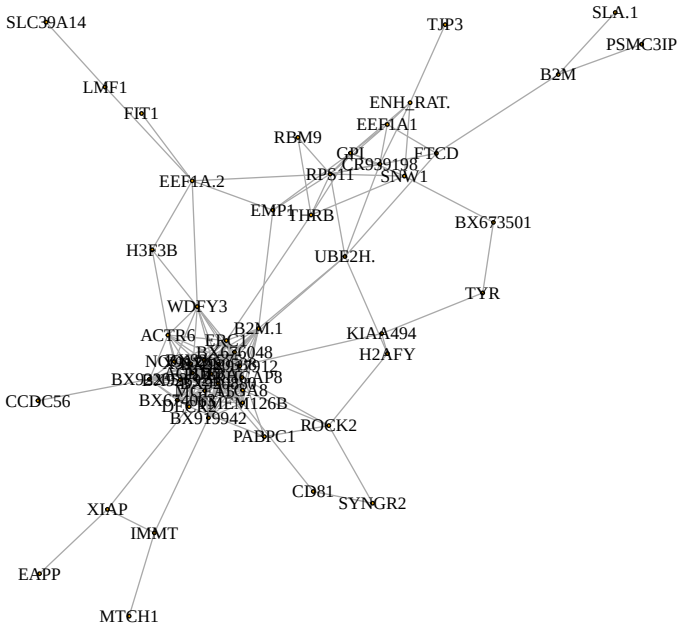


**Fig. 6** Representation of the inferred network with Fruchterman and Reingold force directed placement algorithm.

The free softwares Gephi[4] [3], Tulip[5] [1] or Cytoscape[6] [31], among others, can also be used to visualize a network interactively (they support zooming and panning, among other features).

[4] http://gephi.org
[5] http://tulip.labri.fr
[6] http://www.cytoscape.org

## 4.2 Global characteristics

This section gives the definition of two global numerical characteristics that can help to understand the network structure.

**Definition 1 (density).** The *density* of a network is the number of edges divided by the number of pairs of nodes, $\frac{|E|}{p(p-1)/2}$.

In the toy example given in Figure 7, the number of edges is equal to 4 and the number of pairs of nodes is equal to $\frac{4\times 3}{2} = 6$ so the density is equal to $\frac{4}{6} \simeq 66.7\%$.[7]

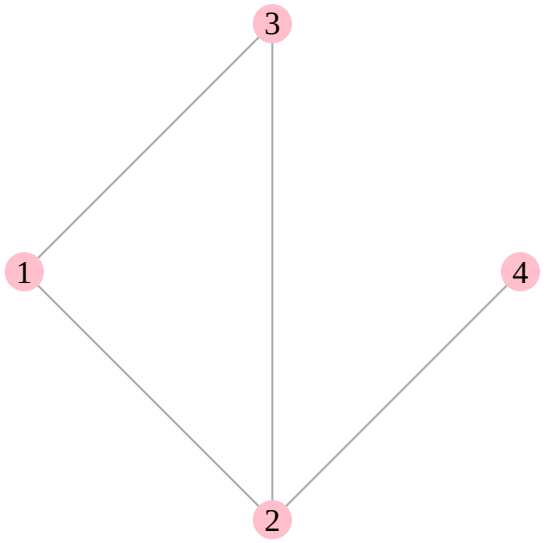


**Fig. 7** Simple network with a transitivity equal to 1/3.

Because it is equal to the frequency of edges over the number of possible edges, the density is a measure of how densely connected the network is.

The "55-eqtl network" has 231 edges for 55 nodes; its density is thus equal to $\frac{231}{55\times 54/2} \simeq 15.6\%$. It can be obtained with the function `edge_density`:

```
edge_density(glassoSubNet)

  ## [1] 0.1555556
```

It is expected that the density tends to decrease with the number of edges (see [9] for examples of real-world networks together with their main characteristics).

[7] The number of pairs for a set of $n$ objets is equal to $\frac{n(n-1)}{2}$.

**Definition 2 (transitivity).** The *transitivity* of a network is the number of triangles in the network divided by the number of triplets of nodes that are connected by at least two edges.

In the toy example given in Figure 7, the transitivity is equal to $\frac{1}{3} \simeq 33.3\%$ (1 triangle linking the nodes $\{1,2,3\}$ and three triplets with at least two edges: $\{1,2,3\}$, $\{2,3,4\}$ and $\{1,2,4\}$.

Speaking in terms of a social network, the transitivity thus measures the probability that two of my friends are also friends. A transitivity which is much larger than the density indicates that the nodes are not connected *at random* but on the contrary that there is a strong local connectivity (a kind of "modular structure"), which is often the case in real-world networks.

The "55-eqtl network" has a transitivity equal to 68.7% that is obtained with the function `transitivity`:

```
transitivity(glassoSubNet)
```

```
## [1] 0.6868448
```

As expected, the transitivity is much larger than the density for the "55-eqtl network" which shows a strong local connectivity.

## *4.3 Individual characteristics*

Once the network structure is analyzed globally, one may want to focus more precisely on nodes individually so as to extract the most "important" ones. Some simple numeric characteristics can be used to do so.

