

22

Case Studies Using Graphs on Biological Data

R. Gentleman, D. Scholtens, B. Ding, V. J. Carey, and W. Huber

Abstract

In this chapter we consider four specific data-analytic and inferential problems that can be addressed using graphs. We demonstrate the use of the software and methods described in Chapters 20 and 21 on real problems in computational biology. We will show how one can investigate relationships between gene expression and protein-protein interaction data, how GO annotations can be used to analyze gene sets, how literature citations can be related to experimental data, and how gene expression data can be mapped on pathways.

22.1 Introduction

In our first example, we demonstrate how graphs can be used to perform an analysis that relates gene expression data to protein complex co-membership data. The question of interest was whether genes in a protein complex are more likely to have a similar pattern of gene expression than genes in different complexes. More details are reported by Balasubramanian et al. (2004), which in turn was based on the work of Ge et al. (2001). Balasubramanian et al. (2004) used two graphs defined on a common set of nodes: the genes present in yeast. The relationship represented by the edges in the first graph is co-membership in a cluster of correlated expression, while the edges in the second graph represent co-membership in a protein complex.

In our second example, we consider sets of genes and use the Hypergeometric distribution to identify GO terms that have an over-representation of the selected genes. Other categorizations, such as pathways, or chromosomal location (e.g., cytochrome band), can be analyzed similarly.

In the third example, data from the National Library of Medicine (NLM) are used to provide links between genes and scientific articles. We note that these relationships can be phrased in terms of a bipartite graph and use that observation together with standard techniques from social networks analysis to identify interesting relationships between genes and papers.

In the fourth example, we explore pathway data and demonstrate one way of relating gene expression data to pathway information. The analysis is mainly exploratory and demonstrates some of the benefits that accrue from linking R and Graphviz.

22.2 Comparing the transcriptome and the interactome

Our title for this section is largely the same as that of Ge et al. (2001); and we will demonstrate how to carry out the bulk of the analysis that they report, using tools in the packages `graph`, `Rgraphviz`, and `RBGL`. We will make use experimental data from the `yeastExpData` package.

The methods that we will consider can be implemented in many other ways but the advantage to using a graph-based approach is the abstraction that it provides. The models are similar to those discussed by Balasubramanian et al. (2004) and we refer the interested reader to the `GraphAT` package which can be used to reproduce their results.

Ge et al. (2001) assembled gene expression data from a yeast cell-cycle experiment (Cho et al., 1998), literature protein-protein interaction (PPI) data, and yeast two-hybrid data. We have curated the data slightly to make it simpler to carry out the analyses. In particular, we reduced the data to the 2885 genes that were common to all experiments.

The relevant data sets are `ccyclered`, which is a dataframe with 11 columns and 2885 rows describing the set of common genes, and `litG`, which is a graph representing the curated set of literature predicted protein-protein interactions. We note that this data set is not up to date, but retain it because it provides answers that coincide with those of Ge et al. (2001).

The information about which cluster a gene is in can be obtained from `ccyclered`. We use that to create a *cluster graph* (see Section 21.2). In the cluster graph, edges are between all genes that are in the same cluster, and no edges connect genes from different clusters. The graph `ccClust` has 30 complete subgraphs.

```
> library("yeastExpData")
> data(ccyclered)
> clusts <- split(ccyclered[["Y.name"]], ccyclered[["Cluster"]])
> cg1 <- new("clusterGraph", clusters = clusts)
> ccClust <- connectedComp(cg1)
```

We next turn our attention to a brief exploration of the literature based collection of protein-protein interactions. We make use of the data in `litG` and examine the *connected components* found therein.

```
> data(litG)
> ccLit <- connectedComp(litG)
> cclens <- listLen(ccLit)
> table(cclens)
```

cclens	1	2	3	4	5	6	7	8	12	13	36	88
	2587	29	10	7	1	1	2	1	1	1	1	1

We see that most of the proteins, 2587, do not have edges to others, and that there are a few, rather large sets of connected proteins. The largest one contains 88, the next largest 36. We plot these in Figures 22.1 and 22.2.

```
> ord <- order(cclens, decreasing = TRUE)
> sG1 <- subGraph(ccLit[[ord[1]]], litG)
> sG2 <- subGraph(ccLit[[ord[2]]], litG)
```

22.2.1 Testing associations

It is now easy to determine how many pairs of genes have both a protein-protein interaction and are found in the same expression cluster. To compute this, we simply find the intersection of the cluster-graph and the literature graph.

```
> commonG <- intersection(cg1, litG)
```

```
A graph with undirected edges
Number of Nodes = 2885
Number of Edges = 42
```

We see there are 42 edges in common. This might seem like a small number, but in fact it is significantly larger than what would be expected by chance. There are several ways to test this. One way is to generate an appropriate null distribution and to compare the observed value, 42, to the values from this distribution. To generate the null distribution, there are some reasons to consider random edge graphs (Erdős and Rényi, 1959), and this is what Ge et al. (2001) did. However, if one examines the random graphs generated using the random edge model, they seldom resemble the structure in the graph based on the observed data. We propose generating the null distribution by permuting the node labels on the observed data graph.

In the next code chunk, we show a small function that performs the node label permutation test. Notice, from Figure 22.3, that the maximum number of edges in the intersection of the permuted graphs is much smaller than that observed in our data, 42. This justifies our assertion that there

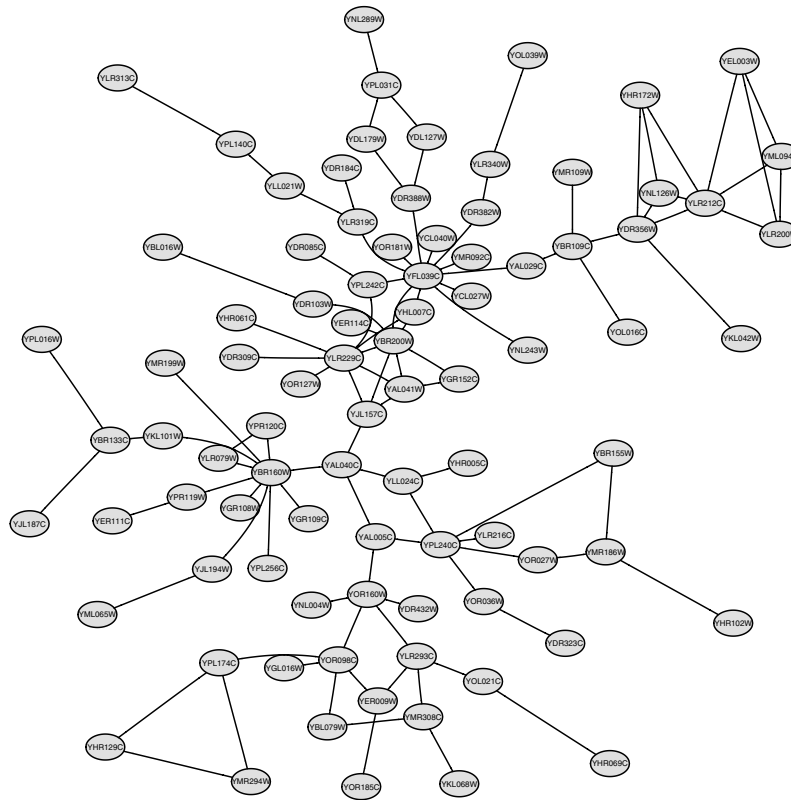


Figure 22.1. The largest PPI connected component.

