Exploring protein interaction data using *ppiStats*

Denise Scholtens and Tony Chiang

Introduction

ppiStats is a package that provides tools for the description and analysis of protein interaction data represented in node-and-edge graph form. The package is specifically designed to address data generated by bait-prey technologies (e.g. yeast-two-hybrid (Y2H) and affinity purification-mass spectrometry (AP-MS)) in which directed edges extend from bait proteins toward detected prey proteins, with the nature of the detected interaction depending on the technology.

This vignette demonstrates the main functionality of the *ppiStats* package including: 1) summarizing observed data, 2) identifying systematically biased proteins in a data set, and 3) estimating stochastic error probabilities. In order to demonstrate *ppiStats*, we will make use of the protein interaction data sets available in *ppiData*.

```
> library("ppiStats")
> library("ppiData")
```

In this vignette, we will use to AP-MS data sets reported by Gavin et al. (2002) (Gavin02) and Gavin et al. (2006) (Gavin06) for *Saccharomyces cerevisiae*. AP-MS technology is a bait-prey system designed to detect complex co-membership on behalf of proteins.

Most of the functions available in *ppiStats* require objects that extend the class `graph`, defined in the *graph* package. For example, `Gavin2002BPGraph` and `Gavin2006BPGraph` are `graphNEL` objects available in *ppiData*. For assistance in converting raw data files into the appropriate object instance, see the help pages for the functions `bpMatrix` and `genBPGraph` in the *ppiStats* package. The *graph* and *RpsiXML* packages also contain helpful functionality.

```
> data(Gavin2002BPGraph)
> data(Gavin2006BPGraph)
> GavinGraphs = list(Gavin02=Gavin2002BPGraph,Gavin06=Gavin2006BPGraph)
> GavinGraphs
```

```
$Gavin02
A graphNEL graph with directed edges
Number of Nodes = 1362
Number of Edges = 3418

$Gavin06
A graphNEL graph with directed edges
Number of Nodes = 2551
Number of Edges = 19105
```

Summarizing observed data

While AP-MS data can in concept assay all possible complex co-membership interactions in a cell, in fact only a certain portion of those interactions are assayed in an experiment. The sampled portion of the 'interactome' depends on which proteins are selected as baits, the subset of those baits that are amenable to the technology, and the set of proteins in the cell that are available for detection as prey under the conditions of the experiment. Chiang et al. (2007) present the notion of 'viability' to summarize these ideas, defining 'viable baits' (VBs) as baits that detect at least one prey and 'viable prey' (VP) as proteins that are detected by at least one bait. It is possible for a protein to be both a viable bait and a viable prey (VBP). These definitions are important since only the edges that extend from VB nodes to VP nodes are deemed 'tested' in the analyses presented here. All other potential interactions (e.g. an edge between two nodes that are VP but not VBP) are treated as untested.

The function `createSummaryTables` lists several summary statistics about the viable protein population in an experiment. In addition to describing the viable protein population, the total number of directed interactions (TI) is reported.

```
> summaryTables = createSummaryTables(GavinGraphs)
> summaryTables
```

```
          VB   VP VBP VBP/VB VP/VB    TI TI/VB
Gavin02  455 1178 271   0.60  2.59  3418  7.51
Gavin06 1752 1790 991   0.57  1.02 19105 10.90
```

When examining the VB and VP populations, it can be helpful to compare the number of represented proteins to the total population in the cell, especially if the experiment is intended to be genome-wide. Figure 1 demonstrates VB and VP coverage for the Gavin02 and Gavin06 experiments, relative to the entire 6000+ genes in yeast.

```
> viabilityCharts(GavinGraphs)
```

Rather than simply enumerating the viable protein population, the function `idViableProteins` identifies the specific VB, VP, and VBP proteins represented in a data set.

```
> GavinViableProteins = lapply(GavinGraphs,FUN=idViableProteins)
> sapply(GavinViableProteins,FUN=function(x) lapply(x,FUN=length))
```

```
    Gavin02 Gavin06
VB  455     1752
VP  1178    1790
VBP 271     991
```

```
> Gavin06VBPproteins =  GavinViableProteins[["Gavin06"]]$VBP
> Gavin06VBPproteins[1:4]
```

```
[1] "YLR274W" "YDR460W" "YGL240W" "YPR189W"
```