**Definition 3 (degree).** The *degree* of a node $v_i$ is the number of edges adjacent to this node: $d_i = |\{(v_i, v_j) \in E : j \neq i\}|$.

Nodes that have a large degree are called *hubs*.

In the toy example given in Figure 7, the degree of node 2 is equal to 3 (three edges are afferent to node 2 linking node 2 to nodes 1, 3 and 4).

The degree is a measure of the node's "popularity". Using the function `degree`, the degrees of all nodes in the "55-eqtl network" can be obtained:

```
head(degree(glassoSubNet), n=5)
```

```
##    THRB PSMC3IP    XIAP ARHGAP8  X91721
##       5       1       3      18      16
```

The degree distribution of the "55-eqtl network" is shown in Figure 8. This figure shows that most of the nodes have a very small degree (smaller than 5) whereas a few nodes have (comparatively) very large degrees (more than 20).

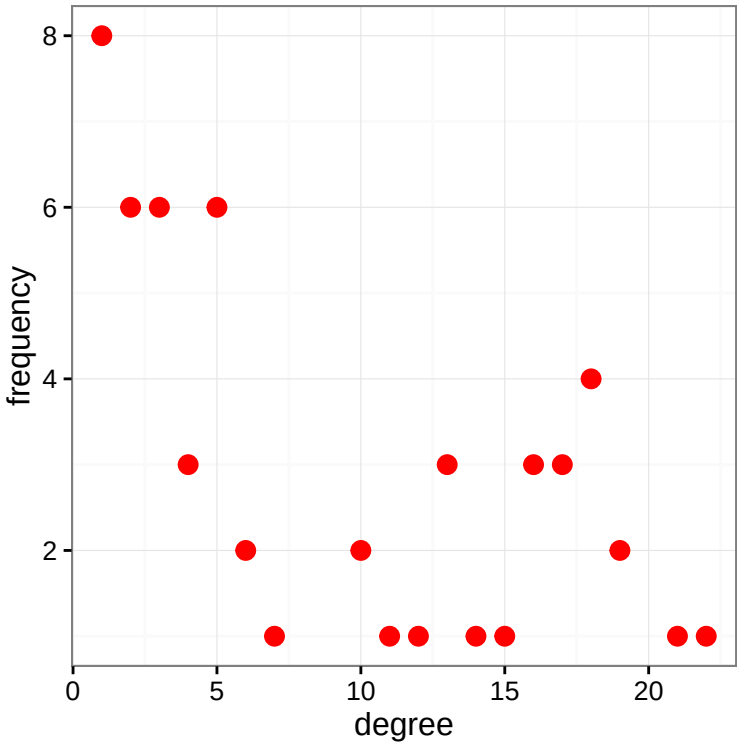


**Fig. 8** Degree distribution for the "55-eqtl network".

Many real-world networks are reported to have a *degree distribution* (*i.e.*, the values $(\mathbb{P}(k))_k$ that counts the number of nodes with a given degree $k$) which fits a *power law*: $\mathbb{P}(k) \sim k^{-\gamma}$ for a given $\gamma > 0$. Thus, degree distributions are often displayed with log-log scales (i.e., $\log \mathbb{P}(k)$ versus $\log k$). In this case, a good linear fit indicates a power law distribution. The "55-eqtl network" is a bit too small to observe such a distribution but nevertheless, the degree distribution is skewed. Also, there is a higher proportion of nodes with a degree between 15 and 20. Looking at Figure 9, we can see that this corresponds to the set of nodes that are highly connected in Figure 9.

**Definition 4 (betweenness).** The *betweenness* of a node $v$ is the number of shortest paths between any pair of nodes that pass through this node.

In the toy example given in Figure 7, the betweenness of node 2 is equal to 2 because the shortest path between nodes 1 and 4 is $1 \to 2 \to 4$ and the shortest path between nodes 3 and 4 is $1 \to 2 \to 4$. All the other nodes have a betweenness equal to 0.

The betweenness is a centrality measure: nodes that have a large betweenness are those that are the most likely to disconnect the network if removed. They may

