Hypergeometric Tests for Gene Lists

The Category and GOstats Packages

Seth Falcon
Fred Hutchinson Cancer Research Center
2007
The Plan

1. GO
2. Data filtering
3. Selecting interesting genes
4. Are any GO terms over represented?
5. Hypergeometric testing
6. Conditional Hypergeometric testing
7. Using Category and GOstats
A Category Database

1. GO
2. KEGG
3. PFAM
4. Chromosome bands

A collection of gene sets
Each GO term has zero or more EntrezGene ID annotations.

Parent terms inherit annotations from children.

In our plots, edges go from children to parents.

DAG: Directed Acyclic Graph
GO Evidence Codes

- **IMP**: inferred from mutant phenotype
- **IGI**: inferred from genetic interaction
- **IPI**: inferred from physical interaction
- **ISS**: inferred from sequence similarity
- **IDA**: inferred from direct assay
- **IEP**: inferred from expression pattern
- **IEA**: inferred from electronic annotation
- **TAS**: traceable author statement
- **NAS**: non-traceable author statement
- **ND**: no biological data available
- **IC**: inferred by curator
The Plan

1. GO
2. **Data filtering**
3. Selecting interesting genes
4. Are any GO terms over represented?
5. Hypergeometric testing
6. Conditional Hypergeometric testing
7. Using Category and GOstats
```
varCut = 0.5
filterAns = nsFilter(bcrAblOrNeg,
    require.entrez = TRUE,
    require.symbol = TRUE,
    require.GOBP = TRUE,
    remove.dupEntrez = TRUE,
    var.func = IQR,
    var.cutoff = varCut)
```
Non-Specific Filtering

- Remove probe sets with no Entrez Gene ID or no GO annotation.
- Compute IQR, and filter out probe sets with little variance across samples.
- If two or more probe sets map to the same Entrez Gene ID, keep only one (the one with largest IQR). This is important to avoid double counting.

```r
varCut = 0.5
filterAns = nsFilter(bcrAblOrNeg,
require.entrez = TRUE,
require.symbol = TRUE,
require.GO BP = TRUE,
remove.dupEntrez = TRUE,
var.func = IQR,
var.cutoff = varCut)
```
The Plan

1. GO
2. Data filtering
3. Selecting interesting genes
4. Are any GO terms over represented?
5. Hypergeometric testing
6. Conditional Hypergeometric testing
7. Using Category and GOstats
Select interesting genes

- Use the method of your choice.
- We will use a t-test for differential expression between two groups.

```r
ttests = rowttests(nsFiltered, "mol.biol")
ttestCutoff = 0.05
smPV = ttests$p.value < ttestCutoff
```
The Plan

1. GO
2. Data filtering
3. Selecting interesting genes
4. Are any GO terms over represented?
5. Hypergeometric testing
6. Conditional Hypergeometric testing
7. Using Category and GOstats
Back to the question

Are there any GO terms that have a larger than expected subset of our selected genes in their annotation list?

If so, these GO terms will give us insight into the functional characteristics of the gene list.

The common test is for *over representation*, but one can also test for *under representation*.
An Urn Model
An Urn Model

- The urn contains a ball for each gene in the gene universe.
- Paint the balls representing genes in our selected list white and paint the rest black.
- Don't forget to label the balls so that you know which gene is which!
The Hypergeometric Distribution

- From Wikipedia, the free encyclopedia:

  ... the hypergeometric distribution is a discrete probability distribution that describes the number of successes in a sequence of $n$ draws from a finite population without replacement.

Quiz: (5 minutes) write a detailed proof that a hypergeometric test is equivalent to a one-tailed Fisher's Exact test.
Testing a GO Term

Testing a GO term amounts to drawing the genes annotated at it from the urn and tallying white and black.
Testing a GO Term

- Testing a GO term amounts to drawing the genes annotated at it from the urn and filling out the table.
- NB: You can apply other two-way table tests besides Fisher's Exact test. For large categories, that may make sense.

<table>
<thead>
<tr>
<th></th>
<th>Selected (white)</th>
<th>Not (black)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In GO</td>
<td>n11</td>
<td>n12</td>
</tr>
<tr>
<td>Not in GO</td>
<td>n21</td>
<td>n22</td>
</tr>
</tbody>
</table>
What's in the universe?

