MDQC: Mahalanobis Distance Quality Control

Gabriela V. Cohen Freue and Justin Harrington

April 21, 2009

Contents

1 Introduction

The process of producing microarray data involves multiple steps, some of which may suffer from technical problems and seriously damage the quality of the data. Thus, it is essential to identify those arrays with low quality. Our Mahalanobis Distance Quality Control (MDQC) is a multivariate quality assessment method for microarrays that is based on the similarity of quality measures across arrays, i.e., on the idea of outlier detection. Intuitively, the "distance" of an array's quality attributes measures the similarity of the quality of that array against the quality of the other arrays. Then, arrays with unusually high distances can be flagged as potentially low-quality. This method computes a distance measure, the Mahalanobis Distance, to summarize the quality measures contained in a quality control (QC) report for each array. The use of this distance allows us to take the correlation structure of the quality measures into account. In addition, by using robust estimators to identify the typical quality measures of good-quality arrays, the evaluation is not affected by the measures of outlying arrays.

We show that computing these distances on subsets of the quality measures contained in the QC report may increase the method's ability to detect unusual arrays and helps to identify possible reasons of the quality problems. Thus, while MDQC can be based on all the quality measures simultaneously (using `method="nogroups"` in `mdqc` function), it is usually recommended to compute the MDs on subsets of them (using `method="apriori"`, `"cluster"`, or `"loading"`), or on a transformed space with a lower dimension (using `method="global"`). In the `"apriori"` approach the user forms groups of quality measures on the basis of an a priori interpretation of them and according to the quality aspect they represent. The `"cluster"` and the `"loading"` methods are two data-driven methods to form the groups. The former groups the quality measures using clustering analysis, and the latter uses the loadings of a robust principal component analysis (PCA) to identify the quality measures that contain similar information and group them. It is important to note that the `"apriori"`, the `"cluster"`, and the `"loading"` methods create groups of the original quality measures of the report and compute one

MD within each group. Finally, the "global" method computes a single MD based on the reduced space of the first k principal components from a robust PCA. The number k of PCs can be chosen using a scree plot. A robust PCA and the associated biplot and scree plot can be obtained using the prcomp.robust function in this package. More details on each method are given in Cohen Freue *et al.* (2007).

The resulting MDs can be used to flag poor-quality arrays as their MDs will be large relative to those of undamaged arrays, i.e., they will be far from the center of the normal arrays. Under usual distributional assumptions, the squared MDs have an approximate Chi-Squared distribution with p degrees of freedom. Thus, using the Chi-Squared distribution we can set a cutoff point to decide if the array is likely to be defective.

Before illustrating the performance of MDQC, we end with some remarks about MDQC. First, it is important to note that although we use MDQC in the context of quality assessment of microarrays, most of the methodologies implemented by MDQC have been widely used in Statistics to detect outliers. Thus, although we illustrate our method using two data sets of Affymetrix GeneChips and their QC reports, all the ideas can be applied to other platforms and/or QC reports as well as for outlier detection outside microarray data. Second, MDQC can only be used if the number of observations (n) is greater than or equal to the number of quality measures in the group (p). Thus, if p>n, the user needs to divide the quality measures into groups (see the apriori, clustering or loading PCA grouping methods in the paper Cohen Freue *et al.* (2007)) so that each group contains less variables than observations. Otherwise, other PCA methods have to be applied to reduce the dimensionality of the data set (see Huber *et al.* (2002)). Finally, as in any robust estimation method, caution should be taken when MDQC is applied to data sets with small number of arrays. In these cases, the robust estimators of location and covariance matrix may not be properly scaled damaging the performance of MDQC.