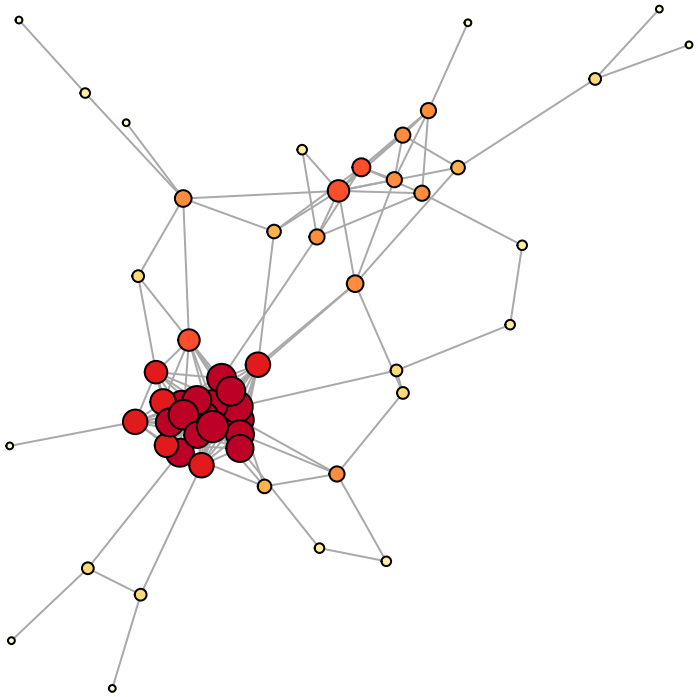

**Fig. 9** "55-eqtl network": the node sizes and their colour intensities are proportional to their degrees.

thus correspond to genes of high importance. Using the function `betweenness`, the betweenness of the 55 nodes of the "55-eqtl network" can be obtained:

```
head(betweenness(glassoSubNet), n=4)
```

```
##       THRB    PSMC3IP       XIAP    ARHGAP8
## 137.41563    0.00000   57.47527   54.33676
```

The betweenness of every node is displayed in Figure 10. It is interesting to note that nodes with high betweenness are not necessarily hubs. The nodes with the highest betweenness are more outside the set of nodes which are highly connected.

## *4.4 Clustering*

Clustering nodes in a network consists of partitioning the network into densely connected groups that we will call *modules* in the sequel. The nodes in a given mod-

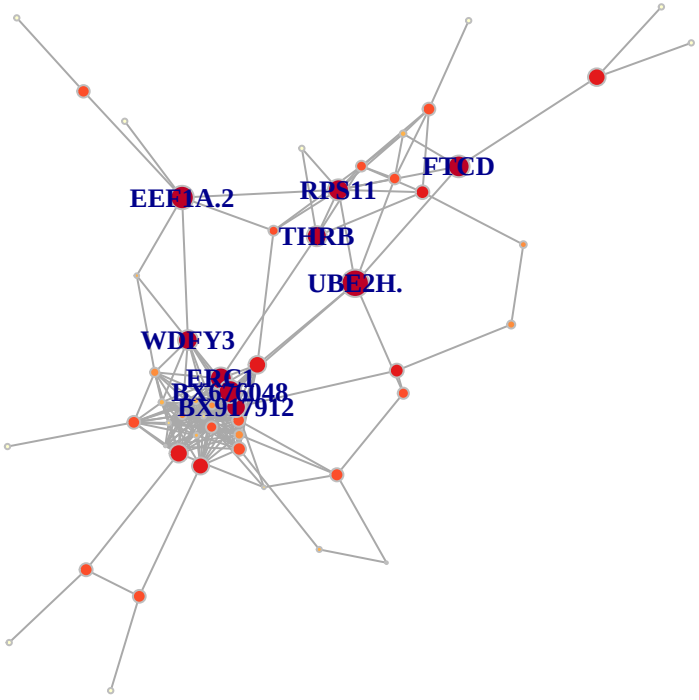


**Fig. 10** "55-eqtl network": the node sizes and their colour intensities are proportional to their betweenness.

ule share a few number of edges (comparatively) with the nodes of other modules. Modules are often called *communities* in social sciences and *clusters* in statistics. A number of methods have been designed to address this issue and this section is much too small to go beyond scratching the surface of this topic. For further references on this topic, we advise the reader to refer to [13, 28].

One of the most popular approaches for node clustering consists of maximizing a quality criterion called *modularity* [25]:

**Definition 5 (modularity).** Given a partition $(\mathscr{C}_1, \ldots, \mathscr{C}_K)$ of the nodes of the graph, the *modularity* of the partition is equal to

$$\mathscr{Q}(\mathscr{C}_1, \ldots, \mathscr{C}_K) = \frac{1}{2m} \sum_{k=1}^{K} \sum_{v_i,\, v_j \in \mathscr{C}_k} \left( \mathbb{I}_{(v_i, v_j) \in E} - P_{ij} \right)$$

where $P_{ij} = \frac{d_i d_j}{2m}$, $d_i$ the degree of node $i$ and $m = |E|$ is the number of edges in the network.

In this definition, $P_{ij}$ plays the role of a probability to have an edge between $v_i$ and $v_j$ according to a "null model". In the "null model", the edges depend only on the degrees of each node and not on the clusters themselves: the larger the modularity, the more the edges are concentrated in the clusters $(\mathscr{C}_j)_j$. This model slightly differs from maximizing the number of edges in the clusters: edges that correspond to nodes with a large degree have a lesser impact in the modularity value: this aims at encompassing in the criterion the notion of *preferential attachment* [2], which is the fact that, in real networks, people tend to connect preferably with people who already have a large number of connections. Hence, the edges of very popular nodes (hubs) seem to be less "significant" (or, in other words, less important to define an homogeneous module). In particular, the modularity is known to better separate hubs (as compared to a naive approach consisting of minimizing the number of edges between clusters, that leads more frequently to have huge clusters and tiny ones with isolated nodes). Also, the modularity is not monotonous in the number of modules: it can thus be useful to decide on an adequate number of clusters. However, it is also known to fail to detect small modules [13]. Several method can be used to find a partition that approximately optimizes the modularity[8]. In the R package **igraph**, several methods are implemented. In the following, we will use the function `cluster_spinglass` which implements the method described in [26] (equivalent in certain cases to modularity optimization) and based on simulated annealing:

```
finalClustering = cluster_spinglass(glassoSubNet)
modularity(finalClustering)

  ## [1] 0.3102359

head(membership(finalClustering))

  ##      THRB   PSMC3IP      XIAP   ARHGAP8    X91721 BX917912
  ##         4         4         5         2         2         2

sizes(finalClustering)

  ## Community sizes
  ##  1  2  3  4  5
  ##  8 21  7 15  4
```

Using this method algorithm[9], the "55-eqtl network" could be partitioned into 5 modules (Figure 11), of 8, 21, 7, 15, 4 nodes respectively. The modularity of this partition is equal to 0.31.

