A short introduction to the object-oriented interface of goCluster

Gunnar Wrobel

November 1, 2004

Biozentrum & Swiss Institute of Bioinformatics Klingelbergstrasse 50-70, CH-4056 Basel, Switzerland,
http://www.gunnarwrobel.de

1 Getting started

This vignette will provide a short overview on how to get started with *goCluster* and will explain how the package can be used to combine clustering with gene ontology term analysis.

The vignette is focused on using the object oriented interface of *goCluster*. I will provide a second vignette that will explain how to use the internal functions of the package directly at a later point.

```
> library("goCluster", verbose = FALSE)

Loading required package: Biobase
Welcome to Bioconductor
         Vignettes contain introductory material.  To view,
         simply type: openVignette()
         For details on reading vignettes, see
         the openVignette help page.
Loading required package: YEAST

> data(benomylsetup)
> benomyldata <- benomylsetup$data$dataset[1:400, ]
```

This first step loads the package together with a configuration for analysis of the "benomyl dataset" (`benomylsetup`). This dataset characterizes the stress response of yeast to the microtubule destabilizing drug benomyl. The original expression values are included in the configuration object. A reduced dataset is being extracted and stored in the variable `benomyldata`. This dataset is an `exprSet` object as defined in the core *Biobase*-package and it will be the sample dataset to demonstrate the *goCluster*-setup and analysis process. The selection of the first four hundred gene is intended to reduce the execution time for this vignette.

At this point it would be possible to run the complete analysis using the following two statements:

```
> benomyl <- new("goCluster")
> execute(benomyl) <- benomylSetup
```

This is possible because the `benomylsetup` variable is a list that holds all necessary configuration information for the *goCluster*-analysis. But since this vignette is meant to explain the usage of *goCluster* it is not very instructive to use a preconfigured setup. Instead I will demonstrate the `config`-method provided by the *goCluster*-package.

goCluster provides functions to interactively configure the analysis. Since interactivity is something the vignette cannot provide the configuration process in the following sections will contain a mockup of the steps to take in order to configure the analysis.

2 Configuring *goCluster*

The following six sections have to be setup for a *goCluster*-analysis:

1. Data
2. Annotation
3. Clustering
4. Significance analysis
5. Statistical analysis
6. Visualization

All steps will be performed using a single *goCluster*-object that will be created with the following statement:

```
> benomyl <- new("goCluster")
```

Using the `config`-method with this object will start the configuration process.

```
> benomylC <- config(benomyl)
```

2.1 Data

The following sections will provide a transcript of the dialog after calling the `config`-method. It has been splitted to the six sections mentioned above for better readability.

```
Please select a name or title for the dataset.

# Benomyl

Please specify the variable name of the dataset to be analyzed.

# benomyldata

Please specify the variable name of the unique ID for the rows
of the dataset. Alternatively you can specify "rownames" and
the current rownames of the dataset will be used as unique ID.

# rownames
```

Here you can specify a title for your analysis. More importantly you need to set the name of the variable holding your dataset. The benomyl dataset has been stored in the `benomyldata`-variable (see above).

Finally the unique ID has to be specified so that *goCluster* will be able to link each gene to its corresponding annotation data. Often the rows of the dataset will be labelled with the unique ID. In that case you can specify *rownames* here and *goCluster* will retrieve the information from the dataset. But you may also specify the name of a character vector that holds the unique IDs.

In case of the benomyl dataset the rownames have already been replaced with the corresponding yeast symbolic names which provides a convenient link to the *YEAST*-annotation-package available on the bioconductor website.

goCluster will check for the correct class of the dataset variable and it will also verify the length of the unique ID vector by comparing it to the size of the dataset.