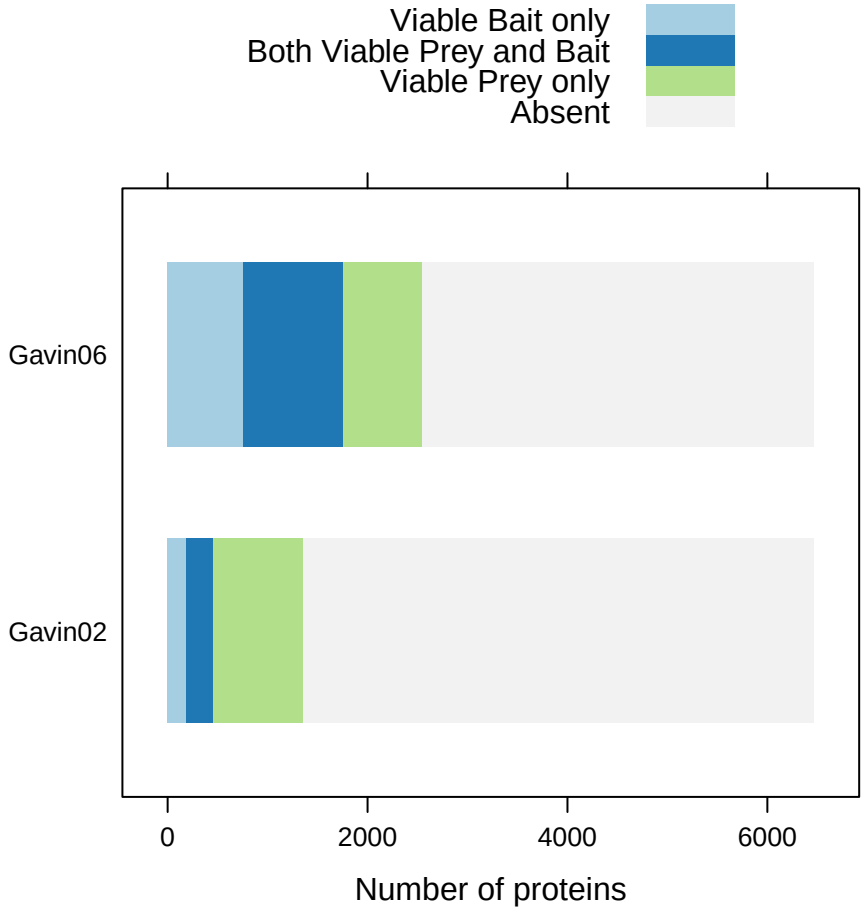

Figure 1: Enumerations of the VB, VP, and VBP populations for the Gavin02 and Gavin06 datasets in comparison with the entire set of 6000+ genes in yeast.

Identifying systematically biased proteins

In some bait-prey experiments, certain proteins are prone to systematic bias. Specifically, in the VBP population, some proteins have a high imblance between in- and out-degree. In-degree in the VBP population represents the number of times a VBP node is detected as prey by another VBP protein. Out-degree represents the number of times that VBP node detects other prey. When in-degree is much higher than out-degree, or vice versa, it may be an indication that the node is subject to more than simply stochastic measurement error in line with the sensitivity and specificity of the technology. The `inOutScatterCharts` function compares in- and out-degree for the VBP nodes and provides boundaries for determining which nodes may be particularly prone to systematic error according to the Binomial model of Chiang et al. (2007).

```
> inOutScatterCharts(GavinGraphs)
```

It appears that the Gavin06 dataset contains more proteins that are subject to systematic error than the Gavin02 dataset. We can identify these proteints using the function `idSystematic`, remove them from the graph, and then proceed with estimation of stochastic error probabilities using the remaining VBP nodes.

```
> systematicVBPsGavin06 = idSystematic(Gavin2006BPGraph,Gavin06VBPproteins,bpGraph=TRUE)
> unbiasedGavin06VBPs = setdiff(Gavin06VBPproteins,systematicVBPsGavin06)
> Gavin06VBPsubgraph = subGraph(unbiasedGavin06VBPs,Gavin2006BPGraph)
> Gavin06VBPsubgraph

A graphNEL graph with directed edges
Number of Nodes = 678
Number of Edges = 1622
```

Estimating stochastic error probabilities

Each edge between the subgraph induced by the VBP nodes is tested twice, once in each direction. With perfectly sensitive and specific technology, each edge would either be observed twice or not observed twice. The presence of unreciprocated edges indicates the presence of measurement error - either a false positive (FP) or a false negative (FN) observation.

