Robust Tools for the Imperfect World ☆

Peter Filzmoser[a,*], Valentin Todorov[b]

[a] *Department of Statistics and Probability Theory, Vienna University of Technology, Wiedner Hauptstr. 8-10, 1040 Vienna, Austria*
[b] *United Nations Industrial Development Organization (UNIDO), Vienna International Centre, P.O. Box 300, A-1400 Vienna, Austria*

Abstract

Data outliers or other data inhomogeneities lead to a violation of the assumptions of traditional statistical estimators and methods. Robust statistics offers tools that can reliably work with contaminated data. Here, outlier detection methods in low and high dimension, as well as important robust estimators and methods for multivariate data are reviewed, and the most important references to the corresponding literature are provided. Algorithms are discussed, and routines in R are provided, allowing for a straightforward application of the robust methods to real data.

Keywords: robustness, MCD, outliers, high breakdown, PCA, statistical design patterns

1. Introduction

In statistical modeling and estimation we are used to work with assumptions like normal distribution or independence. The practice, however, is usually different: practical data sets often do not follow these strict assumptions. There might be several different processes inherent in the data generating process, or other effects that cannot be controlled, or where the person collecting the data is not even aware of. It is then often unclear how reliable the results are, if the model assumptions are violated.

Robust statistics tries to answer the question of the effect of model deviations on a statistical estimator. Tools like the influence function or the breakdown point even serve as a formal approach to describe the robustness properties of an estimator [26]. Generally speaking, robust statistics offers methods that still give reliable results if strict model assumptions are violated. Robust statistical

☆ The views expressed herein are those of the authors and do not necessarily reflect the views of the United Nations Industrial Development Organization.

* Corresponding author: Tel.: +43 1 58801 10733.
Email addresses: P.Filzmoser@tuwien.ac.at (Peter Filzmoser), v.todorov@unido.org (Valentin Todorov)

methods focus on the data majority (where the model assumptions are met), but tolerate deviations of a part of the data. It then depends on the robust estimator which maximum data fraction is tolerated, and how precise the results still are.

Robustness is typically related to the problem of data outliers, and therefore multivariate outlier detection is an important goal of robust statistics. However, also for outlier detection we have a strict model in mind, and outliers are data points that severely deviate from this model. In multivariate statistics one usually assumes that the data majority originates from a multivariate normal distribution. For the conventional multivariate methods (e.g. principal component analysis (PCA)) we also assume that the data majority follows a multivariate normal distribution, and robust principal component analysis downweights data points that deviate from this model distribution. In the context of regression, data outliers would be observations that deviate from the linear trend formed by the data majority.

Different aspects of robust multivariate statistics and different palettes of methods can be found in several recent review papers on this topic. Daszykowski et al. [14] present some basic concepts of robust statistics (multivariate location and scatter estimation, outlier detection and principal component analysis) and discuss their usefulness in chemometric data analysis. Several references to Matlab implementations are given but no practical details about the software or illustrations of its usage are given. Similarly, Frosch-Møller et al. [23] present an overview of common robust chemometric methods (PCA, principal component regression (PCR) and partial least squares (PLS)) but again, software written in Matlab or SAS is only referred to. The most complete set of procedures for robust statistical analysis which is available in the R programming environment is described in detail in its "manual" [64]. In this contribution we focus on the practical aspects of robust statistics, and we limit ourselves to several important methods and tasks of multivariate data analysis. The presented methods are illustrated with easy to use R code examples. Robust methods for high-dimensional data which were recently implemented in R are presented too. Section 2 is devoted to multivariate outlier detection. In Section 3 the problem of robust multivariate location and covariance estimation is treated. Both estimators play an important role for multivariate statistical methods, and the robust estimators can be directly used to robustify such methods. In recent years, many robust statistical methods have been developed for high-dimensional data: Section 4 mentions several approaches for outlier detection in high dimension. All discussed methods and algorithms are freely available in the statistical software environment R [47]. Since the authors of this article contributed with several R extension packages, Section 5 discusses general design patterns for a unified approach to software tools for robust estimation. Different approaches to robustify the most prominent multivariate method, principal component analysis, are topic of Section 6. The final Section 7 concludes.

2. Detection of outliers by robust methods

Researchers in science, industry and economics work with huge amounts of data and this even increases the possibility of anomalous data and makes their (visual) detection more difficult. The most common estimators of multivariate location and scatter are the sample mean $\bar{\boldsymbol{x}}$ and the sample covariance matrix $\mathbf{S}$ given by

$$\begin{aligned} \bar{\boldsymbol{x}} &= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i \\ \boldsymbol{S} &= \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^t. \end{aligned} \tag{1}$$

These estimates are optimal if the data come from a multivariate normal distribution but are extremely sensitive to the presence of even a few outliers (atypical values, anomalous observations, gross errors) in the data. To get a reliable analysis we need robust estimates, say $\boldsymbol{T}$ and $\boldsymbol{C}$ of the multivariate location and scatter which can down-weight the influence of possible outliers in the data. Such an estimator is the Minimum Covariance Determinant (MCD) introduced by Rousseeuw [50] which will be described later.

In contrast to univariate outliers, multivariate outliers are not necessarily extreme along a single coordinate. They could deviate from the multivariate structure formed by the majority of the observations. We illustrate the effect of outliers on classical location and covariance estimates and the performance of the corresponding robust estimates with a small multivariate example with 20 observations in 5 variables. The data set `wood` from the R package **robustbase** contains the well known modified wood specific gravity data set from Rousseeuw and Leroy [52], Table 8, page 243. The raw data are from Draper and Smith [19], p. 227 and were used to determine the influence of anatomical factors of wood specific gravity with five explanatory variables and an intercept. A contaminated version of the data set was created by replacing a few (four) observations by outliers. We consider only the X part consisting of the explanatory variables. The four outliers do not show up in the box plots in the left part of Figure 1 but they are clearly seen in several of the scatter plots shown in the right panel, i.e. they are not univariate outliers and cannot be identified by investigating any of the coordinates separately.

Figure 2 shows the pairwise correlations computed classically as the sample correlation coefficients and computed robustly by applying the Minimum Covariance Determinant (MCD) method. In the upper triangle the corresponding ellipses are shown representing bivariate normal density contours with location and covariance computed classically (sample mean and sample covariance matrix) and robustly (applying MCD). A large positive or negative correlation is

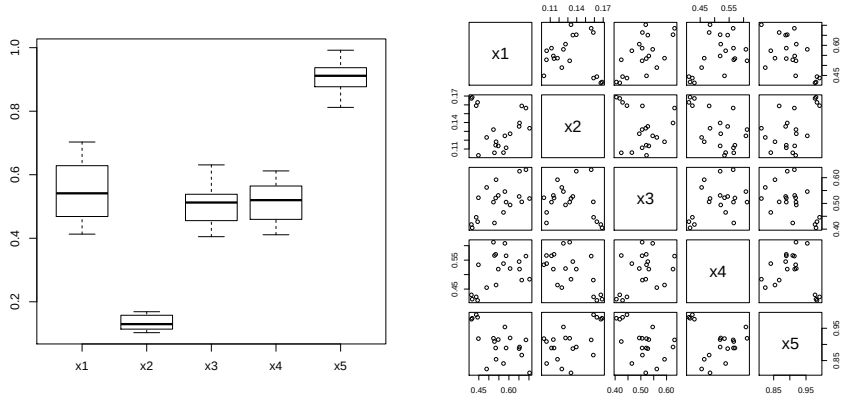

Figure 1: Box plots (left) and pairwise scatter plots (right) for the modified wood gravity data set.

represented by an elongated ellipse with major axis oriented along the ±45 degree direction while near to zero correlation is represented by an almost circular ellipse. The differences between the classical and robust estimates are easily seen visually.

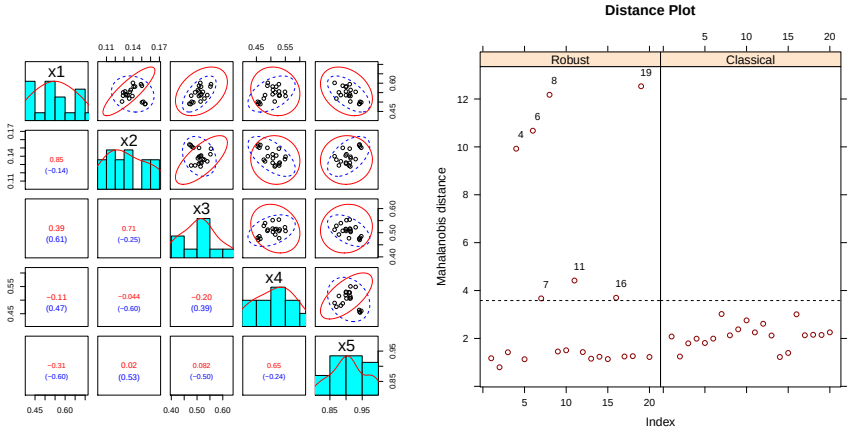

Figure 2: Classical and robust correlations and tolerance ellipses (left) and classical and robust Mahalanobis distances (right) for the modified wood gravity data.

