Repeated double cross validation

Peter Filzmoser[a], Bettina Liebmann[b] and Kurt Varmuza[b]*

---

* *Correspondence to: K. Varmuza, Laboratory for Chemometrics, Institute of Chemical Engineering, Vienna University of Technology, Getreidemarkt 9/166, A-1060 Vienna, Austria*
E-mail: kvarmuza@email.tuwien.ac.at

[a] *P. Filzmoser*
Institute of Statistics and Probability Theory, Vienna University of Technology, Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria
E-mail: P.Filzmoser@tuwien.ac.at

[b] *B. Liebmann, K. Varmuza*
Laboratory for Chemometrics, Institute of Chemical Engineering, Vienna University of Technology, Getreidemarkt 9/166, A-1060 Vienna, Austria
E-mail: liebmann@mail.zserv.tuwien.ac.at

(**Abstract)**

Repeated double cross validation (rdCV) is a strategy for (a) optimizing the complexity of regression models, and (b) for a realistic estimation of prediction errors when the model is applied to new cases (that are within the population of the data used). This strategy is suited for small data sets and is a complementary method to bootstrap methods. rdCV is a formal, partly new combination of known procedures and methods, and has been implemented in a function for the programming environment R, providing several types of plots for model evaluation. The current version of the software is dedicated to regression models obtained by PLS. The applied methods for repeated splits of the data into test sets and calibration sets, as well as for estimation of the optimum number of PLS components, are described. The relevance of some parameters (number of segments in CV, number of repetitions) is investigated. rdCV is applied to two data sets from chemistry: (1) determination of glucose concentrations from NIR data in mash samples from bioethanol production; (2) modeling the gas chromatographic retention indices of polycyclic aromatic compounds from molecular descriptors. Models using all original variables and models using a small subset of the variables, selected by a genetic algorithm, are compared by rdCV.

Keywords: prediction performance; optimum complexity of linear PLS models; cross validation; bootstrap; R

1. INTRODUCTION

Modeling a property y by several variables x is a fundamental task in chemometrics. Widely used are empirical linear models of the form

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_jx_j + \ldots + b_mx_m + e \quad (1)$$

with b_0 for the intercept, b_1 to b_m the regressions coefficients, m the number of variables, and e the error term. For mean-centered data, b_0 becomes zero. The model parameters b_0 and b_1 to b_m are estimated from a calibration set containing n_{CALIB} objects (samples) with the observed values of the x-variables and of y. Principal aim of creating a model is a good prediction performance of the model for new objects, for which an optimum complexity of the model is necessary. In the most used regression method in chemometrics, partial least-squares, PLS [1-3], the complexity is controlled by the number of PLS components, a. An optimum number of components, a_{OPT}, avoids underfitting (if a is too small, the model is too simple) and overfitting (if a is too big, the model is too much adapted to the calibration data). The resulting model has to be evaluated with a test set with observed values for x and y, containing n_{TEST} objects that were not used in generation and optimization of the model.

This implies the necessity of splitting all available objects into a calibration set, which is used for model building, and a test set for model evaluation. For small values of n, as it is often the case with chemometric data sets, resampling

techniques are useful for this task. The (frequently random) split can be performed several times to obtain a reasonable estimation of the optimum model complexity as well as of the range of the prediction errors for new objects.

Widely used for this purpose are traditional versions of cross validation (CV) [4], but also Monte Carlo cross validation [5-7], and moving window cross validation [8], as well as bootstrap techniques [9,10]. It is important, however, to note that CV-based methods are not always optimal, particularly when dealing with data obtained by experimental design [11]. Some evaluation strategies allow estimating the variability of the optimum model complexity and the variability of the prediction performance, and thus point out fundamental limits for model generation, if only small data sets are available.

This contribution focuses on the strategy *repeated double cross validation* (rdCV), suited for small data sets for optimizing the complexity of regression models, and for estimating the prediction errors to be expected for new cases (that are within the population of the data used). rdCV is a formal combination of known procedures and is a complementary method to bootstrap.
This strategy dates back to an early study by Stone [12] . Similar approaches have been described for a binary classification problem in proteomics [13,14], for a discrimination of human sweat samples [15] as well as for principal component analysis [16].

We have implemented rdCV (using PLS regression) in a function for the programming environment R [17], and have created several types of plots for model evaluation [2]. In this paper the algorithm of rdCV, and the *standard error method* [18] for an estimation of the optimum number of components are described in detail. The rdCV strategy is applied to model the glucose concentration in mashes by NIR data, and an example of quantitative structure-property relationships.

2. METHODS

2.1. Overview

Here we give an outline of the following sections, which are organized in a modular system. As appropriate performance criteria are essential for empirical regression models, they will be defined first. Based on the introduced criteria, we present a statistical approach for finding the optimum complexity of a regression model, that is the optimum number of PLS components (Section 2.3). The so called "standard error method" is then integrated as a vital part in the "repeated double cross validation" (rdCV). Eventually, Section 2.4 focuses on a comprehensive description of the rdCV method as a whole. The main distinguishing characteristics of rdCV to simple methods are: Accidental performance results are avoided by – typically 100 – repeated random splits of the data into calibration sets and test sets. The model's complexity is optimized by inner *k*-fold cross validation with each available calibration set. For

estimation of the prediction performance when applying the model to new samples, each model is validated with an independent test set. Depending on the number of repetitions, rdCV yields a large number of residuals from "independent" test sets, which are the basis for measuring the prediction performance. To compare the rdCV method with standard validation approaches, a *k*-fold cross validation as well as a bootstrap technique is described in Section 2.5.

2.2. Performance criteria

A statistical estimation of the prediction performance of a model is based on the prediction errors (residuals)

$$e_i = y_i - \hat{y}_i \qquad (2)$$

with y_i for the given (experimental) value and $\hat{y}_i$ the predicted (modeled) value of an object *i*. The predicted *y*-values are from a test set or obtained with CV or bootstrap. If a rather large number of residuals is available – as it is the case with rdCV or bootstrap - the distribution of the prediction errors gives a comprehensive picture of the model performance. In many cases, the distribution is similar to a normal distribution and has a mean of approximately zero. Such a distribution can be well described by a single parameter that measures the spread. Other types of distributions can, for instance, be

characterized by a tolerance interval comprising 95% of the residuals [19,20]. The criteria used here are defined as follows.

The standard deviation of the prediction errors (in short: standard error of prediction, SEP) is defined by

$$\mathrm{SEP} = \sqrt{\frac{1}{n_{SEP} - 1} \sum_{i=1}^{n_{SEP}} \left(y_i - \hat{y}_i - \mathrm{bias}\right)^2} \qquad (3)$$

$$\mathrm{bias} = \frac{1}{n_{SEP}} \sum_{i=1}^{n_{SEP}} \left(y_i - \hat{y}_i\right) \qquad (4)$$

The number of used $\hat{y}$-values, n_{SEP}, is for simple strategies the number of objects in a single test set or the total number of objects; in rdCV it is the number of objects times the number of repetitions. The bias is the arithmetic mean of the prediction errors; it is near zero (especially for large n_{SEP}) for objects drawn as a random sample as done in rdCV. The bias may be different from zero for data obtained in a different experiment or with a different instrument; in this case, the data do not belong to the population of the data used for model generation and a test set bias (a systematic deviation) has to be considered. The SEP is the standard deviation of the residuals, and thus it is expressed in the same unit as the *y*-variable. We use SEP only for test-set predicted $\hat{y}_i$ (equivalent to $\mathrm{SEP}_{\mathrm{TEST}}$).

The mean squared error (MSE) obtained from CV within a calibration set is used to optimize the number of PLS components by the standard error method ([18], see Section 2.3) and is defined as

$$\mathrm{MSE} = \frac{1}{n_{MSE}} \sum_{i=1}^{n_{MSE}} (y_i - \hat{y}_i)^2 \qquad (5)$$

The number of values considered, n_{MSE}, is the number of objects in the used validation set (see Section 2.3).

The tolerance interval TI_{95} is defined by the difference of the 2.5% and 97.5% percentiles, $q_{0.025}$, and $q_{0.975}$, of the empirical residual distribution; this measure of residual spread is less influenced by extreme outliers and is independent from the shape of the distribution.

2.3. Optimum model complexity

In PLS the model complexity is controlled by the number, a, of used PLS components (linear combinations of the variables). Many different evaluation methods have been proposed for estimating the optimum number of components, a_{OPT}, and almost all of these techniques are based on CV or bootstrap [4,18]. Randomization tests have been applied to assess the statistical significance of adding further components [11,21]. Different approaches for the estimation of the optimum model complexity have been compared for PLS [22] and PCA [23].

The simplest form of CV is to split the data (randomly) into k segments (typically, k is chosen between 5 and 10), to fit regression models for a range of numbers of PLS components to all but one segment, and to evaluate the models on the omitted segment. Since each segment serves once for evaluation, the optimum number of PLS components can be determined. A simple bootstrap version is to generate new data sets of the same size as the original data by randomly sampling objects with replacement from the original data. Regression models for different numbers of PLS components are fit to the data, and afterwards evaluated on objects that were not used in the bootstrap data set. This strategy allows estimating the optimal model complexity; however, the resulting prediction performance is often too optimistic. It is thus recommended to split the data into calibration and test data and to base the final choice of the model complexity solely on results from the calibration data, whereas the prediction performance is based on the test data. However, another random split into calibration and test data could yield a different value for the prediction performance.

The selection of a_{OPT} is based on an error criterion, for instance MSE, obtained by CV for the validation sets (see Section 2.4). MSE usually decreases with increasing a; after a more or less distinct minimum it increases because of overfitting. In a single CV, for each object in the used calibration set an estimation $\hat{y}$ is obtained, and MSE is usually computed from all n_{CALIB} residuals; that means we have one MSE for each value of a. The global minimum of MSE

is in general considered to give a too large value for a_{OPT}, and various heuristic schemes are used to select a somewhat smaller value avoiding overfitting [23,24].

With the rdCV approach as well as with bootstrap we have several values of MSE for each a (1, ..., a_{MAX}) and thus we can apply the statistically based *standard error method* for an estimation of the optimum number of components. This method is briefly described as "one standard error rule" in Figure 3.6 of [18], and we give a detailed description here (Figure 1). In this method for each number of components a, the mean of the MSE values, m_{MSE}, resulting from the validation sets is calculated. In rdCV the number of validation sets is equal to the number of segments in the inner CV loop (Figure 2). Assume the global minimum of the means, ${}_{MIN}m_{MSE}$, is at a_{MIN} components. For this number of components the standard deviation $s({}_{MIN}m_{MSE})$ of ${}_{MIN}m_{MSE}$ is calculated; it is the standard deviation of a mean (standard error) and therefore given by

$$s({}_{MIN}m_{MSE}) = \frac{s_{MSE}}{\sqrt{h}} \qquad (6)$$

with s_{MSE} for the standard deviation of the MSE-values at a_{MIN} components, and h for the number of MSE-values at a_{MIN}. According to the one standard error rule, the optimum, a_{OPT}, is given by the most parsimonious model with $m_{MSE} < {}_{MIN}m_{MSE} + s({}_{MIN}m_{MSE})$. Thus, we consider that the MSE-means have

errors and we take a conservative model to avoid overfitting. The number, *h*, of MSE values must not be too small, say at least four; in the pseudo code below *h* corresponds to SEG_{CALIB}.