2 Acute Lymphoblastic Leukemia Study

To evaluate the performance of MDQC, we use part of an acute lymphoblastic leukemia study described in Ross *et al.* (2003), containing 20 Affymetrix HG-U133Bmicroarrays (see the allQC help file for more details on this data set). However, MDQC can be used on any QC report for other types of microarrays. Bolstad *et al.* (2005) and Brettschneider *et al.* (2007) examined the quality of these arrays using histograms of probe-level data, MA-plots and probe-level model methods described in Bioconductor's affyPLM package. According to their quality assessment, array 2 has a strong spatial artifact on the chip and array 14 presents other evidence of poor quality.

```
> library(mdqc)
> data(allQC)
> dim(allQC)

[1] 20 11

> allQC[1:2, ]

  Scale Factor Percent Present Average Background Minimum Background
1     4.905489       0.2653124           67.34494           62.20386
```

```
2     9.765986       0.2165158           68.18425           52.25200
  Maximum Background     BioB     BioC     BioD     CreX
1           72.16051 9.436160 11.20988 13.19537 14.52620
2          126.21238 9.533749 11.58259 13.52912 15.21234
  AFFX-HSAC07/X00351.3'/5' AFFX-HUMGAPDH/M33197.3'/5'
1                0.3235390                 0.05796629
2                0.9697007                 0.16387418
```

As it was previously mentioned, when the number of quality measures is smaller than the number of observations, one can perform a *multivariate* analysis using MDQC based on *all* the quality measures in the report.

```
> mdout <- mdqc(allQC, method = "nogroups")
> plot(mdout)
> print(mdout)

Method used: nogroups         Number of groups: 1
Robust estimator: S-estimatorMDs exceeding the square root of the  90 % percentile of the Chi-S
[1] 14
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1] 14
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
[1] 14

> summary(mdout)

Summary information for MDQC
Method used: nogroups         Number of groups: 1
Robust estimator: S-estimator
Number of Outliers:
90%        95%        99%
1         1         1
```

Figure 1 shows that using this MDQC approach array 14 is flagged as having potential quality problems and array 2 appears only as a borderline case. In other words, collapsing *all* the quality measures into a single MD downweights array 2's quality problems and masks other outlying observations in the QC report, such as those of arrays 1, 7 or 8. Thus, we study the MDs on groups with a reduced number of variables. These alternative methods reduce the possibility of masking outliers and may give information about the potential source of the quality problem.

Based on the interpretation of the quality measures in the QC report, the user may want to group these measures into the following three groups:

1. Scale Factor, % Present, Avg BG, Min BG, Max BG
2. BioB, BioC, BioD, CreX

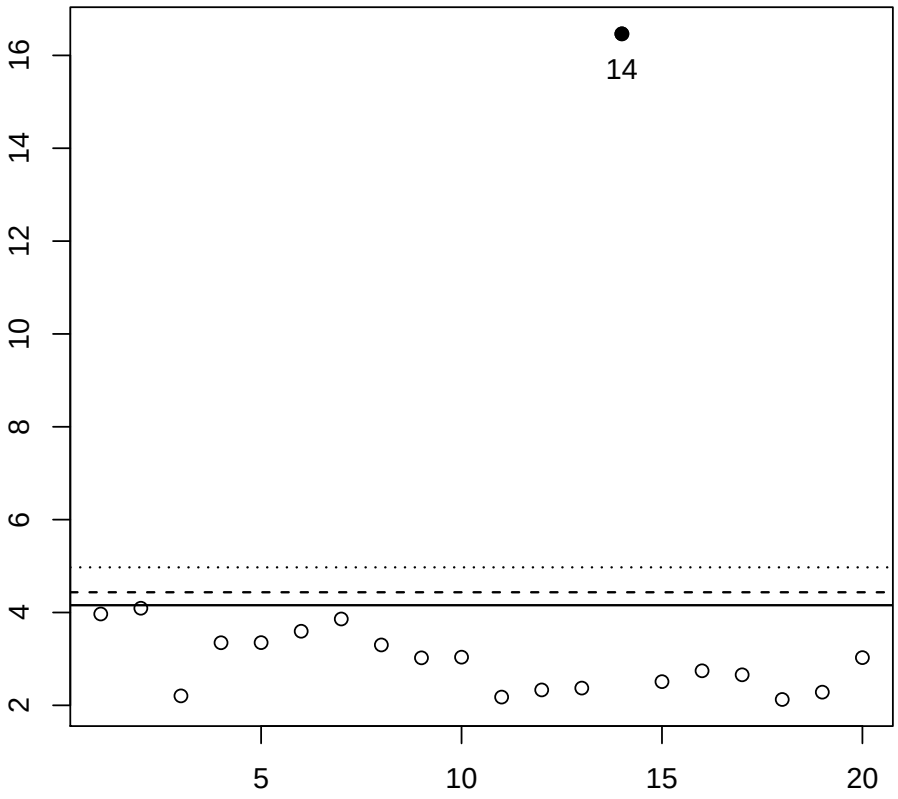

Figure 1: Results of MDQC based on all measures of the QC report. The MDs (y-axis) are computed using the robust S-estimator for each array (x-axis). The solid, dashed and dotted lines correspond to the square root of the 90th, 95th and 99th percentile of the Chi-Squared distribution, respectively. Outlying arrays are identified using solid points.