Consider p-dimensional observations $\boldsymbol{x}_i$ (column vector), for $i = 1, \ldots, n$, which are collected in the rows of the data matrix $\boldsymbol{X}$. A general framework for multi-

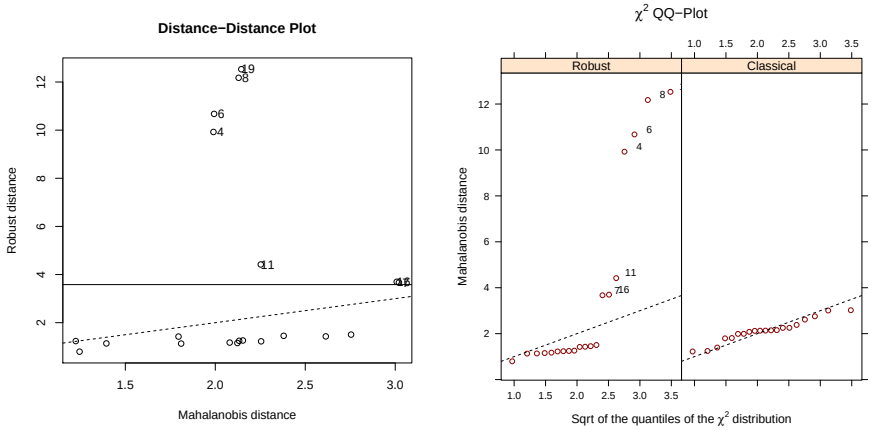

Figure 3: Distance-distance plot (left) and χ^2 quantile-quantile plot (right) of the classical and robust Mahalanobis distances for the modified wood gravity data.

variate outlier identification in $\boldsymbol{X}$ is to compute some measure of the distance of a particular data point from the center of the data and declare as outliers those points which are too far away from the center. Usually, as a measure of "outlyingness" for a data point $\boldsymbol{x}_i, i = 1, \ldots, n$, a robust version of the (squared) Mahalanobis distance RD_i^2 is used, computed relative to high breakdown point robust estimates of location $\boldsymbol{T}$ and covariance $\boldsymbol{C}$ of the data set $\boldsymbol{X}$:

$$RD_i^2 = (\boldsymbol{x}_i - \boldsymbol{T})^t \boldsymbol{C}^{-1} (\boldsymbol{x}_i - \boldsymbol{T}) \tag{2}$$

The most common estimators of multivariate location and scatter are the sample mean $\bar{\boldsymbol{x}}$ and the sample covariance matrix $\boldsymbol{S}$, i.e. the corresponding ML estimates (when the data follow a normal distribution). These estimates are optimal if the data come from a multivariate normal distribution but are extremely sensitive to the presence of even a few outliers in the data. The outlier identification procedure based on $\bar{\boldsymbol{x}}$ and $\boldsymbol{S}$ will suffer from the following two problems [52]:

1. *Masking:* multiple outliers can distort the classical estimates of mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{S}$ in such a way (attracting $\bar{\boldsymbol{x}}$ and inflating $\boldsymbol{S}$) that they do not get necessarily large values of the Mahalanobis distance, and
2. *Swamping:* multiple outliers can distort the classical estimates of mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{S}$ in such a way that observations which are consistent with the majority of the data get large values for the Mahalanobis distance.

The problems of masking and swamping are the reasons why diagnostic tools based on leave-one-out will not work: Leaving one observation out in turn is not protecting against the effects of multiple outliers.

The masking effect is illustrated in the right-hand part of Figure 2 in which the classical Mahalanobis distances are plotted against the observation index.

None of the outliers shows up above the cutoff value (represented by the dashed line) $\sqrt{\chi^2_{p,0.975}} = \sqrt{\chi^2_{5,0.975}} = 3.58$. If robust estimates of mean and covariance are used in Equation (2) the outliers lie clearly above the cutoff value as shown in the left panel of the same figure. These two plots can be combined in one as shown in Figure 3 to produce the distance-distance plot proposed by Rousseeuw and van Zomeren [54]. The right-hand side of Figure 3 shows a χ^2 quantile-quantile plot of the classical and robust Mahalanobis distances.

After having found reliable estimates for the location and covariance matrix of the data set, the second issue is to determine how large the robust distances should be in order to declare a point an outlier. The usual cutoff value is a quantile of the χ^2 distribution, like $D_0 = \chi^2_{p,0.975}$. The reason is that if $\boldsymbol{X}$ follows a multivariate normal distribution, the squared Mahalanobis distances based on the sample mean $\bar{\boldsymbol{x}}$ and sample covariance matrix $\boldsymbol{S}$ follow asymptotically a χ^2_p distribution [see for example 35, p. 189]. This will no more be valid if robust estimators are applied and/or if the data have other than multivariate normal distribution. In Maronna and Zamar [43] it was proposed to use a transformation of the cutoff value which should help the distribution of the squared robust distances RD_i^2 to resemble χ^2 for non-normal original data:

$$D_0 = \frac{\chi^2_{p,0.975} med(RD_1^2, \ldots, RD_n^2)}{\chi^2_{p,0.5}}. \tag{3}$$

For other alternatives which could lead to more accurate cutoff value see Filzmoser et al. [20], Hardin and Rocke [28], Cerioli et al. [9], Riani et al. [48].

There exist also other concepts for outlier detection, like methods based on the notion of depth, or projection based methods, see, e.g., Wilcox [68], and the R code therein. The choice for an appropriate outlier detection method can be based on the *outside rate per observation*, which refers to the expected proportion of observations declared outliers [see 68]. Under normality it can be desirable to choose an outlier detection method that has an outside rate per observation that is close to 0.05.

3. Multivariate location and scatter estimation

In the last several decades much effort was devoted to the development of affine equivariant estimators possessing a high breakdown point. The *affine equivariance* is a desirable property of a multivariate estimator, which makes the analysis independent of translations and rotations of the data. We say that a location estimator $\boldsymbol{T}$ of an $n \times p$ data set $\boldsymbol{X}$ is affine equivariant if

$$\boldsymbol{T}(\boldsymbol{X}\boldsymbol{A} + \mathbf{1}_n \boldsymbol{b}^t) = \boldsymbol{T}(\boldsymbol{X})\boldsymbol{A} + \boldsymbol{b} \tag{4}$$

for all p-dimensional vectors $\boldsymbol{b}$ and all nonsingular $p \times p$ matrices $\boldsymbol{A}$. The vector $\mathbf{1}_n$ is a column vector with all n components equal to 1. A scatter estimator $\boldsymbol{C}$

being a positive-definite symmetric $p \times p$ matrix is affine equivariant if

$$\boldsymbol{C}(\boldsymbol{X}\boldsymbol{A} + \mathbf{1}_n \boldsymbol{b}^t) = \boldsymbol{A}^t \boldsymbol{C}(\boldsymbol{X})\boldsymbol{A} \tag{5}$$

holds, again for all p-dimensional vectors $\boldsymbol{b}$ and all nonsingular $p \times p$ matrices $\boldsymbol{A}$. If an estimator is affine equivariant it transforms properly when the data are translated, rotated or the scale changes and thus the analysis is independent of the measurement scales of the variables or their translations or rotations. The affine equivariance is important also for other multivariate methods like discriminant analysis or canonical correlation analysis which are based on location and covariance estimates. In the case of principal component analysis (PCA) a weaker form of equivariance, namely *orthogonal equivariance* is sufficient. Orthogonal equivariance means that the estimator transforms properly under orthogonal transformations (but not affine transformations) of the data.

A performance measure for the robustness of an estimator is its *breakdown point*. The notion of breakdown point was introduced by Hampel [27] and later reformulated in a finite sample setting by Donoho and Huber [18]. The finite sample breakdown point for an estimator $\hat{\theta}$ at the data set $\mathbf{X}$ can be defined (roughly) as the largest proportion $\epsilon_n^*(\hat{\theta}_n, \mathbf{X})$ of data points that can be arbitrarily replaced by outliers without driving the estimator out of any bounds, i.e. making it useless. For an estimator of the location "useless" means that it approaches ∞ and for an estimator of a covariance matrix - that one of its eigenvalues approaches 0 or ∞. One bad point (outlier) is enough to ruin the sample mean or the sample covariance matrix and thus they have a breakdown point of $1/n$ which is the smallest possible breakdown point. Conversely, the sample median can tolerate up to 50% gross errors before it can be made arbitrarily large, therefore we say that its breakdown point is 50%. No affine equivariant estimator can have a breakdown point higher than 50% since it will become impossible to distinguish between the "good" and "bad" part of data. We are interested in *positive breakdown point* estimators, i.e. estimators with breakdown point $\epsilon_n^*(\hat{\theta}_n, \mathbf{X}) > 0$. Estimators with breakdown point 50% are called *high breakdown point estimators*.

A very important performance criteria of any statistical procedure is its *statistical efficiency* and robust estimators are known in general as not very efficient. One way to increase the statistical efficiency of a high breakdown point estimator is to sacrifice the maximal breakdown point of 50% and work with lower, say 25% which in most of the cases is quite reasonable.

All the desirable features of a robust estimator listed above are useless if the estimator cannot be computed in a reasonable amount of time, also in high dimensions and with large amount of data. Therefore the *computational feasibility* is one of the most important feature for the practical application of any estimator or procedure.

An early approach was that of M-estimation introduced by Maronna [39] which provides robust, affine equivariant and easy to compute estimates, but unfortunately these estimates have an unacceptably low breakdown point of $1/p$. Currently the most widely used estimators of this type are the Minimum Covariance Determinant (MCD) estimator and the Minimum Volume Ellipsoid (MVE) estimator, S-estimators and the Stahel-Donoho estimator (see later in this section). These estimators can be configured in such a way as to achieve the theoretically maximal possible breakdown point of 50% which gives them the ability to detect outliers even if their number is as much as almost half of the sample size. If we give up the requirement for affine equivariance, estimators like the orthogonalized Gnanadesikan-Kettenring (OGK) estimator are available and the reward is an extreme gain in speed. For definitions, algorithms and references to the original papers it is suitable to use Maronna et al. [41]. Most of these methods are implemented in the R statistical environment [47] and are available in the object-oriented framework for robust multivariate analysis [64].

3.1. Minimum covariance determinant (MCD) and Minimum volume (MVE) estimators

The MCD estimator for a data set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ in $\Re^p$ is defined by that subset $\{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_h}\}$ of h observations whose covariance matrix has the smallest determinant among all possible subsets of size h. The MCD location and scatter estimate $\mathbf{T}_{MCD}$ and $\mathbf{C}_{MCD}$ are then given as the arithmetic mean and a multiple of the sample covariance matrix of that subset

$$\begin{aligned} \mathbf{T}_{MCD} &= \frac{1}{h}\sum_{j=1}^{h} \boldsymbol{x}_{i_j} \\ \mathbf{C}_{MCD} &= c_{ccf}\, c_{sscf} \frac{1}{h-1}\sum_{j=1}^{h} (\boldsymbol{x}_{i_j} - \mathbf{T}_{MCD})(\boldsymbol{x}_{i_j} - \mathbf{T}_{MCD})^t. \end{aligned} \tag{6}$$

The multiplication factors c_{ccf} (consistency correction factor) and c_{sscf} (small sample correction factor) are selected so that $\mathbf{C}$ is consistent at the multivariate normal model and unbiased at small samples [see 6, 11, 46, 63]. Note, however, that only for a very small amount of trimming (h close to n), the proposed constant c_{ccf} allows to get a consistent estimator for the determinant and, therefore, for the whole covariance matrix. Otherwise, the use of this constant will produce overestimation. Since usually the amount of contamination in the data is unknown, a value like $h = [0.75\ n]$ is used in practice, resulting in an estimator with good statistical properties of the shape matrix, but not of the covariance matrix.

