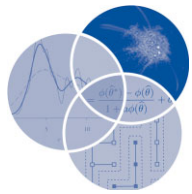
Focus Article



MATLAB library LIBRA

Sabine Verboven[1]* and Mia Hubert[2]

LIBRA stands for 'library for robust analysis.' It is a MATLAB toolbox mainly containing implementations of *robust* statistical methods. Robust statistics is involved with the detection of aberrant observations, also called outliers. The aim of robust statistical methods is to provide estimates which are not affected by the nonregular observations and which are then able to pinpoint the outliers. The robust methods implemented in LIBRA nowadays cover: univariate location, scale (Q_n) and skewness estimation (Medcouple), covariance estimation (FAST MCD), regression (FAST-LTS, MCD regression, depth quantiles), principal component analysis (RAPCA, ROBPCA), principal component regression (RPCR), partial least squares regression (RSIMPLS), classification (RDA, RSIMCA) and several methods to deal with skewed data. Besides that, the toolbox contains various methods for cluster analysis, many graphical diagnostic tools, and classical equivalents of several implemented methods. Only a few of the recently added methods will be highlighted in this paper. The features of the LIBRA functions will be illustrated by means of some real data sets. *WIREs Comp Stat* 2010 2 509–515

The library for robust analysis, LIBRA, contains robust statistical methods developed by the research groups in robust statistics of the Katholieke Universiteit Leuven (Department of Mathematics) and the University of Antwerp (Department of Mathematics and Computer Science). These methods are devised as alternatives to the classical statistical techniques in order to overcome the harmful influence of deviant data points. Data analysts should always be aware that their data often contain atypical observations. Outliers can occur, e.g., because of a human mistake, a change of the process parameters, a technical defect or because some cases belong to another population than the majority of the data. Therefore robust statisticians advocate the use of methods which can identify these outliers.[1–4]

Because MATLAB is a widespread software package used by statisticians, chemometricians, industrial researchers, and engineers, several robust techniques were implemented in the LIBRA toolbox. They include location, scale, and skewness estimation,[5–7] covariance estimation (FAST-MCD)[8], regression (FAST-LTS,[9] MCD regression,[10] depth quantiles[11]), principal component analysis (RAPCA,[12] ROBPCA[13]), principal component regression (RPCR[14]), partial least squares regression (RSIMPLS[15]), classification (RDA,[16] RSIMCA[17]), and several methods to deal with skewed data.[18–21] LIBRA also contains the clustering algorithms as described in Ref 22, many graphical tools to detect and classify the outliers, and classical equivalents of some of the robust methods. The toolbox can be downloaded from the web site `http://wis.kuleuven.be/stat/robust`. The programs are provided free for noncommercial use. They are regularly updated in order to work under the latest MATLAB version (current version, Release 2010a). For compatibility with previous releases and other OS platforms separate zip-files with dll's are also available. Some of the functions require the MATLAB statistics toolbox. An extract of the list of the 35 current main functions in LIBRA is printed in Table 1. In this article three methods are described in more detail: a method for robust principal component analysis (ROBPCA), the partitioning around mediods (PAM) clustering technique and the bagplot. Some other functions are illustrated in Ref 23.

*Correspondence to: sabine.verboven@ua.ac.be

[1]Department of Mathematics and Computer Science, University of Antwerp, Middleheimlaan 1, BE-2020, Antwerp, Belgium

[2]Department of Mathematics, Katholieke Universiteit Leuven, Celestijnenlaan 200B, BE-3001 Leuven, Belgium

