Tutorial for the WGCNA package for R: III. Using simulated data to evaluate different module detection methods and gene screening approaches

4. Standard gene screening

Steve Horvath and Peter Langfelder

December 7, 2011

Contents

0 Setting up the R session

Before starting, the user should choose a working directory, preferably a directory devoted exclusively for this tutorial. After starting an R session, change working directory, load the requisite packages, set standard options, and load the results of previous sections:

```
# Display the current working directory
getwd();
# If necessary, change the path below to the directory where the data files are stored.
# "." means current directory. On Windows use a forward slash / instead of the usual \.
workingDir = ".";
setwd(workingDir);
# Load WGCNA package
library(WGCNA)
# The following setting is important, do not omit.
options(stringsAsFactors = FALSE);
# Load the previously saved data
load("Simulated-dataSimulation.RData");
attach(ModuleEigengeneNetwork1)
```

4 Standard gene screening based on marginal correlation

In this section we use marginal Pearson to relate genes to the trait y.

```
GS1= as.numeric(cor(y, datExpr, use="p"))
# Network terminology: GS1 will be referred to as signed gene significance measure
p.Standard=corPvalueFisher(GS1, nSamples =length(y) )
# since the q-value function has problems with missing data, we use the following trick
p.Standard2=p.Standard
```

```
p.Standard2[is.na(p.Standard)]=1
q.Standard=qvalue(p.Standard2)$qvalues
# Form a data frame to hold the results
StandardGeneScreeningResults=data.frame(GeneName,PearsonCorrelation=GS1, p.Standard, q.Standard)
```

We take a quick look at the content of `StandardGeneScreeningResults`:

```
> head(StandardGeneScreeningResults)
  GeneName PearsonCorrelation p.Standard q.Standard
1    Gene1        -0.01418468  0.9225264  0.9973321
2    Gene2         0.08848892  0.5430281  0.9923225
3    Gene3         0.02078479  0.8866741  0.9973321
4    Gene4        -0.02444026  0.8669079  0.9973321
5    Gene5         0.06000278  0.6804471  0.9923225
6    Gene6        -0.11163446  0.4421721  0.9923225
```

We note that gene screening based on the Pearson correlation GS1 is equivalent to screening based on the Student T test statistic for a fixed sample size. The reason is that the Student T-test statistic is $t = \sqrt{m-2} * GS1/\sqrt{1-GS1^2}$, where m is the number of microarrays. Note that the t statistic is a monotonic function of the correlation GS1 since m is fixed for each gene.

4.a Evaluation of standard screening

We now determine how many noise genes are in the top say 100 genes returned by standard screening. Recall that the eigengene significances were

```
  ESturquoise ESbrown ESgreen ESyellow
1           0    -0.6     0.6        0
```

This implies that grey non-module genes, turquoise genes, the related blue genes, and yellow genes are noise. The following vector indicates which genes are noise (as simulated):

```
NoiseGeneIndicator=is.element( truemodule, c("turquoise", "blue", "yellow", "grey"))+.0
SignalGeneIndicator=1-NoiseGeneIndicator
```

Note the proportion of noise among the top 20 most significant genes:

```
> mean(NoiseGeneIndicator[rank(p.Standard)<=20])
[1] 0.4
> mean(NoiseGeneIndicator[rank(p.Standard)<=200])
[1] 0.585
> mean(NoiseGeneIndicator[rank(p.Standard)<=100])
[1] 0.48
```

This implies that 48% of the top 100 most significant genes are noise. That is quite bad.

How many noise genes have a q-value less than or equal to 0.20? The following code will give the answer:

```
table(q.Standard<.20)
```

```
> table(q.Standard<.20)

FALSE  TRUE
 2997     3
```

```
mean(NoiseGeneIndicator[q.Standard<=0.20])
```

```
> mean(NoiseGeneIndicator[q.Standard<=0.20])
[1] 0.6666667
```

Only 3 genes have a q-value smaller than .2. Two of the three are noise genes.

Discussion of the performance of standard screening. The performance of standard screening is unsatisfactory. For example, among the top 100 most significant genes (lowest p-value), 55% represent noise. One major reason for this poor performance is that a standard analysis ignores the correlations between gene expression profiles. In other words, it ignores the module structure inherent in the data. We find it biologically and statistically more meaningful to carry out a module based analysis, which first identifies modules and next uses module membership to screen for meaningful genes [1, 2]. WGCNA is a systems biologic gene screening method that makes use of module membership information (or equivalently intramodular connectivity). In the following sections, we will review clustering procedures, module detection methods, module eigengenes, module membership, intramodular connectivity and finally network based gene screening methods.
In the following, we describe how to carry out a detailed WGCNA analysis. This analysis reviews basic clustering procedures and provides an in-depth look at important concepts used by WGCNA.
Before we end this section, we save the calculted data for use in following sections:

```
save.image(file = "Simulated-StandardScreening.RData")
```

References

[1] S. Horvath, B. Zhang, M. Carlson, K.V. Lu, S. Zhu, R.M. Felciano, M.F. Laurance, W. Zhao, Q. Shu, Y. Lee, A.C. Scheck, L.M. Liau, H. Wu, D.H. Geschwind, P.G. Febbo, H.I. Kornblum, T.F. Cloughesy, S.F. Nelson, and P.S. Mischel. Analysis of oncogenic signaling networks in glioblastoma identifies aspm as a novel molecular target. *Proc. Natl. Acad. Sci. USA*, 103(46):17402–17407, 2006.

[2] B. Zhang and S. Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1):Article 17, 2005.