

# Supplementary Material for *Creating an in Silico Interactome*

October 3, 2006

## 1 TODO

We need to create a website that has links to the paper, this supplementary material and the html that we can generate for the complex estimates, as well as other materials for the binary interaction data

## 2 Data Processing

We describe our methods used to assemble two different sources of data. To assemble the protein complex interactome we made use of publicly available curated complex data and publicly available experimental data. For the binary interaction data we made use of data from IntAct and from a methodology designed to make predictions of binary interactions.

### 2.1 Processing Protein Complex Data

We first discuss the creation, or estimation of  $I_C$ . Our basic strategy is to assemble a number of software tools, together with available data, from databases or experimental results. We process the data from each database or experiment separately and then sequentially integrate the different protein complex estimates into an estimate. Some investigators may want to restrict their attention to well-known and documented complexes while others will be interested in exploring likely complex co-memberships. For this reason we construct two estimates of  $I_C$  in yeast. One is based only on complexes reported in GO and MIPS (??), while the other extends this first estimate to include data from high through-put co-precipitation experiments.

We remark that the complexes documented within GO and MIP are based on many different technologies such as small scale protein complex verification experiments. Many of these protein complexes have been exhaustively verified.

The choice of input data sources and the manner in which they are processed is subjective and different investigators are likely to make different decisions. We report the sources that we used and the decisions made; others can easily make use of alternative or additional sources. We make use of two rather distinct data sources, one is the set of published and annotated protein complex data that can be obtained from the Gene Ontology (GO), (??), ([www.geneontology.org](http://www.geneontology.org)) and the Munich Information Center for Protein Sequences (MIPS). Additionally we will make use of protein complex estimates from three publicly available wet-lab experiments (???). Rather than make use of these data directly, we first apply the technology of (??) since, as

demonstrated in those references, the resultant data are more consistent with known biological complexes.

In searching for and parsing through information from the two online data repositories, GO and MIPS, we used somewhat indirect methods. For the GO data source, we made use of the Bioconductor metadata package `GO` which contains the pertinent information. For MIPS we downloaded three files: *complexcat.scheme* details the hierarchy classification structure of the MIPS protein complexes; *evidencecat.scheme* contains the evidence code definitions; and *complexcat\_data\_14112005* contains the protein annotations and evidence codes for each protein complex. We the name of the last file has a suffix which is an eight digit number indicating the day/month/year which this file is updated (there is a bi-annual update for this file).

For each data repository, the textual terms used to describe its contents are parsed for specific key words. Using regular expression searches with approximate matching we have searched for all the terms that contain one of the three default character strings listed:

1. the exact word **complex**;
2. one or more words that end with the suffix **ase**;
3. one or more words that end with the suffix **some**.

In particular we use the following three regular expression, **complex**, `\\Base\\b`, and `\\Bsome\\b` in our search. These narrow the pool of return values yet include terms such as **DNA-directed Polymerase II holoenzyme** or **repairosome**.

The software allows users to modify the search criteria to add new terms or to remove any of those default values listed above. Terms selected correspond to protein complexes and these are then collected and the corresponding incidence matrix is computed. We note that both the GO and the MIPS repositories annotate proteins by their systematic gene names, and so the incidence matrix computed has the rows are indexed by the systematic gene names and the columns by the protein complex identification codes (unique to each repository).

We remark that even with the limited return values, the restrictions imposed by the regular expression searches will still contain elements which we do not consider protein complexes. Care must be taken to inspect those obtained elements from each repository.

Elements such as arginase, and CTP synthetase are considered by some to be protein complexes even though they contain one unique protein. Because the data repositories do not include information such as multiplicity, we have decided to remove any protein complexes of size one. In addition, some elements such as (FIXME - look up some examples) are the names of polypeptides and not protein complexes, so they too must be removed from the selected data.

In addition to those exceptions we have just mentioned, the regular expression parsing methodology also obtains those protein complexes derived from high through-put AP-MS experiments and estimated by each respective experimental investigator. Presently, only the MIPS data repository annotates proteins by these high through-put experiments. In constructing the protein complex interactome, we have elected not to retain those complexes whose constituent members are annotated in this manner for both the highly verified  $I_c$  and the the extended  $I_c$ .