is a significant relationship between gene expression pattern and protein complex co-membership, consistent with the findings of Ge et al. (2001).

```
> nodePerm <- function(g1, g2, B = 1000) {
+   n1 <- nodes(g1)
+   sapply(1:B, function(i) {
+     nodes(g1) <- sample(n1)
+     numEdges(intersection(g1, g2))
+   })
+ }
> set.seed(123)

> nPdlist <- nodePerm(litG, cg1)
```

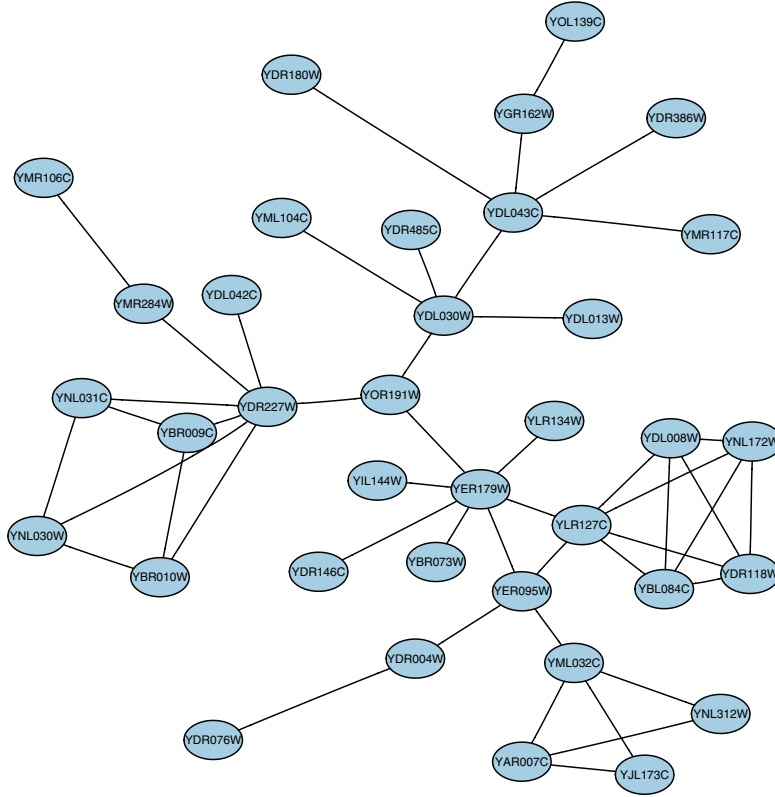


Figure 22.2. Another large PPI connected component.

22.2.2 Data analysis

Now that we have satisfied our testing curiosity, we might want to carry out a little exploratory data analysis. There are clearly some questions that are of interest including:

- Which of the expression clusters have intersections and with which of the literature clusters?
- Are there expression clusters that have a number of literature cluster edges going between them (and hence suggesting that the expression clustering was too fine or that the genes involved in the literature cluster are not cell-cycle regulated).
- Are there known cell-cycle regulated protein complexes, and do the genes involved tend to cluster together in both graphs?

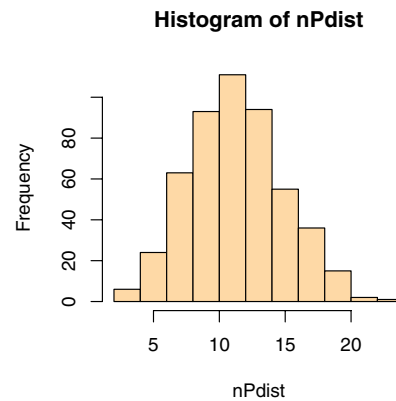


Figure 22.3. A histogram of the number of common edges as computed by a node label permutation model.

- Is the expression behavior of genes that are involved in multiple protein complexes different from that of genes that are involved in only one complex?

Many of these questions require access to more information. For example, we need to know more about the pattern of expression related to each of the gene expression clusters so that we can try to interpret them better. We need to have more information about the likely protein complexes from the literature data so that we can better identify reasonably complete protein complexes and given them, then identify those genes that are involved in more than one complex. But, the most important fact to notice is that all of the substantial calculations and computations (given the meta-data) can be phrased in terms of operations on graphs. This makes it both simple to think about what to do as well as to carry out the operations.

22.3 Using GO

In this section, we consider some of the ways in which data from GO can be used. A fairly extensive description of GO is given in Chapter 7 and we will presume that the reader is familiar with that material. Other more detailed examples involving GO and the analysis of genomic data are available through the vignettes in the `GOstats` package and in reference (Gentleman, 2004).

We make use of the ALL data (Chiaretti et al., 2004) to provide examples of how to make use of GO data in different data analytic situations. We select the B-cell leukemia cases, and from these, we will compare those

with BCR/ABL to those with no observed cytogenetic abnormalities (labeled NEG). To reduce the set of genes for consideration, we applied two different sets of filters. Gene filtering is considered in more detail in Chapter 14 and by von Heydebreck et al. (2004). A non-specific filter was used to remove genes that showed little or no change in expression level across experiments. The resulting data set had 2391 probes remaining. To select genes whose expression values were associated with the phenotypes of interest (BCR/ABL and NEG), we used the `mt.maxT` function from the `multtest` package, which computes a permutation based t -test for comparing two groups.

After adjustment for multiple testing, there were only 19 probes (which correspond to 16 genes) with an adjusted p -value below 0.05. Using those genes, we obtain the set of most-specific GO terms in the MF ontology that they are annotated at. We then use these terms, together with the parent-child relationships, to find the GO graph that contains all less specific terms and we refer to that graph as the *induced* GO graph. This graph is rendered in Figure 22.4. Nodes are labeled by the most specific four digits in their GO label, that is GO:0005125 is labeled as 5125. The most specific terms are at the top of the graph and arrows go from more specific nodes to less specific ones. The node in the bottom center is the MF node. Clearly some sort of interactivity would be beneficial and you might consider using the `imageMap` function from the `Rgraphviz` package.

22.3.1 Finding interesting GO terms

In our example, we have selected a set of genes that are thought to be expressed differently in two subgroups of interest but these same methods apply equally to sets of genes that have been obtained in other ways, say by some form of clustering. Then questions that arise are: whether genes that comprise a cluster have a common function; are involved in common processes; or perhaps, are co-located in some compartment of the cell.

The test is quite straightforward. Given a set of genes and a categorization of those genes, say using one of the three ontologies, we find the set of all unique GO terms within the ontology that are associated with one or more of the genes of interest (i.e., the induced GO graph). Next, for each term, we count the interesting genes annotated at that node and obtain the number of genes assayed that are annotated at the node. Basically, we form the two-way table that identifies a gene as interesting, or not, and as being annotated at the node, or not. The unique LocusLink identifiers, and not the manufacturers identifiers, should be used because there are often multiple probes for a single LocusLink identifier on each chip.