To assess if the modularity is significantly large (and hence if the partition is meaningful), a test of significance has been performed, as described in [27, 24].

[8] The modularity maximization is an intractable problem which can be solved only for small networks. For large networks, fast algorithms are usually used to find an approximate solution.

[9] As the algorithm is partially stochastic, it has been run 100 times and only the best result has been kept.

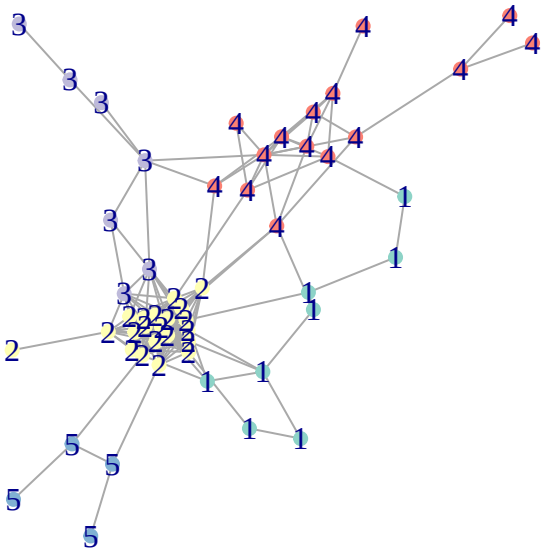

**Fig. 11** Partition of the "55-eqtl network" into 5 modules. The colours and labels indicate module membership.

This test is based on the computation of the maximum modularity for 100 random graphs with the same degree distributions as "55-eqtl network". The distribution of the maximum modularity for the random graphs is compared to the maximum modularity of the "55-eqtl network" in Figure 12.

## 5 Biological mining

Apart from providing easy-to-handle graphical displays, network analysis can be used forward to interpret the data. To that end, the analyst needs to go back to biological knowledge and extract coherent biological findings from statistical results. This analysis can be conducted in 3 steps:

1. gene annotation,
2. biological enrichment, and
3. biological networks.

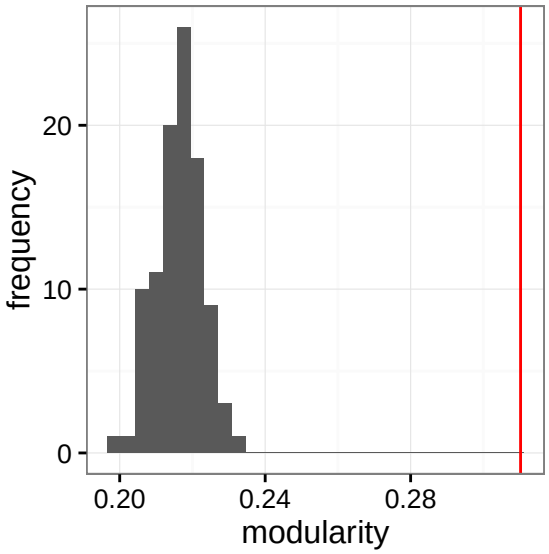


**Fig. 12** Distribution of the maximum modularity over 100 random graphs with the same degree distribution as the "55-eqtl network" compared to the maximum modularity found for this network (red vertical line).

## *5.1 Gene annotation*

In the previous sections, expression data were used without taking into account the biological functions associated with nodes. Nodes are first a DNA sequence coming from RNA sequencing or probes on microarrays. According to the quality of the annotation of the studied genome, only part of the nodes are annotated. One of the advantages of gene network is that all probes, even those that correspond to unannotated probes can be used for the analysis, whereas they are often left aside in other approaches. In the example of the greatest connected component of 55 nodes, 34 nodes were annotated in 2013 [32] while 43 are annotated in 2015 thanks to the progress of the annotation of the pig genome.

Giving access to the original sequences is of prime importance when publishing transcriptomic data (see MIAME, Minimum Information About a Microarray Experiment [5]). Data must be submitted to public repositories such as Gene Expression Omnibus (GEO, NCBI website[10]) or ArrayExpress (EMBL website[11]) and many others allowing the complete access to the probe sequence. At the time of publication, some related information may be associated with the sequence: current annotation with gene name or symbol, gene description, aliases, known orthologs, accession number of the sequence from which the probe has been designed...

