pyEnrichment

Overview

This program package has been implemented by Olivier Fedrigo and Ralph Haygood (© 2009).

It allows for testing gene ontology enrichment from a list of genes associated with a continuous value (e.g. p-value), using Gene Ontology (GO:http://www.geneontology.org/), Protein ANalysis THrough Evolutionary Relationships (PANTHER:http://www.pantherdb.org/), or mouse phenotypic data from Mouse genome Informatics (MGI: http://www.informatics.jax.org/).

Requirements

For running the various tests for category enrichments:

- Python__2.4.x or higher (available at: http://www.python.org)
- PySyck and Syck (available at: http://pyyaml.org/wiki/PySyck)
- R (available at: http://www.r-project.org)

Mapping Data to Categories

Three programs can be used: panther.py, go.py, and mgi.py. They will respectively have access to the PANTHER, GO, and MGI ontology databases and map the list of genes (with values such as pvalues: 0.0 to 1.0) you provided to functional categories of your choice.

python <software>.py -i <inputFile> -o <outputFile> -c <category>

- <software> can either be go, panther, or mgi.
- <inputFile> input file name, tab delimited with a header such as:

```
geneId <tab> geneName <tab> pvalue
gene_001 < tab> SLC2A1 <tab> 0.04
...
```

Each geneId should be unique. Each gene name entry can be a comma-separated list of different name types. This program is able to recognize "hgnc", "entrez", "ensembl", "gene symbol", "refseq_Peptide", "refseq_mRNA", and "uniprot" ids.

- <outputFile> output file name, tab delimited with a header such as:

```
geneId <tab> geneName <tab> value <tab> categories
gene_001 <tab> SLC2A1 <tab> 0.04 <tab> Transport;Glucose
                                                Metabolism
...
```

- <category> Determines what type of ontology you want to use: biological processes("BP"), m("MF"), pathways ("PW"), or cellular components ("CC").

  For PANTHER, permitted values are: "BP", "MF", "PW"
  For GO, permitted values are: "BP", "MF", "CC"
  For MGI, this option is not used

Tests for Category Enrichment

Two tests are available at the moment:

Hypergeometric test (one tail Fisher's exact test)

Wilcoxon test (Mann-whitney U test)

1. Hypergeometric test

python hypergeometric.py -i <inputFile> -o <outputFile> -p <propTop> -k <unclassified>

- <inputFile> input file name, tab delimited with a header such as (=output of go.py, panther.py, or mgi.py):

```
geneId <tab> geneName <tab> pvalue <tab> categories
gene_001 <tab> SLC2A1 <tab> 0.04 <tab> Transport;Glucose
                                                Metabolism
...
```

This file is the output of go.py, panther.py, or mgi.py. But you can also enter any files with three columns (gene name, value, ";" delimited categories). For instance, one could test for chromosome enrichment by replacing category names by chromosome#.

- <outputFile> output file name, tab delimited with a header such as:

```
cat <tab> p-val <tab> top 10% occ <tab> total occ
Transport <tab> 0.005<tab> 10<tab> 40
...
```

`cat` = category name
`p-val` = pvalue for enrichment of this category in the top genes (x% determined by `-p propTop`)
`top x% occ` = occupancy that is the number of genes that belong to this category and

are in the top x%
`total occ` = total occupancy, that is the total number of genes that belong to this category

- <propTop> float value 0.0 ≤ x ≤ 1.0 determining the threshold for the top x% of a list.
- <unclassified> flag to keep unclassified functions or not (yes=will keep them; no=will filter them out)

2. Wilcoxon test

python wilcoxon.py -i <inputFile> -o <outputFile> -k <unclassified>

- <inputFile> input file name, tab delimited with a header such as (=output of go.py, panther.py, or mgi.py):

```
geneID <tab> geneName <tab> pvalue <tab> categories
gene_001 <tab> SLC2A1 <tab> 0.04 <tab> Transport;Glucose
                                                Metabolism
...
```

This file is the output of go.py, panther.py, or mgi.py. But you can also enter any files with three columns (gene name, value, ";" delimited categories). For instance, one could test for chromosome enrichment by replacing category names by chromosome#.