We can ask if there are more interesting genes at the node than one might expect by chance. If that is true, then that term can be thought of as being overrepresented in the data. This question can be answered using a Hypergeometric distribution. The function `GOHyperG`, available in

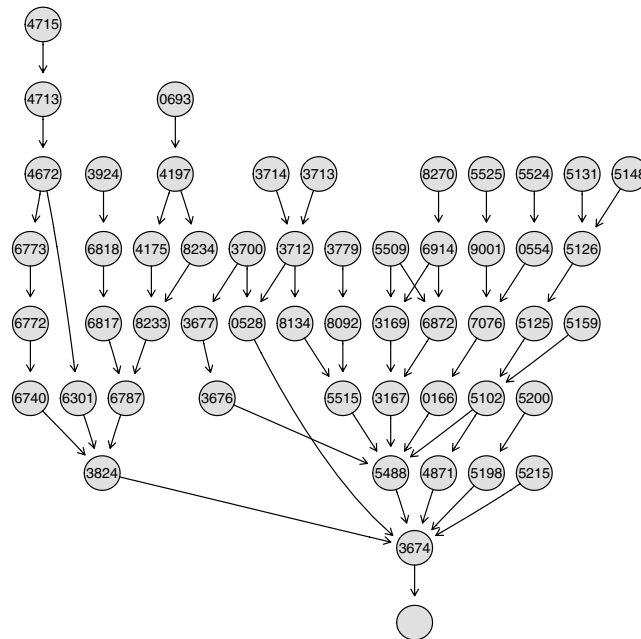


Figure 22.4. The induced GO graph for the selected genes; truncated GO identifiers are used as labels.

the **GOstats** package, takes as input a set of LocusLink identifiers, finds the induced GO graph and performs the Hypergeometric test at each node.

There are some issues that arise in the interpretation of the resultant p -values. First, we note that often very many hypotheses will have been tested and that some form of p -value correction will be needed. However, there is no simple or straightforward way to do that. The different hypotheses are not independent by virtue of the way that GO is structured and even with this difficulty addressed, we are most likely interested in patterns of p -values that correspond to structure in GO rather than single p -values that exceed some threshold. For these reasons, we prefer to report unadjusted p -values and leave corrections to the discretion of the user. These and other issues were considered in more detail by Gentleman (2004), however, much more research in this area is needed.

A second issue that arises is the fact that nodes of the induced GO graph with few genes annotated at them will typically have small p -values. This phenomenon occurs due to the way that we selected nodes for evaluation and the structure of GO. Recall that a gene annotated any node is also annotated at all less specific nodes in the GO hierarchy. Many genes are annotated out quite far into the leaves of the GO graph and hence at

	GO ID	Term	p	n
1	GO:0005131	growth hormone receptor b...	0.002	1
2	GO:0005148	prolactin receptor bindin...	0.004	2
3	GO:0005159	insulin-like growth facto...	0.011	6
4	GO:0003924	GTPase activity	0.014	101
5	GO:0008270	zinc ion binding	0.014	557
6	GO:0030693	caspase activity	0.021	12
7	GO:0004715	non-membrane spanning pro...	0.021	12
8	GO:0046914	transition metal ion bind...	0.026	663
9	GO:0043169	cation binding	0.029	1034
10	GO:0005488	binding	0.04	4825
11	GO:0005525	GTP binding	0.041	181
12	GO:0019001	guanyl nucleotide binding	0.043	187
13	GO:0004713	protein-tyrosine kinase a...	0.043	187
14	GO:0043167	ion binding	0.048	1185
15	GO:0046872	metal ion binding	0.048	1185
16	GO:0005126	hematopoietin/interferon-...	0.065	37
17	GO:0017076	purine nucleotide binding	0.087	976
18	GO:0000166	nucleotide binding	0.091	990

Table 22.1. GO terms, p -values, and numbers of genes for a selection of GO categories.

nodes that have relatively few other genes annotated there. Calculation of the Hypergeometric p -values for these nodes results in very small p -values. Others have dealt with this issue by defining the concept of depth in the GO graph (the number of edges to the root node) and then only using nodes that are neither too deep nor too shallow.

In the next code chunk, we show how to take an induced GO graph, `gGO`, and a set of interesting genes, `gNsLL`, and find the Hypergeometric p -values. This is done using `GOHyperG`. Because the data come from a HG-U95Av2 chip, we use the set of genes on that chip as the set of all genes in the Hypergeometric test. We then make use of the resultant p -values to provide colors for the nodes.

```
> gNsLL <- unique(unlist(mget(names(gde), env = hgu95av2LOCUSID,
+   ifnotfound = NA)))
> gGhyp <- GOHyperG(gNsLL)
```

In Figure 22.5, we reproduce the plot from Figure 22.4 except that we have now colored the nodes according to the p -value obtained from the Hypergeometric test described above. The nodes in Figure 22.5 are colored either dark red or light blue depending on whether the unadjusted Hypergeometric p -value was less than 0.1 or not. The GO terms for the nodes colored red are printed below. The relevant biology suggests that these are quite reasonable.

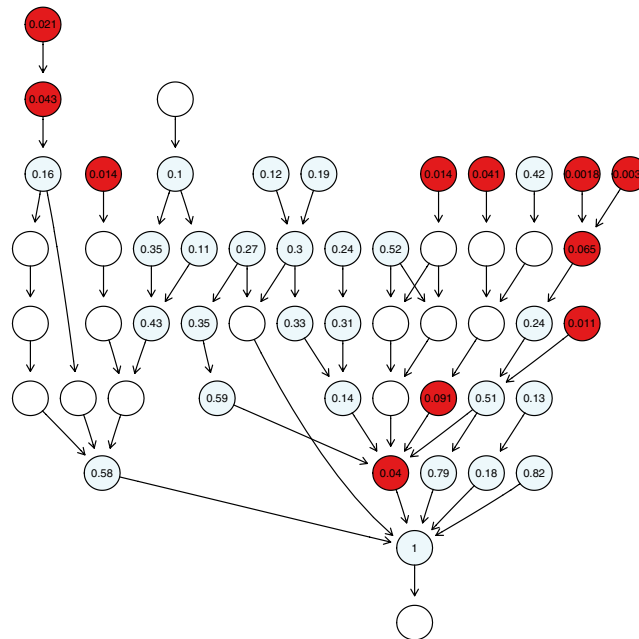


Figure 22.5. The induced GO graph colored according to unadjusted Hypergeometric p -values, whose values are given in the nodes.

We see in Table 22.1 that the nodes with the smallest p -values do tend to be the nodes with few genes annotated at them. However, there are also some nodes with quite small p -values and large counts, such as `GO:0008270` and `GO:0003924`, and these would surely be of some interest in subsequent explorations.

It is interesting to note that we can also ask, and answer, the question about underrepresented GO terms. That is, we can find nodes in the GO graph that, given their size, should have contained one or more interesting genes, under the null hypothesis.