The breakdown point of the estimator is controlled by the parameter h. To achieve the maximal possible BP of the MCD the choice for h is $\lfloor (n + p + 1)/2 \rfloor$, but any integer h within the interval $[(n+p+1)/2, n]$ can be chosen, see Rousseeuw and Leroy [52]. Here $\lfloor z \rfloor$ denotes the integer part of z which is not less than z. If $h = n$ then the MCD location and scatter estimate $\mathbf{T}_{MCD}$ and $\mathbf{C}_{MCD}$ reduce to the sample mean and covariance matrix of the full data set.

The MCD estimator is not very efficient at normal models, especially if h is selected so that maximal BP is achieved. To overcome the low efficiency of the MCD estimator, a reweighed version can be used. For this purpose a weight w_i is assigned to each observation $\boldsymbol{x}_i$, defined as $w_i = 1$ if $(\boldsymbol{x}_i - \mathbf{T}_{MCD})^t \mathbf{C}_{MCD}^{-1} (\boldsymbol{x}_i - \mathbf{T}_{MCD}) \leq \chi^2_{p,0.975}$ and $w_i = 0$ otherwise, relative to the raw MCD estimates $(\mathbf{T}_{MCD}, \mathbf{C}_{MCD})$. Then the reweighted estimates are computed as

$$\begin{aligned} \mathbf{T}_R &= \frac{1}{\nu} \sum_{i=1}^{n} w_i \boldsymbol{x}_i, \\ \mathbf{C}_R &= c_{r.ccf} c_{r.sscf} \frac{1}{\nu - 1} \sum_{i=1}^{n} w_i (\boldsymbol{x}_i - \mathbf{T}_R)(\boldsymbol{x}_i - \mathbf{T}_R)^t, \end{aligned} \tag{7}$$

where ν is the sum of the weights, $\nu = \sum_{i=1}^{n} w_i$. Again, the multiplication factors $c_{r.ccf}$ and $c_{r.sscf}$ are selected so that $\mathbf{C}_R$ is consistent at the multivariate normal model and unbiased at small samples [see 46, 63, and the references therein]. The reweighted estimates $(\mathbf{T}_R, \mathbf{C}_R)$ have the same breakdown point as the initial (raw) *MCD* estimates but better statistical efficiency. The reweighted estimator should not be used when contaminating observations are close to the boundary of the regular observations, because the outliers could then get masked. The R function in package **rrcov** for computing MCD returns the reweighted estimates by default. The actual, "raw" MCD estimates, if necessary, can be obtained by the function `getRaw()`.

Depending on the particular statistical application at hand, we do not always need the maximal breakdown point. If we choose $h = [0.75\ n]$ we will have a breakdown point of 25% which provides sufficient robustness for most statistical applications. This will increase the statistical efficiency of the estimators [see 11].

The computation of the MCD estimator is far from being trivial. The naive algorithm would proceed by exhaustively investigating all subsets of size h out of n to find the subset with the smallest determinant of its covariance matrix, but this will be feasible only for very small data sets. Initially MCD was neglected in favor of MVE because the simple resampling algorithm was more efficient for MVE. Meanwhile several heuristic search algorithms [see 62, 69, 29] and exact algorithms [1] were proposed but now a very fast algorithm due to Rousseeuw and Van Driessen [53] exists and this algorithm is usually used in practice. The algorithm is based on the C-step which moves from one approximation $(\mathbf{T}_1, \mathbf{C}_1)$ of the MCD estimate of a data set $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ to the next one $(\mathbf{T}_2, \mathbf{C}_2)$ with possibly lower determinant $\det(\mathbf{C}_2) \leq \det(\mathbf{C}_1)$ by computing the distances $d_1, \ldots, d_n$ relative to $(\mathbf{T}_1, \mathbf{C}_1)$, i.e.,

$$d_i = \sqrt{(\boldsymbol{x}_i - \mathbf{T}_1)^t \mathbf{C}_1^{-1} (\boldsymbol{x}_i - \mathbf{T}_1)} \tag{8}$$

and then computing $(\mathbf{T}_2, \mathbf{C}_2)$ for those h observations which have smallest distances. "C" in C-step stands for "concentration" since we are looking for a more "concentrated" covariance matrix with lower determinant. Rousseeuw and

Van Driessen [53] have proven a theorem stating that the iteration process given by the C-step converges in a finite number of steps to a (local) minimum. Since there is no guarantee that the global minimum of the MCD objective function will be reached, the iteration must be started many times from different initial subsets, to obtain an approximate solution. The procedure is very fast for small data sets but to make it really "fast" also for large data sets several computational improvements are used.

1. *initial subsets*: It is possible to restart the iterations from randomly generated subsets of size h, but in order to increase the probability of drawing subsets without outliers, $p+1$ points are selected randomly. These $p+1$ points are used to compute $(\mathbf{T}_0, \mathbf{C}_0)$. Then the distances $d_1, \ldots, d_n$ are computed and sorted in increasing order. Finally the first h are selected to form the initial $h-$subset H_0.
2. *reduced number of C-steps*: The C-step involving the computation of the covariance matrix, its determinant and the relative distances, is the most computationally intensive part of the algorithm. Therefore instead of iterating to convergence for each initial subset only two C-steps are performed and the 10 subsets with lowest determinant are kept. Only these subsets are iterated to convergence.
3. *partitioning*: For large n the computation time of the algorithm increases mainly because all n distances given by Equation (8) have to be computed at each iteration. An improvement is to partition the data set into a maximum of say five subsets of approximately equal size (but not larger than say 300) and iterate in each subset separately. The ten best solutions for each data set are kept and finally only those are iterated on the complete data set.
4. *nesting*: Further decrease of the computational time can be achieved for data sets with n larger than say 1500 by drawing 1500 observations without replacement and performing the computations (including the partitioning) on this subset. Only the final iterations are carried out on the complete data set. The number of these iterations depends on the actual size of the data set at hand.

The minimum volume ellipsoid (MVE) estimator introduced by Rousseeuw [50] together with the MCD estimator searches for the ellipsoid of minimal volume containing at least half of the points in the data set $\mathbf{X}$. Then the location estimate is defined as the center of this ellipsoid and the covariance estimate is provided by its shape. Formally the estimate is defined as those $\mathbf{T}_{MVE}, \mathbf{C}_{MVE}$ that minimize $\det(\mathbf{C})$ subject to

$$\#\{i : (\boldsymbol{x}_i - \mathbf{T})^t \mathbf{C}^{-1} (\boldsymbol{x}_i - \mathbf{T}) \leq c^2\} \geq \left\lfloor \frac{n+p+1}{2} \right\rfloor, \tag{9}$$

where # denotes the cardinality. The constant c is chosen as $\chi^2_{p,0.5}$.

The search for the approximate solution is made over ellipsoids determined by the covariance matrix of $p+1$ of the data points and by applying a simple but

effective improvement of the sub-sampling procedure as described in Maronna et al. [41], p. 198. Although there exists no formal proof of this improvement (as for MCD and LTS), simulations show that it can be recommended as an approximation of the MVE. The MVE was the first popular high breakdown point estimator of location and scatter but later it was replaced by the MCD, mainly because of the availability of an efficient algorithm for its computation [53]. Recently the MVE gained importance as initial estimator for S estimation because of its small maximum bias [see 41, Table 6.2, p. 196]. The *maxbias* (maximum asymptotic bias) is a global robustness measure originally defined by Huber [30, 31] for the location model. Nowadays it is frequently used for *maxbias curves*, characterizing important robustness properties of an estimator [51].

3.2. The Stahel-Donoho estimator

The first multivariate equivariant estimator of location and scatter with high breakdown point was proposed by Stahel [59, 60] and Donoho [17] but became better known after the analysis of Maronna and Yohai [42]. For a data set $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ in $\Re^p$ it is defined as a weighted mean and covariance matrix of the form given by Equation (7) where the weight w_i of each observation is inverse proportional to the "outlyingness" of the observation. Let the univariate outlyingness of a point $\boldsymbol{x}_i$ with respect to the data set $\mathbf{X}$ along a vector $\mathbf{a} \in \Re^p, ||\mathbf{a}|| \neq \mathbf{0}$ be given by

$$r(\boldsymbol{x}_i, \mathbf{a}) = \frac{|\boldsymbol{x}^t\mathbf{a} - m(\mathbf{a}^t\mathbf{X})|}{s(\mathbf{a}^t\mathbf{X})} \quad i = 1, \ldots, n \tag{10}$$

where $(\mathbf{a}^t\mathbf{X})$ is the projection of the data set $\mathbf{X}$ on $\mathbf{a}$ and the functions $m()$ and $s()$ are robust univariate location and scale statistics, for example the median and MAD, respectively. When applying this idea to all possible univariate projections, one obtains in a natural way a measure for multivariate outlyingness of $\boldsymbol{x}_i$, which is defined by

$$r_i = r(\boldsymbol{x}_i) = \max_{\mathbf{a}} r(\boldsymbol{x}_i, \mathbf{a}). \tag{11}$$

The weights are computed by $w_i = w(r_i)$ where $w(r)$ is a nonincreasing function of r and $w(r)$ and $w(r)r^2$ are bounded. Maronna and Yohai [42] use the weights

$$w(r) = \min\left(1, \left(\frac{c}{t}\right)^2\right) \tag{12}$$

with $c = \sqrt{\chi^2_{p,\beta}}$ and $\beta = 0.95$, that are known in the literature as "Huber weights".

Exact computation of the estimator is not possible and an approximate solution is found by subsampling a large number of directions $\mathbf{a}$ and computing the outlyingness measures $r_i, i = 1, \ldots, n$ for them. For each subsample of p points the vector $\mathbf{a}$ is taken as the norm 1 vector orthogonal to the hyperplane spanned by these points. It has been shown by simulations [41] that one step reweighting does not improve the estimator.

