How to use clusterProfiler to do GO enrichment analysis with unsupported organisms

Guangchuang Yu
Jinan University
Mar 22, 2012

This vignette is an extension for what already exists in the clusterProfiler.pdf vignette. The clusterProfiler provide *enrichGO* function to do hypergeometric testing with "human", "mouse" and "yeast" organism supported. It is very easy to support other organism provided that the bioconductor annotation package exists.

Most of the software packages for GO enrichment analysis in the Bioconductor project were designed for model organism, and they all rely on the bioconductor annotation packages.

If the organism without annotation package available, it is not easy to employ the existed package to perform such an analysis.

I have extended clusterProfiler to support the unsupported organism.

Here, I will illustrate how to do GO analysis for Streptococcus pyogenes M1 MGAS5005, as an example.

For doing GO analysis, you should have gene and GO mapping data.

I suppose you have nothing in hand, and explain how you get these things in hand.

The whole genome annotation can be downloaded from NCBI. In this example, the M5005 bacteria whole genome annotation file can be downloaded from: ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Streptococcus_pyogenes_MGAS5005_uid58337/

The clusterProfiler package has functions for parsing GFF file. When you have downloaded the gff file, using the following command:

```
> Gff2GeneTable("NC_007297.gff")
```

This function will parse the gff file, and extract information to form a data.frame, and save it as "geneTable.rda" in the current working directory.

```
> load("geneTable.rda")
> head(geneTable)
```

```
   GeneID      GI     seqname start  end strand GeneName        Locus
1  3571011 71909815 NC_007297.1   202 1557      +    dnaA M5005_Spy_0001
3  3571012 71909816 NC_007297.1  1712 2848      +    dnaN M5005_Spy_0002
5  3571013 71909817 NC_007297.1  2923 3120      +       - M5005_Spy_0003
7  3571014 71909818 NC_007297.1  3450 4565      +    ychF M5005_Spy_0004
9  3571015 71909819 NC_007297.1  4635 5204      +     pth M5005_Spy_0005
11 3571016 71909820 NC_007297.1  5207 8710      +    trcF M5005_Spy_0006
>
```

This geneTable is useful for ID mapping, and will be used for mapping GeneID to GeneName if parameter *readable* is set to TRUE when calling *enrichGO*.

```
> eg <- geneTable$GeneID
```

Now you have all GeneID stored in eg, I recommend you using biomaRt package to query GO annotation, and I will demonstrate how to do it.

```
> require(biomaRt)
> bacteria=useMart("bacteria_mart_13")
>  bac = useDataset("str_22007_gene",mart=bacteria)
>  gomap <- getBM(attributes = c("entrezgene",
+             "go_accession"),
+              filters = "entrezgene",
+              values = eg, mart = bac)

> head(gomap)
  entrezgene go_accession
1    3572098   GO:0043565
2    3572008   GO:0006355
3    3572008   GO:0006352
4    3572008   GO:0016987
5    3572008   GO:0003677
6    3572008   GO:0003700
> dim(gomap)
[1] 4025    2
```

You should use other dataset for other bacteria. If the organism is not bacteria, you should use other mart, for instance fungi_mart_13 for fungi.

The gomap only contain GO directly annotation, but undirectly annotation was needed for GO enrichment analysis.

So, clusterProfiler provided another function called *buildGOmap*, for building gomap files needed for analysis.

```
> buildGOmap(gomap)
```

After running this command, *buildGOmap* function generate GO2EG.rda, EG2GO.rda, GO2ALLEG.rda and EG2ALLGO.rda in the working directory.

Providing these files in the working directory. The *enrichGO* function can perform hypergeometric test for this organism.

Suppose the following genes are of interested.

```
gene <- c("3572890","3572609","3572407","3572408","3572333",
        "3572206","3572193","3571922","3571782","3571786",
        "3571624","3571626","3571412","3571413","3571382",
        "3571286","3571289","3571124","3571106","3571029")
> mf = enrichGO(gene, ont="MF", organism="M5005", pvalueCutoff=0.05,
qvalueCutoff=0.1, readable=TRUE)
Loading required package: GO.db
> summary(mf)
                ID
GO:0004312 GO:0004312
GO:0005515 GO:0005515
GO:0004427 GO:0004427
GO:0004807 GO:0004807
GO:0004360 GO:0004360
GO:0008886 GO:0008886
GO:0004585 GO:0004585
GO:0016990 GO:0016990
GO:0004316 GO:0004316
GO:0004356 GO:0004356
GO:0004618 GO:0004618
GO:0003938 GO:0003938
                                                    Description
GO:0004312                                fatty acid synthase activity
GO:0005515                                        protein binding
GO:0004427                          inorganic diphosphatase activity
GO:0004807                        triose-phosphate isomerase activity
GO:0004360          glutamine-fructose-6-phosphate transaminase (isomerizing)
activity
GO:0008886 glyceraldehyde-3-phosphate dehydrogenase (NADP+) (non-
phosphorylating) activity
GO:0004585                  ornithine carbamoyltransferase activity
GO:0016990                        arginine deiminase activity
GO:0004316              3-oxoacyl-[acyl-carrier-protein] reductase (NADPH)
activity
GO:0004356                   glutamate-ammonia ligase activity
GO:0004618                   phosphoglycerate kinase activity
GO:0003938                       IMP dehydrogenase activity
        GeneRatio BgRatio     pvalue    qvalue         geneID Count
GO:0004312     2/20  6/1286 0.003322262 0.09003847       fabF/fabG     2
GO:0005515     3/20 33/1286 0.013038811 0.09003847 groEL/dnaK/nrdE.2
3
```

GO:0004427	1/20	1/1286	0.015552100	0.09003847	ppaC	1
GO:0004807	1/20	1/1286	0.015552100	0.09003847	tpiA	1
GO:0004360	1/20	1/1286	0.015552100	0.09003847	glmS	1
GO:0008886	1/20	1/1286	0.015552100	0.09003847	gapN	1
GO:0004585	1/20	1/1286	0.015552100	0.09003847	arcB	1
GO:0016990	1/20	1/1286	0.015552100	0.09003847	arcA	1
GO:0004316	1/20	1/1286	0.015552100	0.09003847	fabG	1
GO:0004356	1/20	1/1286	0.015552100	0.09003847	glnA	1
GO:0004618	1/20	1/1286	0.015552100	0.09003847	pgk	1
GO:0003938	1/20	1/1286	0.015552100	0.09003847	guaB	1

You can used other tools provided in clusterProfiler, such as *plot* to visualize the result, and *compareCluster* to compare different gene clusters.