There are several reasons why we dis-allow the inclusion of these protein complexes. One is the lack of uniformity by which these complexes were estimated; it appears that the MIPS

repository contains those estimates reported by the investigators of each respective experiment themselves, and this lack of uniformity creates problems in the analysis and validation to the interactome. Another reason is the estimation algorithms themselves; previous analysis (??) of the resulting protein complexes annotated by high through-put experiments shows that in general, the protein complex size tend to be over-estimated, i.e. the overall size distribution is higher than that of the set of well known protein complexes. Lastly, high through-put data is known to have a larger incidence of error than that of small scale systematic experiments, and therefore, caution needs to be taken when dealing with these particular putative protein complexes.

Realizing that there are elements extracted from the data-bases which need to be dis-allowed, we address the systematic and computational methods in excluding individual proteins as well as entire protein complexes. Both GO and MIPS provide additional information on the curation and provenance of the data that they report. This information comes in the form of evidence codes, which are used by both resource, although the codes themselves are unique to the different resources. Users can specify a list of evidence codes and dis-allow those proteins whose annotations are found in the supplied list. Table ?? lists both the GO and MIPS evidence codes we have used to reject protein membership in the various complexes we have extracted:

For the GO data repository, only four evidence codes were used to reject protein membership. Any protein having only these four evidence codes to support its annotation is considered suspect, and we have chosen to remove them from the data. The MIPS evidence codes that are selected is divided into two groups: 1. those that are prefixed by 901 and 2. those that are prefixed by 902. Similar to the chosen GO evidence codes, the MIPS evidence codes beginning with 901 refer to non-experimental methods of inference. Indeed, some of the MIPS codes are similar enough to the GO codes that the curators of MIPS have referenced some GO evidence codes when pertinent. The MIPS evidence codes beginning with 902, however, refer to experimental methods of inference and annotation. Because we have chosen to reject any estimates based on high through-put AP-MS experiments, we have selected any MIPS evidence code that corresponds to any high through-put technology.

Unfortunately, the use of evidence codes is not enough. Even with the selected evidence codes listed above, a large number of high through-put putative protein complexes are still retained. In the addition to the evidence codes, we also make use of the MIPS hierarchy class system in which proteins complexes are grouped. The MIPS data repository groups protein complexes by their functionality, and because the putative, high through put protein complexes have yet to be validated nor been studied, they are grouped into one separate category. This category has a unique prefix code (550), and so we can simply reject any protein complex that has a prefix code 550.

As we have mentioned earlier, the functions of ScISI do have limitations, and so unwanted protein clusters will inevitably make their way into the data-sets we select. To this end, a non-trivial part of the construction process involves manual curation of the protein complexes obtained from the sources.

GO Evidence Codes	GO Evidence Code Definitions
IEA	Inferred from Electronic Annotation
NAS	Non-traceable Author Statement
ND	No Biological Data Available
NR	Not Recorded
MIPS Evidence Codes	MIPS Evidence Code Definitions
901.01.03	Overview information (TAS/NAS)
901.01.03.01	Review
901.01.03.02	Text-book
901.01.04	Personal communication (TAS/NAS)
901.01.04.01	Homepage (Web)
901.01.04.02	E-mail
901.01.05	Closed information (NAS)
901.01.05.01	Institution
901.01.05.02	Private
902.01.01.02.01.01	Co-immunoprecipitation
902.01.01.02.01.01.01	Co-immunoprecipitation, native
902.01.01.02.01.01.02	Co-immunoprecipitation, epitope tag
902.01.01.02.01.02	Affinity chromatography
902.01.01.02.01.02.01	Affinity chromatography, native
902.01.01.02.01.02.02	Affinity chromatography, affinity-tag
902.01.01.04.01	Mass spectrometry (MS)
902.01.01.04.01.01	MS with in-line two-dimensional liquid chromatography (MudPIT)
902.01.01.04.01.02	MS with liquid chromatography coupled to tandem mass spectrometry (LC-MS/MS)
902.01.01.04.01.03	MS with matrix-assisted laser desorption/ionization time-of-light (MALDI TOF MS)
902.01.01.04.02	Immuno detection
901.01.09.02	High throughput experiment

Table 1: This table details the GO evidence codes as well as the MIPS evidence codes. Presently, only the MIPS data repository has annotated genes/proteins via high through-put experimental data; these are the evidence codes prefixed with by 902.

### 2.1.1 AP-MS Experimental Data

Although we have excluded the high through-put AP-MS protein complex estimates from the MIPs repository, we do not reject them universally. We obtained the reported bait to prey results from three high through-put AP-MS experiments (???). These data-sets were then assembled and processed using the technologies reported in (??), in particular using the software packages **apComplex** available from the Bioconductor project.