DOI: 10.1002/wics.96

TABLE 1 | Extract From the List of Current Main Functions in LIBRA

Function	Description
Robust estimators of location/scale/skewness	
mlochuber	M-estimator of location, with Huber psi-function.
unimcd	MCD estimator of location and scale.
mad	Median absolute deviation with finite sample correction factor (estimator of scale).
mscalelogist	M-estimator of scale, with logistic psi-function.
qn	Q_n-estimator of scale.
mc	Medcouple, a robust estimator of skewness.
Robust multivariate analysis	
l1median	L_1 median of multivariate location.
mcdcov	Minimum covariance determinant estimator of multivariate location and covariance.
rapca	Robust principal component analysis (based on projection pursuit).
robpca	Robust principal component analysis (based on projection pursuit and MCD estimation).
rda	Robust linear and quadratic discriminant analysis.
rsimca	Robust soft independent modeling of class analogies
adjustedoutlyingness	Detection of multivariate outliers at skewed data: based on the adjusted outlyingness
halfspacedepth	Halfspacedepth of bivariate data points
bagplot	Draws the bagplot of bivariate data points, based on halfspacedepth or adjusted outlyingness.
Robust regression methods	
ltsregres	Least trimmed squares regression.
mcdregres	Multivariate MCD regression.
rpcr	Robust principal component regression.
rsimpls	Robust partial least squares regression.
cdq	Censored depth quantiles
predict	Regression results for new data based on RPCR or RSIMPLS analysis
Clustering methods	
agnes	Agglomerative nesting
clara	Clustering method for large applications
clusplot	Bivariate clustering plot of output from pam, fanny, or clara
daisy	Computing pairwise dissimilarities
diana	Divisive analysis
fanny	Fuzzy analysis
mona	Monothetic analysis
pam	Partitioning around medoids
tree	Tree plot for the output of agnes or diana

ROBUST PRINCIPAL COMPONENT ANALYSIS (ROBPCA)

Method description

When faced with multivariate data principal component analysis (PCA) is often used for exploratory data analysis or as a dimension reduction technique. In particular, it is very useful when the data set is high-dimensional, i.e., if there are more variables p than observations n. In PCA a set of $k < p$ new uncorrelated variables with a minimal loss of information, called principal components, are constructed. These principal components can reveal latent structures in the data and be used in further analyses, e.g., clustering or regression modeling. In classical PCA, the newly constructed variables are the eigenvectors of the sample covariance matrix and it is well known that this covariance matrix, and hence also the principal components, is highly influenced by outliers in the data.

The robust PCA method ROBPCA, developed by Hubert et al.,[13] is able to resist the bad impact of anomalous observations. The method is a combination of projection pursuit ideas and minimum covariance determinant (MCD) estimation in lower dimensions.[24] An important input parameter of ROBPCA is a lower bound of the percentage of uncontaminated observations in the data, denoted by α. By default the value of α is set to 0.75. This will give accurate results if the data set contains at most 25% of aberrant values, which is a reasonable assumption for most data sets. When more contamination is suspected, the parameter α can be set to any value which is at least 0.5.

Data and results

The Amphora data set was received from Ms. E. Duflou, student at the subfaculty of Archeology in the Katholieke Universiteit Leuven. It contains information on 60 amphora of several sites in the Syrian coast region, from the late Bronze age and the Iron age. The variables represent the curvature of a curve of the upper profiles of the amphora, with 2000 measurement points. They are stored in the matrix X, which is thus of size 60×2000.

To perform a robust PCA on the Amphora data, and to store the results in an output structure 'resultROBPCA,' the following short MATLAB command should be given:

```
>> resultROBPCA=robpca(X)
```

The only required input argument of the 'robpca' function is a matrix X containing the observations in

the rows and the variables in the columns. In this way, all optional input arguments are set to their default values. Here, this implies that

- the number of principal components to retain is set to zero ('k' = 0) such that a scree plot and a robust predicted residual error sum of squares (PRESS) curve are plotted in order to decide on the number of components
- the maximal number of principal components is set to 10 ('kmax' = 10)
- the maximal fraction of outliers ROBPCA is able to resist is 25% ('alpha' = 0.75)
- when the number of variables is small enough ($5p < n$), the MCD approach is used and $k = \text{rank}(X)$ components will be computed ('mcd' = 1)
- plots will be drawn ('plot' = 1) and if the classical outputs were demanded also classical plots will be given
- the three most extreme data points will be labeled on the plots ('labod' = 3, 'labsd' = 3)
- classical PCA will not be computed ('classic' = 0)
- a scree plot will be given ('scree' = 1)
- a plot of the PRESS for each number of components k will be shown ('press' = 1)
- the full ROBPCA approach will be computed instead of some approximated set of eigenvalues and eigenvectors ('robpcamcd' = 1)
- no skewed data are suspected ('skew' = 0)

Recently ROBPCA has the additional input argument 'skew' which allows to perform robust PCA for skewed data.[21] Previously, the algorithm performed best when the regular data were approximately elliptically symmetric distributed. If the variables were skewed, the method tended to point out too many observations as outliers. With the recent changes ROBPCA is able to handle nonsymmetrical data as well.