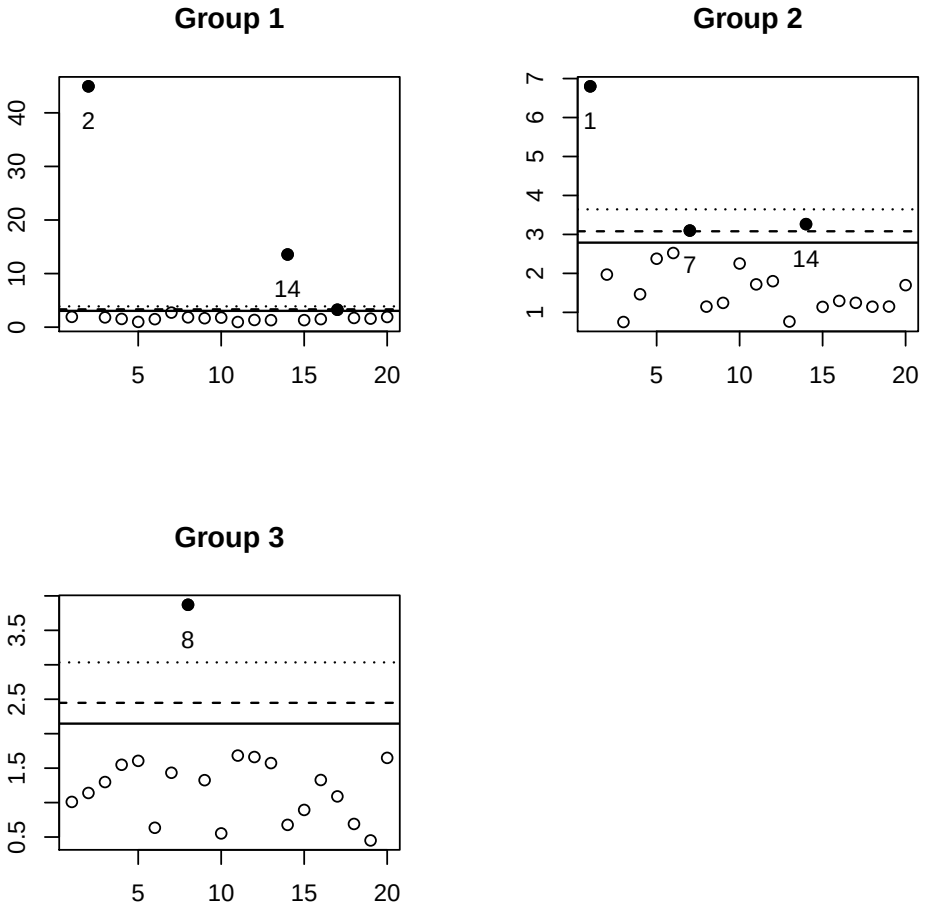

Figure 2: Results of MDQC using the a priori grouping method. The MDs (y-axis) within each group are computed using the robust S-estimator for each array (x-axis). The scale of the y-axis varies from one case to another. The solid, dashed and dotted lines correspond to the square root of the 90th, 95th and 99th percentile of the Chi-Squared distribution, respectively. Outlying arrays are identified using solid points.

3. AFFX-HSAC07/X00351.3'/5', GapDH

This MDQC approach, referred as the **a priori grouping method**, examines one MD for each array and each group. It is important to recall that each group can not contain more quality measures than arrays.

```
> mdout <- mdqc(allQC, method = "apriori", groups = list(1:5, 6:9,
+     10:11))
> plot(mdout)
```