The output of the analysis are putative protein complexes. They come in several different varieties, and we make use exclusively of the multi-bait multi-hit (MBMH) complex estimates.

In using **apComplex** to process the raw, unfiltered AP-MS experimental data, we streamline the estimation process so that high through-put annotated protein complexes depends on a single, unique methodology. This uniformity is beneficial in many ways. It allows for analysis on the putative protein complexes without the use of other statistical mechanisms such as boot-strapping adjustments. The technology of **apComplex** allows the end user to adjust the levels of sensitivity and of specificity, so that different users can generate different estimates based on various error models. Lastly, the statistical model by which **apComplex** processes the data gives a sound and robust mechanism to manage experimental errors (both false positive and false negative observations).

The putative estimates resulting from the **apComplex** processing has shown to be more consistent with known biology. (??) have shown that the MBME putative protein complexes have significant overlap when compared with well known protein complexes and have predicted the exact protein composition of **Arp 2/3** and of **Cleavage Factor IA** complexes from the raw experimental data of (?). In addition, the size distribution of these MBME complexes corresponds more with the distribution of the size of known complexes indicating that these estimates have a better mechanism to filter out false positive protein complex affiliations.

The raw experimental data from (???) have already been processed, and each MBMH estimates are stored as an bipartite graph incidence matrix within the R package **apComplex**. **apComplex** also uses systematic gene names to annotate proteins for complex membership, so that the incidence matrices stored within the package also have the rows indexed by these systematic gene names. Because **apComplex** generates the putative protein complexes each time data is processed, it does not generate any meaningful names for each complex. Rather, **SciSI** creates an ad hoc name for each protein complex based on the experiment by which the complex is derived.

Having obtained the protein complex estimates from **apComplex**, we proceed to the actual construction of the protein complex interactome.

## 3 Binary Interactions

The protein binary interaction data is almost exclusively based on yeast 2-hybrid experimental data, and as a result, we have imposed extra structure to the protein binary interactome by requiring it to contain the bait to prey relationship derived from the Y2H experiments. Imposition of this condition translates the undirected protein-protein interaction graph to a directed bait to prey interaction graph. This relationship is still binary, though non-symmetric, and this will be the model by which we will base  $I_b$ . This additional structure produces an interactome that more accurately reflects the current state of biological knowledge, since the

majority of protein-protein interaction has yet to be tested in a symmetric manner.

### 3.0.2 Obtaining Binary Interaction Information from the IntAct Repository

In beginning the construction of the protein binary interactome from the **IntAct** database, we again parsed for specific key terms. Looking in the molecular interaction (MI) category of the Proteomics Standards Initiative (PSI) (cite PSI), we searched for the three related character strings:

1. y2h;
2. yeast two hybrid;
3. 2-hybrid.

Of the return values from this textual search, we made use of four MI identification codes:

MI Code	Description
MI:0018	Two hybrid;
MI:0397	Two hybrid array;
MI:0398	Two hybrid pooling approach;
MI:0399	Two hybrid fragment pooling approach.

Table 2: The MI codes corresponds to the type of experimentation conducted.

We can use these MI codes to parse one sub-directory of the **IntAct** repository accounting for the various wet-lab experiment information. We found forty-one Intact Acension Codes (AC) whose interaction detection slots contained one of the four MI codes. The forty-one IntAct AC's represented the various wet-lab experiments conducted using the yeast 2-hybrid technology or some derivation thereof.

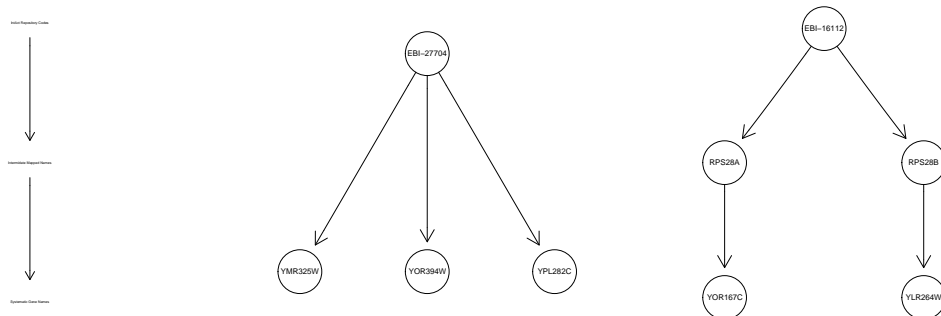
For each 2-hybrid experiment, we first generated a list of the proteins hybridized with a binding domain (these constitute the bait proteins), and for each of these bait proteins, we generated a list of proteins hybridized with an activation domain (the prey proteins) found to experimentally interact with the respective bait protein. In essence, we have created one representation of the bait to prey binary interaction graph.