What genes were candidates for selection?
The choice makes a big impact on Hypergeometric test results.

Possibilities:
- All genes that have a GO annotation
- All genes on the chip
- All genes from the chip that pass a non-specific filter
The Universe Matters

\[
P = \text{function}(\text{size}) \{ \\
    \text{nFound} \leftarrow 10 \\
    \text{nDrawn} \leftarrow 400 \\
    \text{nAtCat} \leftarrow 40 \\
    \text{nNotAtCat} \leftarrow \text{size} - \text{nAtCat} \\
    \text{phyper}(\text{nFound}-1, \text{nAtCat}, \text{nNotAtCat}, \text{nDrawn}, \text{lower.tail}=\text{FALSE}) \\
\}
\]

\[
P(1000) \rightarrow 0.986 \\
P(5000) \rightarrow 0.000914
\]
GO Hypergeometric Testing Hazards

- The Hypergeometric test assumes independence of categories, but...
- Test results often include directly related terms. Is there really evidence for both terms?
- Many tests are performed; p-values must be interpreted with care.
The Plan

1. GO
2. Data filtering
3. Selecting interesting genes
4. Are any GO terms over represented?
5. Hypergeometric testing
6. **Conditional Hypergeometric testing**
7. Using Category and GOstats
GO Testing Philosophy

• More general statements require evidence beyond that which is required to prove more specific statements.

• This is an essential component of the scientific method.

• We only want to call a GO term significant if there is evidence beyond that provided by its significant children.
A Conditional Algorithm

- To test a term conditional on its children, remove genes inherited from significant children before testing.
A Conditional Algorithm

1. Walk leaves of the GO DAG, compute Hypergeometric as usual.
2. When computing the next level, remove genes from significant children.

For further details:
Falcon and Gentleman, Bioinformatics, 2007
Alexa et al, Bioinformatics, 2007
The Plan

1. GO
2. Data filtering
3. Selecting interesting genes
4. Are any GO terms over represented?
5. Hypergeometric testing
6. Conditional Hypergeometric testing
7. Using Category and GOstats
Using the Category and GOstats Packages

First, a short diversion:

Object Oriented Programming in R: the S4 Object System.

- Classes structure data and encapsulate noun concepts (things).
- Methods operate on objects (instances of classes). They do the right thing based on the object's class. They encapsulate verb concepts (actions).
Key classes in Category & GOstats

Inputs:

HyperGParams
GOHyperGParams
KEGGHyperGParams
PFAMHyperGParams

Outputs:

HyperGResult
GOHyperGResult
Input: GOHyperGParams creation

```java
params = new("GOHyperGParams",
genelds = selectedEntrezIds,
universeGenelds = entrezUniverse,
annotation = "hgu95av2",
ontology = "BP",
pvalueCutoff = hgCutoff,
conditional = FALSE,
testDirection = "over")
```
Input: GOHyperGParams
accessors

If \( p \) is a GOHyperGParams instance:

\[
\begin{align*}
\text{geneIds}(p) & \quad \text{testDirection}(p) \\
\text{universeGeneIds}(p) & \quad \text{conditional}(p) \\
\text{annotation}(p) & \quad \text{pvalueCutoff}(p) \\
\text{ontology}(p) & \\
\end{align*}
\]

There are also replacement forms for setting:

\[
\begin{align*}
\text{conditional}(p) &= \text{TRUE} \\
p\text{valueCutoff}(p) &= 0.0000001
\end{align*}
\]
Output: GOHyperGResult

accessors

If \( r \) is a GOHyperGResult instances:

- pvalues
- universeCounts
- summary
- oddsRatios
- geneCounts
- htmlReport
- expectedCounts
- goDag
- geneIdUniverse
- selectedGenes

Most of the accessors for HyperGParams work here too, so you can answer:
Was it conditionl?
Over or under representation?
Etc.
hyperGTest

ans1 = hyperGTest(p1)

p2 = p1
conditional(p2) = TRUE

ans2 = hyperGTest(p)

p can be a:

• GOHyperGParams
• KEGGHyperGParams
• PFAMHyperGParams

Parameter class design makes it easier to run many tests and allows using a single instance as a template for tweaking.
For the lab…

- ALL dataset (ALL1/AF4 vs NEG)
- Work through details of non-specific filtering
- Play with hyperGTest
- Compare standard to conditional tests.