If the user wants to change one or more of the default settings, the input arguments and their new values have to be specified. Assume, e.g., that we have no idea about the amount of contamination and we prefer to apply a highly robust method. If in addition, we might have skewed data, and we are also interested in the results of a classical PCA on the data, we set:

```
>> resultROBPCA=robpca(X,'alpha',0.50,'classic',
1,'skew',1)
```

Similar to all MATLAB built-in graphical functions, most of the LIBRA functions work with variable input arguments. More precisely, the input arguments in the function header consist of N required input arguments and a variable range of optional arguments

```
>> result = functionname(required1,required2,...,
requiredN,varargin).
```

Depending on the application, required input arguments are, e.g., the design matrix X, the response variable y in regression, the group numbers in a classification context, etc. The function call should assign a value to all the required input arguments in the correct order. However, the optional arguments can be omitted (which implies that the defaults are used) or they can be called in an arbitrary order. For example, the commands

```
>> resultROBPCA=robpca(X,'alpha',0.50,'classic',1)
>> resultROBPCA=robpca(X,'classic',1,'alpha',0.50,
'plots',1)
```

produce the same result. The output of the LIBRA functions is a structure containing different fields as described in the help of the evoked function. A structure in MATLAB is an array variable which can contain fields of different types and/or dimensions.

For example, the result of the ROBPCA analysis on the Amphora data with function call

```
>> resultROBPCA=robpca(X,'k',5,'classic',1,'labsd',
8,'labod',12)
```

is the structure 'resultROBPCA' containing the fields

```
      P : [2000 × 5 double]
      L : [0.2970 0.2546 0.1599 0.1251 0.0934]
      M : [1 × 2000 double]
      T : [60 × 5 double]
      k : 5
   kmax : 10
  alpha : 0.7500
      h : 46
Hsubsets : [1 × 1 struct]
     sd : [60 × 1 double]
     od : [60 × 1 double]
 cutoff : [1 × 1 struct]
   flag : [1 × 1 struct]
  class : 'ROBPCA'
classic : [1 × 1 struct]
```

It contains the robust loadings P, the eigenvalues L, the center M, and scores T for a choice of $k = 5$ components, under the assumption that there are at most 25% outliers present in the data. The algorithm then searches for the optimal subset of size $h = 46$. The field 'Hsubsets' contains the indices

TABLE 2 | Overview of the Different Types of Observations Based on Their Score Distance and Their Orthogonal Distance

Distances	Small SD	Large SD
Large OD	Orthogonal outlier	Bad PCA-leverage point
Small OD	Regular observation	Good PCA-leverage point

of several *h*-subsets obtained during the algorithm. It is only needed for further calculations with RPCR or RSIMPLS. Next, the score distance (SD) and the orthogonal distance (OD) of every data point is given. The SD expresses the statistical distance of a data point to the origin in the PCA-subspace. The OD is the orthogonal distance from a data point to the PCA-subspace. For both distances a cutoff value is determined and whenever the cutoff value is exceeded, the corresponding data point will be assigned to one of the three outlier groups: good PCA-leverage points, bad PCA-leverage points or orthogonal outliers. The classification of the observation types is summarized in Table 2. The output component 'resultROBPCA.flag.all' is a vector with value 0 for an outlier, and value 1 otherwise. Moreover orthogonal outliers have 'resultROBPCA.flag.od=0' and 'resultROBPCA.flag.sd=1,' good PCA-leverage points satisfy 'resultROBPCA.flag.od=1' and 'resultROBPCA.flag.sd=0,' and bad PCA-leverage points have 'resultROBPCA.flag.od=0' and 'resultROBPCA.flag.sd=0.'