22.4 Literature co-citation

In this section, we consider the graph structure of literature co-citation data and explore some of the ways it can be used to help add meaning to a data analysis. The basic statistical models and paradigm will be presented first, and subsequently we apply them to co-citation via PubMed; see Chapter 7 for more details on PubMed. There are many different problems that can be addressed using these data, but we will consider only a few of them.

One of the problems in providing concrete recommendations is the lack of a gold standard against which to measure the performance of the various tools. We have used a number of examples where we believe one can make a reasonable statement about whether two genes are related and then contrast the different measures and adjustments with respect to their agreement with this point of view. Of course, your opinion might be different, and in that case you would naturally select a test statistic to use accordingly. More details on the approach and more extensive examples were given by Ding and Gentleman (2004).

One can consider citation in terms of a bipartite graph. The genes represent one type of node and the scientific papers represent the other type of node. An edge exists between a gene and a paper if the gene is cited in the paper. In this graph, there are no edges between papers and no edges between genes. The relationships between genes are mediated by the papers and the relationships between papers are mediated by the genes. From this bipartite graph, we can generate two *one mode* graphs. One is the graph whose nodes are genes and an edge exists between two genes if they are co-cited in one or more papers. Edge weights can be used in this graph to count the number of co-citations. The second type of one mode graph that is of some interest is the graph whose nodes represent papers, and an edge exists between two papers if they co-cite at least one gene. Edge weights can be used to represent the number of genes that have been co-cited.

In the context of a co-citation graph (see Section 20.2.1 for more details), the *actor size* is the number of papers that cite the gene of interest, while the *event size* is the number of genes that are cited by a specific paper. We note that some adjustment for either actor or event size can improve the inference and should be considered; we discuss this in Section 22.4.2. For example co-citation in a paper such as that by Strausberg et al. (2002), which cites more than 15,000 genes, has very little information and one would not generally treat co-citation in this paper as indicating any relationship between genes. On the other hand, co-citation in a paper that discusses only two or three genes is a much stronger indication of an intrinsic biological relationship. Interested readers are referred to Section 22.4.1 below, Chapter 8 of the book by Wasserman and Faust (1994), and the article of Ding and Gentleman (2004) for further considerations.

The concepts of adjacency, reachability, and connectedness can all be applied to bipartite graphs, and hence to affiliation networks. Of these, the strongest and most interpretable property will be adjacency. If we consider the co-citation network, the notion of a relationship based on reachability seems very vague and would be difficult to interpret. Similarly, it will be difficult to place much meaning on the path length between two genes, or two papers. We also note that that notions reachability, diameter, and connectedness in the one mode networks are likely to be of little biological interest. In a co-citation graph, only the direct co-citations are likely to be important. Two genes that are co-cited will share an edge, and it is not clear

	<i>gene</i> ₂		
<i>gene</i> ₁	<i>n</i> ₁₁	<i>n</i> ₁₂	<i>n</i> _{1.}
	<i>n</i> ₂₁	<i>n</i> ₂₂	<i>n</i> _{2.}
	<i>n</i> _{.1}	<i>n</i> _{.2}	<i>n</i>

Table 22.2. Notational conventions for a two-way table.

that the existence of a path, say through some third gene, is any evidence of a relationship. One might be willing to argue that the existence of many, very short paths between two genes of interest constitutes evidence of a relationship, but that requires a different approach.

22.4.1 Statistical development

For many of the two-way tables that arise in bioinformatics, one of the entries in the table is much much larger than the other ones. For example, with co-citation comparisons, or when comparing annotation at a particular GO term with some other property, we find that most genes have neither property and so one entry in the two-way table is very large compared to the other three. To facilitate discussion, we will use the convention that the n_{22} entry in the two-way table is the one with the very large number. To further ease the exposition, we will base some of the discussion on the notion that we want to compare two genes, $gene_1$ and $gene_2$, on the basis of their co-citations in the medical literature.

When one entry in the table (Table 22.2) is much larger than the others, the actual distribution of the test statistics may be quite far from the asymptotic distributions that are commonly used to assess significance. It may be prudent to rely on test statistics that either do not use n_{22} or which do not depend heavily on it. Some of these were studied by Ding and Gentleman (2004) and we discuss their findings here. Ding and Gentleman (2004) considered a wide range of statistics and recommended the three following statistics as performing well, under different situations.

- **Concordance measure**

$$n_{11}$$

- **Jaccard Index** (Jaccard, 1912)

$$\frac{n_{11}}{n_{11} + n_{12} + n_{21}}$$

- **Hubert's Γ** (Hubert, 1987; Good, 1994)

$$\frac{n_{11}n_{22} - n_{12}n_{21}}{\sqrt{(n_{11} + n_{12})(n_{21} + n_{22})(n_{11} + n_{21})(n_{12} + n_{22})}}$$

The range of the Jaccard Index is $[0,1]$ and for Hubert's Γ the range is $[-1,1]$. Hubert's Γ is equivalent to the fourfold correlation coefficient.

Ding and Gentleman (2004) carried out an empirical study of the performance of the different test statistics together with adjustments for both event size and actor size. Although there is no gold standard by which to compare these different test statistics, it is nonetheless important to attempt to understand the properties of the different test statistics. Notions of power and size are therefore approximate, and based on comparisons where a biological association between genes could be determined, and other cases using genes where it was very unlikely that any true biological association exists. That is, Ding and Gentleman (2004) considered genes that are likely to have a biologically meaningful relationship, as well as those that, despite frequent co-citation, are not likely to have a biologically meaningful relationship. They found that the χ^2 and odds ratio based statistics do not, in general, perform as well as the Concordance, Jaccard Index and Hubert's Γ based statistics. They also found that actor size adjustment tends to make tests too conservative, whereas event-size adjusted Concordance and Jaccard Index tend to be too anti-conservative. The necessary software for carrying out these tests is provided in the **CoCiteStats** package.

Let \mathcal{N} denote the set of actors, with cardinality n , and let \mathcal{M} denote the set of events, with cardinality m . We denote the affiliation matrix as \mathbf{A} , where $\mathbf{A}_{i,j}$ is 1 if actor i was present at event j . The corresponding one mode networks can be then be found as $\mathbf{X}^{\mathcal{N}} = \mathbf{A}\mathbf{A}'$ and $\mathbf{X}^{\mathcal{M}} = \mathbf{A}'\mathbf{A}$. Note that in $\mathbf{X}^{\mathcal{M}}$ the i, j entry is the number of events that both actor i and actor j attended. In some cases we will be interested in Boolean versions of these matrices, that is versions of $\mathbf{X}^{\mathcal{M}}$ and $\mathbf{X}^{\mathcal{N}}$ that have entries that are zero or one, which indicate whether actors i and j attended one or more events together.

We return to the subject of actor and event size adjustments. We note that a very large event (a paper that cites very many genes) is likely to co-cite two genes, but the information about their relationship is weaker than if they were co-cited in a paper that cited only a small number of genes. Considering, instead, actors we see that an actor (or gene) that attends many events is much more likely to be affiliated with other actors than an actor that attends few events. In the context of co-citation, this says that a well studied gene is more likely to be associated with other genes than a recently discovered, or recently studied, gene.