- <outputFile> output file name, tab delimited with a header such as:

```
cat <tab> p-val <tab> <tab> occ
Transport <tab> 0.005<tab> 40
...
```

`cat` = category name
`p-val` = pvalue for enrichment of this category
`occ` = total occupancy, that is the total number of genes that belong to this category

- <unclassified> flag to keep unclassified functions or not (yes=will keep them; no=will filter them out)

Tools

Other useful tools are also available for:

List genes in each category

Adjust p-values

Cluster categories

1. List genes in each category (parseCat.py)

This program takes the output from panther.py, go.py, or mgi.py (list of genes mapped to categories) and convert it to a list of categories with their associated genes.

python parseCat.py -i <inputFile> -o <outputFile>

- <inputFile> input file name, tab delimited with a header such as (=output of go.py, panther.py, or mgi.py):

```
geneId <tab> geneName <tab> pvalue <tab> categories
gene_001 <tab> SLC2A1 <tab> 0.04 <tab> Transport;Glucose
                                                Metabolism
gene_002 <tab> SLC2A2 <tab> 0.05 <tab> Transport;Glucose
                                                Metabolism
gene_003 <tab> SLC2A3 <tab> 0.08 <tab> Transport;Glucose
                                                Metabolism
...
```

- <outputFile> output file name, tab delimited with a header such as:

```
cat <tab> geneNames
Transport<tab> gene_001;gene_002;gene_003
Glucose Metabolism <tab> gene_001;gene_002;gene_003
...
```

2. Adjust p-values (adjust.py)

This program takes the output from hypergeometric.py or wilcoxon.py (list of pvalues for each category) and adjust the pvalues for multiple comparisons with different methods.

python adjust.py -i <inputFile> -o <outputFile> -m <method> -c <occupancy>

- <inputFile> input file name, output file from hypergeometric.py or wilcoxon.py with a header such as:

```
cat <tab> p-val <tab> <tab> occ
Transport <tab> 0.005<tab> 40
...
```

or

```
cat <tab> p-val <tab> top 10% occ <tab> total occ
Transport <tab> 0.005<tab> 10<tab> 40
...
```

- <outputFile> output file name, tab delimited with a header such as:

```
cat <tab> fdr <tab> pval <tab> occ
Transport<tab> 0.1 <tab> 0.005 <tab> 40
...
```

or

```
cat <tab> fdr <tab> pval <tab> occ  <tab> total occ
Transport<tab> 0.1 <tab> 0.005 <tab> 10 <tab> 40
...
```

- <method> one of the following adjustment methods:

  homl = "holm"

  Hochberg = "hochberg"

  Hommel = "hommel"

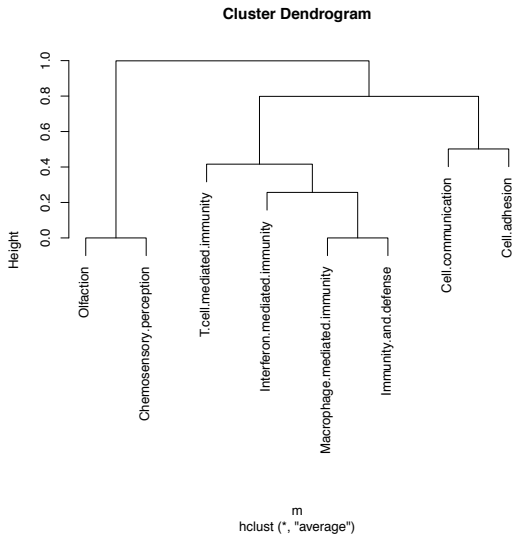
  Bonferroni = "bonferroni"

  Benjamini & Hochberg = "BH"

  Benjamini & Yekutieli = "BY"

  False discovery rate = "fdr"

- <occupancy> minimum category occupancy (minimum number of genes in this category - default value = 1)



3. Cluster categories (makeDendrogram.py)

In order to reflect the hierarchical and overlapping nature of gene ontologies (i.e. some categories may "pull" others). One can clustered significant categories based on a dissimilarity matrix of assigned genes. First, the program determines a subset of significant categories given a p-value or adjusted p-value threshold (output from hypergeometric.py, wilcoxon.py, or adjust.py). Second, genes are assigned to these categories using the output from go.py, panther.py, or mgi.py and a dissimilarity matrix is calculated. Finally, the data is clustered ("average" method) and a dendrogram is plotted and saved as a pdf file.

python makeDendrogram.py -i <inputFile> -o<outputFile> -t<threshold> -m <mappedFile>

- <inputFile> input file name, tab delimited with a header such as (=output of hypergeometric.py, wilcoxon.py. or adjust.py):

```
cat <tab> pvalue <tab> ...
transport <tab> 0.006 <tab> ...
...
```

- <outputFile> output file name, a pdf graphic file.
- <threshold> a p-value or adjusted p-value threshold.
- <mappedFile> input file name, tab delimited with a header such as (=output of panther.py, go.py, or mgi.py):

```
geneId <tab> geneName <tab> pvalue <tab> categories
```

```
gene_001 <tab> SLC2A1 <tab> 0.04 <tab> Transport;Glucose
                                                Metabolism
...
```