An outlier map as given in Figure 1a visualizes the ODs versus the SDs. For the Amphora data we see that the observations 52–58 are classified as bad PCA-leverage points. It turned out that these seven amphora all came from the same site of Tell Sukas. There are also some small orthogonal outliers (7, 23, 24, 27, 47) and one good leverage point (2). Note that several of these cases are lying close to the cutoff lines and therefore must not to be seen as real outliers.

The plots can also be reproduced with the function 'makeplot.' This function was build to facilitate the plotting of all possible graphs from the output of one of the main functions in the toolbox. When the function call

```
>> makeplot(resultROBPCA,'labsd',8,'labod',12)
```

is made, a menu is displayed on which all the relevant plots are mentioned. By choosing the requested outlier map, the 8 data points with largest SD and the 12 data points with largest OD will be labeled, as seen in Figure 1a. Note that the command

```
>> makeplot(resultROBPCA,'nameplot','pcadiag',
'labsd',8,'labod',12)
```

directly invokes the same graph without interference of the small menu. As classical PCA was also computed ('classic' = 1), the outlier map resulting from classical PCA is also shown (Figure 1b). We see that classical PCA has been attracted by the bad leverage points and has turned several of them into good leverage points. Note that the results of this classical analysis are stored in the output field 'resultROBPCA.classic.'

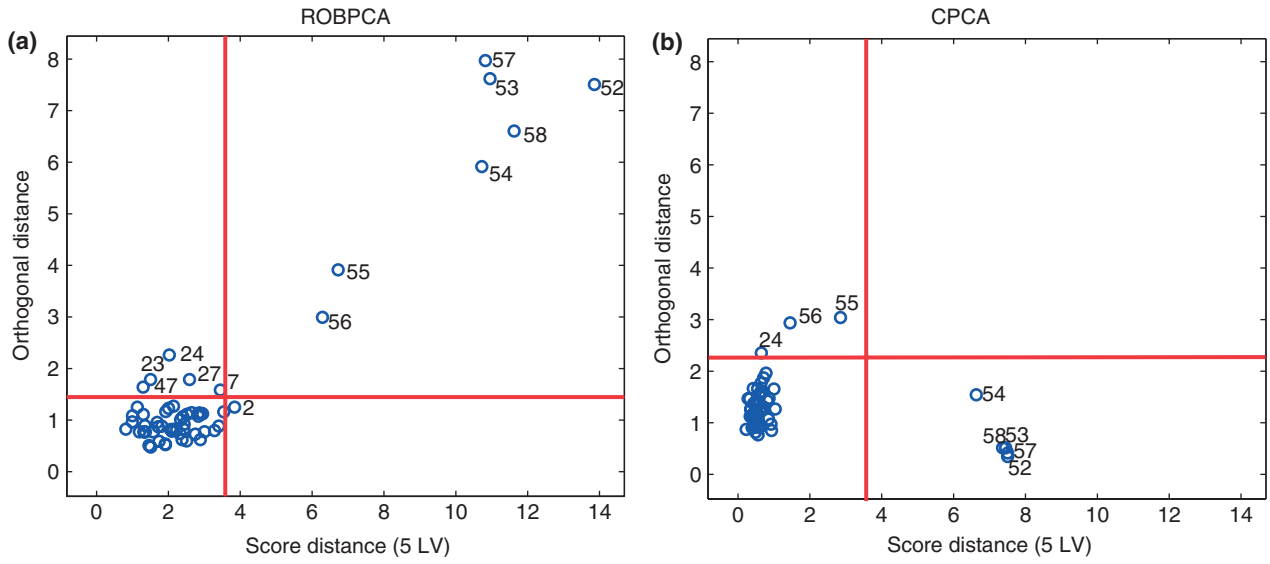


FIGURE 1 | ROBPCA and classical PCA outlier maps for the Amphora data set. (a) ROBPCA outlier map and (b) classical PCA outlier map.

CLUSTERING WITH PAM

Method description

Clustering is often performed at the exploratory data phase. Cluster analysis encompasses a number of different classification algorithms. The goal of clustering is to organize data into a meaningful structure, i.e., to develop taxonomies. LIBRA contains implementations of the cluster methods described in Ref 22. It contains the partitioning methods PAM, FANNY, and CLARA which divide the data into k user specified clusters. The hierarchical methods MONA, AGNES, and DIANA yield an entire hierarchy of clusterings of the data set. The function DAISY computes dissimilarities for different types of observations, which can be used as input for the partitioning methods.