3.3. Orthogonalized Gnanadesikan/Kettenring

The MCD estimator and all other known affine equivariant high-breakdown point estimates are solutions to a highly non-convex optimization problem and as such pose a serious computational challenge. Much faster estimates with high breakdown point can be computed if one gives up the requirements of affine equivariance of the covariance matrix. Such an algorithm was proposed by Maronna and Zamar [43] which is based on the very simple robust bivariate covariance estimator s_{jk} proposed by Gnanadesikan and Kettenring [25] and studied by Devlin et al. [16]. For a pair of random variables Y_j and Y_k and a standard deviation function $\sigma()$, s_{jk} is defined as

$$s_{jk} = \frac{1}{4}\left(\sigma\left(\frac{Y_j}{\sigma(Y_j)} + \frac{Y_k}{\sigma(Y_k)}\right)^2 - \sigma\left(\frac{Y_j}{\sigma(Y_j)} - \frac{Y_k}{\sigma(Y_k)}\right)^2\right). \tag{13}$$

If a robust function is chosen for $\sigma()$ then s_{jk} is also robust and an estimate of the covariance matrix can be obtained by computing each of its elements s_{jk} for each $j = 1, \ldots, p$ and $k = 1, \ldots, p$ using Equation (13). This estimator does not necessarily produce a positive definite matrix (although symmetric) and it is not affine equivariant. Maronna and Zamar [43] overcome the lack of positive definiteness by the following steps:

1. Define $\boldsymbol{y}_i = \boldsymbol{D}^{-1}\boldsymbol{x}_i, i = 1, \ldots, n$ with $\mathbf{D} = \text{diag}(\sigma(X_1), \ldots, \sigma(X_p))$ where $X_l, l = 1, \ldots, p$ are the columns of the data matrix $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. Thus a normalized data matrix $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ is computed.
2. Compute the matrix $\boldsymbol{U} = (u_{jk})$ as $u_{jk} = s_{jk} = s(Y_j, Y_k)$ if $j \neq k$ or $u_{jk} = 1$ otherwise. Here $Y_l, l = 1, \ldots, p$ are the columns of the transformed data matrix $\boldsymbol{Y}$ and $s(.,.)$ is a robust estimate of the covariance of two random variables like the one in Equation (13).
3. Obtain the "principal component decomposition" of $\boldsymbol{Y}$ by decomposing $\boldsymbol{U} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^t$ where $\boldsymbol{\Lambda}$ is a diagonal matrix $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p)$ with the eigenvalues λ_j of $\boldsymbol{U}$ and $\boldsymbol{E}$ is a matrix with columns the eigenvalues $\boldsymbol{e}_j$ of $\boldsymbol{U}$.
4. Define $\boldsymbol{z}_i = \boldsymbol{E}^t\boldsymbol{y}_i = \boldsymbol{E}^t\boldsymbol{D}^{-1}\boldsymbol{x}_i$ and $\boldsymbol{A} = \boldsymbol{D}\boldsymbol{E}$. Then the estimator of $\boldsymbol{\Sigma}$ is $\boldsymbol{C}_{OGK} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^t$ where $\boldsymbol{\Gamma} = \text{diag}(\sigma(Z_j)^2), j = 1, \ldots, p$ and the location estimator is $\boldsymbol{T}_{OGK} = \boldsymbol{A}\boldsymbol{m}$ where $\boldsymbol{m} = m(\boldsymbol{z}_i) = (m(Z_1), \ldots, m(Z_p))$ is a robust mean function.

This can be iterated by computing $\boldsymbol{C}_{OGK}$ and $\boldsymbol{T}_{OGK}$ for $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ obtained in the last step of the procedure and then transforming back to the original coordinate system. Simulations [43] show that iterations beyond the second did not lead to improvement.

Similarly as for the MCD estimator a one-step reweighting can be performed using Equations (7) but the weights w_i are based on the 0.9 quantile of the χ^2_p distribution (instead of 0.975) and the correction factors $c_r.ccf$ and $c_r.sscf$ are not used.

In order to complete the algorithm we need a robust and efficient location function $m()$ and scale function $\sigma()$, and one proposal is given in Maronna and

Zamar [43]. Further, the robust estimate of covariance between two random vectors $s()$ given by Equation (13) can be replaced by another one. This *OGK* algorithm preserves the positive definiteness of the covariance matrix and is "almost affine equivariant". Even faster versions of this algorithm were proposed by Alqallaf et al. [3].

3.4. S estimates and MM estimates

S estimators of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ were introduced by Davies [15] and further studied by Lopuhaä [38] [see also 52, p. 263]. For a data set of p-variate observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ an S estimate $(\mathbf{T}, \mathbf{C})$ is defined as the solution of $\sigma(d_1, \ldots, d_n) = \min$ where $d_i = (\boldsymbol{x} - \mathbf{T})^t \mathbf{C}^{-1} (\boldsymbol{x} - \mathbf{T})$ and $\det(\mathbf{C}) = 1$. Here $\sigma = \sigma(\boldsymbol{z})$ is the M-scale estimate of a data set $\boldsymbol{z} = \{z_1, \ldots, z_n\}$ defined as the solution of $\frac{1}{n}\Sigma\rho(z/\sigma) = \delta$ where ρ is nondecreasing, $\rho(0) = 0$ and $\rho(\infty) = 1$ and $\delta \in (0, 1)$. An equivalent definition is to find the vector $\mathbf{T}$ and a positive definite symmetric matrix $\mathbf{C}$ that minimizes $\det(\mathbf{C})$ subject to

$$\frac{1}{n} \sum_{i=1}^{n} \rho(d_i) = b_0 \tag{14}$$

where d_i and ρ are defined as above and the constant b_0 is chosen for consistency of the estimator.

As shown by Lopuhaä [38] S estimators have a close connection to the M estimators and the solution $(\mathbf{T}, \mathbf{C})$ is also a solution to an equation defining an M estimator as well as a weighted sample mean and covariance matrix:

$$\begin{aligned} d_i^j &= [(\boldsymbol{x}_i - \mathbf{T}^{(j-1)})^t (\mathbf{C}^{(j-1)})^{-1} (\boldsymbol{x} - \mathbf{T}^{(j-1)})]^{1/2} \\ \mathbf{T}^{(j)} &= \frac{\Sigma w(d_i^{(j)}) \boldsymbol{x}_i}{\Sigma w(d_i^{(j)})} \\ \mathbf{C}^{(j)} &= \frac{\Sigma w(d_i^{(j)})(\boldsymbol{x}_i - \mathbf{T}^{(j)})(\boldsymbol{x}_i - \mathbf{T}^{(j)})^t}{\Sigma w(d_i^{(j)})} \end{aligned} \tag{15}$$

There are several algorithms for computing the S estimates: (i) *SURREAL* proposed by Ruppert [55] as an analog to the algorithm proposed by the same author for computing S estimators of regression; (ii)*Bisquare S estimation with HBDP start*: as described in Maronna et al. [41]; (iii) *Rocke type S estimates* [49] and (iv) *Fast S estimates* proposed by Salibian-Barrera and Yohai [57]. For more details about the computation of these estimates see Todorov and Filzmoser [64]. Rocke [49] warns that when using S estimators in high dimension, they can fail to reject as outliers points that are large distances from the main mass of points, although attaining a breakdown point approaching 50%,

Tatsuoka and Tyler [61] introduced the multivariate MM estimators together with the broader class of estimators which they call "multivariate M-estimators with auxiliary scale". They estimate the scale by means of a very robust S estimator, and then estimate the location and covariance using a different ρ-function

with better efficiency at the normal model. The location and covariance estimates inherit the breakdown point of the auxiliary scale and can be seen as a generalization of the regression MM estimators of Yohai [70].

4. Outlier detection in high dimensions

High-dimensional data are typical in many contemporary applications in scientific areas like genetics, spectral analysis, data mining, image processing, etc. The advances of the computer and information science have enabled the researchers to store huge amounts of data acquired usually automatically. High dimensional data introduce new challenges to the traditional analytical methods. First of all the computational effort for the anyway computationally intensive robust algorithms increases with increasing n and p towards the limits of feasibility. The computational time of projection pursuit algorithms like Stahel-Donoho estimator increases very rapidly with higher dimensions. Even the much faster pairwise algorithms like OGK have their limits.

In the following, some methods for high-dimensional outlier detection are listed that are implemented in R. There is a lot of activity in this relatively young field of science. Another promising proposal is from Alqallaf et al. [4] on the propagation of outliers, corresponding to the bad performance of standard high-breakdown affine equivariant estimators under componentwise contamination.

4.1. The PCDist Algorithm

Shieh and Hung [58] considered the problem of outlier detection in the context of multi-class microarray data which contains gene expression levels for a number of samples each of which is assigned to a biological class (e.g. presence or absence of tumor, tumor type, etc). In this case we can have two types of outliers - either samples which were misclassified (assigned to the wrong class) or samples that do not belong to any of the classes present in the data (due to poor class definition, undiscovered biological class, etc.). The first type are referred to as mislabeled samples and are relatively easily discovered by classification methods and the second they call abnormal samples. The proposed method iterates through the classes present in the data, separates each class from the rest and identifies the outliers relative to this class, thus treating both types of outliers, the mislabeled and the abnormal samples in a homogenous way. The algorithm can be summarized as follows:

1. Dimension reduction. Since microarray data usually contain a large number of genes p, much larger than the number of observations n, the usual low-dimensional methods described in Section 2 will not work. Therefore the first step of the algorithm is, ignoring the class structure, to reduce the dimensionality by classical PCA which is the most widely used unsupervised dimension reduction method.
2. Selecting the number of principal components to use. We expect that only a few k components are sufficient to explain most of the variability of the data. For selecting the optimal number of components Shieh and

Hung [58] use an automatic selection method proposed by Zhu and Ghodsi [71] which assumes that the distribution of the variances changes at the elbow point k of the scree plot of the principal components and finds this elbow point by maximizing the corresponding profile log-likelihood function. This selection method is fast and has been shown to perform well in many practical situations. Alternatively one could select the number of components which explain a given fraction of the variance (e.g. 95%).