One argument that is often made in social network theory is that the measure of association between actors should be logically independent of the event size. When the data are presented in the form of a two-way table, the odds ratio is one measure of association that is logically independent of group size. An alternative discussed in Wasserman and Faust (1994) is to normalize either of $\mathbf{X}^{\mathcal{M}}$ or $\mathbf{X}^{\mathcal{N}}$ so that all row and column totals are equal; this idea will not be explored here.

The use of $\mathbf{X}^{\mathcal{N}}$ intrinsically assumes equal weighting of papers. The size of the papers, however, may also play a key role in deciding significance of association between genes and some adjustment may be needed. There are

various ways of doing this, but in principle one should down-weight large papers as their information content is less. We consider a weight equal to the inverse of the number of genes cited for each paper, i.e., paper size.

22.4.2 Comparisons of interest

Now we revise the gene-gene contingency table, Table 22.2, for the case where the comparison of interest is between two genes, $gene_1$ and $gene_2$, on the basis of co-citation. We let $n'_{ij} = \sum_{l \in Pub_{ij}} 1/N_l$, $i, j = 1, 2$ where N_l is the size of paper l , i.e., the number of genes cited by PubMed l , Pub_{11} is the set of papers citing both genes; Pub_{12} citing $gene_1$ but not $gene_2$, Pub_{21} citing $gene_2$ but not $gene_1$, and Pub_{22} are those citing neither genes. Hence n'_{ij} is a weighted version of n_{ij} where the weight depends on the number of genes cited by each paper. We can then use the three statistics proposed in Section 22.4.1, Concordance, Jaccard's index, and Hubert's Γ , with n_{ij} replaced by n'_{ij} .

22.4.3 Examples

We begin with a small example to clarify some of the relevant issues. TRO and BYSL form a complex mediating cell adhesion. Suzuki et al. (1999) studied expression of these two genes in human placenta. These two genes are the only two human gene products referred to in this paper (PMID: 10026108). Conversely, they were also co-cited in Strausberg et al. (2002) (PMID: 12477932) where ESTs were generated from libraries enriched for full-length cDNAs; there is no direct association between the genes they have cited other than the fact that their cDNA sequences can be obtained. So we can see that the paper by Suzuki et al. (1999) is very informative about these two genes, and their potential relationship, while that by Strausberg et al. (2002) is not.

We consider the Concordance measure, Hubert's Γ , and the Jaccard Index. For all three we also consider gene size adjustments, paper size adjustments and both gene and paper size adjustments, thus yielding four statistics for each of these.

Example 1

We first look at the association between two genes, BYSL with LocusLink ID 705 and TRO with LocusLink ID 7216. As noted above, they have been co-cited twice (PMID: 12477932, 10026108) where the second paper cited only these two genes and the first one cited 14596 genes. Even though one of the papers citing both is general (PMID: 12477932), the other (PMID: 10026108) is a very specific paper discussing the two genes. Moreover, the two genes were cited in only 4 and 8 papers respectively, hence we believe that there is an association between them and we would like to use a test statistic that is capable of detecting that relationship.

		7216	
705	2	2	4
	6	74666	74672
	8	74668	74676

	Concordance	Jaccard	Hubert
None	2.0000 (0.0000)	0.2000 (0.0600)	0.3535 (0.0800)
GS	0.9911 (0.1000)	0.9824 (0.1000)	0.9822 (0.1000)
PS	0.5001 (0.0000)	0.0832 (0.0000)	0.1579 (0.0000)
BOTH	0.9855 (0.0800)	0.9715 (0.0800)	0.9710 (0.0800)

Table 22.3. PubMed co-citation: Locuslink ID 705 and 7216.

Using a Hypergeometric distribution the exact p -value for testing the null hypothesis that gene 705 and 7216 are not related is 0.377 when no edge weights are considered, indicating no significant association between them. Failure to account for the edge weights may offer an explanation.

Table 22.3 reports the results for the three statistics from Section 22.4.1. For each statistic, we also considered four versions: no adjustment (None), gene size adjustment (GS), paper size adjustment (PS) and both gene and paper size adjustment (Both). The numbers listed in each entry are the score and p -value (in parentheses).

Results from Concordance, Jaccard Index, and Hubert's Γ are quite consistent, the original Concordance statistic and paper size adjusted Concordance, Jaccard Index, and Hubert's Γ are significant at 0.05 level. This suggests that paper size adjustment is useful especially as one of the papers under investigation is extremely large in size. The adjustments for gene size all lead to non-significant results.

An analysis using GO by Ding and Gentleman (2004) indicated that the two genes are highly significantly related in their biological processes.

Example 2

The previous example suggests that both the number of co-citations and the paper size are important in determining the level of significance. To see this more clearly, we consider genes 10038 (ADPRTL2) and 10039 (ADPRTL3) which are co-cited four times. The sizes of the papers citing 10038 and 10039 are 3,2,2,2, all relatively small compared with previous examples. Moreover, the genes were cited 7 and 8 times respectively.

	10039		
10038	4	3	7
	4	74665	74669
	8	74668	74676

	Concordance	Jaccard	Hubert
None	4.0000 (0.0000)	0.3636 (0.0000)	0.5345 (0.0000)
GS	0.9937 (0.0000)	0.9875 (0.0000)	0.9874 (0.0000)
PS	1.8333 (0.0000)	0.3771 (0.0000)	0.5476 (0.0000)
BOTH	0.9956 (0.0000)	0.9913 (0.0000)	0.9913 (0.0000)

Table 22.4. PubMed co-citation: Locuslink ID 10038 and 10039.

All results reported in Table 22.4 are significant. This suggests that if paper size is small then there is no obvious need for paper size adjustment; almost all the statistics, with or without adjustment, yield similar results.

Application to gene lists. Here we use the test statistics, suggested above, but aggregate them over the set of genes in the gene list or over the boundary of the gene list.

Given a list of genes, D , one can find the boundary of that list, with respect to the one mode co-citation graph \mathbf{X}^N . This boundary is simply the set of genes that were co-cited one or more times with the genes in D . Because there are many papers that cite thousands of genes, the boundary itself will not be very interesting, and we will typically restrict our attention to those genes where the sum of the edge weights exceeds some threshold. This cut-off can be determined empirically.

Once the boundary has been determined, we might want to find those genes that have a particularly strong association with the genes in D . While parametric tests are not generally available, a resampling test can be used to assess significance. Alternatively, we can compute pairwise relationships between the members of D itself. These distances, could then be analyzed, using multidimensional scaling or they could form the basis for yet another graph.