The function `estimateCCMErrorRates` estimates stochastic error probabilities, specifically the false positive (p_{FP}) and false negative (p_{FN}) probabilities, using the method of moments approach described by Chiang et al. (2007). `estimateCCMErrorRates` calls `estErrProbMethodOfMoments` to compute a solution manifold and then selects a single pair of estimates for p_{FP} and p_{FN} using a gold standard approach as described in Scholtens et al. (2008). In the example below, we use the `ScISIC` dataset from the package *ScISI* as the gold standard. This is a set of manurally curated complexes culled from GO and MIPS and serves to define a set of edges that should be detected by AP-MS technology.

```
> data(ScISIC)
> Gavin06m = as(Gavin06VBPsubgraph,"matrix")
```

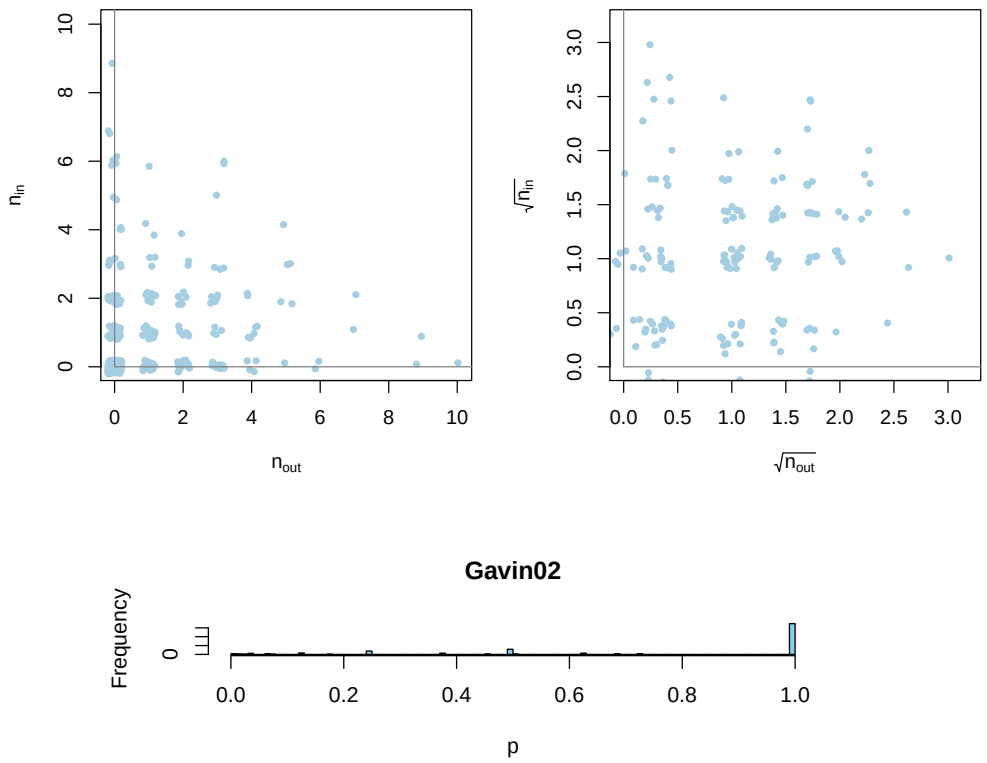

Figure 2: Scatter plots corresponding to the in- and out-degree (and the square root of each) for the VBP nodes in Gavin02. The points within the diagonal bands represent those proteins for which there is not enough evidence to reject the null hypothesis that the protein might be affected by some systematic bias while the points outside the bands are rejected at the 0.01 p-value. The bottom plot describes the distribution of the p-values for each data point. While the majority of proteins are not rejected, we can see that a substantial number proteins are. Those proteins which are rejected should not be analyzed using the standard statistical methods to model stochastic variability nor should they be used to estimate the underlying features within the data.

