Testing Gene Lists for GO Term Association using GOstats

S. Falcon and R. Gentleman

November 27, 2006

1 Introduction

TheGOstats package has extensive facilities for testing the association of Gene Ontology (GO) The Gene Ontology Consortium (2000) terms among a given gene list. You can test for both over and under representation of GO terms using either the standard Hypergeometric test or a conditional Hypergeometric test that uses the relationships among the GO terms for conditioning (similar to that presented in Alexa et al. (2006)).

In this vignette we describe the preprocessing required to construct inputs for the main testing function, `hyperGTest`, the algorithms used, and the structure of the return value. We use a microarray data set (Chiaretti et al., 2004) from a clinical trial in acute lymphoblastic leukemia (ALL) to work an example analysis. In the ALL data, we focus on the patients with B-cell derived ALL, and in particular on comparing the group with ALL1/AF4 to those with no observed cytogenetic abnormalities.

2 Preprocessing and Inputs

To perform an analysis using the Hypergeometric-based tests, one needs to define a *gene universe* (usually conceptualized as the number of balls in an urn) and a list of selected genes from the universe. While it is clear that the selected gene list determines to a large degree the results of the analysis, the fact that the universe has a large effect on the conclusions is, perhaps, less obvious.

For microarray data, one can use the unique gene identifiers assayed in the experiment as the gene universe. However, the presence of a gene on the array does not necessarily mean much. Some arrays, such as those from Affymetrix, attempt to include probes for as much of the genome as possible. Since not all genes will be expressed under all conditions (a widely held belief is that about 40% of the genome is expressed in any tissue), it may be sensible to reduce the universe to those that are expressed.

To identify the set of expressed genes from a microarray experiment, we propose that a non-specific filter be applied and that the genes that pass the filter be used to form the universe for any subsequent functional analyses. Below, we outline the non-specific filtering procedure used for the example analysis.

Once a gene universe has been established, one can apply any number of methods to select genes. For the example analysis we use a simple t-test to identify differentially expressed genes among the two subgroups in the sample population.

It is worth noting that the effect of increasing the universe size with genes that are irrelevant to the questions at hand, in general, has the effect of making the resultant p-values look more significant. For example, in a universe of 1000 genes where 400 have been selected, suppose that a GO term has 40 gene annotations from the universe of 1000. If 10 of the genes in the selected gene list are among the 40 genes annotated at this category, then the Hypergeometric p-value is 0.99. However, if the gene universe contained 5000 genes, the p-value would drop to 0.001.

2.1 Non-specific filtering

Our non-specific filtering procedure removed probes missing either Entrez Gene identifies or mappings to GO terms. Because of an imbalance of men and women by group, probes measuring genes on the Y chromosome were dropped. The inter-quartile range was used with a cutoff of 0.5 to select probes with sufficient variability across samples to be informative; probes with little variability across all samples are inherently uninteresting. Finally, the set of remaining probes was refined by ensuring that each probe maps to exactly one Entrez Gene identifier. For those probes mapping to the same Entrez Gene ID, the probe with largest IQR was selected.

Producing a set of Entrez Gene identifiers that map to a unique set of probes at the non-specific filtering stage is important because genes are mapped to GO categories using Entrez Gene IDs and we want to avoid double counting any GO categories. In all, the filtering left 3248 genes.

2.2 Gene selection via t-test

After applying the non-specific filtering described above, a standard t-test was used to identify a set of genes with differential expression between the ALL1/AF4 and NEG groups. There were 641 genes with p-values less than 0.05. We did not make use of any p-value correction methods since we are interested in a relatively long gene list.

A detail often omitted from GO association analyses is the fact that the t-test, and most similar statistics, are directional. For a given gene, average expression might be higher in the ALL1/AF4 group than in the NEG group, whereas for a different gene it might be the NEG group that shows the increased expression. By only looking at the p-values for the test statistics, the directionality is lost. The danger is that an association with a GO category may be found where the genes are not differentially expressed in the same direction. One way to tackle this problem is by separating the selected gene list into two lists according to direction and running two analyses. A more elegant approach is the subject of further research.