We return to the ALL example begun in Section 22.3. In that example, we selected genes whose expression values were associated with the phenotypes of interest (BCR/ABL and NEG) using a permutation-based t -test to compare the two groups. We found 19 probes, corresponding to 16 genes, that had adjusted p -values below 0.05. Suppose that we wanted to find out whether there are subsets of these genes that are closely related, according to co-citation. We can also ask if there are other genes that are

closely related to the selected genes that we did not find. We first obtain the unique LocusLink identifiers and then map these to the set of papers that cite the genes. We begin with the data object `intLLc` that contains the LocusLink identifiers for the selected genes. For each of these we first obtain the number of citations for each gene.

```
> papersByLL <- mget(intLLc, humanLLMappingsLL2PMID,
+   ifnotfound = NA)
> ncit <- sapply(papersByLL, length)
> ncit
```

25	687	195	2534	23145	7277	841	4599	2273	87
68	5	7	28	3	10	94	24	10	6
6935	9697	9900	3937	1396	8835				
10	5	4	11	4	12				

We see that the number of citations ranges from 94 to 3. Next, we can construct a simple co-citation graph, on these genes and here we need only concern ourselves with this rather small set of papers. The paper sizes were also computed and they range from 14596 to 1.

```
> num <- length(papersByLL)
> grels <- vector("list", length = num)
> names(grels) <- names(papersByLL)
> for (i in 1:num) {
+   curr <- papersByLL[[i]]
+   grels[[i]] <- lapply(papersByLL, function(x) {
+     mt <- match(x, curr, 0)
+     if (any(mt > 0))
+       curr[mt]
+     else NULL
+   })
+ }
> for (i in 1:num) grels[[i]] <- grels[[i]][-i]
```

We have now computed the edges that are present in our graph. Next we want to see which papers co-cite genes from among our list.

```
> gr2 <- lapply(grels, function(x) {
+   slen <- sapply(x, length)
+   x[slen > 0]
+ })
> table(unlist(gr2))
```

12477932	14702039
132	30

We notice that all of the co-citations between the genes we have selected are due to two papers, one by Strausberg et al. (2002) and a similar one by Ota et al., and hence there is no information about relationships between these genes to be gleaned from the currently available medical literature.

We can take a more exploratory approach. For instance, starting with the same set of genes, the boundary of their co-citation graph can be examined. That is, we are looking for all genes that have a co-citation with one or more of the genes in our list. We will need to discount the very large papers, and hence we will make use of edge weights in constructing our graph and subsequently will trim those elements of the boundary with edge weights that are small.

Finding the boundary is relatively straightforward. Given our list of genes, we first find their citations, and using those citations we find the information on genes cited in those papers. In the next code chunk, a simple function, `LL2wts`, that carries out this computation is provided. Given a set of LocusLink IDs it finds all papers that cite these genes. Then, taking those papers, it finds all genes they cite and creates a weight vector, where the weights are 1 over the papers sizes. Finally, a list of the named weight vectors is output.

```
> LL2wts <- function(inList) {
+   pBLL <- mget(inList, humanLLMappingsLL2PMID,
+               ifnotfound = NA)
+   numL <- length(inList)
+   ans <- NULL
+   for (i in 1:numL) {
+     lls <- mget(as.character(pBLL[[i]]),
+               humanLLMappingsPMID2LL,
+               ifnotfound = NA)
+     lens <- sapply(lls, length)
+     names(lens) <- NULL
+     wts <- rep(1/lens, lens)
+     wtsbyg <- split(wts, unlist(lls, use.names = FALSE))
+     ans[[i]] <- sapply(wtsbyg, sum)
+   }
+   ans
+ }
> vv <- LL2wts(intLLc)
```

Given `vv`, we can answer a number of questions. For example, we can find which of the elements of `vv` have the largest weights, we can see which genes are connected to more than one gene in our list of interesting genes, and of those, which have relatively high weights.

```
> allLL <- unique(unlist(sapply(vv, names)))
> bdrywts <- rep(0, length(allLL))
> names(bdrywts) <- allLL
> for (wvec in vv) bdrywts[names(wvec)] <- bdrywts[names(wvec)] +
+   wvec
> wts <- bdrywts[!(allLL %in% intLLc)]
> sum(wts > 1)
```

```
[1] 20
```

```
> range(wts[wts > 1])
```

```
[1] 1.08 9.00
```

We can see that there are 20 genes that have weights that are larger than 1 and hence might warrant further study. We can find those that are on the HG-U95Av2 chip by using the chip-specific annotation package, `hgu95av2`.

```
> LL95 <- unlist(as.list(hgu95av2LOCUSID))
> bdryLL <- names(wts[wts > 1])
> onC <- match(bdryLL, LL95, 0)
> unlist(mget(names(LL95[onC]), hgu95av2SYMBOL))
```

517_at	1084_at	2043_s_at	1441_s_at	2024_s_at
"SHFM3P1"	"ABL2"	"BCR"	"FAS"	"LYN"
879_at	32725_at	38350_f_at	40567_at	34448_s_at
"MX2"	"BID"	"TUBA2"	"TUBA3"	"CASP2"
36143_at	38281_at	486_at	1765_at	38755_at
"CASP3"	"CASP7"	"CASP9"	"CASP10"	"FADD"
1867_at	40969_at	35681_r_at		
"CFLAR"	"SOCS3"	"ZFXH1B"		

22.5 Pathways

In this section, we consider some uses of pathway information in the analysis of gene expression data. Although the concept of a pathway does not have a rigorous definition, the general concept is widely used. For example, the biological process ontology from GO describes itself as being less than a pathway.

Associating gene expression data with pathways has been considered by many others, including Doniger et al. (2003). In some applications, one might render a pathway and color the nodes (genes) according to changes in expression across experimental conditions. Although this approach has some appeal, there are other uses for pathway data. Pathways can be used to perform subgroup analysis where interest is restricted to a set of genes that are associated with a particular pathway. However, there are many situations where one would not expect the expression levels to change. For example, many signal transduction pathways are known to end in the activation of a transcription factor. Thus, to know if the pathway is active, it seems more reasonable to study the targets of the transcription factor than the constituent elements of the pathway.

In our first example, we consider the network structure of the pathways themselves. We make use of the bipartite graph that relates genes and pathways and study the one mode network on pathways that results from it. In our second example, we take a single pathway, the integrin-mediated

cell-adhesion pathway, and render it in different ways, using gene expression data to modify the outputs.

22.5.1 *The graph structure of pathways*

Consider the bipartite graph where one set of nodes are genes and the other set of nodes are pathways. We are interested in understanding the relationships between pathways due to shared genes, or shared sets of genes. We represent the bipartite graph in terms of an incidence matrix; see Section 20.2.1 for more details.