Once we have obtained the bait to prey data from **IntAct**, we need to modify this list because all the proteins are denoted by its correspond IntAct accession code. We want to translate these codes into the systematic gene names so that we have a uniform system of nomenclature.

## 3.1 Obtaining Membrane Bound Binary Interaction Information

The membranous bound protein interaction action data is given by (cite Miller et al) in the supplementary section of their manuscript. Again, we keep the bait to prey interaction relationships and create the list of reflecting this association.

Unlike the protein interactions obtained from **IntAct**, (cite Miller) have given definitive SGN's for all interacting membranous proteins, and we make use of those denoted names.



(a) Mappings from IntAct to Systematic Gene Name  
(b) Mapping from IntAct directly to systematic genes  
(c) Mapping from IntAct through an intermediary to systematic genes

Figure 1: The mappings from IntAct to the systematic genes takes a tree like structure. The first diagram shows what must happen to each IntAct ID in the mapping process. The middle diagram shows that sometimes the intermediary mappings are trivial and the process is direct (though the mapping may still be one to many). The diagram on the right shows the mapping to an intermediate. The number of intermediary mappings tend to be either zero or one from the Y2H data, but might be more than one for other data sets (there might even be no mapping available so that we keep only the IntAct ID). All the mappings are, in general, one to one with a few that are one to many.

### 3.1.1 Translating protein names

While each of the data repositories hosting the complex membership data used the systematic gene names to identify the constituent protein members, **IntAct** uses its own unique nomenclature system to identify proteins involved in the Y2H experimental data. With this additional nomenclature, there is a need to translate from the IntAct codes to the systematic gene names for yeast.

IntAct faces the same problems of identifiability as we do. They have streamlined this issue by inventing a new nomenclature which is consistent in their repository. Each protein is given a particular IntAct accession code, and for each of these protein accession code, the protein names (common, systematic, or any other defined nomenclature) is annotated. Therefore, some IntAct protein accession codes might map directly to the SGNs, while others might map to other nomenclatures that need to be further mapped to obtain SGNs (though none of these maps need necessarily be one to one). Some IntAct protein codes cannot be mapped to any SGN, and for these proteins, the IntAct codes is retained.

Because the mappings from an Intact protein code need not be one to one, we need to chose which specific systematic gene to use. The mapping from the IntAct protein code to the systematic gene names in the software induces a unique and stable path structure, a rooted, directed tree (figure 1(b) and figure 1(c)). The choice will be made to take the SGN that is the destination of the left-most path in this tree: in figure 1(b), EBI-27704 will map directly to YMR325W, while in figure 1(c), EBI-16112 will map to RPS28A and then to YOR167C. Again this choice is arbitrary, and other users can define which path to take when conducting the

mappings. Defining this choice, we have now finished the construction of a representation of the protein binary interactome on each experimental data set, and we continue with generating the cumulative protein binary interactome  $I_b$ .

### 3.2 Binary Protein Interaction data

The identifiability problem plays an important role in  $I_b$ . Mapping from the IntAct accession codes identifies 1914 bait proteins and 3826 prey proteins by their respective SGNs. These un-defined mappings percolate through  $I_b$ , but we have left them in the interactome because their inclusion has no effect on our computational analysis (we remark that removal is trivial and the choice is left to the user).

### 3.3 Summary Statistics

As stated above, our definition of a multiprotein complex presumes that every member is physically linked to at least one other member of the complex. Stated in graph theoretic terms we assume that the graph of a complex where nodes represent proteins and edges represent physical interactions is connected.

To develop appropriate statistical methods we describe the same setting in a slightly more abstract notation. For a given complex,  $C$ , say, consisting of  $n$  proteins the graph that represents this complex must be connected, by our definition of a complex. Hence, there are between  $n - 1$  and  $n(n - 1)/2$  edges. We can conceptualize this in the following way. Consider an urn that contains balls, each ball represents on possible edge, or binary interaction. For  $C$  there are  $n(n - 1)/2$  possible edges and hence there are that many balls in the urn. Some of the balls represent edges that truly exist in  $C$ , and these are colored black, the remainder of the balls are white.