[10] http://www.ncbi.nlm.nih.gov/geo

[11] https://www.ebi.ac.uk/arrayexpress

Functional information could be associated to each gene product. A consortium tries to attribute functional terms with a curated approach (controlled vocabulary) named Gene Ontology (GO[12]). The biology is cleaved in three domains: Biological Processes (*e.g.*, glycolytic process), Molecular Function (*e.g.*, acetyl-CoA transporter activity) and Cellular Component (*e.g.*, glycosome). Other reliable functions may be obtained with KEGG (Kyoto Encyclopedia of Genes and Genomes[13]). KEGG is a database which gives access to many well documented pathways as signaling (*e.g.*, PI3K-Akt signaling pathway), metabolism (*e.g.*, Lipid metabolism) or biological processes (*e.g.*, cell growth and death).

Functional information for a full list of genes can be obtained from databases like DAVID (Database for Annotation, Visualization and Integrated Discovery[14]) with the downloadable application EASE (Expression Analysis Systematic Explorer[15]) or "Ensembl" with BioMart[16]. Care must be taken if an updated version is available. For instance, current annotation in Ensembl is the release 81 - July 2015 at the time of this review. Also, the user has to carefully make the choice of the genome annotation to which to refer. For instance, for the pig genome, two genome annotations can be used: the one of the pig or the one of the human. At the date of this review, in BioMart:

- *Pig genome*: there are 18466 Ensembl gene ID (from 21630) with at least one GO and a total of 180197 GO Term Accessions. One gene is associated to 0 to 246 GO Term Accessions (the average is about 8 GO per Ensembl gene ID).
- *Human genome*: there are 20632 Ensembl gene ID (from 22699) with at least one GO and a total of 774505 GO Term Accessions. One gene is associated to 0 to 1849 GO Term Accessions (the average is about 31 GO per Ensembl gene ID).

For genes in the same family, the gene annotation may be ambiguous between species, with possible false contributions to a function when using the Human genome instead of the Pig genome. However, using the Human genome strongly increases the number of associated functions. For this reason, the Human genome is preferred in the sequel, as a referenced mammalian genome. The lists of genes obtained from the different clusters obtained in Section 4.4 will be further studied. For instance, Table 1 shows an extract of some related functions for four of the 43 annotated genes. No functional information could be retrieved for the *ACBD5* gene while the *PDE8A* gene is much better annotated.

[12] http://geneontology.org
[13] http://www.genome.jp/kegg
[14] https://david.ncifcrf.gov
[15] https://david.ncifcrf.gov/ease/ease1.htm
[16] http://www.ensembl.org/biomart/martview/79399dc2f5745752a66a5a4a43f32a38

**Table 1** Example of some systematic functional annotation for four genes out of the 43 that are annotated. These results were obtained with the EASE application.

| Gene Symbol | GO Biological Process | GO Cellular Component | GO Molecular Function | KEGG pathway |
|---|---|---|---|---|
| ACBD5 | | | | |
| DECR2 | alcohol; metabolism | peroxisome | oxidoreductase activity | |
| ITGA8 | cell adhesion | plasma membrane | cell adhesion; molecule activity | |
| PDE8A | cell communication | insoluble fraction | transition; metal ion binding | Purine metabolism |

## *5.2 Biological enrichment*

Here, the reference genome for the pig species is the human genome in order to obtain richer biological information related to each gene. Another reliable step of the analysis of large transcriptomic data or of the clustering of coexpressed genes consists in identifying enriched biological functions associated with a set of selected genes.

Many free softwares (STRING[17], GeneCodis[18], WebGestalt[19], DAVID[20] among others) or softwares under license as Ingenuity Pathway Analysis (IPA[21] and others) are available to obtain enriched biological functions under different terms. The overall process is most often the same:

1. The first step is to attribute known biology terms for each gene from several databases (see Section 5.1). The most usual ones can be found below, but other reference databases may be more relevant to the studied species:
   - Gene Ontology[22];
   - KEGG [23];
   - Transcription factors[24] may give information about the transcription regulation of the targeted gene in the reference genome based on the known cis-regulatory element. This information could be particularly interesting with a co-expression analysis but must be used with care when dealing with data from homologous species;

[17] http://string-db.org
[18] http://genecodis.cnb.csic.es
[19] http://bioinfo.vanderbilt.edu/webgestalt
[20] https://david.ncifcrf.gov
[21] http://www.ingenuity.com/products/ipa
[22] http://geneontology.org
[23] http://www.genome.jp/kegg
[24] http://www.broadinstitute.org/gsea/msigdb

- Others, such as Omic Tools[25], are useful for retrieving regulating miRNA or other non-coding RNA, common protein domain, co-cited in publications...

2. The second step is to identify the terms from the above lists and count the number of genes for each term [18]. A statistical test will then give the significance of the enrichment (Fisher's exact tests based on hypergeometric distribution [12] and correction for multiple testing[4]).

With the 43 annotated nodes provided in this example, Webgestalt[26] recognized 40 unique genes with *e.g.*, "RNA transport" pathway significantly enriched (related to 3 nodes/genes, *PABPC1*, *EEF1A1*, *EEF1A2*). With GeneCodis[27], co-occurrence findings are possible: three genes (*EEF1A1*, *NCOA2*, *THRB*) are significantly associated with "regulation of transcription, DNA-dependent (BP), nucleus (CC), protein binding (MF), V$MAZ_Q6" (transcription factor targets) meaning that the products of these three genes are localized in the nucleus with protein binding activity to regulate the transcription. The transcription factor MAZ (MYC-associated zinc finger protein (purine-binding transcription factor)) was demonstrated to be able to regulate the expression of these three genes.

