goProfiles: an R package for the Statistical Analysis of Functional Profiles

Alex Sánchez, Jordi Ocaña and Miquel Salicrú

November 1, 2022

1 Introduction

This document presents an introduction to the use of *goProfiles*, an R package for the analysis of lists of genes based on their projection at a given level of the Gene Ontology following the methodology developed by [2, 3].

2 Requirements

To use the package one must have R (2.7 or greater) installed. Bioconductor 2.2 or greater is also needed.

3 Quick start

For the impatient user the following lines provide a quick and simple example on the use of the package to explore and compare two experimental datasets obtained from two prostate cancer experiments ([4, 1]). Detailed explanations about the computations can be found in the following chapters and in the package help.

The analysis proceeds as follows:

- First a dataset is loaded into memory. This dataset contains several lists of genes, from two different studies, selected as being differentially expressed in prostate cancer. `help(prostateIds)` will provide extra information about the lists of genes.

```
> require(goProfiles)
> data(prostateIds)
```

- Next a functional profile is build for each list. For simplicity it is build for the MF ontology only.

```
> welsh.MF <- basicProfile (welsh01EntrezIDs[1:100], onto="MF", level=2, orgPack
> singh.MF <- basicProfile (singh01EntrezIDs[1:100], onto="MF", level=2, orgPack
> welsh.singh.MF <-mergeProfilesLists(welsh.MF, singh.MF, profNames=c("Welsh", "
> printProfiles(welsh.singh.MF, percentage=TRUE)

Functional Profile
==================
[1] "MF ontology"
                    Description       GOID Welsh Singh
23    ATP-dependent activity... GO:0140657   4.0     6
6           antioxidant activity GO:0016209   1.0     2
5                       binding GO:0005488  99.0    95
2             catalytic activity GO:0003824  20.2    40
1  cytoskeletal motor activi... GO:0003774   1.0     0
```

```
13 molecular adaptor activit... GO:0060090   6.1     8
16 molecular carrier activit... GO:0140104   1.0     1
15 molecular function regula... GO:0098772  14.1    14
20 molecular sequestering ac... GO:0140313   1.0     1
12 molecular transducer acti... GO:0060089   5.1     3
9  protein folding chaperone... GO:0044183   0.0     3
25 protein-containing comple... GO:0140776   0.0     1
3  structural molecule activ... GO:0005198  21.2    11
17 transcription regulator a... GO:0140110   8.1    10
10 translation regulator act... GO:0045182   0.0     2
4           transporter activity GO:0005215   7.1     6
```

```
> plotProfiles (welsh.MF, aTitle="Welsh (2001). Prostate cancer data")
```