The standard error method is applied within rdCV as follows (in pseudo programming code):

(1) Use a calibration set with n_{CALIB} objects.

(2) Split the calibration set randomly into SEG_{CALIB} (≥ 4) segments.

(3) **FOR** κ = 1 TO SEG_{CALIB} (loop through all segments)

(3.1) Validation set = objects in segment κ (n_{VAL} objects)

(3.2) Training set = other objects ($n_{TRAIN} = n_{CALIB} - n_{VAL}$ objects)

(3.3) Make PLS models from the training set, varying the number of components a = 1, ..., a_{MAX}.

(3.4) Apply the models to the objects of the validation set, resulting in CV-predicted values $\hat{y}_{CV}(i)$, i = 1, ..., n_{VAL}

(3.5) Compute MSE(κ, a), which are the MSE-values for segment κ for a = 1, ..., a_{MAX}.

NEXT κ

(4) Estimate the optimum number of components, a_{OPT}, by the standard error method

(4.1) We have the values MSE(κ, a), κ = 1, ..., SEG_{CALIB}; a = 1, ..., a_{MAX}

(4.2) Compute the means, m_{MSE}, of the MSE-values for each a.

(4.3) Search the global minimum, $_{MIN}m_{MSE}$, of m_{MSE}; it appears at a_{MIN} components.

(4.4) Compute the standard deviation s_{MSE} of the MSE-values at a_{MIN} .

(4.5) The optimum number of components, a_{OPT}, for the calibration set is the smallest number of components fitting the inequality (7)

$$m_{MSE} < {}_{MIN}m_{MSE} + \frac{\pi \cdot s_{MSE}}{\sqrt{SEG_{CALIB}}} \qquad (7)$$

The *parsimony factor* π controls the selection of the optimum number of components. A value of zero would chose the global minimum of MSE; $\pi = 1$ gives the one standard error rule; $\pi = 2$ gives a *two standard error rule* and considers a 95% confidence interval of the minimum MSE, resulting in a model with a very small number of components. Mostly, $\pi = 1$ gives best results; for atypical shapes of the relation m_{MSE} versus a, various additional heuristics may be necessary to avoid useless results for a_{OPT}.

2.4. Repeated double cross validation (rdCV)

Double cross validation consists of two nested loops. In the *outer loop* the objects are split randomly into test sets and calibration sets (one segment as test set, and the others as calibration set); it is used to estimate the prediction performance by application of models made solely from calibration set data to the test sets. After finishing the outer loop, for each object a test set predicted *y*-

value is available. The *inner loop* works with a calibration set as defined in the outer loop. The inner loop again consists of a CV for finding the optimum complexity of the model (for PLS the optimum number of components, applying the standard error method described in Section 2.3). In repeated double cross validation (rdCV), a double CV is repeated many times (typically 100 times) in an additional *repetition loop* with different random splits into test sets and calibrations sets. Thus, the number of available test set predicted *y*-values is increased; the prediction performance can be better estimated, as well as its variability. Furthermore, the variability of the optimum number of components can be estimated from the results and a final optimum number of components can be derived for a final model from all objects.

Here, rdCV is applied to calibration models created by PLS; a modification for classification problems would be straightforward. Furthermore, diagnostic plots - based on rdCV results - are presented to evaluate the model complexity and model performance. Our implementation of rdCV in R-functions makes this method available to all interested persons.

The rdCV strategy, written here in pseudo programming code, has been realized as follows (see also Figure 2). For the repetition loop we use index ρ (1, ... , n_{REP}), for the outer loop (test sets) index τ (1, ..., SEG_{TEST}); and for the inner loop (calibration sets) index κ (1, ... SEG_{CALIB}).

Repetition loop: **FOR** ρ = 1 TO n_{REP}

(1) Split all n objects randomly into SEG_{TEST} segments (typ. 3-10) of approximately equal size.

(2) Outer loop: **FOR** τ = 1 TO SEG_{TEST}

(a) Test set = segment with number τ (n_{TEST} objects)

(b) Calibration set = other SEG_{TEST} - 1 segments (n_{CALIB} objects)

(c) Split calibration set into SEG_{CALIB} segments (typ. 4-10) of approximately equal size.

(d) Inner loop: **FOR** κ = 1 TO SEG_{CALIB}

(i) Validation set = segment with number κ (n_{VAL} objects)

(ii) Training set = other SEG_{CALIB} - 1 segments (n_{TRAIN} objects)

(iii) Make PLS models from the training set, with a = 1, ..., a_{MAX} components

(iiii) Apply the PLS models to the validation set, resulting in $\hat{y}_{CV}$ for the objects in segment κ for a = 1, ..., a_{MAX}

NEXT κ

(e) Estimate optimum number of components, a_{OPT}, from $\hat{y}_{CV}$ of the calibration set by the standard error method (Section 2.3), giving $a_{OPT}(\tau)$ for this outer loop.

(f) Make a PLS model from the whole calibration set using $a_{OPT}(\tau)$ components

(g) Apply the model to the current test set, resulting in test-set predicted $\hat{y}$ for n_{TEST} test set objects.

NEXT τ

(3) After completing the outer loop, we have one test-set predicted $\hat{y}$ for each of the n objects

NEXT π

After a complete rdCV run, a variety of numbers is available that can be exploited for model diagnostics and characterization. In each repetition, we have SEG_{TEST} different test sets and the same number of (partly overlapping) calibration sets. From each calibration set an optimum number of PLS components, a_{OPT}, has been estimated. In total SEG_{TEST} times n_{REP} values for a_{OPT} have been estimated from which a final optimum number of PLS components, a_{FINAL}, is derived. In this work, we simply choose the value with the highest frequency (in the case of equal frequencies, the lower value); see Figure 3. If the frequency distribution of the a_{OPT} values does not show a clear maximum, a heuristic algorithm or a selection by the user is necessary to obtain a parsimonious but good model complexity.

A 3-dimensional data array $\boldsymbol{E}$ contains the residuals (prediction errors) e (i, a, ρ) for objects i (i = 1, ..., n), obtained from models with a (a = 1, ... a_{MAX}) components, in repetitions ρ (ρ = 1, ... , n_{REP}), see Figure 4. A corresponding array with the same dimensions contains the predicted values $\hat{y}$ (i, a, ρ).

Of special interest are the data in the slice for $a = a_{FINAL}$. The standard deviation of the n times n_{REP} residuals gives a final performance measure SEP_{FINAL}. The distribution of these residuals gives a picture of the model performance, and the

2.5% and 97.5% percentiles define the tolerance interval TI_{95}. For each repetition a separate SEP value can be computed from n residuals; the distribution of these n_{REP} SEP values - for instance presented in a boxplot (see Section 4.3) - describes the variation of this performance measure. For each object n_{REP} residuals are available; a scatter plot of these residual versus y or the object number indicates objects that give often erroneous $\hat{y}$, and indicates a potential dependence of the residuals on y (see Section 4.4).

The data in array $\boldsymbol{E}$ for a selected repetition ρ can be used to display the SEP as a function of the number of components. Aggregating the graphs for all repetitions in one plot shows the fluctuations of SEP; in particular at too large values of a, which indicates overfitting (see Section 4.3).

2.5. Other strategies

We already mentioned that the simplest forms of CV and bootstrap are usually too optimistic. Thus, another strategy to apply these techniques is to split the data randomly into calibration and test set, to build the regression models with the calibration data, and to evaluate on the test data. This heuristic is frequently applied, and we will use it for comparing the results with rdCV.

More specifically, a test set of a quarter of the n objects is randomly selected. This will allow for a more consistent comparison with rdCV where we proposed to use four segments in the outer loop. Thus, the calibration set with n_{CALIB}

objects consists of the remaining three quarters of the data. Using these calibration data, optimized PLS models have been created by two different approaches:

- **Seven-fold CV:** Using seven segments for CV makes this evaluation better comparable with the proposed rdCV procedure, where we will also use seven segments in the inner loop. By omitting each segment in turn, PLS models with $a = 1, \ldots, a_{MAX}$ components are fitted, and applied to the omitted segment. This yields residuals for each sample in the calibration set. The one standard error rule (Section 2.3) is used by summarizing the MSEs of each segment with mean and standard error; the result is the optimal number a_{OPT} of PLS components. A model with a_{OPT} components is computed from the whole calibration set and is applied to the test set samples. From these results, the prediction performance (SEP) is computed.
- **Bootstrap:** By random sampling with replacement from the calibration data a bootstrap sample (training set) of the same size as the calibration set is generated, and PLS models with $a = 1, \ldots, a_{MAX}$ components are fitted. The models are then applied to those samples from the calibration set that have not been used in the training set, and the MSE is computed. Repeating this procedure 100 times gives 100 MSEs for each model complexity, and a_{OPT} is chosen by applying the one standard error rule. A model with a_{OPT} components is computed from the whole calibration set and is applied to the test set; from the test set predictions SEP is computed.

Although calibration set and test set have been selected randomly, the resulting SEP_{TEST} values could be (just by chance) too optimistic or too pessimistic, depending on how representative this separation was. Therefore, each of the four quarters of the data serves once as a test set, and the remaining three quarters as calibration set. This finally gives four values of a_{OPT} and SEP_{TEST} for CV and bootstrap, which can be compared with results from rdCV.

3. DATA AND SOFTWARE

3.1. Data

For a demonstration of the use of rdCV and a comparison of different validation strategies two data sets from chemistry were investigated in this work.

GLC: The first data set is from 120 mash samples from bioethanol production with different concentrations of glucose [25]. Sample variations are not only related to glucose concentration, but also to enzymatic pretreatment and type of feedstock (wheat or rye) in the fermentation process. The first derivatives of near infrared (NIR) absorbance spectra in the wavelength range of 1100 to 2300 nm provided the *x*-variables, while glucose concentrations in g/l (*y*-variable) were determined by HPLC. In the model building and evaluation process three different variable sets were used: (1) The first variable set contains all 235 *x*-variables available; (2) a subset of 15 features has been

selected by a genetic algorithm (GA) using software MobyDigs [26] on the entire data set; (3) furthermore, 15 variables have been randomly selected from all but the GA selected ones.

PAC: The second data set, which is available in the R-package "chemometrics", is from 209 polycyclic aromatic compounds with their gas chromatographic retention indices as dependent *y*-variable [27]. A set of 467 molecular descriptors [28] was used as *x*-variables to model the retention indices. The descriptors have been calculated by the software Dragon [29] from 3-dimensional chemical structures with all hydrogen atoms explicitly encoded (created by the software Corina [30]). The original 1630 descriptors were treated by a simple variable selection in two steps resulting in 467 variables: (a) elimination of constant or almost constant variables (all but a maximum of five values constant); (b) elimination of variables with a correlation coefficient >0.95 to another variable. Again three different variable sets were used: (1) The first variable set contains all 467 *x*-variables available; (2) a subset of 13 features has been selected by a genetic algorithm (GA) using the software MobyDigs [26] on the entire data set; (3) furthermore, 15 variables have been randomly selected from all but the GA selected ones.

