How to use the OutlierD Package

HyungJun Cho

October 1, 2012

Contents

1 Introduction

It is important to preprocess high-throughput data generated from microarray or mass spectrometry experiments in order to obtain a successful analysis. Outlier detection is an important preprocessing step. For outlier detection, upper and lower fences ($Q3+1.5IQR$ and $Q1-1.5IQR$) of the differences are often used in statistics, where $Q1$=lower 25% quantile, $Q3$=upper 25% quantile, and $IQR = Q3 - Q1$. However, heterogenous variability is often observed in high-throughput data. By ignoring heterogenous variability and using the fences that are constant over all values, we may often miss true outliers and detect false outliers. Therefore, the *OutlierD* package provides various quantile regression techniques (constant, linear, nonlinear, and nonparametric quantile estimators) on the M-A scatterplots, accounting for heterogenous variability to detect outliers with low false positive and negative rates in high-throughput data. The *OutlierD* package employs libraries *quantreg* and *Biobase*, which must be installed in advance.

2 Software Description

We use the duplicates (denoted by X_{1i} and X_{2i}) from the experiments replicated under the same biological and experimental condition, where $i = 1, 2, \ldots, n$ and n is the number of peptides. X_{1i} and X_{2i} are theoretically identical, but in practice have variability, which

is particularly heterogeneous. The heterogeneity of variability can be seen in a MA plot (Figure 1). In the MA plot, M is the difference between duplicates and A is the average of them, i.e., $M_i = \log(X_{1i}/X_{2i}) = \log(X_{1i}) - \log(X_{2i})$ and $A_i = (1/2)\log(X_{1i}X_{2i}) = (\log(X_{1i}) + \log(X_{2i}))/2$. As shown in the first panel of Figure 1, the constant fences are not enough to detect outliers correctly. Using the constant fences $Q3 + 1.5IQR$ and $Q1 - 1.5IQR$ for the differences, we can miss many true outliers in high levels and select many false outliers in low levels.

To account for the heterogeneity of variability, we utilize quantile regression on a M-A scatterplot. The q-quantile linear regression with $\{(A_i, M_i), i = 1, \ldots, n\}$ is to find the parameters minimizing

$$\sum_{\{i:M_i \geq \theta_i\}} q|M_i - \theta_i| + \sum_{\{i:M_i < \theta_i\}} (1-q)|M_i - \theta_i|,$$

where $0 < q < 1$ and $\theta_i = \beta_0 + \beta_1 A_i$. By applying the regression, we compute the 0.25 and 0.75 quantile estimates, $Q1(A)$ and $Q3(A)$, of the differences, M, depending on the levels, A. Then we construct the lower and upper fences: $Q1(A) - 1.5IQR(A)$ and $Q3(A) + 1.5IQR(A)$, where $IQR(A) = Q3(A) - Q1(A)$. To obtain the quantile estimates depending on the levels more flexibly, we can also utilize nonlinear and non-parametric quantile regression approaches. Thus, our developed software **OutlierD** provides intensity-level adaptive fences, which are built from four quantile regression approaches, including constant quantile regression.

3 Example: LCMS Data

We demonstrate the use of the package with a LC/MS data set. This real data set consists of intensity values for 922 peptides and 2 samples. To run *OutlierD*, the data can be prepared as follows.