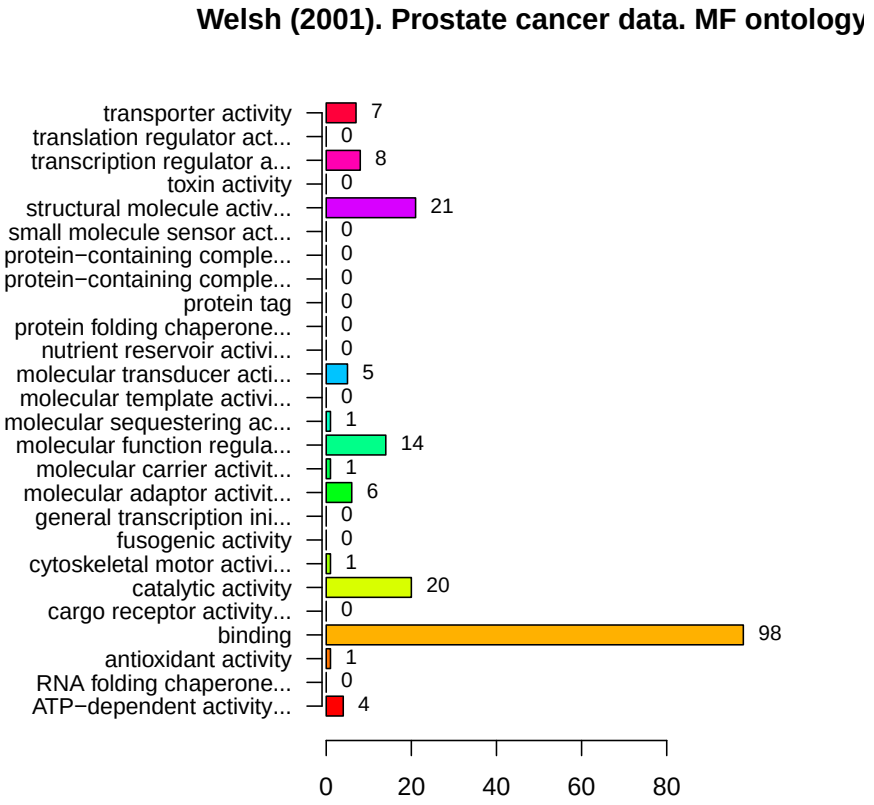

Figure 1: Basic profile for a dataset at the second level of the MF ontology

Changing the parameter `onto` to '`ANY`' instead of '`MF`' will compute profiles for the three ontologies.

```
> welsh <- basicProfile (welsh01EntrezIDs[1:100], onto="ANY", level=2, orgPackage
```

- A visual comparison of profiles can be useful to give hints about the difference between them.
- Finally a numerical comparison of both profiles is performed and its summary is printed. Notice that the comparison rebuilds the profiles, that is the input for the computation are the two lists of genes not the profiles.

```
> plotProfiles (welsh.singh.MF, percentage=T,aTitle="Welsh vs Singh", legend=T)
```

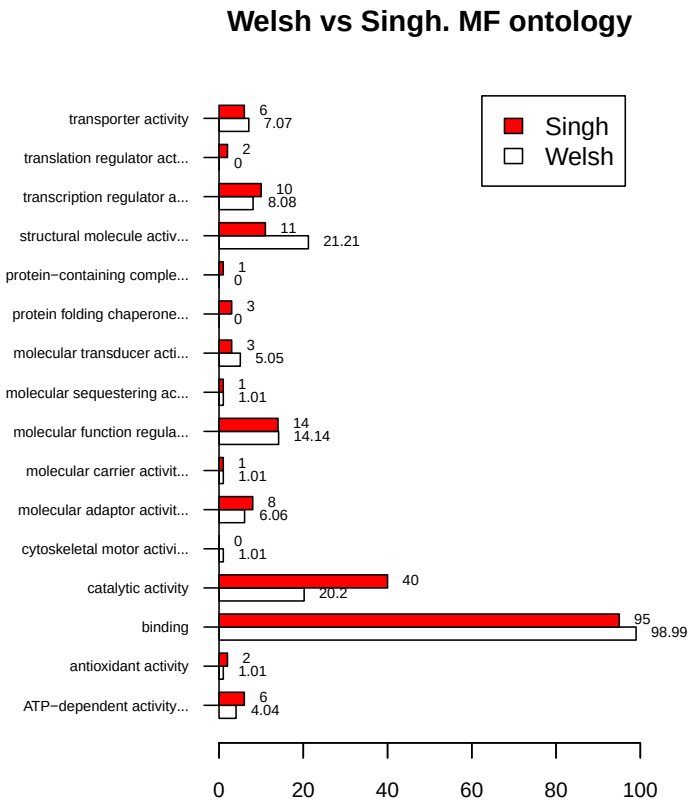

Figure 2: Comparison between two profiles at the second level of the MF ontology

```
> compared.welsh.singh.01.MF <- compareGeneLists (welsh01EntrezIDs[1:100], singh0
> print(compSummary(compared.welsh.singh.01.MF))

Sqr.Euc.Dist       StdErr       pValue   0.95CI.low
    0.057709     0.029974     0.002376    -0.001038
   0.95CI.up
    0.116457
```