3.2. Software

R is a language and environment for statistical computing and graphics [17]. It is free software licensed under the GNU General Public License (GPL) and

available from the Comprehensive R Archive Network (CRAN). R provides many statistical techniques and graphical facilities. Most importantly, it can be extended easily by packages such as “pls” for principal component regression (PCR) and partial least-squares regression (PLS) [31]. For this work the newly developed R-package “chemometrics” [2] has been used; it provides the function “mvr_dcv” for rdCV. This function handles the data as well as settings for the nested loops in rdCV, e. g. number of segments for creation of test, calibration and validation sets. For generation of PLS models, the function "mvr" of the mentioned "pls" package is used with method "simpls". A typical call of “mvr_dcv” is:

```
library(chemometrics)      # load package “chemometrics”
data(PAC)                  # load PAC dataset
result <- mvr_dcv(y~X, data=PAC, ncomp=50, method="simpls")
                           # call rdCV function
```

By default, no scaling of the data is provided; the number of repetitions is 100; the data set is split into four segments in the outer and ten segments in the inner loop. This call will consider models up to 50 PLS components. Further parameters of "mvr_dcv" are explained in the help file of the function.

Several diagnostic plots can be generated from the "mvr_dcv" result for visual inspection of validation results (see Section 2.4). The function “plotSEPmvr” creates a plot with the SEP-values of all repetitions versus the number of PLS

components; function "plotpredmvr" plots the predicted versus the experimental *y*-values; function "plotresmvr" displays the frequency distribution of residuals.

For variable selection by a genetic algorithm (GA) [32] the software MobyDigs [26] has been used. It performs ordinary least-squares regression (OLS) with a leave-one-out (LOO) cross validation for testing variable subsets created by the GA. As fitness criterion the squared adjusted correlation coefficient, R^2_{ADJ}, [2] between experimental and LOO-predicted values was chosen. After more than a million iterations (with a computation time of some hours on a standard personal computer), the variables included in the best model were selected for further analysis. The long computation time of GA prohibits its use within the loops of rdCV. Therefore, variable selection has been performed with the entire data set, knowing that models obtained from variables selected in this way may show an overestimated performance. An alternative would be to use a fast - and much less powerful - variable selection for each calibration set. It is out of scope of this work to compare the advantages and drawbacks of different strategies for variables selection, we aim on comparing the performances of models obtained from given data, regardless of the origin of the variables.

Chemical structures were handled in Molfile format. Approximate 3-dimensional structures with all hydrogen atoms explicitly given have been created by the software Corina [30]. Molecular descriptors have been generated by the software Dragon [29]; the output file of Dragon has been imported by an R-function for further use within R.

4. RESULTS

4.1. Number of segments

In rdCV the size of test and validation sets can be manipulated by user-defined parameters, that is the number of segments in the outer loop and in the inner loop, respectively. To ensure realistic results for the optimum number of components, a_{OPT}, with the standard error method (Section 2.3), the number of segments in the inner loop, SEG_{CALIB}, should be at least four. Two contrary effects influence the stability of a_{OPT}. On one hand, an increase of SEG_{CALIB} increases the number of values for computation of the mean and the standard deviation, which are required in the standard error method. On the other hand, increasing SEG_{CALIB} lowers the number of values in each segment from which MSE is calculated.

The more segments in the outer loop, the more values a_{OPT} are obtainable for the frequency distribution, from which a single final optimum number of components, a_{FINAL}, is derived. Note that additional segments always implicate longer computing time. For example, an rdCV run with the PAC data takes about seven minutes (on a standard personal computer) with SEG_{TEST} and SEG_{CAL} set both to four, but 40 minutes if the number of segments is increased to 10 for both segmentations. Therefore, a trade-off between computing time and segmentation is inevitable.

We studied the influence of different segmentations on the stability of two measures for PLS regression models, a_{FINAL} and SEP_{FINAL}, for the GLC and PAC data (Figure 5). Since both values exhibit only slight variations for each data set, it stands to reason that the number of outer and inner segments is secondary; in particular for smaller sets of variables. Interestingly, a random selection of *x*-variables also yields stable results. For randomly selected variables, the final standard errors of prediction are two to three times higher than for all variables, whereas the optimum number of PLS components for these models with randomly selected variables is lower than for models with all variables or GA-selected variables. Thus, overfitting is avoided by the rdCV approach.

Remarkable is one result for the final number of components, a_{FINAL}, for the PAC data using all *x*-variables and four segments in both loops. In this case, the frequency distribution of the obtained 400 values for a_{OPT} shows three maxima of almost equal heights at 11, 19, and 24 components; selecting the value 11, which has maximum frequency, is somewhat arbitrary. Anyway, it may be instructive to learn from the frequency plot that the optimum number of components is not unique in this example. For such cases, a heuristics has to be added to the algorithm including a user-definable parameter that controls the desired parsimony.

The influence of the number of segments on the shape of the frequency distribution of the optimum number of components is presented in Figure 6 for the GLC data. In the left hand side plot the number of segments in the outer loop, SEG_{TEST}, is fixed to four; in the right hand side plot to 10. In both plots, the number of segments in the inner loop, SEG_{CALIB}, is varied with the values 4, 5, 7, and 10. Most noticeably, all distributions have distinct maxima at almost the same horizontal position (a_{OPT}), indicating the minor influence of the number of segments on the optimum number of components. The more segments in the inner loop (SEG_{CALIB}), the narrower the distribution becomes. This small effect can be explained with more stable results yielded by the one standard error rule for a larger number of MSE values used for mean and standard deviation.

Because of this investigation, we suggest to use $SEG_{TEST} = 4$, and $SEG_{CALIB} = 7$ for rdCV calculations as a sensitive compromise between computational effort and validation results.

4.2. Number of repetitions

The repetition loop in rdCV with n_{REP} passes provides the large number of values for the optimum number of components (a_{OPT}) and the residuals from test set objects, necessary for a reasonable evaluation of models from the used data set. The computing time is proportional to n_{REP}, for instance for 100 repetitions, $SEG_{TEST} = 4$, and $SEG_{CALIB} = 7$ with the GLC data set (235 variables) 4 minutes and with the PAC data set (467 variables) 10 minutes. Thus, the

influence of the number of repetitions on the stability of the result is of interest. For n_{REP} = 5, 20, and 100 the rdCV was repeated 100 times. The variations of the 100 values for final measures a_{FINAL} and SEP_{FINAL} (see Figures 3 and 4) have been expressed as relative standard deviations in percent of the means (Table I). As expected, the variations decrease with increasing number of repetitions. With only 5 or 20 repetitions, the variations are considerably higher than with 100 repetitions; therefore, the latter value is recommended.

4.3. Variation of model performance

The performance of a model is measured by SEP, the standard deviation of the prediction errors (residuals) obtained from objects in test sets (Section 2.2). This measure depends on the random split of the n available objects into test sets and calibration sets (outer loop in rdCV). The results from the n_{REP} repetitions in rdCV allow estimating how much SEP varies for different random splits. The distribution of the n_{REP} SEP values for a model complexity with a_{FINAL} components can be represented in a boxplot. Figure 7 compares the distributions of SEP for different variable sets in the data GLC and PAC. The number of repetitions was 100, and the number of segments four in the outer loop and seven in the inner loop. Variable subsets selected by GA gave the smallest SEP values for both data sets, whereas the 15 randomly selected variables yielded the highest SEP values. The results for the GA-selected variables may be too optimistic because variable selection has been performed with the entire data set.

The graphical impression of a difference in SEP values resulting from two models (e.g. using all variables or a selection of the variables) could also be confirmed by a statistical test, like the Mann-Whitney U-test for comparing the medians. Hence, rdCV also supports model selection because the performance measure corresponding to a model is not one value but a distribution. In addition, rdCV also provides the value SEP_{FINAL} that is usually located close to the median of the SEP values.

For comparison, the SEP values obtained by the simpler strategies described in Section 2.5 are included in the plots. The *n* objects have been split into four parts, and each served as a test set. Consequently, we obtain four values for SEP, either using 7-fold CV or bootstrap in the calibration sets for estimating the optimum number of components. For both data sets and all variable sets, these eight values for SEP show a much larger variation than the rdCV results. One can conclude that a single split into a test set and a calibration set may yield very misleading results.

For the GLC data, Figure 8 presents the SEP values as a function of the number of components for 100 repetitions (left), and a scatter plot with the corresponding predicted versus the experimental *y*-values (right). Two facts are clearly visible in the left plot: (a) A slightly increasing variation of the SEP with increasing number of components; and (b) an outlying curve for one of the repetitions. Actually, in this repetition six outlying objects were by chance put

into one of the test sets, which gave high values for SEP. The predicted *y*-values resulting from this repetition are considerably lower than the experimentally determined values, as can be seen in the plot on the right hand side. It is an advantage of rdCV's repeated random sample selection that a few extremely pessimistic test sets do not deteriorate the final regression result.

4.4. Residual plots

The rdCV procedure yields test set predicted *y*-values for each object, each repetition, and all considered numbers of components, which gives an array with n times n_{REP}, times a_{MAX} data (see Figure 4) with usually some thousand values. Likewise, an array comprising the corresponding residuals (prediction errors for test set objects) is available.

Of special interest are the residuals for the final model complexity with a_{FINAL} components. In Figure 9 probability density functions of prediction errors are given for the GLC data set, using all 235 variables (left) and 15 variables selected by GA (right), respectively. rdCV was applied with $SEG_{TEST} = 4$; $SEG_{CALIB} = 7$; $n_{REP} = 100$. The data are from models with a_{FINAL} components, that is 14 for the data set with all variables, and 15 for the 15 selected variables. In the latter case the variable selection by GA yielded less correlated variables [25], and of course the PLS model is equivalent to an OLS model. The black lines are for the distributions calculated from all 12,000 available residuals; the gray lines show the distribution for each repetition. As these residual

distributions do not differ markedly from normal distributions, they can be characterized by the standard deviation of the residuals, equivalent to SEP. The plots evidently show that the distributions from models with all variables are wider and more varying than the distributions from models using the selected 15 variables. Furthermore, models with all variables give some very large negative prediction errors, illustrated by the tailing of the curves in the left figure.

Residual plots are very common as diagnostic tools for regression models; often the residuals are plotted versus the experimental *y*. In Figure 10 residuals are plotted versus the sample number for the PAC data set, using models with 13 GA-selected variables, and rdCV applied with SEG_{TEST} = 4, SEG_{CALIB} = 7, and n_{REP} = 100; data are from models with a_{FINAL} = 9 components and refer to objects in test sets. The gray symbols are the 100 predicted *y* for the 100 repetitions; the black symbols are their means; the dashed horizontal lines show the approximate 95% tolerance interval $\pm$ 2 SEP_{FINAL}. In general, no systematic dependence of the residuals on sample number or experimental *y* (the objects are sorted here by increasing *y*) is visible. However, some compounds show many large prediction errors. Particularly, for object numbers 12 and 102 all residuals are below the lower tolerance boundary; these two compounds can be considered as structural outliers because they do not contain condensed aromatic rings. Number 12 is azulene (with a 7-ring condensed to a 5- ring); number 102 is thianthrene (two benzene rings connected by two S-bridges). Two structures have large positive prediction errors: number 140 is dimethyl-pyrene and 143 is 1,1'-binaphthyl; both are not structural outliers. For the

identification of "difficult to predict" objects the many (n_{REP}) residuals for each object are very helpful.