In Table 2, from the 11 recognized genes (column "List size"), out of the 21 nodes of cluster 5 (see Figure 11), two gene products (DECR2 and ACBD5, column "Support") are associated with a peroxisome localization in the cell. This function was said enriched compared to the 105 genes (column "Reference support"), which are localized in the peroxisome, among the 34208 genes (column "Reference size") of the Human genome. To evaluate this enrichment a p-value based on hypergeometric distribution (column "p-value") and its corresponding corrected p-value (column "adj. p-value") were calculated.

[25] http://omictools.com/transcriptomics-c1178-p1.html
[26] http://bioinfo.vanderbilt.edu/webgestalt
[27] http://genecodis.cnb.csic.es/analysis

**Table 2** Enrichment analysis of the 21 nodes of the fifth cluster. This result was obtained with GeneCodis and the Human genome as reference. To read the table, see the explanation in the text.

| Items | Details | Support | List size | Ref. support | Ref. size | p-value | adj. p-value | Genes |
|---|---|---|---|---|---|---|---|---|
| GO:0006355 | regulation of transcription, DNA-dependent (BP) | 3 | 11 | 1609 | 34208 | 0.01290 | 0.01934 | PDE8A, NCOA2, ERC1 |
| GO:0005777 | peroxisome (CC) | 2 | 11 | 105 | 34208 | 0.00050 | 0.01462 | DECR2, ACBD5 |
| GO:0016020 | membrane (CC) | 5 | 11 | 4065 | 34208 | 0.00588 | 0.01763 | TMEM126B, CCDC56, ERC1, ITGA8, ACBD5 |
| GO:0016021, GO:0016020 | integral to membrane (CC), membrane (CC) | 4 | 11 | 2933 | 34208 | 0.01088 | 0.02177 | TMEM126B, CCDC56, ITGA8, ACBD5 |
| V$PAX4_03 | V$PAX4_03 | 3 | 11 | 1033 | 34208 | 0.00378 | 0.02267 | ARHGAP8, ACBD5, MGEA5 |
| GO:0016020 | membrane (CC) | 5 | 11 | 4065 | 34208 | 0.01588 | 0.04262 | TMEM126B, CCDC56, ERC1, ITGA8, ACBD5 |
| GO:0000139 | Golgi membrane (CC) | 2 | 11 | 420 | 34208 | 0.00769 | 0.04458 | ERC1, B2M |
| GO:0007275 | multicellular organismal development (BP) | 2 | 11 | 945 | 34208 | 0.03554 | 0.0485 | ERC1, ITGA8 |

### 5.3 Biological networks

Biological networks can be constructed with free software like STRING (http://string-db.org) for functional association networks mainly based on Known and Predicted Protein-Protein Interactions but using also indirect (functional) associations (conserved co-expression data) or previous knowledge from literature.

Another software is Ingenuity Pathway Analysis (IPA), under license, which not only allows the user to find enrichment for the called canonical pathways or biofunctions and others, but also extracts biological networks based on all possible relationships across many databases and literature. IPA can propose networks with a limited total number of nodes (35, 70 or 140 nodes) including the best interactions between the input genes (in priority) and additional genes to obtain significant networks ranked with an associated score. Biological functions are associated with the proposed networks. In our example, cluster 2 contains 21 nodes, out of which five genes had an associated Biological Process enriched with GeneCodis and only 2 genes with Webgestalt. Only 50% of the nodes were used to find associated biological functions because of the limitation of annotation and there was available biological information for about 10-35% of the nodes.

The Ingenuity Pathway Analysis recognized the 11 annotated genes. IPA possessed a rich Ingenuity Knowledge Base with automated and manually curated information from all the databases presented before and also referenced all gene by possible gene interaction. Figure 13 shows the IPA network including all the 11 an-

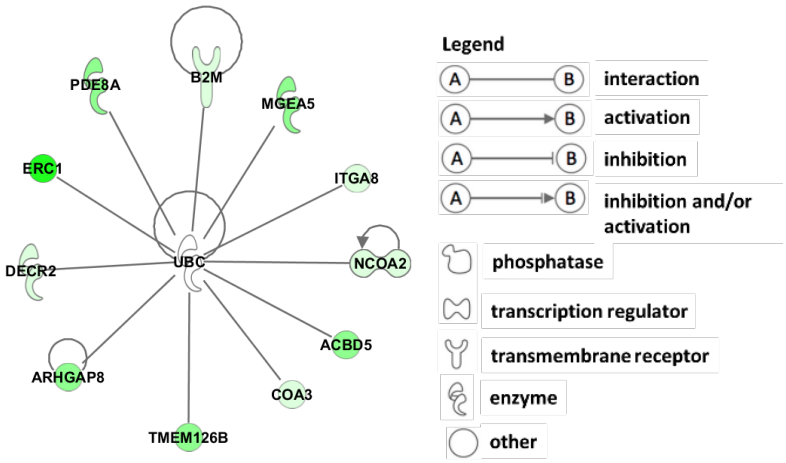


**Fig. 13** IPA network including all the 11 annotated genes of cluster 2.

notated genes of cluster 2. Associated functions are Organismal Survival (4 genes), Development (3 genes), Expression regulation (2 genes). The colour code is related to the betweenness centrality of the node in the largest connected component be-

fore clustering (highest for *ERC1*). Figure 14 shows the network as displayed by