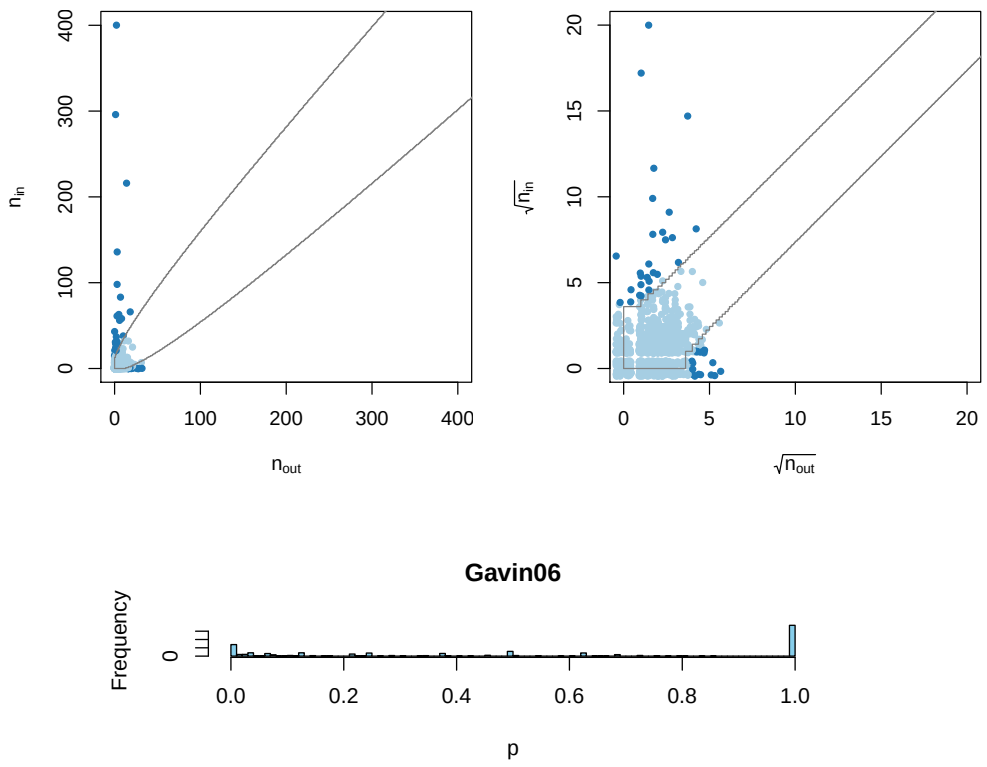

Figure 3: The same plots as described in Figure 2 presented here for the Gavin06 data.