4.5. Summary

The recommended way for creation of a final PLS regression model from given data ***X*** and ***y*** by using the rdCV approach can be summarized as follows: Perform rdCV with SEG_{TEST} = 4 and SEG_{CALIB} = 7 and 100 repetitions; for a first test one may use only 20 repetitions. The results from rdCV give the optimum number of components for the final model, a_{FINAL}, and the standard deviation of prediction errors, SEP_{FINAL}, when applied to new samples from the same population. A tolerance interval for the prediction errors can be deduced from the distribution of the residuals. If this distribution is similar to the normal distribution, an interval of $\pm$ 2 SEP_{FINAL} gives the range for 95% of the prediction errors; otherwise the 2.5% and 97.5% percentiles define the 95 % tolerance interval. Finally, a model from all objects with a_{FINAL} components is built for future use. Assuming that all new samples are from the same data population as the samples used for model creation, the prediction errors can be expected in the same range as estimated by rdCV.

Table II summarizes parameters and performance measures for the PLS models created for the GLC and PAC data set. All results have been obtained with SEG_{TEST} = 4 and SEG_{CALIB} = 7 and 100 repetitions. The final PLS models have been created from all objects with a_{FINAL} components using the GA

selected variables. By application of these models to the samples used for model creation, a standard error of calibration (SEC) of 1.7 for the GLC data and 7.3 for the PAC data, respectively, is computed. These values are within the range of SEP as estimated by rdCV.

5. CONCLUSIONS

Model evaluation is of utmost importance in chemometrics. This study is devoted to the problem of choosing the optimal number of components for partial least squares (PLS) regression, although the strategy is also applicable to other regression methods with a tunable parameter. The optimal model aims at maximizing the prediction performance for new test data. In many papers only a single number, for instance the standard error of prediction (SEP), is presented as a measure of prediction performance. Depending on the evaluation method, this number can reflect the reality, but it can also be too optimistic, or sometimes even too pessimistic. To overcome this problem, we presented a strategy based on repeated double cross validation (rdCV). It includes a statistically based method to find the optimum number of PLS components as well as a careful estimation of the range of prediction errors to be expected for new cases.

This paper provides a comprehensive description of the rdCV procedure. Furthermore, we specify the “standard error method” [18], which is used for determining the optimal number of PLS components. Using two real data sets

from chemistry, the stability of rdCV with respect to the number of repetitions and the number of segments in the inner and outer CV is investigated. As a result, we propose to use 100 repetitions, four segments in the outer CV and seven segments in the inner CV. This choice requires moderate computational effort, and leads to stable results for a final SEP value and for a final number of PLS components.

In addition to these final model parameters, each repetition of the rdCV strategy returns residuals, which can be summarized by a separate SEP value. A graphical presentation of the SEP values from all repetitions provides detailed information on the distribution of the prediction performance measure and the distribution of the number of PLS components. Further diagnostic plots reveal not only the overall prediction quality, but they give insight into the prediction quality of individual objects.

Also the bootstrap technique produces a large number of residuals, but the number of predicted values is in general not equal for the objects; the same holds for Monte Carlo CV. If the residuals have a distribution similar to a normal distribution, the model performance can be approximated by a single number, for instance the standard deviation of the residuals (SEP); otherwise a tolerance interval can be defined for the expected prediction errors by the 2.5% and 97.5% percentiles. These measures for the prediction performance are realistic estimations, as long as new objects are from the same statistical population as the objects used in rdCV.

The rdCV procedure has been implemented by the function “mvr_dcv” in the freely available R package “chemometrics”. Applications to data sets with up to about 200 objects and up to about 500 variables require a computation time of a few minutes on a standard personal computer. The resulting performance measures and data arrays can be simply exported for use by other software.

Acknowledgements

This work was partly funded by the Austrian Research Promotion Agency (FFG), BRIDGE program, project no. 812097/11126.

REFERENCES

1. Martens H, Naes T. Multivariate calibration. Wiley: Chichester, United Kingdom, 1989.
2. Varmuza K, Filzmoser P. Introduction to multivariate statistical analysis in chemometrics. Francis & Taylor, CRC Press: Boca Raton, FL, USA, 2009.
3. Wold S, Ruhe A, Wold H, Dunn WJI. The collinearity problem in linear regression. The partial least squares approach to generalized inverses. *SIAM J. Sci. Stat. Comput.* 1984; **5**: 735-743.

4. Vandeginste BGM, Massart DL, Buydens LCM, De Jong S, Smeyers-Verbeke J. Handbook of chemometrics and qualimetrics: Part B. Elsevier: Amsterdam, The Netherlands, 1998.

5. Boulesteix AL. WilcoxCV: a R package for fast variable selection in cross-validation. *Bioinformatics* 2007; **23**: 1702-1704.

6. Konovalov DA, Sim N, Deconinck E, Vander Heyden Y, Coomans D. Statistical confidence for variable selection in QSAR models via Monte Carlo cross-validation. *J. Chem. Inf. Model.* 2008; **48**: 370-383.

7. Xu QS, Liang YZ, Du YP. Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. *J. Chemometr.* 2004; **18**: 112-120.

8. Kasemsumran S, Du YP, Li BY, Maruo K, Ozaki Y. Moving window cross validation: a new cross validation method for the selection of a rational number of components in a partial least squares calibration model. *Analyst* 2006; **131**: 529-537.

9. Efron B, Tibshirani RJ. An introduction to the bootstrap. Chapman & Hall: London, United Kingdom, 1993.

10. Wehrens R, Putter H, Buydens LMC. The bootstrap: a tutorial. *Chemometr. Intell. Lab. Syst.* 2000; **54**: 35-52.

11. Faber NM, Rajko R. How to avoid over-fitting in multivariate calibration - The conventional validation approach and an alternative. *Anal. Chim. Acta* 2007; **595**: 98-106.

12. Stone M. Cross-validatory choice and assessment of statistical predictions. *J. R. Statist. Soc. B* 1974; **36**: 111-147.

13. Smit S, Hoefsloot HCJ, Smilde AK. Statistical data processing in clinical proteomics. *J. Chromatogr. B* 2008; **866**: 77-88.

14. Smit S, van Breemen MJ, Hoefsloot HCJ, Smilde AK, Aerts JMFG, de Koster CG. Assessing the statistical validity of proteomics based biomarkers. *Anal. Chim. Acta* 2007; **592**: 210-217.

15. Dixon SJ, Xu Y, Brereton RG, Soini HA, Novotny MV, Oberzaucher E, Grammer K, Penn DJ. Pattern recognition of gas chromatography mass spectrometry of human volatiles in sweat to distinguish the sex of subjects and determine potential discriminatory marker peaks. *Chemometr. Intell. Lab. Syst.* 2007; **87**: 161-172.

16. Forina M, Lanteri S, Boggia R, Bertran E. Double cross full validation. *Quimica Analitica (Barcelona, Spain)* 1993; **12**: 128-135.

17. R. A language and environment for statistical computing. R Development Core Team, Foundation for Statistical Computing, www.r-project.org: Vienna, Austria, 2008.

18. Hastie T, Tibshirani RJ, Friedman J. The elements of statistical learning. Springer: New York, NY, USA, 2001.

19. Naes T, Ellekjaer MR. The relation between SEP and confidence intervals. *NIR news* 1992; **3**(6): 6-7.

20. Naes T, Isaksson T, Fearn T, Davies T. A user-friendly guide to multivariate calibration and classification. NIR Publications: Chichester, United Kingdom, 2004.

21. Wiklund S, Nilsson D, Eriksson L, Sjöström M, Wold H, Faber K. A randomization test for PLS component selection. *J. Chemometr.* 2008; **21**: 427-439.

22. Gomez-Carracedo MP, Andrade JM, Rutledge DN, Faber NM. Selecting the optimum number of partial least squares components for the calibration of attenuated total reflectance-mid-infrared spectra of undesigned kerosene samples. *Anal. Chim. Acta* 2007; **585**: 253-265.

23. Bro R, Kjeldahl K, Smilde AK, Kiers HAL. Cross-validation of component models: A critical look at current methods. *Anal. Bioanal. Chem.* 2008; **390**: 1241-1251.

24. Martens HA, Dardenne P. Validation and verification of regression in small data sets. *Chemometr. Intell. Lab. Syst.* 1998; **44**: 99-121.

25. Liebmann B, Friedl A, Varmuza K. Determination of glucose and ethanol in bioethanol production by near infrared spectroscopy and chemometrics. *Anal. Chim. Acta* 2009: in print.

26. MobyDigs. Software. Talete srl, www.talete.it: Milan, Italy, 2004.

27. Lee ML, Vassilaros DL, White CM, Novotny M. Retention indices for programmed-temperature capillary-column gas chromatography of polycyclic aromatic hydrocarbons. *Anal. Chem.* 1979; **51**: 768-773.

28. Todeschini R, Consonni V. Handbook of molecular descriptors. Wiley-VCH: Weinheim, Germany, 2000.

29. Dragon. Software for calculation of molecular descriptors, by Todeschini R., Consonni V., Mauri A., Pavan M. Talete srl, www.talete.mi.it: Milan, Italy, 2004.

30. Corina. Software for the generation of high-quality three-dimensional molecular models. Molecular Networks GmbH Computerchemie, www.mol-net.de: Erlangen, Germany, 2004.

31. Mevik BH, Wehrens R. The pls package: Principal component and partial least squares regression in R. *J. Stat. Software* 2007; **18**(2): 1-24.

32. Leardi R, Boggia R, Terrile M. Genetic algorithms as a strategy for feature selection. *J. Chemometr.* 1992; **6**: 267-281.

Figure 1. Determination of the optimum number of components by the standard error method (schematically) [9].The calibration set is split into SEG_{CALIB} segments, and CV is applied with each segment being a validation set once, and the others the training set. For each segment the error measure MSE is computed for each number of components a (1, ..., a_{MAX}). The means of the MSE values, m_{MSE} are plotted versus a; the minimum is at a_{MIN}. Depending on the size of a parsimony parameter, π, the optimum number of components, a_{OPT} is obtained by Equation (7).

Figure 2. Repeated double cross validation (rdCV).

Figure 3. Determination of the final optimum number of PLS components, a_{FINAL}. rdCV gives SEG_{TEST} times n_{REP} values for the optimum number of components. The distribution of these values shows a distinct maximum and the value with the highest frequency is taken as a_{FINAL}.

Figure 4. Data from rdCV for diagnostics and evaluation of PLS regression models.

Figure 5. Influence of the number of segments on the optimum number of components, a_{FINAL}, and the final standard error of prediction, SEP_{FINAL}, for the GLC and PAC data with each three different variable sets used.

Figure 6. Frequency distribution of the optimum number of components, a_{OPT}, for varying segmentation in the inner and outer loop of rdCV, using the GLC data. Left, SEG_{TEST} = 4; right, SEG_{TEST} = 10. For SEG_{CALIB} the values 4 (solid line), 5 (dashed line), 7 (dotted line), and 10 (dotdashed line) have been used.