In Group 1, arrays 2 and 14 are *both* flagged as potentially defective, and array 17 as a borderline case. In Group 2, array 1 has a MD exceeding the 99% cutoff and arrays 7 and 14 have MDs exceeding the 95% cutoff line. Finally, array 8 is the only one flagged in Group 3. Thus, the MDs based on groups of lower dimension flag both arrays 2 and 14, which is consistent with the results in Bolstad *et al.* (2005) and Brettschneider *et al.* (2007). In addition, arrays 1 and 8 are flagged as potentially low quality and arrays 7 and 17 as borderline quality. Moreover, based on the interpretability of the groups, the problems in array 2 are most likely due to defects in the chip as this array is only identified in Group 1. Similarly, since arrays 1 and 14 are flagged in Group 2, their low quality is most likely due to low quality of the sample. Note that although array 14 is also flagged in Group 1, this can be still due to quality problems in the sample. Finally, array 8 is flagged only in Group 3, suggesting potential problems in the RNA quality.

While in the apriori grouping method, the groups are created by the user, the **clustering grouping method** and the **loading PCA grouping method** are two data-driven methods used to create the groups of quality measures used to compute the MDs. The **clustering grouping method** uses the Partitioning Around Medoids (pam) clustering algorithm based on the distance constructed from the correlation matrix (i.e., the distance between two variables x and y is given by d(x,y)=(1-cor(x,y))/2) to group the quality measures of the QC report. As the data set may contain outlying observation, using the distance based on the robust estimator of the correlation matrix as a dissimilarity measure makes it more robust. We use the S-estimator (Lopuhaä (1989)) with 25% breakdown point to estimate the correlation matrix of the quality measures. This estimator has demonstrated a good performance in studies with small number of arrays, however, other robust estimators can be used (e.g., MCD). Although in this method the user can choose the number of clusters (k), the number of returned quality measures in each group (cluster) can not exceed the number of arrays in the study. If so, the user needs to increase the number of clusters.

```
> mdout <- mdqc(allQC, method = "cluster", k = 3)
> plot(mdout)
> print(mdout)

Method used: cluster         Number of groups: 3
Robust estimator: S-estimatorGroup 1  - Columns
[1] 1 6 7 8 9
MDs exceeding the square root of the  90 % percentile of the Chi-Square distribution
[1]  1  5 14
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1] 1 5
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
[1] 1

Group 2  - Columns
[1] 2 3 4 5
MDs exceeding the square root of the  90 % percentile of the Chi-Square distribution
[1]  2 14
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1]  2 14
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
```