We construct the graph based on the data available from the HG-U95Av2 GeneChip array from Affymetrix. It might be of more interest to consider the construction of this graph based on all mappings for a given organism rather than restricting our attention to a particular chip, but this restriction makes the computations manageable. The construction is considered in some detail as readers are likely to find it useful for creating their own bipartite graphs. There are two relevant mappings, those from probes to pathways, and the converse, from pathways to probes. For the HG-U95Av2 chips these are available as `hgu95av2PATH`, which holds the mappings from probesets to the pathways, and `hgu95av2PATH2PROBE`, which contains the mappings from pathways to probesets. We first load the necessary libraries and then look to see how many pathways different genes are annotated at.

```
> library("hgu95av2")
> library("annotate")
> gene1 <- unlist(eapply(hgu95av2PATH, length))
> table(gene1)
```

gene1	1	2	3	4	5	6	7	8	10	11
11264	635	363	208	71	33	6	10	7	10	
	12	13	15							
	7	3	8							

We see that some genes are annotated at many pathways, while most are annotated at only one. Since genes are annotated at pathways using LocusLink identifiers we next reduce the data by removing any duplicate probes.

```
> pathLL <- eapply(hgu95av2PATH2PROBE, function(x) {
+   LLs <- getLL(x, "hgu95av2")
+   unique(LLs)
+ })
> pLens <- sapply(pathLL, length)
> range(pLens)
```

```
[1] 1 219
```

```
> uniqLL <- unique(unlist(pathLL, use.names = FALSE))
```

We see that pathway sizes are between 1 and 219 for LocusLink identifiers from this chip. We note that these sizes are with respect to the set of genes that we have information on. The actual size (number of genes) in a pathway could be quite different, and for some calculations we will want the actual set of genes, but for others we will need to focus on those genes for which we have data.

Now that we have computed `pathLL`, that is really all that is needed. We can find out how many pathways there are (136), and how many unique LocusLink identifiers there are (2297). In the incidence matrix representation of our bipartite graph, we let LocusLink identifiers denote the rows and pathways denote the columns. The data in `pathLL` are easily transformed to an adjacency matrix where the pathways are the columns, and the genes are the rows.

```
> Amat <- sapply(pathLL, function(x) {
+   mtch <- match(x, uniqLL)
+   zeros <- rep(0, length(uniqLL))
+   zeros[mtch] <- 1
+   zeros
+ })
```

Now that we have an incidence matrix for the pathways, we can construct the one mode graphs for genes and for pathways. We leave the gene graph for the reader to explore and instead consider the pathway graph. The diagonal entries of `pwGmat` will be the counts of the number of genes in each pathway. We set these to zero so that they do not get interpreted as self-loops.

```
> pwGmat <- t(Amat) %*% Amat
> diag(pwGmat) <- 0
> pwG <- as(pwGmat, "graphNEL")
```

Although we could use `Rgraphviz` to lay out the graph, it has too many nodes and edges to provide a meaningful visualization using standard layout methodologies. Further research is needed to develop good layout strategies for this graph. However, we can examine some of the basic characteristics of the graph.

We can find the connected components.

```
> ccpwG <- connectedComp(pwG)
> sapply(ccpwG, length)
```

```
   1   2   3   4   5
132   1   1   1   1
```

We see that there are four singletons, and otherwise all the pathways are connected by the genes that are assayed on the HG-U95Av2 chip. In the next code chunk we find and print the names of the singletons.

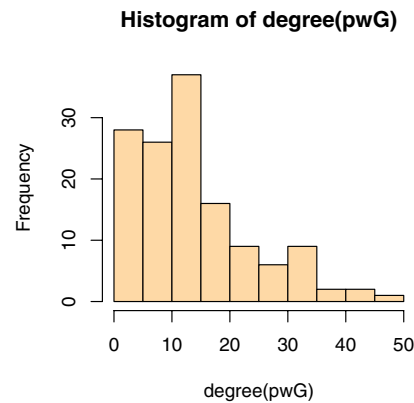


Figure 22.6. The degree distribution of the pathway graph.

```
> library("KEGG")
> for (i in ccpwG) {
+   if (length(i) == 1)
+     cat(get(i, KEGGPATHID2NAME), "\n")
+ }
```

```
Basal transcription factors
Retinol metabolism
Proteasome
Chondroitin / Heparan sulfate biosynthesis
```

These pathways might be connected to each other, or to other pathways, through genes that were not assayed.

We computed the degree distribution of the pathway graph and plotted a histogram in Figure 22.6. Pathways are the nodes in this graph, and so we see that some pathways have many edges to other pathways, and hence are quite central. It might be useful to use edge weights to indicate the number of shared genes, and this could then be used in coloring the edges or perhaps in thresholding them.

Other analyses might focus on finding shared components, for example finding out whether one pathway is wholly contained within another. We will need good layout algorithms for single pathways. We will also need layout mechanisms for joining together different pathways.

22.5.2 Relating expression data to pathways

We now consider a method for relating gene expression data to pathways. Other approaches have been considered, in particular by the GenMAPP project (Doniger et al., 2003), and some of our own work has been reported

in R News (Gentry et al., 2004). We consider the integrin-mediated cell-adhesion pathway, as represented at *KEGG*. The KEGG pathway label is `hsa04510` and the graphical representation from KEGG was shown in Figure 19.1. Users can either access the KEGG Web site directly, or they can use the *KEGGSOAP* package to obtain more information about this pathway. For any microarray experiment, Bioconductor meta-data packages can be used to find associations between probes and the genes involved in different KEGG pathways.

To obtain the pathway graph, you have several different options. You can construct one yourself, based on the available data and potentially expert biological advice, or you can make use of the information from the *cMAP* project, which is available in the *cMAP* package. For this particular pathway, we have already taken the information available in KEGG and used that to construct a graph representation of the pathway. The relevant data structures are constructed from two objects in the *graph* package. The object `IMCAGraph` is an instance of the *graphNEL* class, representing the pathway as a mathematical graph with named nodes and directed edges. The object `IMCAAttrs` is a list of plotting attributes for each node in the graph, such as the color.

We return to the ALL data and ask whether or not there are differences between the two groups (BCR/ABL and NEG) with respect to expression levels of genes in this pathway. We use the subset of the ALL data computed in Section 22.3. However, we do not carry out any gene selection, instead we consider the expression levels of the different genes in this pathway, and how those levels depend on phenotype (whether the samples are BCR/ABL or NEG).

Next, we obtain the mapping between the probes on the Affymetrix array and the genes in the pathway.

```
> hsa04510 <- hgu95av2PATH2PROBE$"04510"
> hsaLLs <- getLL(hsa04510, "hgu95av2")
```

There are 52 nodes in this pathway, and of these 45 represent genes. We find that there are 114 probesets for these genes on the HG-U95Av2 chip. There are many different ways to deal with the duplicate probesets, and here we take the simplistic approach of just selecting the first match. We note that an appropriate investigation of these data would involve a more detailed consideration of how to deal with multiple probes per gene.