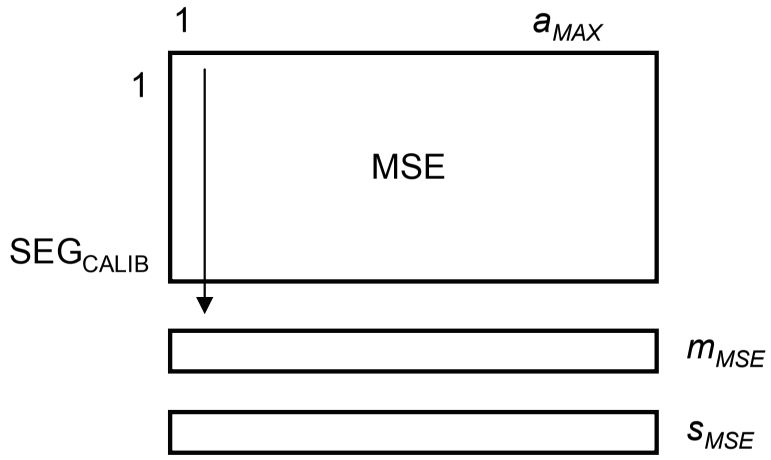
Figure 7. Distribution of SEP values obtained by rdCV in n_{REP} = 100 repetitions. For comparison, the values obtained with four test sets are displayed (with estimation of the number of components either by 7-fold CV or by bootstrap).

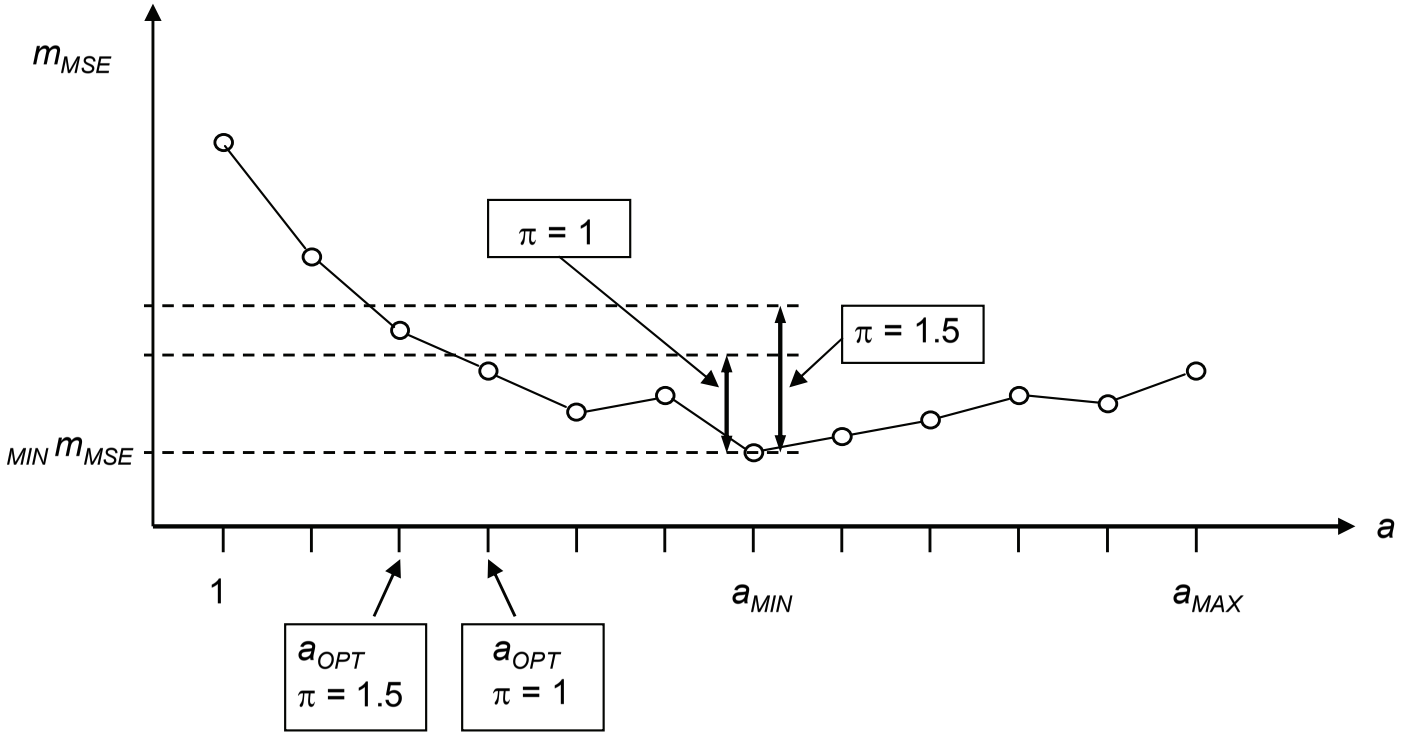
Figure 8. Left: SEP as a function of the number of PLS components for n_{REP} = 100 repetitions (gray lines). The horizontal dashed line indicates SEP_{FINAL}, the vertical dashed line a_{FINAL}. The black line is the mean of the 100 gray lines. One repetition shows extraordinarily high SEP values arising from an accidentally created test set with samples giving high prediction errors. Right: Predicted *y* for all 100 repetitions versus experimental *y* (gray symbols). The mentioned outlying samples show large negative residuals (in the *y*-range 0 to 8, and around 34). The black symbols are the means of 100 predicted values. The GLC data have been used with all variables; the number of segments was 5 in both rdCV loops.

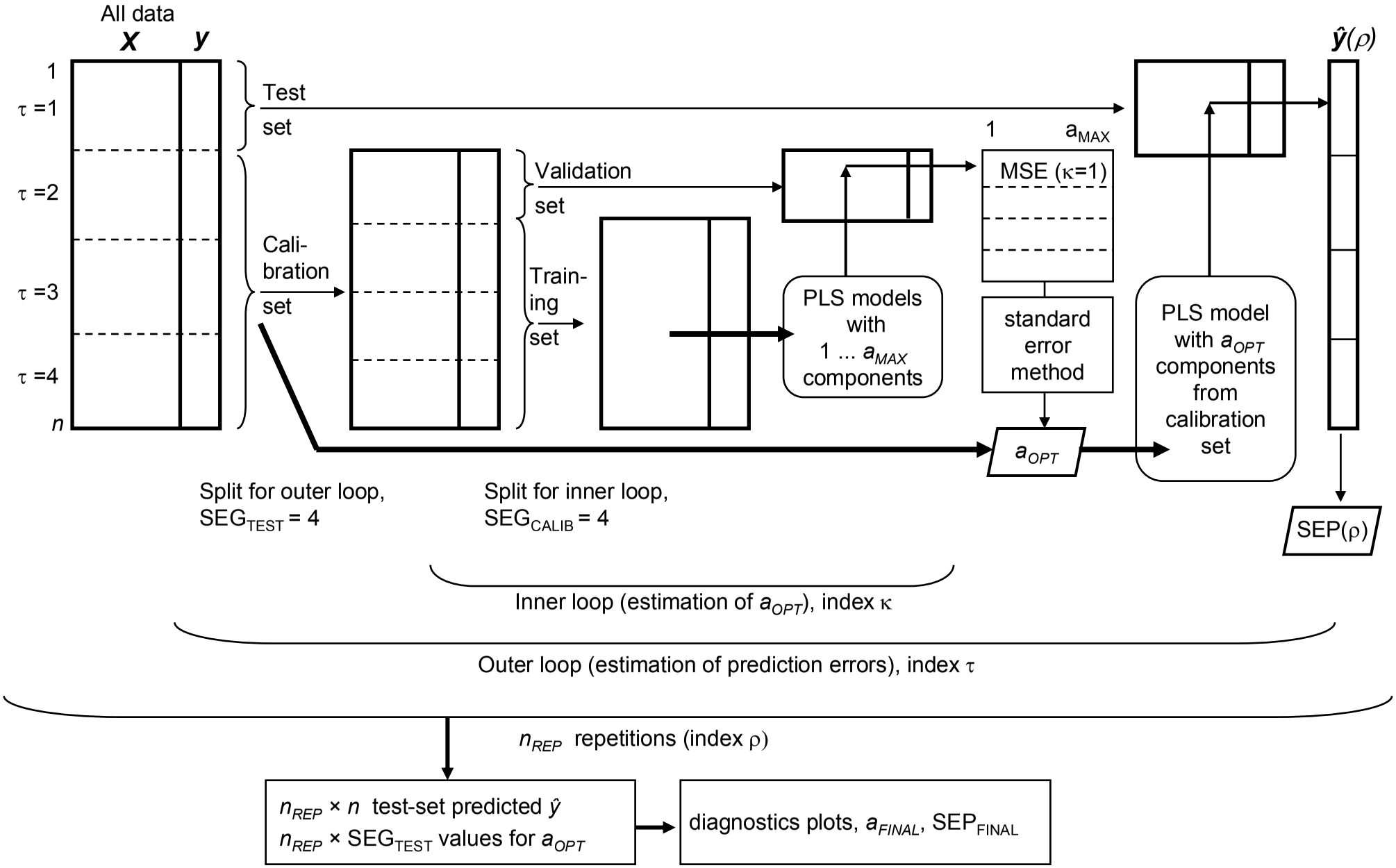
Figure 9. Distributions of prediction errors for modeling the glucose concentration in mash samples by using NIR data (GLC data). Results for all 235 variables (left) and a subset of 15 variables (right), selected by GA, are

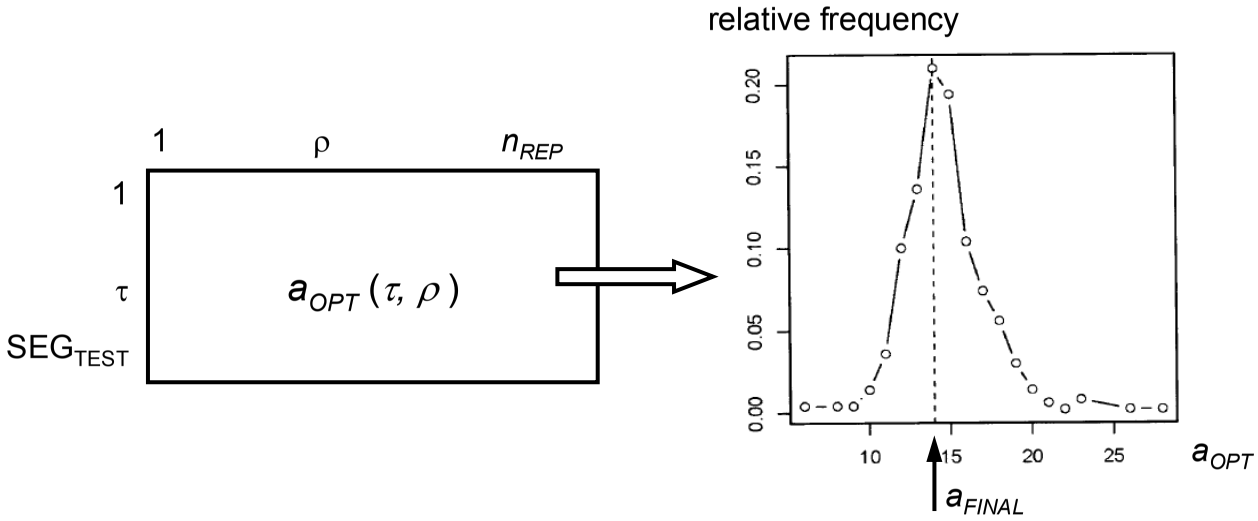
compared. rdCV was applied with SEG_{TEST} = 4, SEG_{CALIB} = 7, and n_{REP} = 100. The gray curves are from the 100 repetitions; the black curves are the distributions for all 12,000 (*n* times n_{REP}) residuals.

