

Getting Y2H Data from Intact

T Chiang

April 4, 2006

This document will illustrate how one could use the Intact repository to download yeast 2-hybrid bait to prey affiliation data. There is no necessity that the data be y2h as the `collectIntactData` function can be called to gather any information that relates to a bait to prey(s) affiliation.

```
> library(y2hStat)
```

```
Loading required package: GO  
Loading required package: GOstats  
Loading required package: graph  
Loading required package: Ruuid  
Loading required package: annotate  
Loading required package: Biobase
```

```
Welcome to Bioconductor
```

```
Vignettes contain introductory material.  
To view, simply type 'openVignette()' or start with 'help(Biobase)'.  
For details on reading vignettes, see the openVignette help page.
```

```
Loading required package: RBGL  
Loading required package: xtable
```

```
Attaching package: 'xtable'
```

```
The following object(s) are masked from package:graph :
```

```
label
```

```
Loading required package: Biobase  
Loading required package: genefilter  
Loading required package: survival  
Loading required package: splines  
Loading required package: multtest
```

```

Loading required package: YEAST
Loading required package: ScISI
Loading required package: apComplex
Loading required package: graph
Loading required package: Rgraphviz

```

The first function that we will call will be the `collectIntactData` function. As the name implies, this function goes into the Intact repository, which exists within this package as a modified XML file, and collects the relevant data given a particular experiment.

```

> dataList <- collectIntactData()
> names(dataList)

[1] "allBaits"          "allPreys"          "indexSetAll"       "baitsSystematic"
[5] "preysSystematic" "shortL"

```

The function is hardcoded to collect data on the 41 yeast 2-hybrid experiments; if we wanted to collect data on a subset of this set or on any other experiments, we would simply enter a character vector of the Intact ID's for the relevant experiments. For example, Ito's 2001 experiment would be referenced by Intact as "EBI-375746". For now, we will only concentrate with the `indexSetAll` sub-sections of the output of `collectIntactData`, as this is the main portion to creating the binary interactome.

```

> bpList <- createBPList(dataList[["indexSetAll"]], dataList[["baitsSystematic"]],
+   dataList[["preysSystematic"]])
> bpList[[3]]

$`EBI-295695`
  EBI-295687
"EBI-295687"

$`EBI-295769`
  EBI-295788
"EBI-295788"

$`EBI-295780`
  EBI-295788
"EBI-295788"

$YKR082W
  EBI-12345
"YJR042W"

```

The `createBPList` takes the set of the bait to prey ordered pairs that `indexSetAll` documents and creates a list of lists. The list, here called `bpList`, is a named list referenced by the first author's name and the date of the experiment. Each entry of `bpList` contains n_i

more lists (where n_i is the number of baits reported by Intact for experiment i); each list is referenced by a bait protein and the entry to each list is a character vector of the preys found by the bait within that particular experiment.

We note that the majority of the proteins have the systematic nomenclature, but some do retain the Intact ID's. We have tried to be exhaustive in mapping the Intact ID's to the systematic names. Two mappings are possible: 1. the Intact ID's are sent to SGD ID's (common gene names) if the mapping is well defined, and from there the SGD ID's are sent to the systematic gene names via the `YEASTCOMMON2SYSTEMATIC` environment of the `YEAST` package; 2. the Intact ID's are sent to SwissProt Accension ID's if this mapping is well defined, and from there the SwissProt ID's are sent to the systematic gene names from a list obtained from http://www.rzpd.de/info/LinkOut/rzpdLinkList_sc.txt. Even with these two different mappings, there are still some Intact ID's which are not defined as systematic gene names, and we leave those Intact ID's.