Data and results

The Agriculture dataset was acquired from Eurostat,[25] the European statistical agency. It contains the gross national product (GNP) per capita and the percentage of the population employed in agriculture (AGRI) of 12 countries belonging to the European Union back in 1993. With the commands

```
>> resultPAM=pam(Agriculture,2,[4 4],0,1,1);
>> labels={'B','DK','D','GR','E','F','IRL','I','L',
    'NL','P','UK'};
>> makeplot(resultPAM, 'labclus', labels)
```

the PAM algorithm is applied on the Agriculture data set by searching for $k = 2$ clusters. The following input parameters indicate that both variables are quantitative, no standardization needs to be performed, Euclidean distances should be computed between the observations and plots will be made. Next, labels are created and a function call is made for graphical output of the clustering algorithm. Note that the 'makeplot' command is not requested here, as a menu will already be prompted when executing 'pam.' However by plugging a column cell array with the corresponding labels into the 'labclus' option of the makeplot function, the data points on the clusplot are labeled accordingly, as can be seen in Figure 2b.

The first resulting figure (Figure 2a) is the silhouette plot, which shows for every observation a bar whose length corresponds with its silhouette width, as well as the average silhouette width. For the definition of the silhouette width, we refer to Kaufman and Rousseeuw.[22] If the silhouette width of an observation is close to one, the corresponding sample is well classified into this cluster. A value close to zero indicates that it is not clear to which cluster the data point belongs, whereas a negative value points to a badly classified observation. Here, we clearly see that there are two clusters: one formed around Germany (obs. 3) and the other one formed around Portugal (obs. 11). The average silhouette width equals 0.6314 which indicates that overall a reasonable structure has been found.

The second figure (Figure 2b) is the clusplot.[26] For a bivariate data set the clusplot draws the data points and they are enclosed by ellipses that represent the clusters found with PAM. On this clusplot it is clear that the group around Portugal is formed by

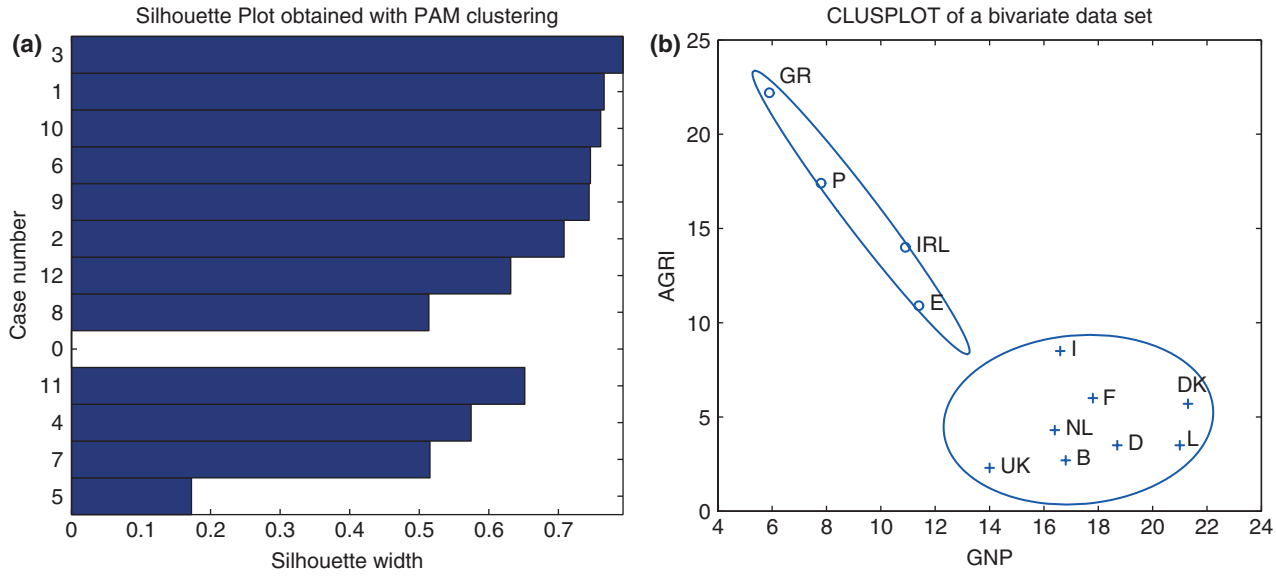


FIGURE 2 | The clustering output visualized for the Agriculture data. (a) Silhouette plot and (b) Clusplot.