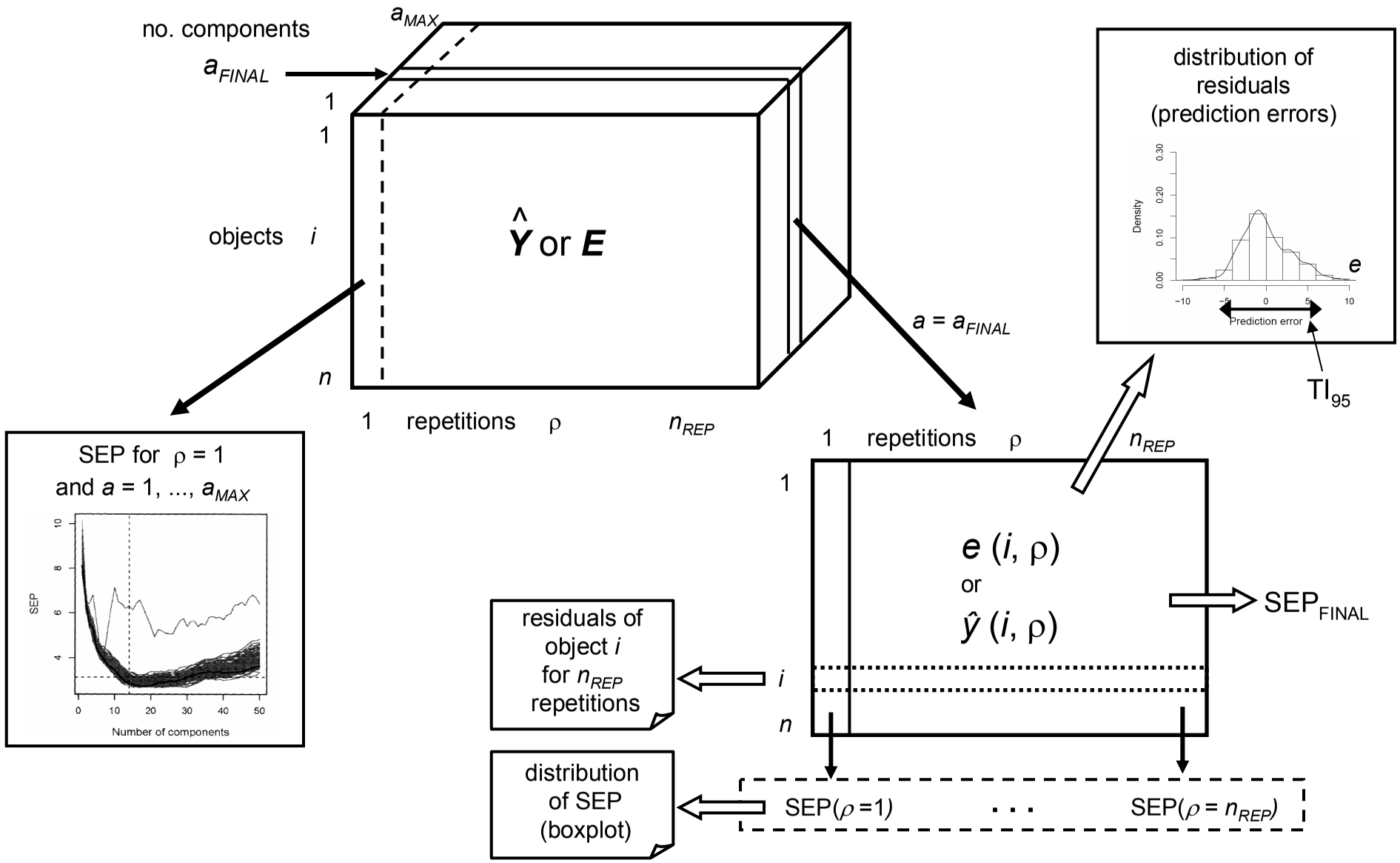
Figure 10. Residuals for modeling the GC retention index from molecular descriptors (PAC data). A set of 13 variables, selected by GA, has been used. rdCV was applied with SEG_{TEST} = 4; SEG_{CALIB} = 7; n_{REP} = 100. The gray symbols are the 100 predicted *y* for the 100 repetitions; the black symbols are their means. Some compounds show large prediction errors. The dashed horizontal lines indicate the tolerance interval $\pm$ 2 SEP_{FINAL}.