In the next code chunk, we extract the LocusLink identifiers associated with each node in the graph and then for each of these take the first probeset that maps to it. We also check to see which of the genes in the pathway have no probes associated with them; these will have a value of `NA` in `whProbe`.

```
> LLs <- unlist(sapply(IMCAAttrs$LocusLink, function(x) x[1]))
> whProbe <- match(LLs, hsaLLs)
> probeNames <- names(hsaLLs)[whProbe]
```

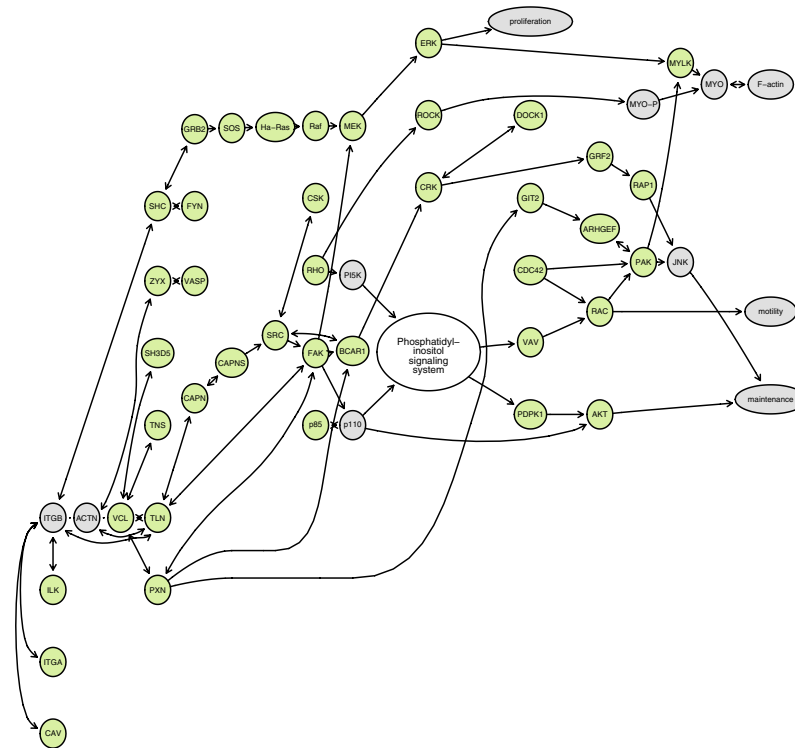


Figure 22.7. The integrin-mediated cell-adhesion network.

```
> names(probeNames) <- names(LLs)
> pN <- probeNames[!is.na(probeNames)]
```

We lay out the graph using `agopen`, as we want to render the same graph several times.

```
> IMCg <- agopen(IMCAGraph, "", attrs = IMCAAttrs$defAttrs,
+   nodeAttrs = IMCAAttrs$nodeAttrs, subGList = IMCAAttrs$subGList)
> plot(IMCg)
```

In Figure 22.7 we see the pathway laid out, with nodes that represent genes colored green. Now that we have found a set of probes that map to each gene in the pathway, we split the data into those with BCR/ABL and those that have no abnormalities and render the pathway, once for each group. For each group, we will plot a pie chart for each node. The pie chart will reflect a split, across the gene, of the samples for that gene. We will use splits of $(0, 6]$, for low, $(6, 8.5]$ for moderate and $(8.5, \infty]$ for high,

levels of expression. This visualization is different from one that colors nodes according to whether the genes are more highly expressed in one group than the other. It allows the reader to compare the distribution of expression, for each gene, between the two phenotypes.

Now that we have found the expression levels and computed the counts for each of the probes, we are ready to layout the graph and then render it, once for each phenotype we are interested in. The resulting plots are shown in Figure 22.8. Using pie charts for the nodes in the graph is easily done, and the procedure is documented in the **Rgraphviz** package. We note that due to the modular nature of the graph drawing procedures in **Rgraphviz**, virtually any R plot can be used for the nodes in a graph; see also Section 21.4.4. It is also easy to simply color the nodes according to which group has higher levels of expression, as is done by many others.

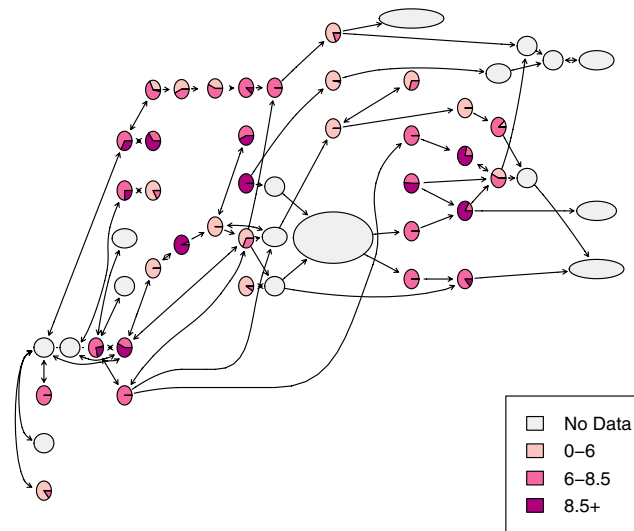
The graphs themselves are quite interesting. The similarity in distribution of expression levels, especially for those genes on the right half of the graph is remarkable. On the left side, we draw your attention to FYN, which has about 3/4 of the samples in the high range for BCR/ABL while for the NEG samples about 3/4 of the samples are moderate.

22.6 Concluding remarks

In this chapter, we have presented four case studies that made use of the tools that were introduced in the earlier chapters of this section. Our purpose was not to promulgate the examples themselves, but rather to demonstrate the flexibility of the software tools that are available and to emphasize that virtually any analysis can be undertaken, with a small amount of additional programming. You should only be limited by your ideas and the available data.

There are still many questions to answer, and much software needs to be written. We will need specialized graph algorithms to deal with the fact that many biological relationships are measured with error, and hence usual constructs and algorithms may fail or be unusable when false negative and false positive relationships exist. Visualizing graphs, as opposed to layout, is a difficult problem and one that is starting to get some attention. We hope that the tool kit of graph algorithms and methods described here, linked to the R statistical computing framework, will foster many new developments.

a) pie chart graph for BCR/ABL



b) pie chart graph for NEG

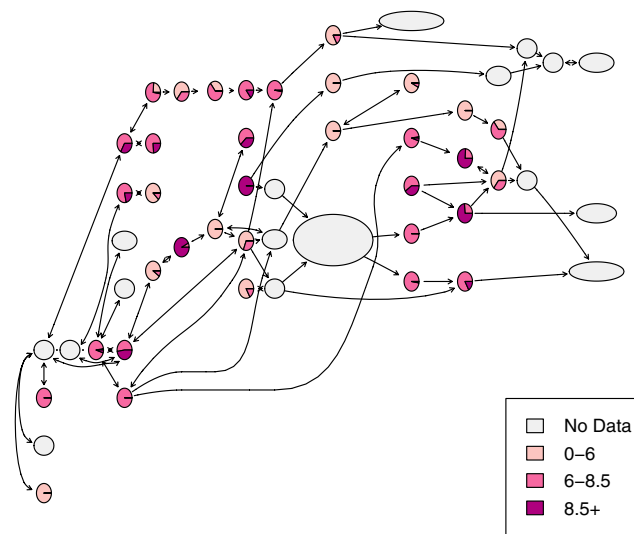


Figure 22.8. Pie chart graphs representing gene expression data for a) BCR/ABL samples, b) NEG samples.

Bioinformatics and Computational Biology Solutions Using R
and Bioconductor

Gentleman, R.; Carey, V.; Huber, W.; Irizarry, R.; Dudoit, S.
(Eds.)

2005, XIX, 473 p. 128 illus. in color., Hardcover

ISBN: 978-0-387-25146-2