2.2 Annotation

```
Please select one of the
following classes for the
annotation data:

1 ) Chromosome-abp
2 ) GO-abp

# 2

Please specify the name of the Bioconductor
meta package that applies to your dataset.
This has to be an AnnBuilder package that
holds all information to your array.
If you are using Affymetrix arrays there is
a high chance you will find a prepackaged
dataset at the Bioconductor website and for
spotted array you will have to create your
own AnnBuilder package.

# YEAST

If the unique id you specified for the
dataset is not the true linker between
genes and annotation, please specify the
linking annotation here. This is the
case for packages like hgu95av2 where the
affymetrix probe ids are the ids that
link all annotations together, but the locus
link ids are the actual ids that have been
used to establish the package in the first
place. The name you specify here needs to
yield the name of an environment together with
the package you specified the step before.
If you don't specify anthing this option is disabled.

#

Please choose the type of ontology to use.
You can use the following abbreviations:
Molecular function (MF),
Biological Process (BP),
Cellular component (CC),
all three (GO).
You can select several ontologi
by seperating them with a comma.

# GO

Please choose the type of GO annotations
```

```
that you are willing to accept:
IC:  inferred by curator,
IDA: inferred from direct assay,
IEA: inferred from electronic annotation,
IEP: inferred from expression pattern,
IGI: inferred from genetic interaction,
IMP: inferred from mutant phenotype,
IPI: inferred from physical interaction,
ISS: inferred from sequence or structural similarity,
NAS: non-traceable author statement,
ND:  no biological data available,
TAS: traceable author statement,
NR:  not recorded,
ALL: accept all types given above.
You can select several evidence types
by separating them with a comma. If
you need more information on the
different types, please read this page:
http://www.geneontology.org/GO.evidence.html

# ALL
```

The initial class selection lets you specify the type of annotation you want to use for the analysis. The "abp"-suffix signifies that the annotation type is dependant on an *AnnBuilder*-package corresponding to the array type the data was generated with. In the second step you will need to specify the name of this *AnnBuilder*-package.

As mentioned in the previous section the correct package for the benomyl dataset is the *YEAST*-meta-package. The gene ontology has been selected as the annotation type and the corresponding *goCluster*-class allows to exclude certain parts of the gene ontology. But here we chose to include all available information.

goCluster will automatically validate that the meta package you specified provides the necessary environments that hold the requested type of annotation data.

2.3 Clustering

```
Please select one of the
following classes for the
gene selection or clustering algorithm:

1 ) Clara
2 ) Hclust
3 ) Kmeans
4 ) Pam

# 4

Please select the distance measure to be
used. See the PAM documentation for a detailed
explanation (?pam).
1 ) euclidean
2 ) manhattan
```

```
# 1

Please select the number of clusters
for the PAM clustering. [2, 10000]

# 4
```

This section allows to specify the algorithm employed in order to partition the genes into groups. While *goCluster* currently only provides four different clustering algorithms it can easily be extended to utilize any kind of gene selection algorithm that will return one or several gene lists. These can then be analysed within the *goCluster*-framework.

Here we select the PAM clustering algorithm (partitioning around medoids). The distance measure is set to euclidian and pam is set to return 4 clusters which will result in about 50 genes per cluster for the four hundred genes we have in our initial dataset.

2.4 Significance Analysis

```
Please select one of the
following classes for the
significance analysis:

1 ) Base
2 ) Bonferroni
3 ) FDR

# 3

Please select how often randomization
should be performed. [0, 10000]

# 4

Please select the false discovery rate
for the selection of GO terms. [0, 1]

# 0.05
```

Here we choose the false discovery rate in order to correct for the massive multiple testing performed when testing the gene ontology terms for each of the clusters identified.

The procedure will replace each of the clusters identified with the same number of randomly selected genes. This will be repeated as often as given in the second parameter. We selected only four repetitions here in order to increase the speed of calculation but in a standard analysis the randomization should be performed around a hundred times.

The significance analysis will use the statistical function configured in the next section in order to establish a histogram over the p-values obtained. This distribution can then be combined with the FDR threshold specified by the user in order to select significant annotation terms.

Please note that this type of statistical approach is not entirely valid for an annotation type like the gene ontology (directed acyclic graph). The results should be considered to be an indication rather than mathematically precise values (the interested reader is referred to a discussion of the problem written by R. Gentleman [`http://bioconductor.org/Docs/Papers/2003/Compendium/GOstats.pdf`]). This problem does not occur in case you use an annotation type that provides terms that are independant of each other.

2.5 Significance Analysis

```
Please select one of the
following classes for the
statistical analysis algorithm:

1 ) Hyper

# 1
```

Currently you can only choose the hypergeometric distribution in order to identify the enrichment of functionally related genes within the clusters.

2.6 Visualization

```
Please select one of the
following classes for the
visualization:

1 ) Heatmap
2 ) None

# 1
```

Currently the visualization methods provided by *goCluster* are still very much in development. But the heatmap function already provides one important feature: It considers every selected annotation term and identifies all genes responsible for the selection of this term. We will use this information in order to visualize the results as a heatmap.