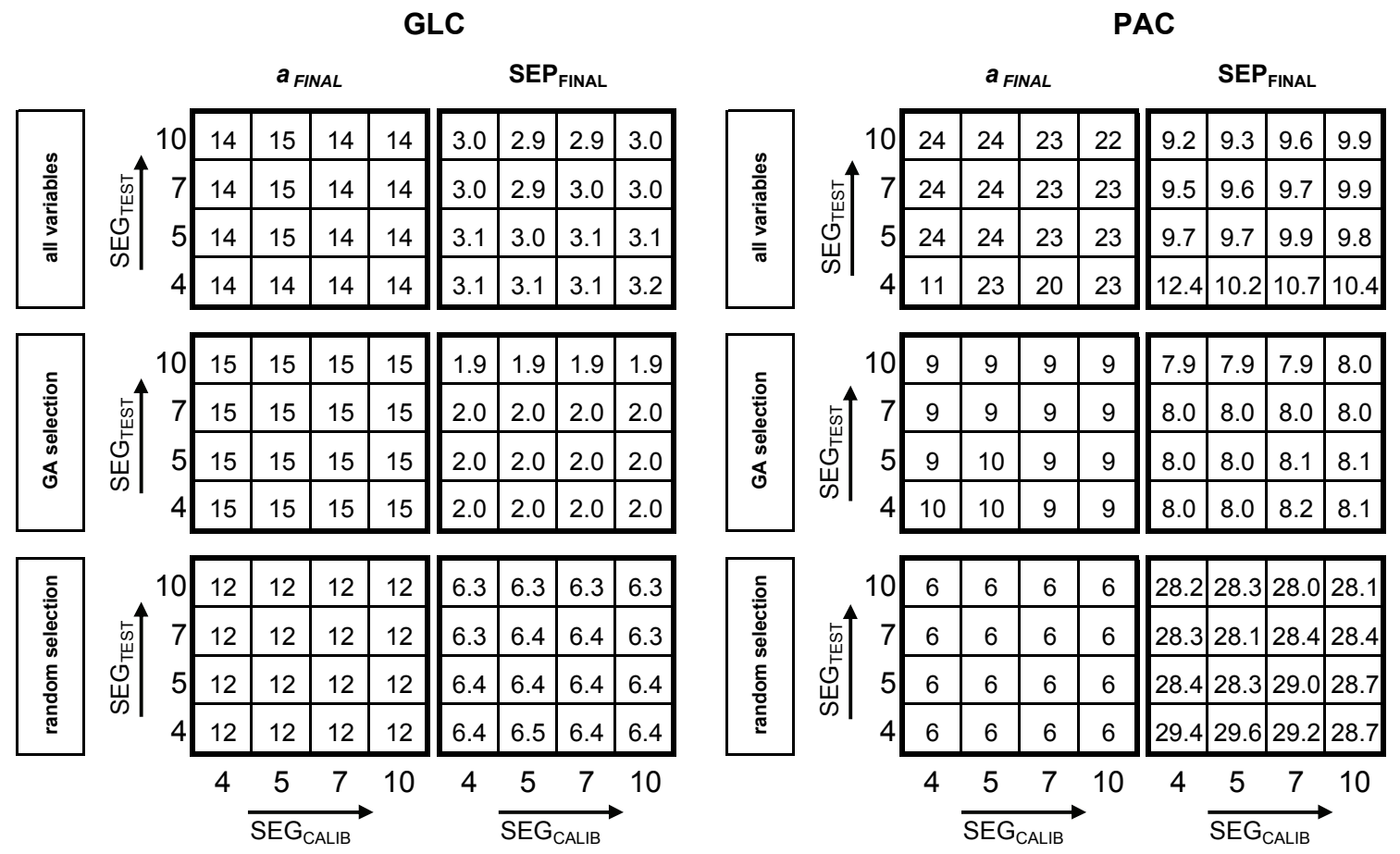

1
a_{MAX}
1
MSE
SEG_{CALIB}
m_{MSE}
s_{MSE}

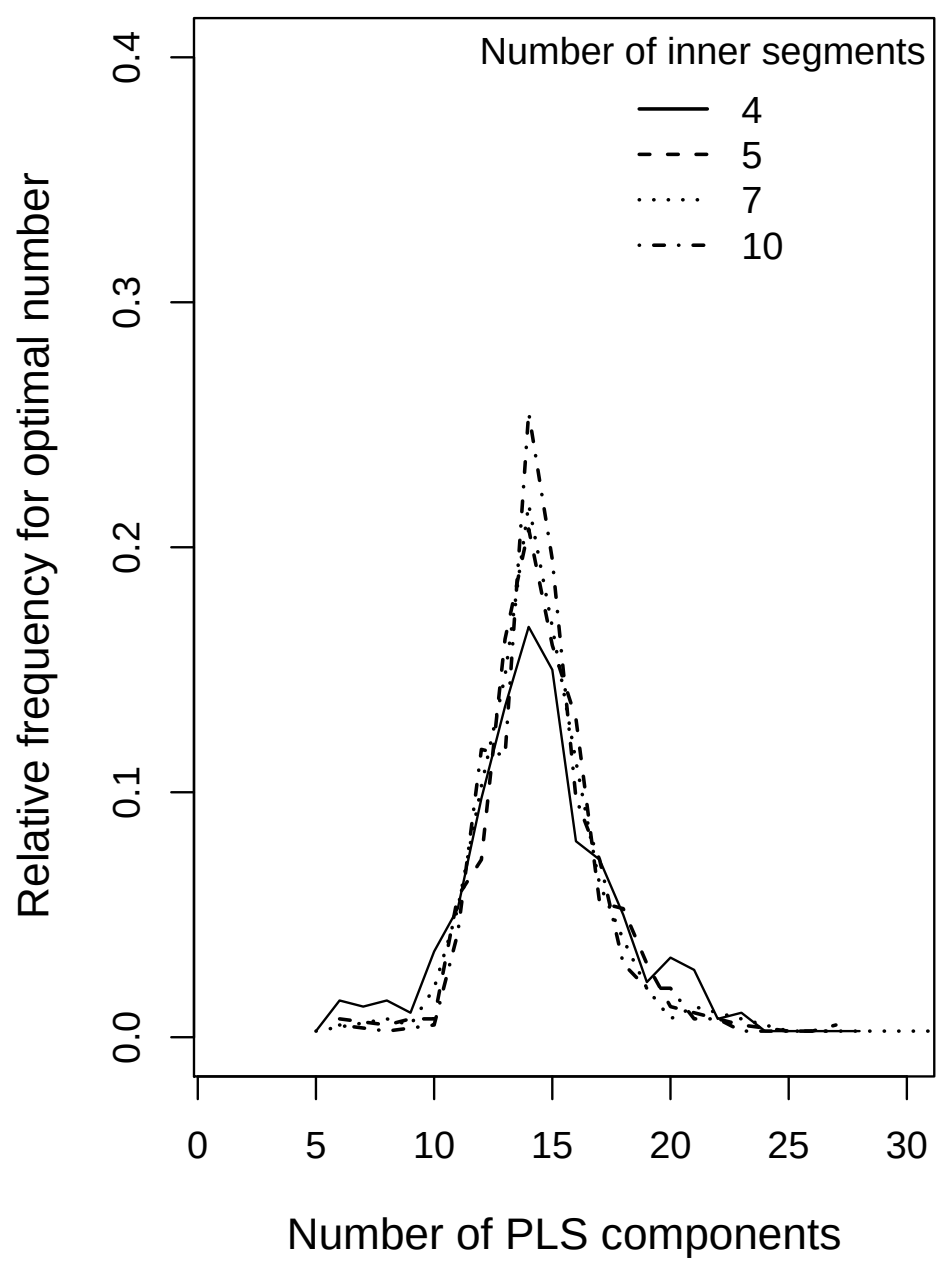

m_{MSE}
π = 1
π = 1.5
$_{MIN}\,m_{MSE}$
a
1
a_{MIN}
a_{MAX}
a_{OPT}
π = 1.5
a_{OPT}
π = 1

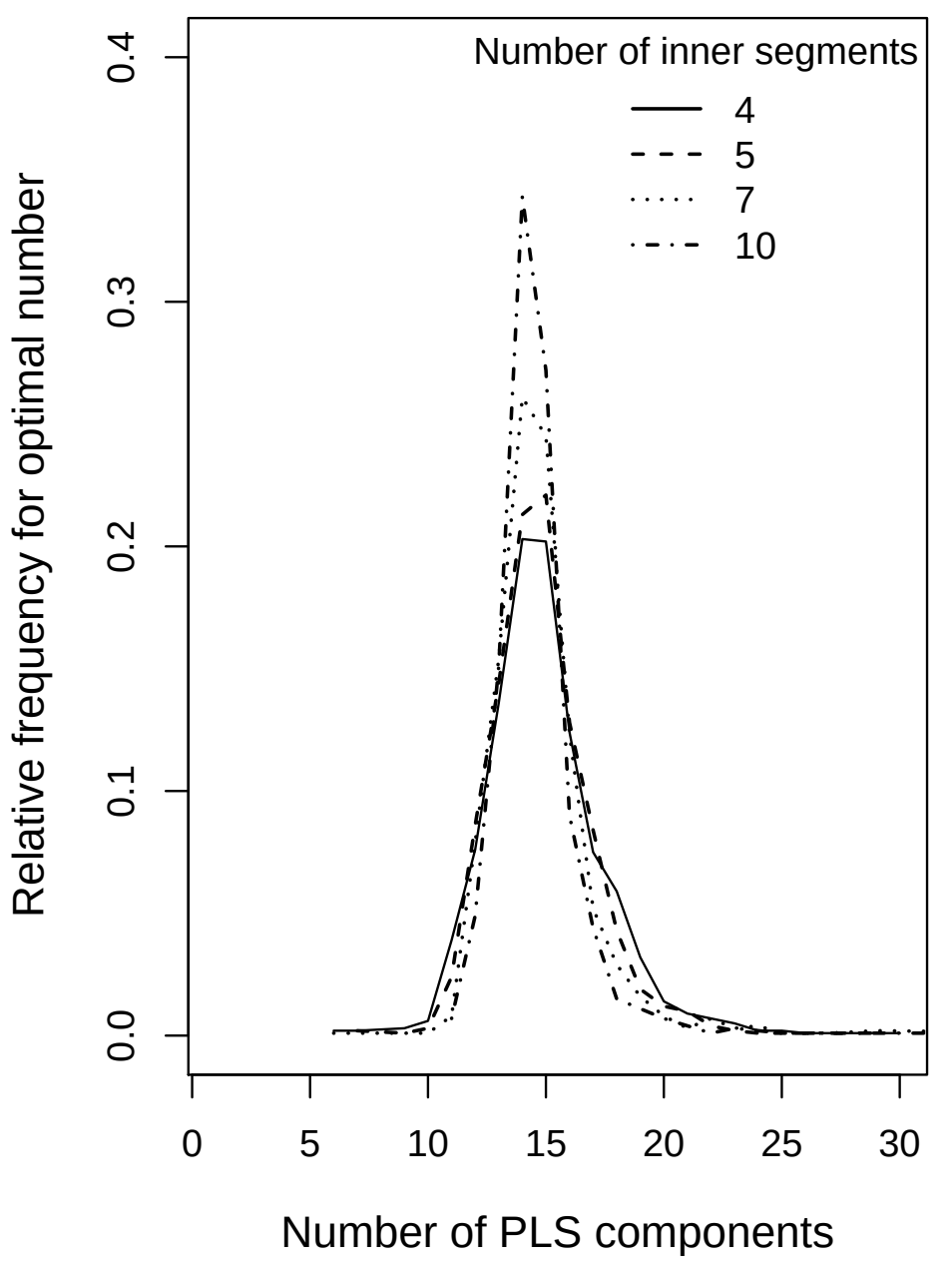

All data
X
y
1
τ =1
τ =2
τ =3
τ =4
n
Test set
Cali-bration set
Validation set
Train-ing set
PLS models with 1 ... a_{MAX} components
1
a_{MAX}
MSE (κ=1)
standard error method
a_{OPT}
PLS model with a_{OPT} components from calibration set
$\hat{y}(\rho)$
SEP(ρ)
Split for outer loop, SEG_{TEST} = 4
Split for inner loop, SEG_{CALIB} = 4
Inner loop (estimation of a_{OPT}), index κ
Outer loop (estimation of prediction errors), index τ
n_{REP} repetitions (index ρ)
n_{REP} × n test-set predicted $\hat{y}$
n_{REP} × SEG_{TEST} values for a_{OPT}
diagnostics plots, a_{FINAL}, SEP_{FINAL}

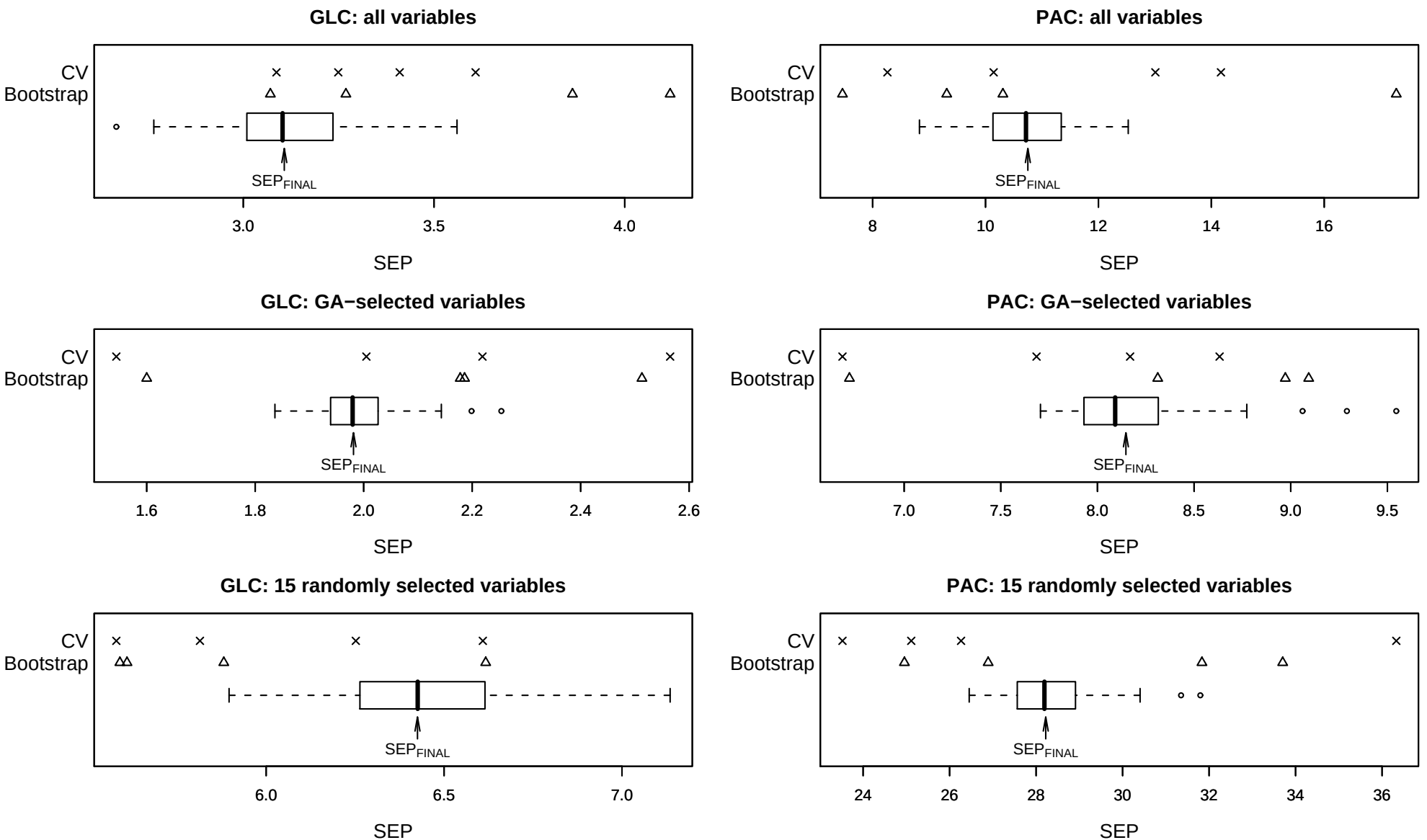

1
ρ
n_{REP}
1
τ
SEG_{TEST}
$a_{OPT}(\tau, \rho)$
relative frequency
0.00
0.05
0.10
0.15
0.20
10
15
20
25
a_{OPT}
a_{FINAL}