- If one finds that the lists are globally different it may be interesting to check out to which categories may be this difference attributed. Recent versions of *goProfiles* (from 1.11.02) allow to do a Fisher test on every category in the level of the profile.

```
> list1 <- welsh01EntrezIDs[1:100]
> list2 <- singh01EntrezIDs[1:100]
> commProf <- basicProfile(intersect(list1, list2), onto="MF", level=2, orgPackag
> fisherGOProfiles(welsh.MF$MF, singh.MF$MF, commProf, method="holm")

GO:0140657 GO:0016209 GO:0005488 GO:0003824 GO:0003774
  1.000000   1.000000   1.000000   0.074802   1.000000
GO:0060090 GO:0140104 GO:0098772 GO:0140313 GO:0060089
  1.000000   1.000000   1.000000   1.000000   1.000000
GO:0044183 GO:0140776 GO:0005198 GO:0140110 GO:0045182
  1.000000   1.000000   1.000000   1.000000   1.000000
GO:0005215
  1.000000
attr(,"unadjusted")
GO:0140657 GO:0016209 GO:0005488 GO:0003824 GO:0003774
 0.7490194  0.4980570  0.2132657  0.0046751  0.4818653
GO:0060090 GO:0140104 GO:0098772 GO:0140313 GO:0060089
 0.5710163  1.0000000  0.8366508  1.0000000  0.4852225
GO:0044183 GO:0140776 GO:0005198 GO:0140110 GO:0045182
 0.2470855  1.0000000  0.0771238  0.8077512  0.4980570
GO:0005215
 0.7772129
```

- The reason to compare two lists from similar studies may be to combine them. In this case what is more meaningful to do is an equivalence test, instead of a difference test as above.

```
> data(prostateIds)
> expandedWelsh <- expandedProfile(welsh01EntrezIDs[1:100], onto="MF",
+                          level=2, orgPackage="org.Hs.eg.db")
> expandedSingh <- expandedProfile(singh01EntrezIDs[1:100], onto="MF",
+                          level=2, orgPackage="org.Hs.eg.db")
> commonGenes <- intersect(welsh01EntrezIDs[1:100], singh01EntrezIDs[1:100])
> commonExpanded <- expandedProfile(commonGenes, onto="MF", level=2, orgPackage=
> equivMF <-equivalentGOProfiles (expandedWelsh[['MF']],
+                            qm  = expandedSingh[['MF']],
+                            pqn0= commonExpanded[['MF']])
> print(equivSummary(equivMF, decs=5))

             Sqr.Euc.Dist                    StdErr
                  0.05771                   0.02997
                   pValue                     CI.up
                  0.72268                   0.10701
                       d0 Equivalent? (1=yes,0=not)
                  0.04000                   0.00000
```

>

4 User guide

This introduction has simply shown the possibilities of the program. A more complete user guide can be found at the program's wiki in github:
https://github.com/alexsanchezpla/goProfiles/wiki

References

[1] Singh, Dinesh., William R Sellers, Philip K. Febbo, Kenneth Ross, Donald G Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A Renshaw, Anthony V D'Amico, Jerome P Richie, Eric S Lander, Massimo Loda, Philip W Kantoff, and Todd R Golub. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.

[2] A. Sánchez-Pla, M. Salicrú, and J. Ocaña. Statistical methods for the analysis of high-throughput data based on functional profiles derived from the gene ontology. *Journal of Statistical Planning and Inference*, 137(12):3975–3989, 2007.

[3] M. Salicrú, J. Ocaña and A. Sánchez-Pla. Comparison of Gene Lists based on Functional Profiles. *BMC Bioinformatics*, DOI: 10.1186/1471-2105-12-401, 2011.

[4] J.B. Welsh, Lisa M. Sapinoso, Andrew I. Su, Suzanne G. Kern, Jessica Wang-Rodriguez, Christopher A. Moskaluk-Jr., Henry F. Frierson, and Hampton Garret M. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res*, 61:5974–5978, 2001.