Ireland, Greece, and Spain. They all have a low GNP and a rather high percentage of people working in agriculture. The group around Germany contains Italy, France, Belgium, United Kingdom, The Netherlands, Luxembourg and Denmark. Their GNP is higher and they have a rather low percentage of people working in agriculture. Note that when the number of variables is larger than two, clusplot first applies PCA on the data matrix, or multidimensional scaling (if the input of the cluster algorithm consists of dissimilarities).

BAGPLOT: A BIVARIATE BOXPLOT

Method description

The bagplot is a bivariate generalization of the univariate boxplot, based on the halfspacedepth.[19] A bagplot displays a bag containing 50% of the most central data points, a fence separating inliers from outliers, and a loop indicating the points outside the bag but inside the fence. The LIBRA 'bagplot' function not only draws a bagplot but also yields the half-space depth of all observations as well as the Tukey median, which is the observation with largest depth.

As the computation of the halfspacedepth is rather time-consuming, it is recommended at large data sets to perform the computations on a random subset of the data. The size of this subset can be chosen by the user. Alternatively a bagplot can be computed based on the adjusted outlyingness.[20]

Data and results

The cars data set was obtained from Ref 27 and consists of a representative sample of 804 used General Motor cars. We illustrate the bagplot on the bivariate data set with variables 'retail price' and 'mileage.'

A bagplot (based on the halfspacedepth) is constructed with the command

```
>> bagplot(X,'colorbag',[1  0  0]);
```

Figure 3 shows the resulting bagplot. The observation in the center (indicated with a cross) is the Tukey median, with largest halfspacedepth. The bag is the polygon drawn as a full line with red interior and contains the 50% cases with largest depth. The fence is the region indicated by the light gray loop. The observations outside the fence labeled with black stars are the outliers. They mainly have an unusual mileage and/or a unusual retail price. Note that the optional input parameters 'databag' and 'datafence' can be put to zero, in order not to display all observations inside the bag or inside the fence. Also the size of the subsample on which the computations are based, can be changed with the 'sizesubset' parameter.

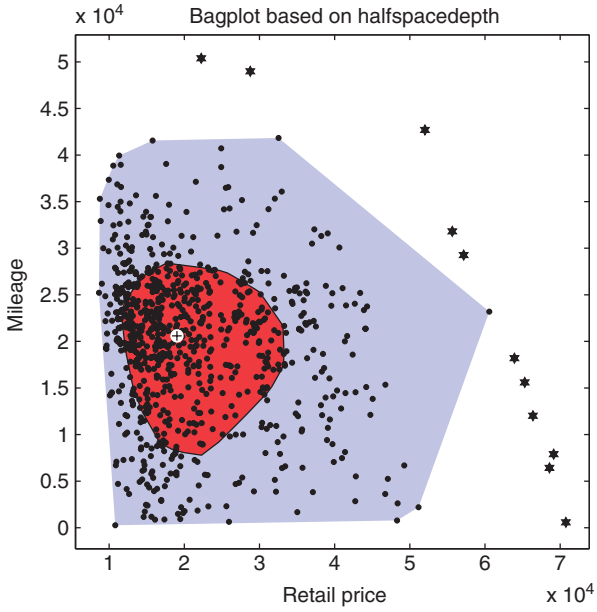


FIGURE 3 | Bagplot of the Cars data set based on the halfspacedepth.

CONCLUSION

In this article we have highlighted the main features of LIBRA, the MATLAB library for robust analysis. Illustrated on three multivariate methods (PCA, clustering, bagplot) we have shown the user-friendly input and output structure of the functions, the variety of graphical diagnostic tools, and the possibility for comparison with non-robust approaches. Also an overview of the current main functions is given.