2.3 Inputs

Often one wishes to perform many similar analyses using slightly different sets of parameters and to facilitate this pattern of usage the main interface to the Hypergeometric tests, `hyperGTest`, takes a single parameter object as its argument. This argument is an instance of class *GOHyperGParams*. Using a parameter class instead of individual arguments makes it easier to organize and execute a series of related analyses. For example, one can create a list of *GOHyperGParams* instances and perform the Hypergeometric test on each using R's `lapply` function:

```
resultList <- lapply(lisOfParamObjs, hyperGTest)
```

Below, we create a parameter instance by specifying the gene list, the universe, the name of the annotation data package, and the GO ontology we wish to interrogate. For the example analysis, we have stored the vector of Entrez Gene identifiers making up the gene universe in `entrezUniverse`. The selected genes are stored in `selectedEntrezIds`. In addition, users can specify a p-value cutoff, a flag to indicate whether the conditional Hypergeometric calculation should be used, and indicate whether the test should evaluate over or under-representation of GO terms.

```
> hgCutoff <- 0.001
> params <- new("GOHyperGParams", geneIds = selectedEntrezIds,
+     universeGeneIds = entrezUniverse, annotation = "hgu95av2",
+     ontology = "BP", pvalueCutoff = hgCutoff, conditional = FALSE,
+     testDirection = "over")
```

3 GOstats Capabilities

In the Hypergeometric model, each term is treated as an independent classification. Each gene is cross-classified according to whether or not it has been selected and whether or not it is annotated, not necessarily specifically annotated, at a particular term. A Hypergeometric probability is computed to assess whether the number of selected genes associated with the term is larger than expected.

The `hyperGTest` function provides an implementation of the commonly applied Hypergeometric calculation for over or under-representation of GO terms in a specified gene list. This computation ignores the structure of the GO terms, treating each term as independent from all other terms.

Often an analysis for GO term associations results in the identification of directly related GO terms with considerable overlap of genes. This is because each GO term inherits all annotations from its more specific descendants. To alleviate this problem, we have implemented a method which conditions on all child terms that are themselves significant at a specified p-value cutoff. Given a subgraph of one of the three GO ontologies, we test the leaves of the graph, that is, those terms with no child terms. Before testing the terms whose children have already been tested, we remove all genes annotated at significant children from the parent's gene list. This continues until all terms have been tested.

4 Outputs

The `hyperGTest` function returns an instance of class *GOHyperGResult*. Printing the result at the R prompt provides a brief summary of the test performed and the number of significant terms found.

```
> hgOver <- hyperGTest(params)
> conditional(params) <- TRUE
> hgCondOver <- hyperGTest(params)
```

```
> hgOver

Gene to GO BP  test for over-representation
1217 GO BP ids tested (22 have p < 0.001)
Selected gene set size: 582
    Gene universe size: 2915
    Annotation package: hgu95av2
```

The *GOHyperGResult* instance returned by `hyperGTest` contains the p-value, odds ratio, expected gene count, and actual gene count for each term tested along with the vector of gene identifiers annotated at each term. It is also possible to retrieve a *graph* instance representing the GO DAG for further computation. All result components can be accessed programatically using accessor functions (see the manual page for the *GOHyperGResult* class for details). Calling `summary` on the result produces a `data.frame` summarizing the results which can optionally be limited to a user-specified minimum p-value and/or minimum gene count for the terms. To make it easy for non-technical users to review the results, the `htmlReport` function generates an HTML file that can be viewed in any web browser. The output generated by `htmlReport` as called below is available at FIXMEPOSTURL.

```
> htmlReport(hgCondOver, file = "ALL_hgco.html")
```

```
> toLatex(sessionInfo())
```

- R version 2.4.0 (2006-10-03), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US;LC_NUMERIC=C;LC_TIME=en_US;LC_COLLATE=en_US;LC_MONETARY=en_US;LC_ME`
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, tools, utils
- Other packages: ALL 1.4.1, annotate 1.12.0, Biobase 1.12.2, Category 2.0.3, genefilter 1.12.0, geneplotter 1.12.0, GO 1.14.1, GOstats 2.0.4, graph 1.12.0, hgu95av2 1.14.0, KEGG 1.14.1, RBGL 1.10.0, RColorBrewer 0.2-3, Rgraphviz 1.12.1, survival 2.29, xtable 1.4-2

References

Adrian Alexa, Jorg Rahnenfuhrer, and Thomas Lengauer. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, 22 (13):1600–7, 2006.

S. Chiaretti, X Li, R Gentleman, A Vitale, M. Vignetti, F. Mandelli, J. Ritz, , and R. Foa. Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103:2771–2778, 2004.

The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.