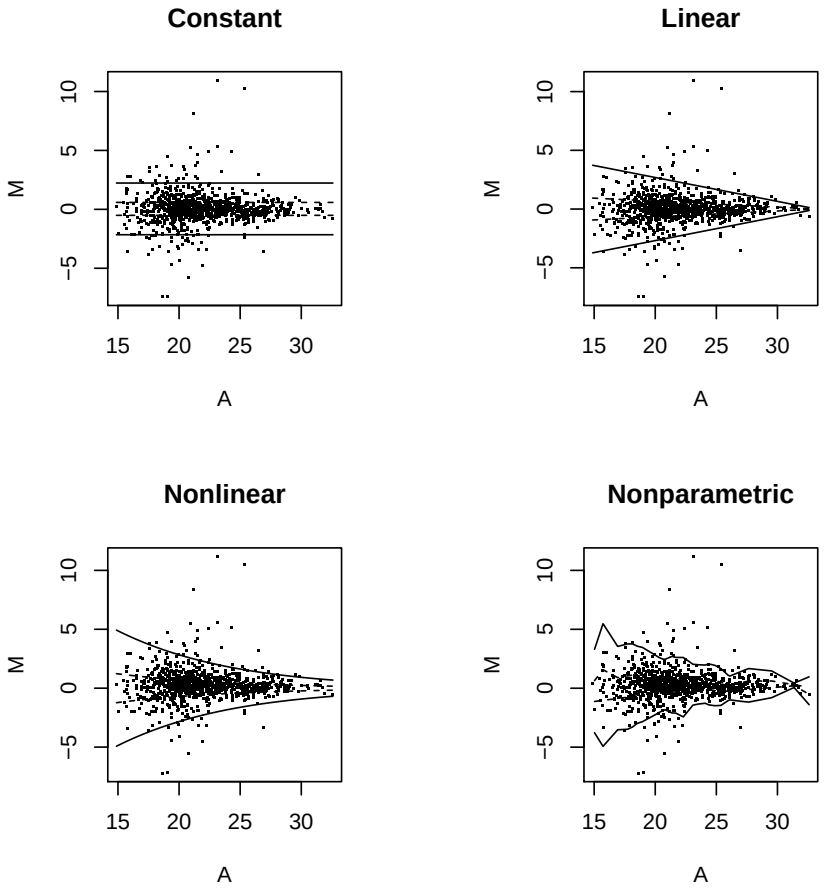
```
> library(OutlierD)
> data(lcms)
> x <- log2(lcms)
> dim(x)

[1] 922   2
```

We here took log2-transformation without any other normalizations. An appropriate normalization can be taken if needed. If the data is ready, *OutlierD* can be run as follows.

```
> fit1 <- OutlierD(x1=x[,1], x2=x[,2], k=1.5, method="constant")

Please wait... Done.
```

```
> fit2 <- OutlierD(x1=x[,1], x2=x[,2], k=1.5, method="linear")

Please wait... Done.

> fit3 <- OutlierD(x1=x[,1], x2=x[,2], k=1.5, method="nonlin")

Please wait... Done.

> fit4 <- OutlierD(x1=x[,1], x2=x[,2], k=1.5, method="nonpar")

Please wait... Done.
```

The arguments $x1$ and $x2$ are two data vectors for samples, consisting of many elements for peptides or proteins. The argument $k(> 0)$ is a constant in $Q1 - k * IQR$ and $Q3 + k * IQR$. To be stringent, give a bigger value for k. A user can choose one of four different quantile regression estimators: constant, linear, nonlin, and nonpar. Using the options show the following plots.



Constant quantile regression is the simplest, whereas nonparametric quantile regression is the most complex. The above objects (fit1, fit2, fit3, and fit4) contain the outputs. To see outliers detected, for example, type:

```
> fit3$n.outliers

[1] 52

> dim(fit3$x)

[1] 922   9

> head(fit3$x)

  Outlier       X1       X2        A           M         Q1        Q3        LB
1   FALSE 21.58628 21.67638 21.63133 -0.09025246 -0.5918062 0.5918062 -2.367225
2   FALSE 17.30741 14.24971 15.77856  2.95421054 -1.1198610 1.1198610 -4.479444
3   FALSE 22.55693 22.72186 22.63940 -0.16497390 -0.5290621 0.5290621 -2.116248
4   FALSE 25.70870 26.51004 26.10937 -0.80134111 -0.3584835 0.3584835 -1.433934
5    TRUE 22.26959 18.44030 20.35495  3.82855048 -0.6814970 0.6814970 -2.725988
6   FALSE 25.09663 25.63526 25.36594 -0.53862554 -0.3898117 0.3898117 -1.559247
        UB
1 2.367225
2 4.479444
3 2.116248
4 1.433934
5 2.725988
6 1.559247
```

The outliers detected by nonlinear quantile regression are indicated in the first column. Using nonlinear quantile regression, 14 outliers were detected.

4 Conclusion

This package is designed to detect outliers using quantile regression on the M-A scatterplots of high-throughput data. According to the degree of heterogeneous variability, one of constant, linear, nonlinear, and nonparametric quantile estimators can be chosen.