REFERENCES

1. Huber PJ. *Robust Statistics*. New York: John Wiley & Sons; 1981.
2. Hampel FR, Ronchetti EM, Rousseeuw PJ, Stahel WA. *Robust Statistics: The Approach Based on Influence Functions*. New York: John Wiley & Sons; 1986.
3. Rousseeuw PJ, Leroy AM. *Robust Regression and Outlier Detection*. New York: John Wiley & Sons; 1987.

4. Maronna RA, Martin DR, Yohai VJ. *Robust Statistics: Theory and Methods*. New York: John Wiley & Sons; 2006.
5. Rousseeuw PJ, Verboven S. Robust estimation in very small samples. *Comput Stat Data Anal* 2002, 40: 741–758.
6. Rousseeuw PJ, Croux C. Alternatives to the median absolute deviation. *J Am Stat Assoc* 1993, 88:1273–1283.
7. Brys G, Hubert M, Struyf A. A robust measure of skewness. *J Comput Graph Stat* 2004, 13:996–1017.
8. Rousseeuw PJ, Van Driessen K. A Fast algorithm for the minimum covariance determinant estimator. *Technometrics* 1999, 41:212–223.
9. Rousseeuw PJ, Van Driessen K. In: Gaul W, Opitz O, SchaderM, eds, *Data Analysis: Scientific Modeling and Practical Application*. New York: Springer-Verlag; 2000, 335–346.
10. Rousseeuw PJ, Van Aelst S, Van Driessen K, Agullo A. Robust multivariate regression. *Technometrics* 2004, 46:293–305.
11. Debruyne M, Hubert M, Portnoy S, Vanden Branden K. Censored depth quantiles. *Comput Stat Data Anal* 2008, 52:1604–1614.
12. Hubert M, Rousseeuw PJ, Verboven S. A Fast method for robust principal components with applications to chemometrics. *Chemom Intell Lab Syst* 2002, 60: 101–111.
13. Hubert M, Rousseeuw PJ, Vanden Branden K. ROBPCA: a new approach to robust principal component analysis. *Technometrics* 2005, 47:64–79.
14. Hubert M, Verboven S. A robust PCR method for high-dimensional regressors. *J Chemom* 2003, 17:438–452.
15. Hubert M, Vanden Branden K. Robust methods for partial least squares regression. *J Chemom* 2003, 17: 537–549.
16. Hubert M, Van Driessen K. Fast and robust discriminant analysis. *Comput Stat Data Anal* 2004, 45:301–320.
17. Vanden Branden K, Hubert M. Robust classification in high dimensions based on the SIMCA method. *Chemom Intell Lab Syst* 2005, 79:10–21.
18. Hubert M, Vandervieren E. An adjusted boxplot for skewed distributions. *Comput Stat Data Anal* 2008, 52:5186–5201.
19. Rousseeuw PJ, Ruts I, Tukey JW. The bagplot: a bivariate boxplot. *Am Stat* 1999, 53:382–387.
20. Hubert M, Van der Veeken S. Outlier detection for skewed data. *J Chemom* 2008, 22:235–246.
21. Hubert M, Rousseeuw PJ, Verdonck T. Robust PCA for skewed data and its outlier map. *Comput Stat Data Anal* 2009, 53:2264–2274.
22. Kaufman L, Rousseeuw PJ. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & Sons; 1990.
23. Verboven S, Hubert M. LIBRA: a Matlab library for robust analysis. *Chemom Intell Lab Syst* 2005, 75: 127–136.
24. Rousseeuw PJ. Least median of squares regression. *J Am Stat Assoc* 1984, 79:871–880.
25. Eurostat. *Cijfers en feiten: Een statistisch portret van de Europese Unie*. 1994.
26. Pison G, Struyf A, Rousseeuw PJ. Displaying a clustering with CLUSPLOT. *Comput Stat Data Anal* 1999, 30:381–392.
27. Kuiper S. Introduction to multiple regression: how much is your car worth? *J Stat Educ* 2008, 16(3). www.amstat.org/publications/jse/v16n3/datasets.kuiper.html.