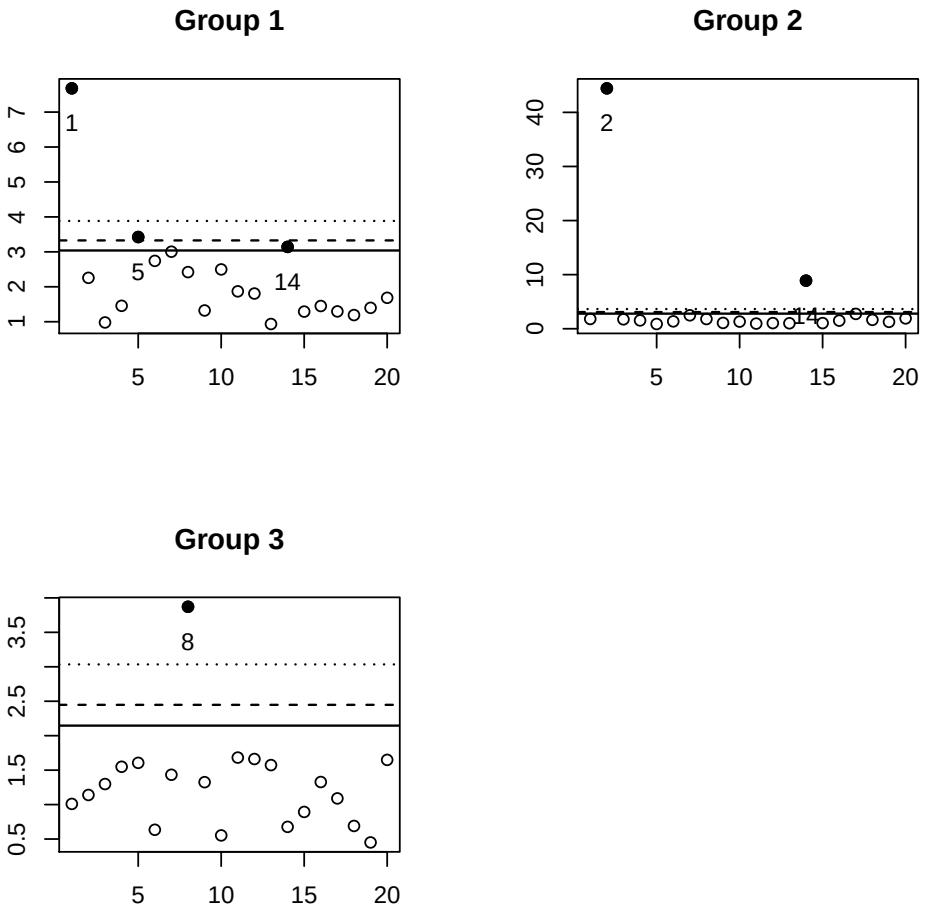

Figure 3: Results of MDQC using the clustering and the loading grouping methods.

```
[1]  2 14

Group 3  - Columns
[1] 10 11
MDs exceeding the square root of the  90 % percentile of the Chi-Square distribution
[1] 8
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1] 8
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
[1] 8
```

The **loading PCA grouping method** uses a robust PCA to group the quality measures in the report according to their contribution to the first k principal components. As each loading vector derived from a robust PCA shows the contribution of each quality variable to the first k PCs, these vectors can

be used to group the quality variables based on the similarity of their contribution to the model. We can apply a clustering method, such as pam-algorithm or a hierarchical clustering method, to group the p quality variables based on their contribution to the first k principal components. As in previous method, the number of quality measures in each group (cluster) can not exceed the number of arrays in the study. Thus, if this is the case, the user needs to increase the number of clusters (k) used in the loading space. For more details on the robust PCA, see the prcomp.robust function in this package and Cohen Freue *et al.* (2007).

```
> mdout <- mdqc(allQC, method = "loading", k = 3, pc = 4)
> plot(mdout)
> print(mdout)

Method used: loading         Number of groups: 3
Robust estimator: S-estimator        Number of Principal Components: 4
Group 1  - Columns
[1] 1 6 7 8 9
MDs exceeding the square root of the  90 % percentile of the Chi-Square distribution
[1]  1  5 14
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1] 1 5
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
[1] 1

Group 2  - Columns
[1] 2 3 4 5
MDs exceeding the square root of the  90 % percentile of the Chi-Square distribution
[1]  2 14
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1]  2 14
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
[1]  2 14

Group 3  - Columns
[1] 10 11
MDs exceeding the square root of the  90 % percentile of the Chi-Square distribution
[1] 8
MDs exceeding the square root of the  95 % percentile of the Chi-Square distribution
[1] 8
MDs exceeding the square root of the  99 % percentile of the Chi-Square distribution
[1] 8
```

It is interesting to see that this approach leads to the same groups as the clustering grouping method, as given in Figure 3.

Finally, we examine the performance of MDQC using the **global PCA method** to reduce the dimensionality of the data. MDQC performs a robust PCA which requires more observations (arrays) than variables (quality measures). If this is not the case, the user needs to apply other PCA methods (see

for example Huber *et al.* (2002)) or group the variables using previously described methods. Using the scree plot, we retain $k = 4$ principal components in this analysis. Figure 4 shows the results of the MDQC when a single MD is calculated based on the first 4 principal components derived from a robust PCA based on robustly standardized data (see Cohen Freue *et al.* (2007) for more details). We note that this approach still flags arrays 2, 8 and 14 as having potential quality problems. However, the first two appear only as borderline cases. In addition, arrays 1, 7 and 17 are still masked using this method.

```
> mdout <- mdqc(allQC, method = "global", pc = 4)
> plot(mdout)
```