3. For each class j in the selected low dimensional space:
   - Compute highly robust multivariate location and covariance matrix $(\mathbf{T}, \mathbf{C})$ using S estimators, see Section 3.4.
   - Compute robust distances RD_i^2 using Equation (2)
   - Compare these robust distances to the threshold D_0 given by Equation (3) and declare outliers in class j those observations for which $RD_i^2 > D_0$
4. Pool the j lists of outliers together and report them.

The effectiveness of this simple outlier detection method based on PCA and robust estimation was demonstrated on real data sets and a limited simulation study. A possible improvement could be to use supervised dimension reduction in the first step, i.e. partial least squares (PLS) instead of PCA. Another possible improvement is the use of robust PCA for dimension reduction.

4.2. The PCOut Algorithm

The *PCOut* algorithm proposed by Filzmoser et al. [21] combines two complementary measures of outlyingness in two subsequent steps. In the first step the target are the location outliers (mean-shift outliers, described by a different location vector) and in the second step the aim is to detect scatter outliers (variance inflation/deflation outliers, which possess a different scatter matrix than the rest of the data). The algorithm thus provides a final score that allows the ranking of the observations according to their deviation from the bulk of the data. A brief sketch of the algorithm will be presented in the following sections, all details and many examples are available in the original paper.

4.2.1. Preprocessing

The algorithm starts by several preprocessing steps, the first of which is robust rescaling each component by the coordinate-wise median and MAD,

$$x_{ij}^* = \frac{x_{ij} - med(x_{1j}, \ldots, x_{nj})}{MAD(x_{1j}, \ldots, x_{nj})}, j = 1, \ldots, p. \tag{16}$$

In order to be able to perform this rescaling it is necessary either to omit the dimensions with MAD equal to zero or to use another measure. From this rescaled data the covariance matrix is calculated and eigen-decomposition is performed which results in a semi robust PCA. Only those p^* components which amount to at least 99% of the total variance (see Section 6 for details on PCA)

are retained. Skipping out the components which contribute only useless noise, a representation in a lower dimensional p^* space, $p^* < p$ is obtained. This step solves also the problem with $p \gg n$ since we can select $p^* < n$. This decomposition can be represented by

$$\mathbf{Z} = \mathbf{X}^*\mathbf{V} \tag{17}$$

where $\mathbf{V}$ is the matrix of eigenvectors and $\mathbf{X}^*$ is the matrix with components x^*_{ij}.

These principal components are rescaled by the median and the MAD similarly as above,

$$z^*_{ij} = \frac{z_{ij} - med(z_{1j}, \ldots, z_{nj})}{MAD(z_{1j}, \ldots, z_{nj})}, j = 1, \ldots, p^*. \tag{18}$$

The resulting matrix $\mathbf{Z}^* = (z^*_{ij})$ is the input for the next two steps of the algorithm.

4.2.2. Detection of location outliers

The detection of location outliers starts by calculation of componentwise robust kurtosis measure according to:

$$w_j = \left| \frac{1}{n} \sum_{i=1}^{n} \frac{(z^*_{ij} - med(z^*_{1j}, \ldots, z^*_{nj}))^4}{MAD(z^*_{1j}, \ldots, z^*_{nj})^4} - 3 \right|, j = 1, \ldots, p^*. \tag{19}$$

where z^*_{ij} are the rescaled principal components from Equation (18). Equation (19) assigns higher weights to the components where outliers clearly stand out. If no outliers are present in a given component then the principal component is expected to be approximately normally distributed and the kurtosis will be close to zero. Therefore each dimension $q = 1, \ldots, p^*$ is weighted proportionally to the absolute value of the kurtosis given by Equation (19). Since the components are uncorrelated it is possible to calculate a robust Mahalanobis distance utilizing the distance from the median (as scaled by MAD):

$$RD_i = \sqrt{\sum_{j=1}^{p^*} (z^*_{ij} w^*_j)^2}. \tag{20}$$

which then are translated to

$$d_i = RD_i \frac{\sqrt{\chi^2_{p^*,0.5}}}{med(RD_1, \ldots, RD_n)} \text{ for } i = 1, \ldots, n \tag{21}$$

in order to bring the empirical distances $\{d_i\}$ closer to $\chi^2_{p^*}$. These distances are used in the translated biweight function [49] to assign weights to each observation. These weights are used as a measure of outlyingness. Filzmoser et al. [21]

claim that the translated biweight function (shown below) has certain advantages over other weighting schemes they have experimented with. The weights for each observation are calculated as follows

$$w_i = \begin{cases} 0, & d_i \geq c \\ \left(1 - \left(\frac{d_i - M}{c - M}\right)^2\right)^2, & M < d_i < c, \\ 1, & d_i \leq M \end{cases} \tag{22}$$

where $i = 1, \ldots, n$ and c is given by

$$c = med(d_1, \ldots, d_n) + 2.5.MAD(d_1, \ldots, d_n). \tag{23}$$

M is found by sorting the distances $\{d_1, \ldots, d_n\}$ in ascending order and taking M equal to the distance at position $\lfloor n/3 \rfloor$. These weights $w_{1i}, i = 1, \ldots, n$ are kept to combine with the result from step two to obtain the final weights.

4.2.3. Detection of scatter outliers

The scatter outliers are searched for in the space defined by $\mathbf{Z}^*$ in Equation (18) by calculating the Euclidean norm for data in principal component space (which is equivalent to the Mahalanobis distance in the original data space but is much faster to compute). The weights $w_{2i}, i = 1, \ldots, n$ of the second step are calculated using again the translated biweight function from Equation (22) and setting $M^2 = \chi^2_{p^*,0.25}$ and $c = \chi^2_{p^*,0.99}$.

4.2.4. Computation of final weights

The weights resulting from the two phases of the algorithm are combined into final weights according to

$$w_i = \frac{(w_{1i} + s)(w_{2i} + s)}{(1 + s)^2}, i = 1, \ldots, n. \tag{24}$$

where typically $s = 0.25$. Outliers are then identified as points having weights $w_i < 0.25$.

5. Statistical Design Patterns and computing in R

The routine use of the robust methods in a wide area of application domains is unthinkable without the computational power of today's personal computers and the availability of ready to use implementations of the algorithms. Several commercial statistical software packages currently include some basic robust procedures like *MCD*, *LTS* and *MM*-regression(S-Plus, SAS). The toolbox Libra for Matlab [67] features many of the robust multivariate statistical methods described here. Recently several robust algorithms were implemented also in the statistical package Stata. However, the most complete set of procedures for robust statistical analysis is available in the R programming environment. Several

packages including **robustbase**, **rrcov**, **rrcovHD** and **rrcovNA** contribute to fill the gap between theoretically developed robust methods and their availability in standard statistical software. This allows both basic and advanced methods to be used by a broad range of researchers. Todorov and Filzmoser [64] provide an implementation of an object oriented framework for multivariate analysis in which the object oriented programming paradigm is leveraged to create a unique user friendly and extendable tool covering multivariate location and scatter estimation, outlier detection, principal component analysis, linear and quadratic discriminant analysis, and multivariate tests.

In classical multivariate statistics we rely on parametric models based on assumptions about the structural and the stochastic parts of the model for which optimal procedures are derived, like the least squares estimators and the maximum likelihood estimators. The corresponding robust methods can be seen as extensions to the classical ones which can cope with deviations from the stochastic assumptions thus mitigating the dangers for classical estimators. The developed statistical procedures will remain reliable and reasonably efficient even when such deviations are present. For example in the case of location and covariance estimation the classical theory yields the sample mean $\bar{\boldsymbol{x}}$ and the sample covariance matrix $\mathbf{S}$, i.e., the corresponding MLE estimates as an optimal solution. One (out of many) robust alternatives is the minimum covariance determinant estimator. When we consider this situation from an object oriented design point of view we can think of an abstract base class representing the estimation problem, a concrete realization of this object—the classical estimates, and a second concrete derivative of the base class representing the MCD estimates. Since there exist many other robust estimators of multivariate location and covariance which share common characteristics we would prefer to add one more level of abstraction by defining an abstract "robust" object from which all other robust estimators are derived. We encounter a similar pattern in most of the other multivariate statistical methods like principal component analysis, linear and quadratic discriminant analysis, etc. and we will call it a *statistical design pattern*. Design patterns are usually defined as general solutions to recurring design problems and refer to both the description of a solution and an instance of that solution solving a particular problem. The current use of the term design patterns originates in the writings of the architect Christopher Alexander devoted to urban planning and building architecture [2] but it was brought to the software development community by the seminal book of Gamma et al. [24]. A design pattern can be seen as a template for how to solve a problem which can be used in many different situations. Object-oriented design patterns are about classes and the relationships between classes or objects at abstract level, without defining the final classes or objects of the particular application. In order to be usable, design patterns must be defined formally and the documentation, including a preferably evocative name, describes the context in which the pattern is used, the pattern structure, the participants and collaboration, thus presenting the suggested solution. Design patterns are not limited to architecture or software development but can be applied in any domain where solutions are searched for.

In the framework presented here design patterns are used to support the usage, experimentation, development and testing of robust multivariate methods as well as to simplify the comparisons with related methods. It is intended to reduce the effort for performing any of the following tasks: (i) application of the already existing robust multivariate methods for practical data analysis; (ii) implementation of new robust multivariate methods or variants of the existing ones; (iii) evaluation of existing and new methods by carrying out comparison studies. Software design patterns are usually modeled and documented in natural languages and visual languages, such as the *Unified Modeling Language (UML)* [7]. UML is a standardized general-purpose language used to specify, visualize, construct and document the artifacts of an object-oriented software system. Figure 5 presents visually, as an *UML class diagram*, the example of classical and robust multivariate location and covariance estimation which was described above. A class corresponds to an algorithm and the diagram shows the classes, their attributes, and the relationships among the classes.

