Using graphs to relate expression data and protein-protein interaction data

R. Gentleman and D. Scholtens

September 8, 2007

Introduction

In Ge et al. (2001) the authors consider an interesting question. They assemble 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 reported in Ge et al. (2001). In particular we reduced the data to the 2885 genes that were common to all experiments. We have also represented most of the data in terms of graphs (nodes and edges) since that is the form we will use for most of our analyses.

The results reported in this document are based on version 0.9.9 of the *yeastExpData* data package.

0.1 Graphs

The cell-cycle data is stored in `ccyclered` and from this it is easy to create a *cluster*-graph. A cluster graph is simply a graph that is created from clustered data. In this graph all genes in the same cluster have edges between them and there are no between cluster edges.

```
> clusts = split(ccyclered[["Y.name"]], ccyclered[["Cluster"]])
> cg1 = new("clusterGraph", clusters = lapply(clusts, as.character))
> ccClust = connComp(cg1)
```

Next we are interested in looking for associations between the clusters in this graph (genes that show similar patterns of expression) and the literature curated set of protein-protein interactions.

We note that these are not really literature purported protein complexes (readers might want to use the output of a procedure like that in *apComplex* applied to TAP data to explore protein complex data). We also note that the data used to create these graphs is already somewhat out of date and that there is new data available from MIPS.

We have chosen to continue with these data since they align closely with the report in Ge et al. (2001).

```
> data(litG)
> ccLit = connectedComp(litG)
> cclens = sapply(ccLit, length)
> 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
```

```
> ccL2 = ccLit[cclens > 1]
> ccl2 = cclens[cclens > 1]
```

We see that there are most of the proteins do not have edges to others and that there are a few, rather large sets of connected proteins.

We can plot a few of those.

```
> sG1 = subGraph(ccL2[[5]], litG)
> sG2 = subGraph(ccL2[[1]], litG)
```

It is worth noting that the structure in Figures 1 and 2 is suggestive of collections of proteins that form cohesive subgroups that work to achieve particular objectives.

An open problem is the development of algorithms (and ultimately software and statistical models) capable of reliably identifying the constituent components. Among the important aspects of such a decomposition is the notion that some proteins will be involved in multiple complexes.

1 Testing Associations

The hypothesis presented in **?** was to investigate a potential correlation between expression clusters and interaction clusters. The devised a strategy called the *transcriptome-interactome correlation mapping strategy* to assess the level of dependence. Here we reformulate their ideas (a more extensive discussion with some extensions is provided in Balasubraminian et al. (2004)).

The basic idea is to represent the different sets of interactions (relationships) in terms of graphs. Each graph is defined on the same set of nodes (the genes/proteins that are common to all experiments) and the specific set of relationships are represented as edges in the appropriate graph. Above we created one graph where the edges represent known (literature based) interactions, `litG` and the second is based on clustered gene expression data, `cg1`.

We can now determine how many edges are in common between the two graphs by simply computing their intersection.

