

Seminar III: R/Bioconductor qvalue package

Marie Lisandra Zepeda Mendoza

lzepeda@lcg.unam.mx

3 de septiembre de 2009

Resumen

I will talk to you about another package of Bioconductor named "qvalue"

1. What is qvalue for?

First we have to define what a q-value is: The q-value of a test measures the proportion of false positives incurred (called the false discovery rate) when that particular test is called significant.

The qvalue package has functions for the computation and presentation of q-values for features from an experiment with multiple comparisons.

It automatically generates various plots, allowing us to make sensible significance cut-offs.

2. Why is it interesting?

Why I liked this package is because it can be applied to problems in genomics, brain imaging, astrophysics, and data mining... that is a really wide range of different application fields. And also because we already know what an e-value and a p-value is, but I had not heard before of a named q-value, although the basic concept of what it implies was already known.

Recalling the Statistics subject of the last semester, we saw the types of errors: false positives (also called α or type I), and false negatives (also called β or type II). This is a package to measure those false positives, not just by hand as we learned before.

To install qvalue, start R and enter:

```
> source("http://bioconductor.org/biocLite.R")  
> biocLite("qvalue")
```

3. What other packages does it depend on?

It only depends on one package, the tcltk; but two packages depend on it, which are SSPA (Sample size and Power Analysis for Microarray Data) and webbioc (Bioconductor Web Interface)

4. What can you do with it?

It has four functions:
qvalue: For computing q-values from p-values with multiple comparisons.
qplot: For visualizing the results with graphs
qsummary: For visualizing the results with a table
qwrite: For writing the output to file
It also has a qvalue.gui

5. Examples using this package

We will use a real case study: the breast cancer dataset of Hedenfalk

```
> library(qvalue)
> data(hedenfalk)
> head(hedenfalk, 5)

[1] 0.01212618 0.07502524 0.99492110 0.04178549 0.84581390

> coso <- qvalue(hedenfalk)
> class(coso)

[1] "qvalue"

> head(coso, 2)

$call
qvalue(p = hedenfalk)

$pi0
[1] 0.6635185

pi0 is the proportion of true null hypotheses. The 3er row is all the qvalues.

> qsummary(coso)

Call:
qvalue(p = hedenfalk)

pi0:          0.6635185

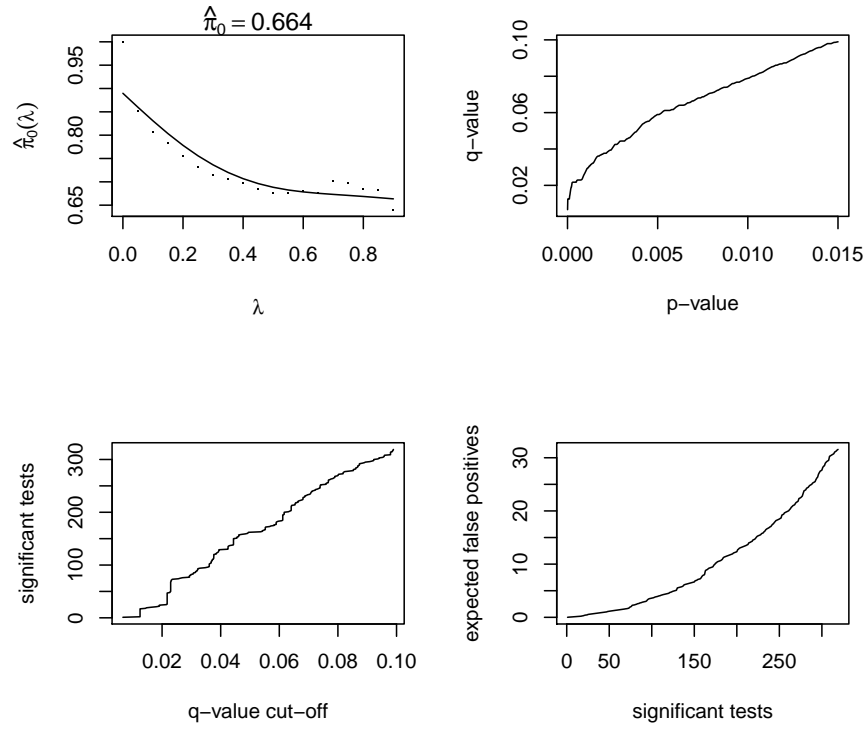
Cumulative number of significant calls:

      <1e-04 <0.001 <0.01 <0.025 <0.05 <0.1 <1
p-value   15     76    265    424    605    868 3170
q-value    0      0      1     73    162    319 3170
```

qplot is the one that I would call the most usefull function of this package. It produces four plots.

The first one is estimated pi0 vs the tuning parameter lambda, comparing your estimate of pi0 to this plot allows you to estimate its quality. The remaining three plots show how many tests are significant, as well as how many false positives to expect for each q-value cut-off.

```
> qplot(coso)
```



```
> qwrite(coso)
```