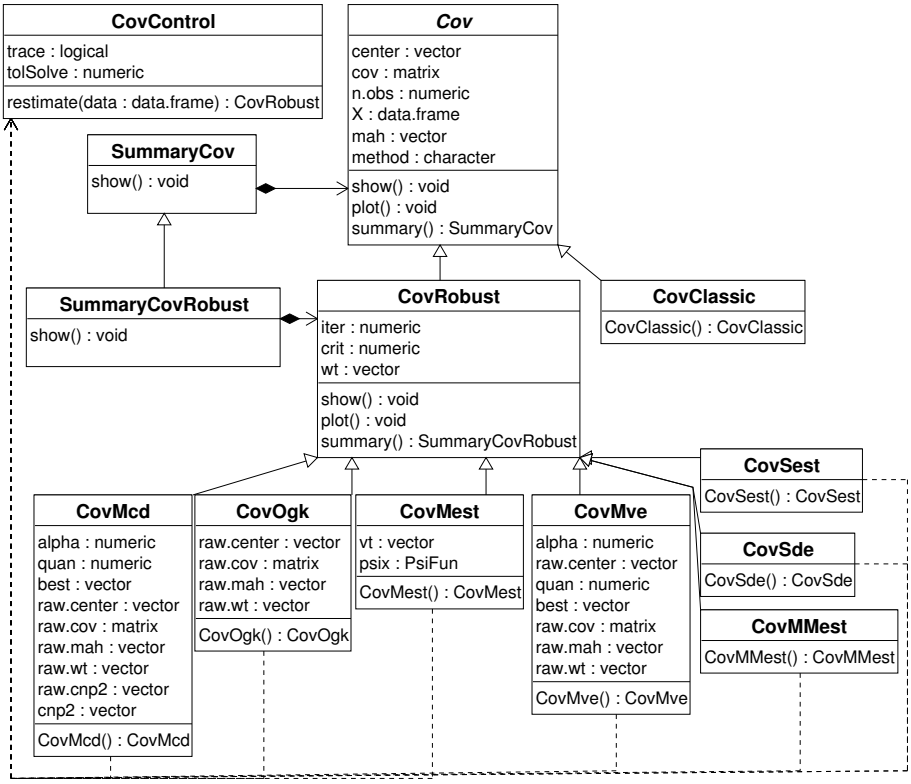

Figure 4: Class diagram of the statistical design pattern for robust estimation of multivariate location and covariance matrix methods.

Currently the framework spreads over the packages **rrcov**, **robustbase**, **rrcovNA** and **rrcovHD** and is used by many other packages extending its functionality.

6. Principal Component Analysis

Principal component analysis (PCA) is a widely used technique for dimension reduction achieved by finding a smaller number k of linear combinations of the originally observed p variables and retaining most of the variability of the data. These new variables, referred to as *principal components* are uncorrelated with each other and account for decreasing amount of the total variance, i.e. the first principal component explains the maximum variance in the data, the second principal component explains the maximum variance in the data that has not been explained by the first principal component and so on. Dimension reduction by PCA is mainly used for: (i) visualization of multivariate data by scatter plots (in a lower dimensional space); (ii) transformation of highly correlated variables into a smaller set of uncorrelated variables which can be used by other methods (e.g. multiple or multivariate regression); (iii) combination of several variables characterizing a given process into a single or a few *characteristic* variables or *indicators*.

The classical approach to PCA measures the variability through the empirical variance and is essentially based on computation of eigenvalues and eigenvectors of the sample covariance or correlation matrix. Therefore the results may be extremely sensitive to the presence of even a few atypical observations in the data. The outliers could artificially increase the variance in an otherwise uninformative direction and this direction will be determined as a PC direction. These discrepancies will carry over to any subsequent analysis and to any graphical display related to the principal components such as the biplot.

Consider an $n \times p$ data matrix $\boldsymbol{X}$. Further, $\boldsymbol{m}$ denotes the (robust) center of the data and $\mathbf{1}_n$ is a column vector with all n components equal to 1. We are looking for linear combinations $\boldsymbol{t}_j$ that result from a projection of the centered data on a direction $\boldsymbol{p}_j$,

$$\boldsymbol{t}_j = (\boldsymbol{X} - \mathbf{1}_n \boldsymbol{m}^t)\boldsymbol{p}_j \tag{25}$$

such that

$$\boldsymbol{p}_j = \underset{\boldsymbol{p}}{\operatorname{argmax}} \operatorname{Var}(\boldsymbol{X}\boldsymbol{p}) \tag{26}$$

subject to $\|\boldsymbol{p}_j\| = 1$ and $\operatorname{Cov}(\boldsymbol{X}\boldsymbol{p}_j, \boldsymbol{X}\boldsymbol{p}_l) = 0$ for $l < j$ and $j = 1, \ldots, k$ with $k \leq \min(n, p)$. The solutions of these maximization problems are obtained by solving a Lagrangian problem, and the result is that the principal components of $\boldsymbol{X}$ are the eigenvectors of the covariance matrix $\operatorname{Cov}(\boldsymbol{X})$, and the variances are the corresponding eigenvalues $l_j = \operatorname{Var}(\boldsymbol{X}\boldsymbol{p}_j)$. Classical PCA is obtained if the sample covariance matrix $\boldsymbol{S}$ given by Equation (1) is used for "Cov". PCA based on robust covariance estimation will be discussed in Section 6.1. Here, not only "Cov" but also the data center $\boldsymbol{m}$ need to be estimated robustly.

Usually the eigenvectors are sorted in decreasing order of the eigenvalues and hence the first k principal components are the most important ones in terms of explained variance. Finally, the vectors $\boldsymbol{t}_j$ are collected as columns in the $n \times k$ *scores* matrix $\boldsymbol{T}$, and the vectors $\boldsymbol{p}_j$ as columns in the *loadings* matrix $\boldsymbol{P}$. The eigenvalues l_j are arranged in the diagonal of the $k \times k$ diagonal matrix $\boldsymbol{\Lambda}$. This allows to represent the covariance matrix as

$$\mathrm{Cov}(\boldsymbol{X}) = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^t. \tag{27}$$

The original $\boldsymbol{X}$ matrix can be reconstructed from the scores $\boldsymbol{T}$ in the original coordinate system (using k principal components) preserving the main structure of the data:

$$\boldsymbol{X} = \mathbf{1}\boldsymbol{m}^t + \boldsymbol{T}\boldsymbol{P}^t + \boldsymbol{E}, \tag{28}$$

where the error or residual matrix $\boldsymbol{E}$ will be zero if all principal components are used.

PCA was probably the first multivariate technique subjected to robustification, either by simply computing the eigenvalues and eigenvectors of a robust estimate of the covariance matrix or directly by estimating each principal component in a robust manner. The different approaches to robust PCA are presented in the next sections and examples are given how these robust analyses can be carried out in R. Details about the methods and algorithms can be found in the corresponding references.

6.1. PCA based on robust covariance matrix (MCD, OGK, MVE, etc.)

The most straightforward and intuitive method to obtain robust PCA is to replace the classical estimates of location and covariance by their robust analogues. In the earlier works M estimators of location and scatter were used for this purpose [see 16, 8] but these estimators have the disadvantage of low breakdown point in high dimensions. To cope with this problem Naga and Antille [44] used the MVE estimator and Todorov et al. [65] used the MCD estimator. Croux and Haesbroeck [12] investigated the properties of the MCD estimator and computed its influence function and efficiency.

6.2. PCA based on projection pursuit

The second approach to robust PCA uses *projection pursuit* (PP) and calculates directly the robust estimates of the eigenvalues and eigenvectors without passing by a robust covariance estimation. Directions are sought for, which maximize the variance of the data projected onto them. The advantage of this approach is that the principal components can be computed sequentially, and that one can stop after $k < p$ components have been extracted. Thus, this approach is appealing for high-dimensional data, in particular for problems with $p > n$.

Using the empirical variance in the maximization problem would lead to classical PCA, and robust scale estimators result in robust PCA. Such a method was first introduced by Li and Chen [36] using an M estimator of scale. They

showed that the PCA estimates inherit the robustness properties of the scale estimator. Unfortunately, in spite of the good statistical properties of the method, the algorithm they proposed was too complicated to be used in practice. A more tractable algorithm in these lines was proposed by Croux and Ruiz-Gazen [13], and they also completed the theoretical results of Li and Chen [36] by computing the influence functions of the estimators of the eigenvalues, eigenvectors and associated covariance matrix as well as by computing the asymptotic variances.

The first step of the algorithm is centering of the data by subtracting the centers of the variables from the columns of the data matrix. A multivariate generalization of the median, the so called geometric median (also known as spatial median or L_1–median) is used for this purpose. For a $n \times p$ data set $\boldsymbol{X}$ the L_1–median $\hat{\boldsymbol{\mu}}$ is defined as:

$$\hat{\boldsymbol{\mu}} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{\mu}|| \tag{29}$$

where $||.||$ denotes the Euclidean norm. It is fast to compute and has a 50 % breakdown point [see e.g. 22]. When solving the maximization problem, the algorithm does not investigate all possible directions but considers only those defined by a data point and the robust center of the data. The robust variance estimate is computed for the data points projected on these n directions and the direction corresponding to the maximum of the variance is the searched approximation of the first principal component. After that the search continues in the same way in the space orthogonal to the first component. An improved version of this algorithm, being more precise especially for high-dimensional data, was proposed by Croux et al. [10]. The space of all possible directions is scanned more thoroughly. This is done by restricting the search for an optimal direction on a regular grid in a plane.

The PCA projection pursuit algorithms Croux and Ruiz-Gazen [13] and Croux et al. [10] are represented in R by the classes `PcaProj` and `PcaGrid`, respectively. Their generating functions provide simple wrappers around the original functions from the package **pcaPP** and return objects of the corresponding class, derived from `PcaRobust`.

```
> pc <- PcaGrid(X, k=2, scale=mad)
>     # k=2 PCs are computed, MAD is the robust scale measure
> P  <- getLoadings(pc)        # robust PCA loadings
> T  <- getScores(pc)          # robust PCA scores
```

6.3. The method ROBPCA

This robust PCA method proposed by Hubert et al. [32] tries to combine the advantages of both approaches—PCA based on a robust covariance matrix and PCA based on projection pursuit. A brief description of the algorithm follows, for details see the relevant references [33]. First SVD is applied to express the information in the n-dimensional space (useful if $p > n$). Then for each observation a measure of "outlyingness" is computed. The h data points

with smallest outlyingness measure are used to compute the robust covariance matrix and to select the number k of principal components to retain. With an eigen-decomposition of this covariance matrix, the space spanned by the first k eigenvectors is used to project all data points. Finally, location and covariance of the projected data are computed using the reweighted MCD estimator, and the eigenvectors of this scatter matrix yield the robust principal components.