We will, in some cases, use the notation  $G_C = (V_C, E_C)$ , to denote the graph induced by  $C$ . We note that in this graph the edges are not directed and there are no self-loops. That is, we assume that no protein in  $C$  has an edge to itself, in part this makes the math simpler and in part it reflects the technology. While AP-MS can determine the constituent elements of complex it cannot directly ascertain their multiplicity.

The first problem we address is estimating the number of black balls in the urn. We label this unknown quantity  $X$ . The basis for this estimation is the sampling of some nodes of  $C$  and determining which other members of  $C$  they are observed to be connected to. We will presume that the nodes that are sampled are a simple random sample from the population  $V_C$ , but note that this is not always the case in real experiments. Let  $k$  denote the number of nodes sampled and let  $n = |V_C|$ , when needed we will also use  $n_C$ . Further, let  $x$  denote the number of distinct edges found based on the sampling of  $k$  nodes.

To determine how many edges were tested we note that the first node is compared to the remaining  $n - 1$ , the second to the remaining  $n - 2$  and so on. So, the number of edges tested is  $[(n - 1) + (n - 2) + \dots + (n - k)]$ . If we sample all nodes then the sequence is,

$$\sum_{j=1}^{n-1} (n - j) = \sum_{j=1}^{n-1} j = \frac{(n - 1)n}{2},$$



where we have made use of the well known relationship regarding the sum of the first  $n$  integers. And hence, sampling all nodes does in fact result in the inspection of all edges.

A widely used estimate of  $X$  arises from equating the observed proportion of edges in the sample, with the unknown proportion in the population. That is,

$$\begin{aligned}\frac{\hat{X}}{n(n-1)/2} &= \frac{x}{[(n-1) + (n-2) + \dots + (n-k)]} \\ \hat{X} &= \frac{x(n)(n-1)}{2[(n-1) + (n-2) + \dots + (n-k)]}\end{aligned}$$

A second estimator can be derived from the following argument. If the nodes that are sampled represent a random sample from the population of nodes,  $V_C$ , then the observed mean degree is an unbiased estimate of the population mean degree. The population mean degree times  $n$  divided by 2 is an estimate of the number of edges. For each sampled node we let  $d_i$  denote its observed degree. So, we have

$$\tilde{X} = \frac{n \sum_{i=1}^k d_i}{2k}.$$

The two estimators are quite similar, and sometimes identical.

At this point we have merely estimated the number of edges in  $G_C$ , from that we make the next step of assessing whether or not  $G_C$  is connected. There are a number of factors that influence that determination and we provide simulation examples and evidence of how these different factors can be used.

First, the larger  $X$  the more likely the graph is connected. Next, we can examine the number of unique nodes that were selected either as part of the  $k$  nodes used to probe the graph or as the other ends of the detected edges. The more nodes detected this way, the higher the probability that the graph is connected. And finally, we can take the observed subgraph, induced by our sampling, and determine how many more edges are needed to obtain a connected graph; the fewer the more likely it is that the true graph is connected.

We also explore a simulation approach to assessing whether the underlying graph is connected. Given the observed edges for the  $k$  query nodes and the estimated number of edges in the graph,  $X$ , one can simulate the remainder of the graph by drawing from the remaining edges. For each sample, one can assess whether the graph is connected. Repeating this a large number of times yields some number of connected graphs and some number of unconnected graphs. The relative proportions can be used to assign a probability that the underlying graph is connected. This approach can also be loosened if desired and other measures made on the simulated graphs.

Once we address real experimental data, the situation changes and becomes more problematic. Some alternative approaches will be needed. In addition we propose four different summary statistics and investigate their behavior using the ScISI and the available Y2H data.

1. For a given protein complex,  $C_i$ , find all complex members,  $P_j$  that were used as a bait in some Y2H experiment. For these, compute the proportion that found at least one other member of the complex. If  $P_j$  was used as a bait in more than one Y2H experiment do not double count, but take any positive result as positive.

2. For a given protein complex,  $C_i$ , find the average out-degree of all bait proteins, again avoid double counting, in this case by taking the maximum out-degree. Divide this by the complex size.
3. Given the number of proteins in a complex that are detected as either bait or prey, find the number that are connected to at least one other complex co-member. [Should this be the proportion of complex members that are connected to at least one other complex member?]
4. Compute the ratio of the number of edges needed to make the complex connected, by adding to the observed edges, divided by the minimum number of edges needed to create a connected graph from the complex. This is essentially a measure of incompleteness.