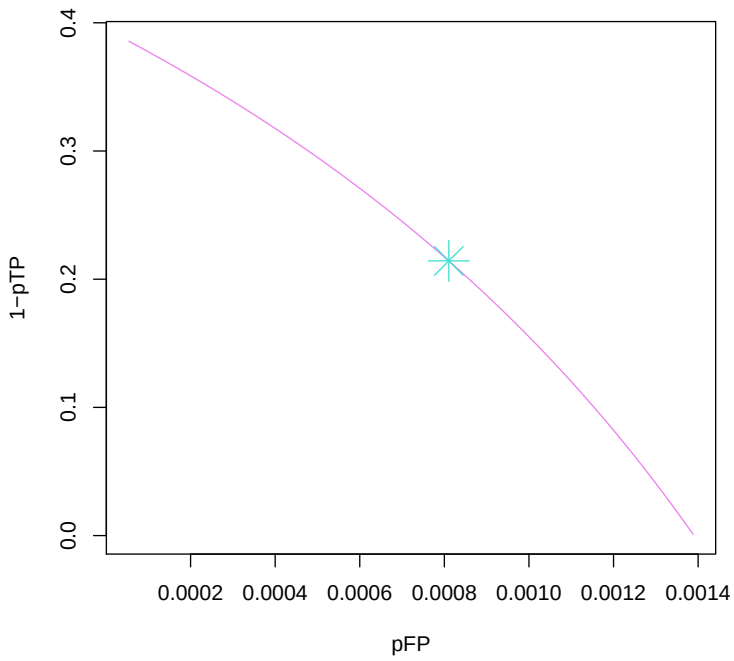

Figure 4: A plot of the solution manifold for estimates of p_{FP} and p_{FN} with the star denoting the gold standard point estimate.

```
> errorProbabilities = estimateCCMErrorRates(Gavin06m,ScISIC)
> pTPestimate = errorProbabilities$globalpTP
> pFPestimate =errorProbabilities$globalpFP
> solutionManifold = errorProbabilities$probPairs
> pTPestimate

[1] 0.7857143

> pFPestimate

[1] 0.0008106258
```

Figure 4 plots the calculated solution manifold and highlights the gold standard estimates.

```
> plot(solutionManifold[,"pFPs"],solutionManifold[,"pFNs"],type="l",col="violet",xlab="pFP",y
> points(pFPestimate,1-pTPestimate,cex=3,pch=8,col="turquoise")
```

Estimates of p_{FP} and p_{FN} are helpful in using statistical models to estimate features and parameters for that dataset. One popular statistic is node degree, def. the number of interacting partners for a protein. Nodes in an experimental AP-MS graph have an unknown true degree. An AP-MS experiment yields an observed number of reciprocated edges, as well as unreciprocated in- and out-edges. A few examples are listed below.

```
> Gavin06VBPobserved = calcInOutDegStats(Gavin06VBPsubgraph)
> Gavin06VBPobserved$unrecipInDegree[5:10]

YJR041C YLR357W YGL128C YER022W YDR167W YMR089C
      0       4       0       2       1       0

> Gavin06VBPobserved$unrecipOutDegree[5:10]

YJR041C YLR357W YGL128C YER022W YDR167W YMR089C
      0       1       1       0       5       1

> Gavin06VBPobserved$recipIn[5:10]

YJR041C YLR357W YGL128C YER022W YDR167W YMR089C
      1       6       4       2       1       1
```

The function `degreeEstimates` estimates degree for VBP nodes according to the parametric model described by Scholtens et al. (2008) using the observed data and the stochastic error probability estimates.

```
> Gavin06VBPdegree = degreeEstimates(Gavin06m,pTPestimate,pFPestimate)
> Gavin06VBPdegree[5:10]

YJR041C YLR357W YGL128C YER022W YDR167W YMR089C
      1      10       5       3       5       1

>
```

Conclusion

The *ppiStats* package is useful for describing data in graphs and is aimed at bait-prey systems. It is useful for summarizing graph data, identifying systematically biased baits and estimating stochastic error probabilities in preparation for estimation of relevant graph features.

```
> sessionInfo()

R version 3.3.0 (2016-05-03)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.4 LTS

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:
[1] parallel  stats4    stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] ppiStats_1.38.0      ppiData_0.9.0        lattice_0.20-33
 [4] ScISI_1.44.0         apComplex_2.38.0     RpsiXML_2.14.0
 [7] hypergraph_1.44.0    RBGL_1.48.0          graph_1.50.0
[10] annotate_1.50.0      XML_3.98-1.4         GO.db_3.3.0
[13] AnnotationDbi_1.34.0 IRanges_2.6.0        S4Vectors_0.10.0
[16] Biobase_2.32.0       BiocGenerics_0.18.0

loaded via a namespace (and not attached):
 [1] splines_3.3.0       xtable_1.8-2        Category_2.38.0
 [4] tools_3.3.0         grid_3.3.0          DBI_0.4
 [7] genefilter_1.54.0   survival_2.39-2     GSEABase_1.34.0
[10] Matrix_1.2-6        org.Sc.sgd.db_3.3.0 RColorBrewer_1.1-2
[13] Rgraphviz_2.16.0    RSQLite_1.0.0
```

References

A. C. Gavin et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

A. C. Gavin et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440:631–636, 2006.

T. Chiang et al. Coverage and error models of protein-protein interaction data by directed graph analysis. *Genome Biology*, 8:R186, 2007.

D. Scholtens et al. Estimating node degree in bait-prey graphs. *Bioinformatics*, 24:218-224, 2008.