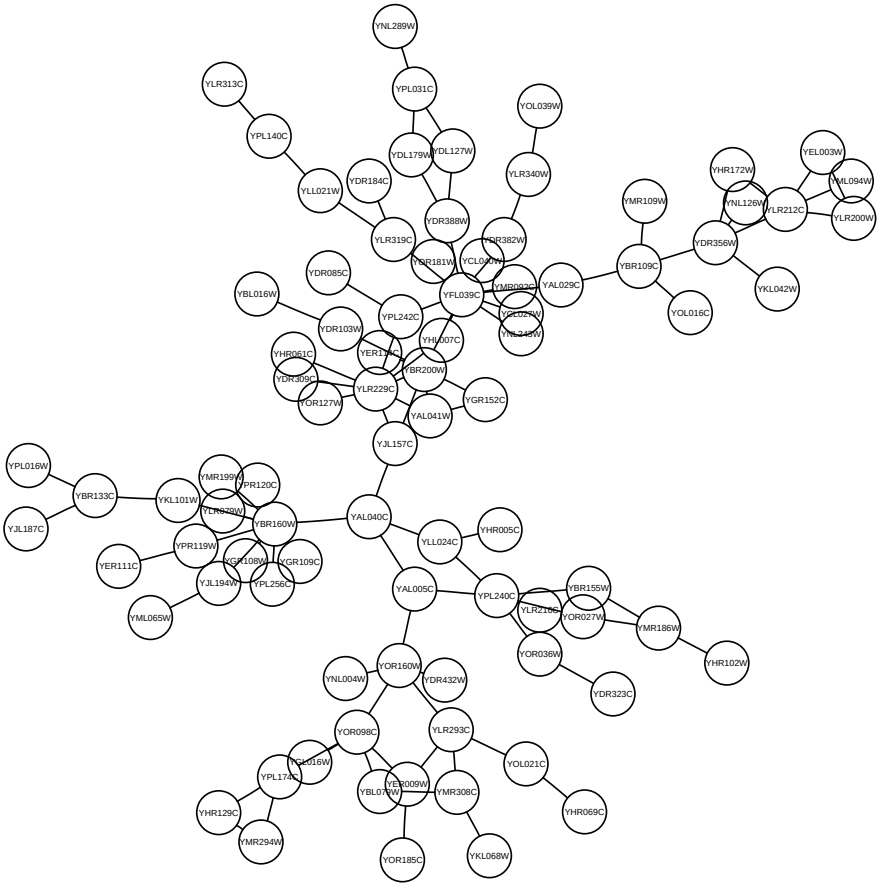

Figure 1: One of the larger PPI graphs

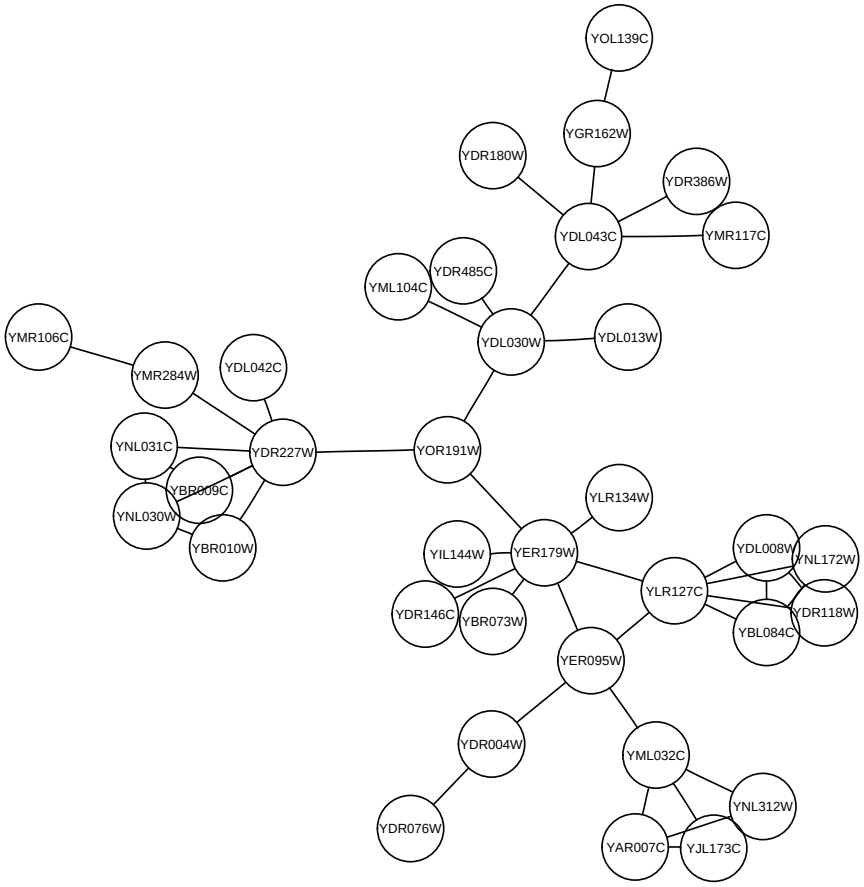

Figure 2: Another of the larger PPI connected components.

```
> commonG = intersection(litG, cg1)
> commonG

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

And we can see that there are 42 edges in common. Since the literature graph has 315 and the gene expression graph has 156205 we might wonder whether 42 is remarkably large or remarkably small.

One way to test this assumption is to generate an appropriate null distribution and to compare the observed value (42) to the values from this distribution. While there are some reasons to consider a random edge model (as proposed by Erdos and Renyi) there are more compelling reasons to condition on the structure in the graph and to use a node label permutation distribution. This is demonstrated below. Note that this does take quite some time to run though.

```
> ePerm = function(g1, g2, B = 500) {
+     ans = rep(NA, B)
+     n1 = nodes(g1)
+     for (i in 1:B) {
+         nodes(g1) = sample(n1)
+         ans[i] = numEdges(intersection(g1, g2))
+     }
+     return(ans)
+ }
> set.seed(123)
> data(nPdist)
> max(nPdist)

[1] 23
```

2 Data Analysis

Now that we have satisfied our testing curiosity we might want to start to carry out a little exploratory data analysis. There are clearly some questions that are of interest. They include the following:

- 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?
- Is the expression behavior of genes that are involved in multiple literature clusters (or at least that we suspect of being so involved) different from that of genes that are involved in only one cluster?

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) are very simple to put in graph theoretic terms.

References

R. Balasubraminian, T. LaFramboise, D. Scholtens, and R. Gentleman. Integrating disparate sources of protein–protein interaction data. Technical report, Bioconductor, 2004.

R.C. Cho et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2:65–73, 1998.

H. Ge, Z. Liu, G. M. Church, and M. Vidal. Correlation between transcriptome and interactome mapping data from saccharomyces cerevisiae. *Nature Genetics*, 29:482–486, 2001.