2.7 Retrieving the configuration

The `setup`-method allows you to retrieve the configuration stored in a *goCluster*-object. The value returned will be a list that contains all relevant parameters. The following command demonstrates how you can retrieve the options set for the clustering algorithm:

```
> setup(benomylC)[["algo"]]

$distance
[1] "euclidean"

$clusters
[1] 4
```

3 Executing *goCluster*

The configured object can be analysed by calling its `execute`-method.

```
> benomylE <- execute(benomylC)
```

```
Loading required package: GO
Building GO annotation...
Done!
Loading required package: cluster
Starting to cluster the dataset...
Finished...
Randomizing groups.
25 %..50 %..75 %..100 %..
Analyzing random data.

Going to analyze 1600  groups.
This might take a while...
33 %..67 %..100 %..
Analyzing original data.

Going to analyze 400  groups.
This might take a while...
Selecting significant annotations.
```

The selected (significant) gene ontology terms can be found in the `selection`-slot of the significance analysis object. This object is stored in the `sign`-slot of the main `goCluster`-object. The following code will only display the gene ontology terms of the “Molecular function” branch that were identified in the third cluster.

```
> GOterms <- benomylE@sign@selection
> GOterms[[3]][[1]]
```

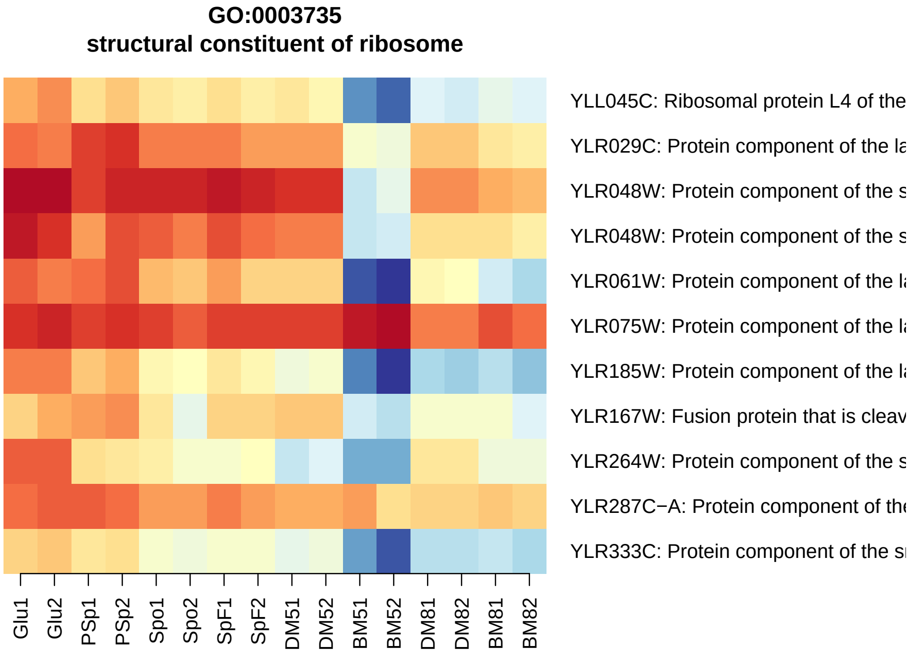
```
  GO:0003735   GO:0005198
3.045828e-10 8.208859e-08
```

Here we select one of the most significant gene ontology terms identified (GO:0003735 - structural constituent of ribosome) and display the genes associated with this term as a heatmap.

You can use the function `selectTerms` in order to select one or several annotation terms from a *goCluster*-result in case you chose the heatmap visualization. The result will be returned in a list format.

```
> GO <- "GO:0003735"
> result <- selectTerms(benomylE, GO)
```

The data section of the `result` variable can be easily rendered as a heatmap.


GO:0003735
structural constituent of ribosome
YLL045C: Ribosomal protein L4 of the
YLR029C: Protein component of the la
YLR048W: Protein component of the s
YLR048W: Protein component of the s
YLR061W: Protein component of the la
YLR075W: Protein component of the la
YLR185W: Protein component of the la
YLR167W: Fusion protein that is cleav
YLR264W: Protein component of the s
YLR287C−A: Protein component of the
YLR333C: Protein component of the sı
Glu1
Glu2
PSp1
PSp2
Spo1
Spo2
SpF1
SpF2
DM51
DM52
BM51
BM52
DM81
DM82
BM81
BM82