The algorithm ROBCA is implemented in the R package (rrcov) as the function `PcaHubert()`.

6.4. Spherical principal components (SPC)

The spherical principal components procedure was first proposed by Locantore et al. [37] as a method for functional data analysis. The idea is to perform classical PCA on the data, projected onto a unit sphere. The estimates of the eigenvectors are consistent if the data are elliptically distributed [see 5] and the procedure is extremely fast. Although not much is known about the efficiency of this method, the simulations of Maronna [40] show that it has very good performance. If each coordinate of the data is normalized using some kind of robust scale, like for example the MAD, and then SPC is applied, we obtain "elliptical PCA", but unfortunately this procedure is not consistent.

6.5. Example: Robust PCA in R

Hubert and Van Driessen [34] used a data set containing the spectra of three different cultivars of the same fruit. The three cultivars (groups) are named D, M and HA, and their sample sizes are 490, 106 and 500 observations, respectively. The spectra are measured at 256 wavelengths. The fruit data is thus a high-dimensional data set which was used by Hubert and Van Driessen [34] to illustrate a new approach for robust linear discriminant analysis, and it was studied again by Vanden Branden and Hubert [66]. From these studies it is known that the first two cultivars D and M are relatively homogenous and do not contain atypical observations, but the third group HA contains a subgroup of 180 observations which were obtained with a different illumination system. For our example illustrating robust PCA we will use only the HA group with 500 observations.

We assume that the package **rrcov** has been loaded and that `X` contains the data. Depending on the PCA algorithm, $k = 2$ to $k = 4$ components explain more than 99% of the data variability. For reasons of comparability we use $k = 4$ for all methods.

```
> pcC <- PcaClassic(X, k=4)              # classical PCA
> pcG <- PcaGrid(X, k=4)                 # see Section 6.2
> pcH <- PcaHubert(X, k=4, alpha=0.5)    # see Section 6.3
> pcL <- PcaLocantore(X, k=4)            # see Section 6.4
```

Details of the PCA results can be seen with the function `summary()` applied to the result objects. A scatter plot of the first two PCA scores (first two columns of $\mathbf{T}$) can be seen with

```
> pca.scoreplot(pcC,main="(a) Classical PCA")
> pca.scoreplot(pcG,main="(b) PCA based on PP")
> pca.scoreplot(pcH,main="(c) ROBPCA")
> pca.scoreplot(pcL,main="(d) Spherical PCA")
```

and the results are shown in Figure 5. All PCA methods show that there are two data groups, but only the robust methods identify one of the groups as outliers.

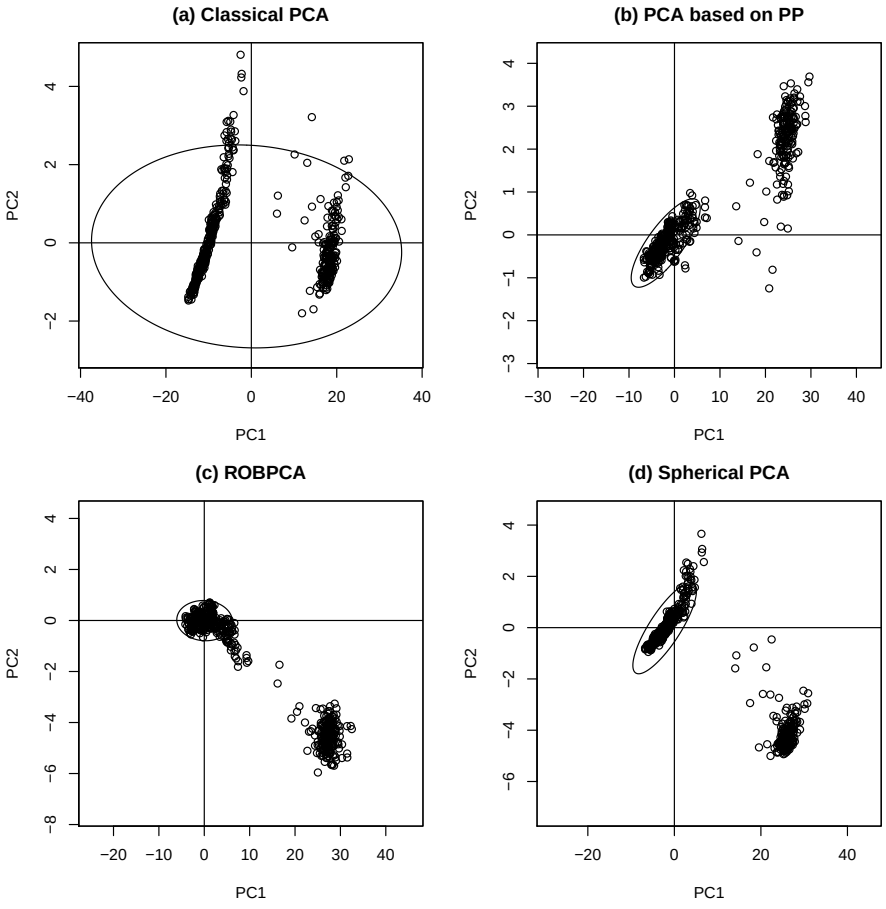

Figure 5: Plots of the first two PCA scores for classical PCA (a), and robust PCA based on projection pursuit (b), for the ROBCA algorithm (c), and for spherical PCA (d).

The *diagnostic plot* proposed by [32] which is based on the *score distances* and *orthogonal distances* computed for each observation is especially useful for identifying outlying observations. The *score distance* is defined by

$$SD_i = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{l_j}}, \quad i = 1, \ldots, n, \tag{30}$$

where t_{ij} are the elements of the score matrix $\boldsymbol{T}$. It is a Mahalanobis-like measure of distance of an observation within the k-dimensional principal component space. The *orthogonal distance* is defined by

$$OD_i = ||\boldsymbol{x}_i - \boldsymbol{m} - \boldsymbol{P}\boldsymbol{t}_i||, \quad i = 1, \ldots, n \tag{31}$$

where $\boldsymbol{t}_i$ is the ith row of the score matrix $\boldsymbol{T}$ and m is the estimated center of the data. This measure corresponds to the distance between the observation and its projection into the space spanned by the first k principal components. The *diagnostic plot* shows the orthogonal distance versus the score distance, and indicates with a horizontal and vertical line the cut-off values that allow to distinguish regular observations from the two types of outliers [for details, see 32].

The diagnostic plots for the resulting PCA objects are shown by

```
> plot(pcC,main="(a) Classical PCA")
> plot(pcG,main="(b) PCA based on PP")
> plot(pcH,main="(c) ROBPCA")
> plot(pcL,main="(d) Spherical PCA")
```

which gives the plots in Figure 6. It can be seen that classical PCA (a) shows some outliers, but also regular observations are declared as outliers. In addition, there is no clear grouping structure visible. The robust PCA methods all show two groups and in addition some deviating data points. PCA based on PP using the algorithm of Croux et al. [10] clearly flags the group with the different detector efficiency as outliers in terms of both the orthogonal and the score distance and ROBPCA finds almost the same answer. For the result of spherical PCA, the score distance is reliable but the orthogonal distance is misleading.

7. Conclusions

Robust statistical methods are available nowadays in several software packages. R has the most comprehensive collection of tools for robust estimation. Following certain statistical design patterns (Section 5), some of the contributed R packages for robustness have a unified design structure, as well as unified input/output structure. This should make it easier for the user to apply robust methods to real data sets and for the developer to experiment with and implement new algorithms as well as to maintain the software.

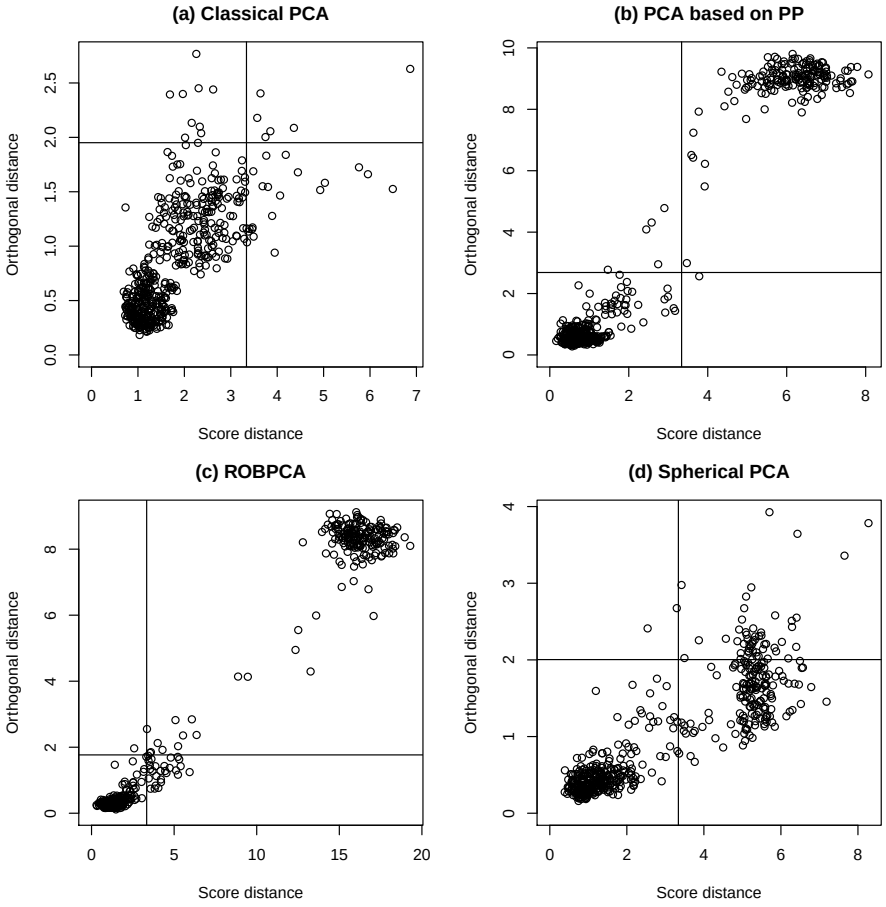

Figure 6: Diagnostic plots for classical PCA (a), and robust PCA based on projection pursuit (b), for the ROBCA algorithm (c), and for spherical PCA (d).