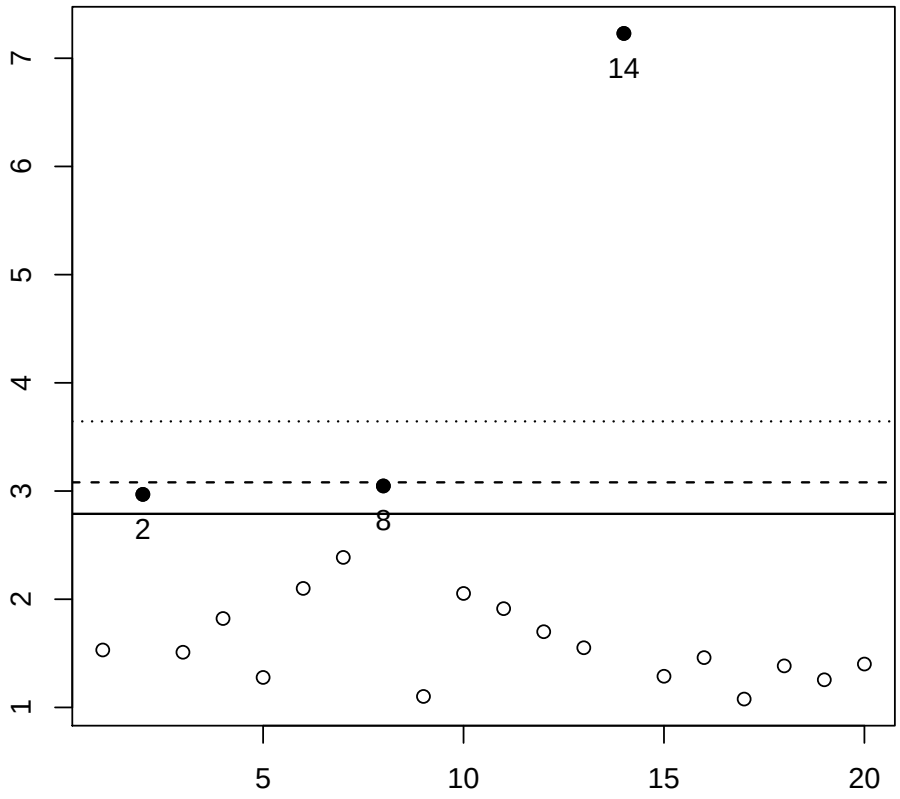

Figure 4: Results of MDQC using the global PCA method. The MDs (y-axis) are computed on the first four principal components for each array (x-axis). The solid, dashed and dotted lines indicate the square root of the 90th, 95th and 99th percentile of the Chi-Squared distribution, respectively. Outlying arrays are identified using solid points.

3 Conclusion

The previous example shows that all five approaches of MDQC (i.e., all variables, a priori, clustering, loading and global PCA) identify the problematic arrays 2 and 14. However, the a priori grouping method outstands the problem of array 2, unmasks other potentially low quality arrays and provides possible explanations of the quality problems.

In summary, MDQC has a clear statistical foundation, it performs a robust multivariate analysis of the quality measures provided in the QC report while taking into account their correlation structure, it is easy to apply, and it is computationally lightweight. These properties make MDQC a useful diagnostic technique suitable for large data sets.

References

Bolstad, B. M.; Collin, F.; Brettschneider, J.; Simpson, K.; Cope, L.; Irizarry R. A.; and Speed T. P. (2005) Quality assessment of Affymetrix GeneChip data. In Gentleman, R.; Carey, C. J.; Huber, W.; Irizarry, R. A.; and Dudoit, S. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York: Springer.

Brettschneider, J.; Collin, F.; Bolstad, B. M.; and Speed, T. P. (2007) Quality assessment for short oligonucleotide arrays. Forthcoming in *Technometrics* (with Discussion).

Cohen Freue, G. V. and Hollander, Z. and Shen, E. and Zamar, R. H. and Balshaw, R. and Scherer, A. and McManus, B. and Keown, P. and McMaster, W. R. and Ng, R. T. (2007) MDQC: A New Quality Assessment Method for Microarrays Based on Quality Control Reports. *Bioinformatics* **23** 3162 - 3169.

Huber, M.; Rousseeuw, P.; and Verboven, S. (2002) A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, **60**, 101-111.

Lopuhaä, H. P. (1989) On the Relation between S-Estimators and M-Estimators of Multivariate Location and Covariance. *Ann. Statist.*, **17**, 1662-1683.

Ross, M. E.; Zhou, X.; Song, G.; Shurtleff, S. A.; Girtman, K.; Williams, W. K.; Liu, H.; Mahfouz, R.; Raimondi, S. C.; Lenny, N.; Patel, A.; and Downing, J. R. (2003) Classification of pediatric acute lymphoblastic leukemia by gene expression profiling. *Blood* **102**, 2951-9.