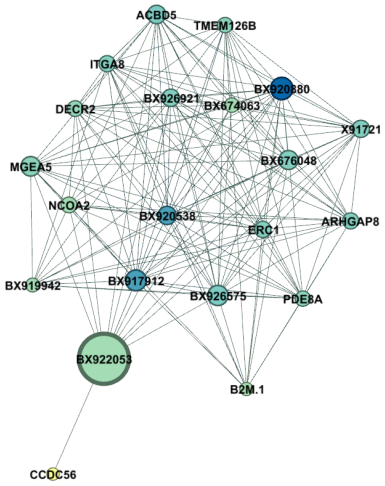


**Fig. 14** Cluster 2 as displayed by Gephi.

Gephi[28][3] (this software easily imports graphs in graphml format as described in Section 4.4). The node size corresponds to the betweenness centrality and the colour intensity corresponds to the node degree, both restricted to the subgraph induced by the nodes in cluster 2.

Figures 13 and 14 correspond to two representations of the same cluster 2. The first one used the available biological information to propose an optimized network. The second one is built with the initial information on co-expression without prior biological knowledge. As observed in our previous work [32], every cluster was associated with only one IPA network. In this case, 100% of the annotated genes of cluster 2 are included in the same IPA network (it was only about 80% for all clusters in our original work). Compared to the original paper [32], it has to be noted that the initial annotation of *CCDC56* was changed into *COA3* (cytochrome c oxidase assembly protein 3) by IPA: both names are indeed aliases. This simple example shows that a careful control of all the steps of functional annotation has to be performed. Finally a biological hypothesis could be proposed for cluster 2, the density of which (0.74) is much higher than that of the entire network (0.15). Cluster 2 was found to correspond to the Ubiquitin Proteasome Pathway (see `http://www.genome.jp/kegg-bin/show_pathway?hsa03050` for details) where the Ubiquitin protein binds most substrate proteins before their degradation by the proteasome.

These tools may be useful to help biologists to explore list of genes or proteins coming from high throughput technologies or lists coming from co-expression networks to explore associated functions with each community/cluster/module. However, the biologist must not forget his/her original biological question. In [32], the aim was to identify key genes being regulated by a cis-eQTL and to underline pos-

[28] `https://gephi.github.io`

sible important relationships between the original list of genes. Key genes could be unknown genes important from an eQTL point of view or important in the network. Such insights may encourage further biological analyses. Taken altogether, this complete set of tools may be powerful to decipher the biological mechanisms and the genetics regulating the biology of a tissue and underlying complex traits of interest in an agronomic context.

## 6 Link with a phenotype

Since an eQTL study is not a differential study, links of the genes with eQTLs and any phenotype are expected to be erratic a priori. In the pig example, let us consider the meat pH as a phenotype of interest: it is linked with meat quality. No high correlation was found between pH and gene expressions. A finer analysis is hence needed. The idea is to link the network structure with the phenotype of interest using spatial statistical tools. On average are the genes of one cluster more correlated to the pH? Which genes are particularly correlated to the pH as well as their neighbouring genes on the network? Using spatial statistics, it is possible to detect modules and specific genes that are linked with a terminal phenotype. This analysis is not detailed in the present chapter and we encourage the interested readers to refer to [32].

## 7 Conclusion

The prime objective was to decipher the processes underlying a list a genes whose expression is (partially) under genetic control. Due to an incomplete annotation of mammalian genomes, we proposed a statistical approach based on Gaussian graphical models for estimating and mining co-expression of a list of genes. This has led us to highlight a small subset of interesting genes (genes that are highly linked or central in the graph structure), and modules of densely connected genes. Roughly speaking, these modules were enriched in a single biological function, leading to a better clarity in the biological interpretation of the complex system under study. Last but not least, all these meaningful results are the consequence of a joint work between statisticians and biologists, which proves the importance of the collaboration between the two fields.

## References

1. Auber, D.: Tulip: a huge graph visualisation framework. In: P. Mutzel, M. Jünger (eds.) Graph Drawing Softwares, Mathematics and Visualization, pp. 105–126. Springer-Verlag (2003)
2. Barabási, A., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)

3. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: E.e.a. Adar (ed.) Proceedings of the Third International AAAI Conference on Weblogs and Social Media, pp. 361–362. Menlo Park: AAAI Press, 2009 (2009). URL http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154
4. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B **57**, 289–300 (1995)
5. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C., Causton, H., Gaasterland, T., Glenisson, P., Holstege, F., Kim, I., Markowitz, V., Matese, J., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., Vingron, M.: Minimum information about a microarray experiment (miame)-toward standards for microarray data. Nature Genetics **29**(4), 365–371 (2001)
6. Butte, A., Kohane, I.: Unsupervised knowledge discovery in medical databases using relevance networks. In: Proceedings of the AMIA Symposium, pp. 711–715 (1999)
7. Butte, A., Kohane, I.: Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 418–429 (2000)
8. Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal **Complex Systems** (2006). URL http://igraph.sf.net
9. Dorogovtsev, S., Mendes, J.: Evolution of Networks. From biological Nets to the Internet and WWW. Oxford University Press (2003)
10. Dozmorov, M., Giles, C., Wren, J.: Predicting gene ontology from a global meta-analysis of 1-color microarray experiments. BMC Bioinformatics **12**(Supp 10), S14 (2011)
11. Edwards, D.: Introduction to Graphical Modelling. Springer, New York (1995)
12. Fisher, R.: On the interpretation of $\chi^2$ from contingency tables, and the calculation of P. Journal of the Royal Statistical Society **85**(1), 87–94 (1922). DOI 10.2307/2340521. JSTOR2340521
13. Fortunato, S., Barthélémy, M.: Resolution limit in community detection. In: Proceedings of the National Academy of Sciences, vol. 104, pp. 36–41 (2007). Doi:10.1073/pnas.0605965104; URL: http://www.pnas.org/content/104/1/36.abstract
14. Foygel, R., Drton, M.: Extended Bayesian information criteria for Gaussian graphical models. In: Proceedings of Neural Information Processing Systems (NIPS 2010), pp. 604–612. Vancouver, Canada (2010)
15. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**(3), 432–441 (2008)
16. Fruchterman, T., Reingold, B.: Graph drawing by force-directed placement. Software, Practice and Experience **21**, 1129–1164 (1991)
17. Gillis, J., Pavlidis, P.: "guilt by association" is the exception rather than the rule in gene networks. PlLoS Computational Biology **8**(3), e1002,444 (2012)
18. da Huang, W., Sherman, B., Lempicki, R.: Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Research **37**(1), 1–13 (2009)
19. Kogelman, L., Zhernakova, D., Westra, H., Cirera, S., Fredholm, M., Franke, L., Kadamideen, H.: An integrative systems genetics approach reveals potential causal genes and pathways related to obesity. Genome Medecine **7**, 105 (2015). DOI 10.1186/s13073-015-0229-0
20. Liaubet, L., Lobjois, V., Faraut, T., Tircazes, A., Benne, F., Iannuccelli, N., Pires, J., Glénisson, J., Robic, A., Le Roy, P., SanCristobal, M., Cherel, P.: Genetic variability or transcript abundance in pig peri-mortem skeletal muscle: eQTL localized genes involved in stress response, cell death, muscle disorders and metabolism. BMC Genomics **12**(548), 548 (2011)
21. Liu, H., Roeber, K., Wasserman, L.: Stability approach to regularization selection (StARS) for high dimensional graphical models. In: Proceedings of Neural Information Processing Systems (NIPS 2010), vol. 23, pp. 1432–1440. Vancouver, Canada (2010). URL http://machinelearning.wustl.edu/mlpapers/papers/NIPS2010_0834
22. Lysen, S.: Permuted inclusion criterion: A variable selection technique. Ph.D. thesis, University of Pennsylvania (2009)

23. Meinshausen, N., Bühlmann, P.: High dimensional graphs and variable selection with the lasso. Annals of Statistic **34**(3), 1436–1462 (2006)
24. Montastier, E., Villa-Vialaneix, N., Caspar-Bauguil, S., Hlavaty, P., Tvrzicka, E., Gonzalez, I., Saris, W., Langin, D., Kunesova, M., Viguerie, N.: System model network for adipose tissue signatures related to weight changes in response to calorie restriction and subsequent weight maintenance. PLoS Computational Biology **11**(1), e1004,047 (2015). DOI doi:10.1371/journal.pcbi.1004047. First co-author.
25. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Physical Review, E **69**, 026,113 (2004). DOI 10.1103/PhysRevE.69.026113.
URL http://www.citebase.org/abstract?id=oai%3AarXiv.org%3Acond-mat%2F0308217
26. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Physical Review, E **74**(016110) (2006)
27. Rossi, F., Villa-Vialaneix, N.: Représentation d'un grand réseau à partir d'une classification hiérarchique de ses sommets. Journal de la Société Française de Statistique **152**(3), 34–65 (2011).
URL http://publications-sfds.math.cnrs.fr/index.php/J-SFdS/article/view/82/73
28. Schaeffer, S.: Graph clustering. Computer Science Review **1**(1), 27–64 (2007)
29. Schäfer, J., Strimmer, K.: An empirical bayes approach to inferring large-scale gene association networks. Bioinformatics **21**(6), 754–764 (2005). DOI 10.1093/bioinformatics/bti062
30. Schäfer, J., Strimmer, K.: A shrinkage approach to large-scale covariance matrix estimation and implication for functional genomics. Statistical Applications in Genetics and Molecular Biology **4**, 1–32 (2005)
31. Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research **13**(11), 2498–2504 (2003)
32. Villa-Vialaneix, N., Liaubet, L., Laurent, T., Cherel, P., Gamot, A., San Cristobal, M.: The structure of a gene co-expression network reveals biological functions underlying eQTLs. PLoS ONE **8**(4), e60,045 (2013). DOI 10.1371/journal.pone.0060045