Various robust estimators for multivariate location and covariance have been implemented in the package **rrcov** [see 64]. The appropriate choice of an algorithm mainly depends on the data structure (e.g. more variables than observations), but also on the theoretical robustness properties of the corresponding estimator. Since many multivariate statistical methods are based on location and covariance estimation, it is straightforward to robustify these methods by plugging in the robust estimators. In Section 6 we mentioned this approach in the context of robust PCA. There are, however, other approaches to robust PCA, like the projection-pursuit approach, which has the advantage that the components are computed sequentially. This allows to stop after a desired number of extracted components, which is very useful in case of high-dimensional data (see Section 6).

There exist many more robust statistical methods and implementations in R,

like robust regression (**robustbase**), robust discriminant analysis (**rrcov**), multivariate inference by robust bootstrap by Salibian-Barrera et al. [56] (**FRB**), robust methods for analyzing compositional data (**robCompositions**), which are not treated here because of space limitations. The general philosophy of robust estimation is the same: The model is estimated for the data majority satisfying the model, and the influence of deviating data points on the estimation is reduced. Maronna et al. [41] discuss the most important robust methods and estimators.

References

[1] J.J. Agulló, in: A. Prat (Ed.), Proceedings in Computational Statistics, COMPSTAT'96, Physica Verlag, Heidelberg, 1996, pp. 175–180.

[2] C. Alexander, S. Ishikawa, M. Silverstein, A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series), Oxford University Press, 1977.

[3] F.A. Alqallaf, K.P. Konis, R.D. Martin, R.H. Zamar, in: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, 2002, pp. 14–23.

[4] F. Alqallaf, S. Van Aelst, V.J. Yohai, R.H. Zamar, The Annals of Statistics 37 (2009) 311–331.

[5] G. Boente, R. Fraiman, Test 8 (1999) 1–28.

[6] R. Butler, P. Davies, M. Jhun, The Annals of Statistics 21 (1993) 1385–1401.

[7] G. Booch, J. Rumbaugh, I. Jacobson, The unified modeling language user guide, Addison-Wesley, 2005.

[8] N.A. Campbell, Applied Statistics 29 (1980) 231–237.

[9] A. Cerioli, M. Riani, A.C. Atkinson, Statistics and Computing 19 (2009) 341–353.

[10] C. Croux, P. Filzmoser, M. Oliveira, Chemometrics and Intelligent Laboratory Systems 87 (2007) 218–225.

[11] C. Croux, G. Haesbroeck, Journal of Multivariate Analysis 71 (1999) 161–190.

[12] C. Croux, G. Haesbroeck, Biometrika 87 (2000) 603–618.

[13] C. Croux, A. Ruiz-Gazen, Journal of Multivariate Analysis 95 (2005) 206–226.

[14] M. Daszykowski, K. Kaczmarek, Y.V. Heyden, B. Walczak, Chemometrics and Intelligent Laboratory Systems 85 (2007) 203–219.

[15] P. Davies, The Annals of Statistics 15 (1987) 1269–1292.

[16] S.J. Devlin, R. Gnanadesikan, J.R. Kettenring, Journal of the American Statistical Association 76 (1981) 354–362.

[17] D.L. Donoho, Breakdown Properties of Multivariate Location Estimators, Technical Report, Harvard University, Boston, 1982.

[18] D.L. Donoho, P.J. Huber, in: P. Bickel, K. Doksum, J.L. Hodges (Eds.), A Festschrift for Erich Lehmann, Wadsworth, Belmont, CA, 1983, pp. 157–184.

[19] N.R. Draper, H. Smith, Applied Regression Analysis, John Wiley & Sons, New York, 1966.

[20] P. Filzmoser, R.G. Garrett, C. Reimann, Computers & Geosciences 31 (2005) 579–587.

[21] P. Filzmoser, R. Maronna, M. Werner, Computational Statistics & Data Analysis 52 (2008) 1694–1711.

[22] H. Fritz, P. Filzmoser, C. Croux, Computational Statistics (2012). To appear.

[23] S. Frosch-Møller, J. von Frese, R. Bro, Journal of Chemometrics 19 (2005) 549–563.

[24] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software, Addison-Wesley, Reading, 1995.

[25] R. Gnanadesikan, J.R. Kettenring, Biometrics 28 (1972) 81–124.

[26] F. Hampel, E. Ronchetti, P. Rousseeuw, W. Stahel, Robust Statistics. The Approach Based on Infuence Functions, John Wiley & Sons, 1986.

[27] F.R. Hampel, The Annals of Mathematical Statistics 42 (1971) 1887–1896.

[28] J. Hardin, D.M. Rocke, Journal of Computational and Graphical Statistics 14 (2005) 910–927.

[29] D.M. Hawkins, Computational Statistics & Data Analysis 17 (1994) 197–210.

[30] P.J. Huber, The Annals of Mathematical Statistics 35 (1964) 73–101.

[31] P.J. Huber, Robust Statistics, John Wiley & Sons,New York, 1981.

[32] M. Hubert, P. Rousseeuw, K. Vanden Branden, Technometrics 47 (2005) 64–79.

[33] M. Hubert, P.J. Rousseeuw, S. van Aelst, Statistical Science 23 (2008) 92–119.

[34] M. Hubert, K. Van Driessen, Computational Statistics & Data Analysis 45 (2004) 301–320.

[35] R.A. Johnson, D.W. Wichern, Applied Multivariate Statistical Analysis, Prentice Hall, International, 2002. Fifth edition.

[36] G. Li, Z. Chen, Journal of the American Statistical Association 80 (1985) 759–766.

[37] N. Locantore, J. Marron, D. Simpson, N. Tripoli, J. Zhang, K. Cohen, Test 8 (1999) 1–28.

[38] H.P. Lopuhaä, The Annals of Statistics 17 (1989) 1662–1683.

[39] R.A. Maronna, The Annals of Statistics 1 (1976) 51–67.

[40] R.A. Maronna, Technometrics 47 (2005) 264–273.

[41] R.A. Maronna, D. Martin, V. Yohai, Robust Statistics: Theory and Methods, John Wiley & Sons, New York, 2006.

[42] R.A. Maronna, V.J. Yohai, Journal of the American Statistical Association 90 (1995) 330–341.

[43] R.A. Maronna, R.H. Zamar, Technometrics 44 (2002) 307–317.

[44] R. Naga, G. Antille, Computational Statistics & Data Analysis 10 (1990) 169–174.

[45] D.J. Olive and D.M. Hawkins, Robust Multivariate Location and Dispersion, Dept. of Mathematics, Southern Illinois University, Carbondale, IL, USA, 2010.

[46] G. Pison, S. Van Aelst, G. Willems, Metrika 55 (2002) 111–123.

[47] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[48] M. Riani, A.C. Atkinson, A. Cerioli, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71 (2009) 447–466.

[49] D.M. Rocke, The Annals of Statistics 24 (1996) 1327–1345.

[50] P.J. Rousseeuw, in: W. Grossmann, G. Pflug, I. Vincze, W. Wertz (Eds.), Mathematical Statistics and Applications Vol. B, Reidel Publishing, Dordrecht, 1985, pp. 283–297.

[51] P.J. Rousseeuw, in: S. Kotz, C. Read, D. Banks (Eds.), Encyclopedia of Statistical Sciences Update Volume 3, Wiley, New York, 1999, pp. 441–443.

[52] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, John Wiley & Sons, New York, 1987.

[53] P.J. Rousseeuw, K. Van Driessen, Technometrics 41 (1999) 212–223.

[54] P.J. Rousseeuw, B.C. van Zomeren, Journal of the American Statistical Association 85 (1990) 633–651.

[55] D. Ruppert, Journal of Computational and Graphical Statistics 1 (1992) 253–270.

[56] M. Salibian-Barrera, S. Van Aelst, G. Willems, Journal of the American Statistical Association 101 (2006) 1198–1211.

[57] M. Salibian-Barrera, V.J. Yohai, Journal of Computational and Graphical Statistics 15 (2006) 414–427.

[58] A.D. Shieh, Y.S. Hung, Statistical Applications in Genetics and Molecular Biology 8 (2009).

[59] W.A. Stahel, Breakdown of Covariance Estimators, Research Report 31, ETH Zurich, 1981. Fachgruppe für Statistik.

[60] W.A. Stahel, Robuste Schätzungen: Infinitesimale Optimalität und Schätzungen von Kovarianzmatrizen, Ph.D. thesis no. 6881, Swiss Federal Institute of Technology (ETH), Zürich, 1981.

[61] K. Tatsuoka, D. Tyler, The Annals of Statistics 28 (2000) 1219–1243.

[62] V. Todorov, Computational Statistics & Data Analysis 14 (1992) 515–525.

[63] V. Todorov, A note on the MCD consistency and small sample correction factors, 2008. Unpublished manuscript, in preparation.

[64] V. Todorov, P. Filzmoser, Journal of Statistical Software 32 (2009) 1–47.

[65] V. Todorov, N. Neykov, P. Neytchev, in: R. Dutter, W. Grossmann (Eds.), Short Communications in Computational Statistics, COMPSTAT 1994, Physica Verlag, Heidelberg, 1994, pp. 90–92.

[66] K. Vanden Branden, M. Hubert, Chemometrics and Intellegent Laboratory Systems 79 (2005) 10–21.

[67] S. Verboven, M. Hubert, Chemometrics and Intelligent Laboratory Systems 75 (2005) 127–136.

[68] R.R. Wilcox Introduction to Robust Estimation and Hypothesis Testing, Elsevier Academic Press, Burlington, MA, USA, 2005.

[69] D.L. Woodruff, D.M. Rocke, Journal of the American Statistical Association 89 (1994) 888–896.

[70] V.J. Yohai, The Annals of Statistics 15 (1987) 642–656.

[71] M. Zhu, A. Ghodsi, Computational Statistics and Data Analysis 51 (2006) 918–930.