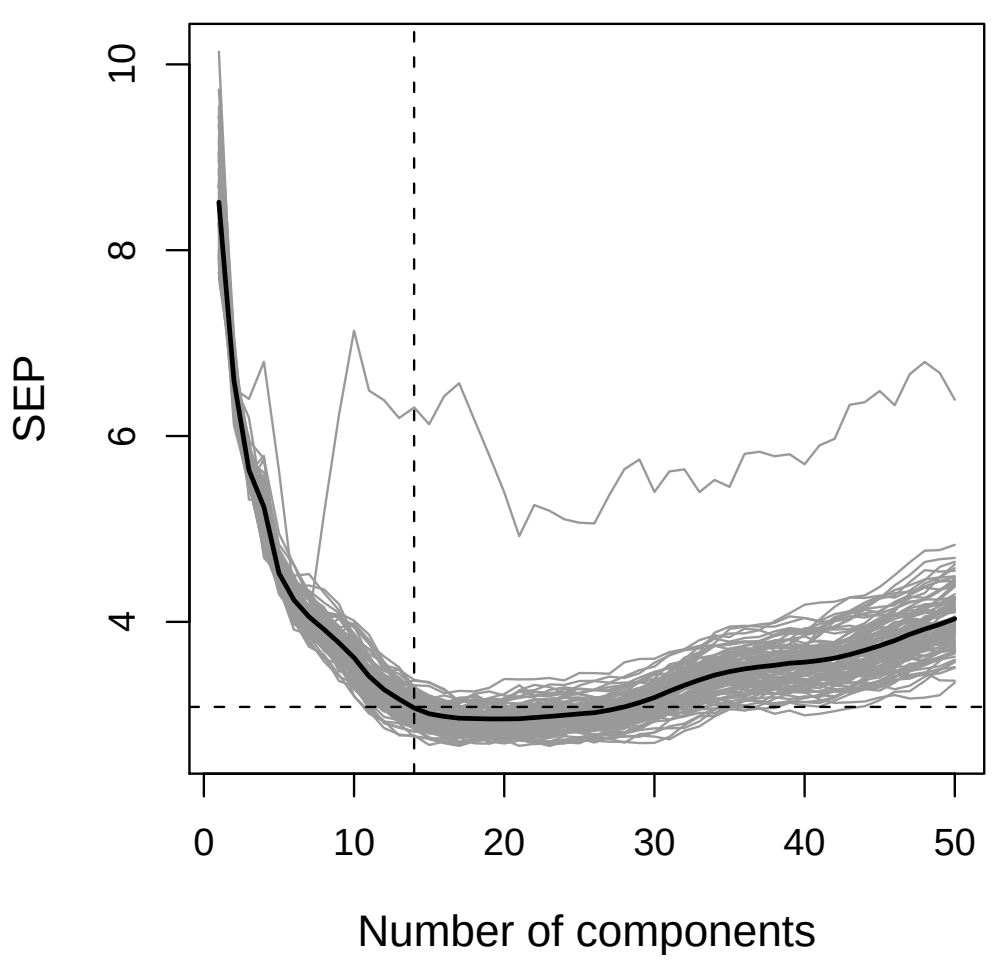

no. components
a_{MAX}
a_{FINAL}
1
1
objects i
$\hat{Y}$ or E
n
1 repetitions ρ n_{REP}
$a = a_{FINAL}$
distribution of residuals (prediction errors)
Density
0.00
0.10
0.20
0.30
−10
−5
0
5
10
Prediction error
e
TI_{95}
SEP for $\rho = 1$
and $a = 1, ..., a_{MAX}$
SEP
4
6
8
10
0
10
20
30
40
50
Number of components
1 repetitions ρ n_{REP}
1
$e\ (i, \rho)$
or
$\hat{y}\ (i, \rho)$
SEP_{FINAL}
residuals of object i for n_{REP} repetitions
i
n
distribution of SEP (boxplot)
SEP(ρ =1) . . . SEP($\rho = n_{REP}$)

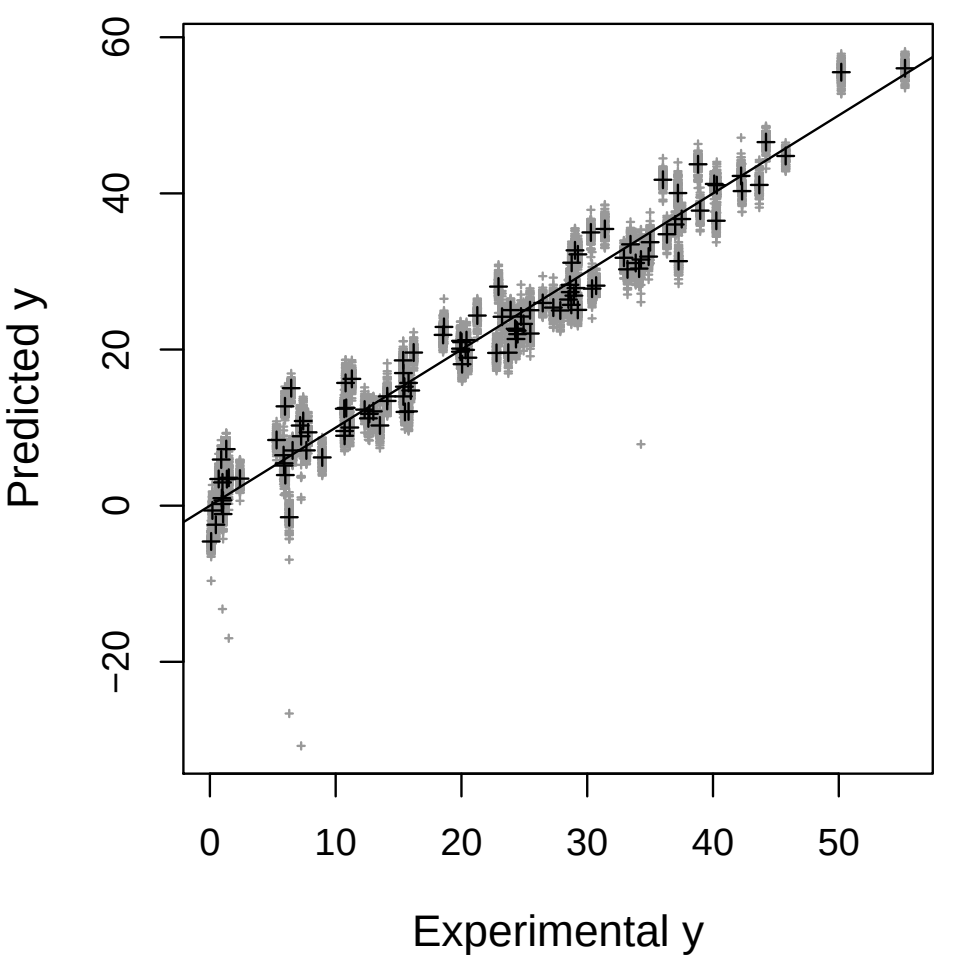

GLC
PAC
a FINAL
SEP FINAL
all variables
GA selection
random selection
SEG TEST
SEG CALIB
10
7
5
4
14 15 14 14 | 3.0 2.9 2.9 3.0
14 15 14 14 | 3.0 2.9 3.0 3.0
14 15 14 14 | 3.1 3.0 3.1 3.1
14 14 14 14 | 3.1 3.1 3.1 3.2
15 15 15 15 | 1.9 1.9 1.9 1.9
15 15 15 15 | 2.0 2.0 2.0 2.0
15 15 15 15 | 2.0 2.0 2.0 2.0
15 15 15 15 | 2.0 2.0 2.0 2.0
12 12 12 12 | 6.3 6.3 6.3 6.3
12 12 12 12 | 6.3 6.4 6.4 6.3
12 12 12 12 | 6.4 6.4 6.4 6.4
12 12 12 12 | 6.4 6.5 6.4 6.4
24 24 23 22 | 9.2 9.3 9.6 9.9
24 24 23 23 | 9.5 9.6 9.7 9.9
24 24 23 23 | 9.7 9.7 9.9 9.8
11 23 20 23 | 12.4 10.2 10.7 10.4
9 9 9 9 | 7.9 7.9 7.9 8.0
9 9 9 9 | 8.0 8.0 8.0 8.0
9 10 9 9 | 8.0 8.0 8.1 8.1
10 10 9 9 | 8.0 8.0 8.2 8.1
6 6 6 6 | 28.2 28.3 28.0 28.1
6 6 6 6 | 28.3 28.1 28.4 28.4
6 6 6 6 | 28.4 28.3 29.0 28.7
6 6 6 6 | 29.4 29.6 29.2 28.7
4 5 7 10
SEG CALIB

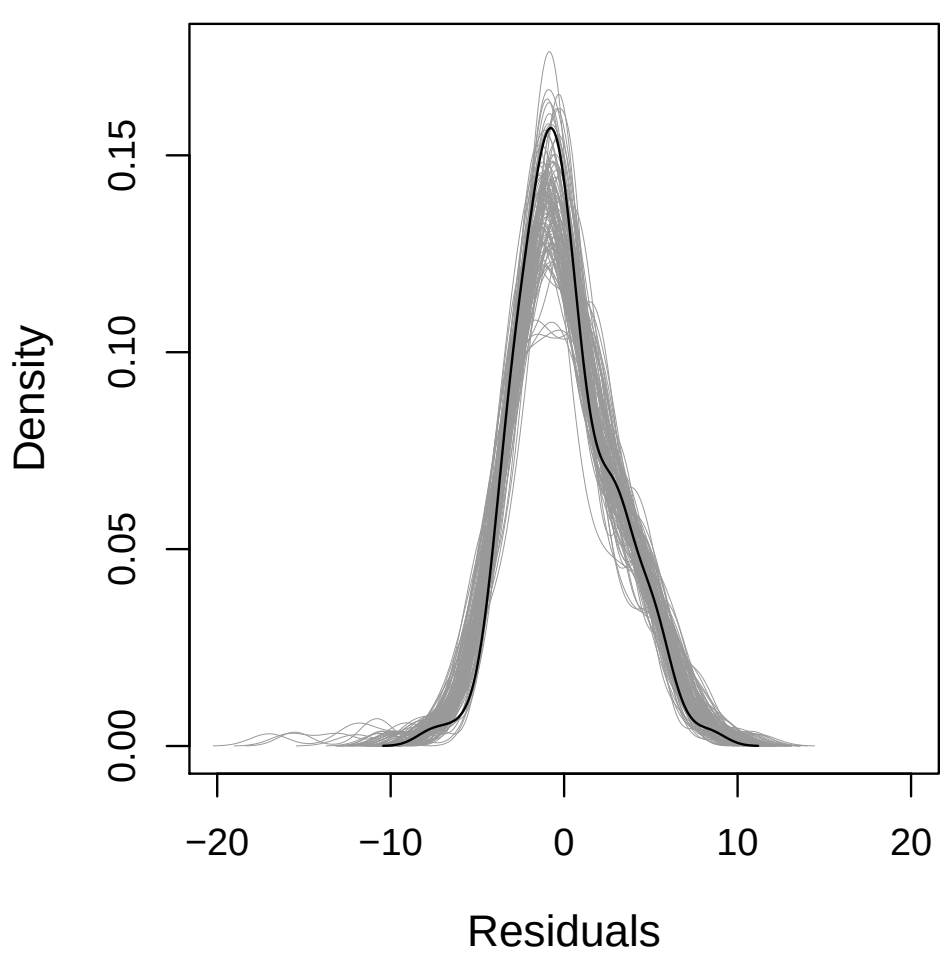

Number of inner segments
4
5
7
10
Relative frequency for optimal number
0.0
0.1
0.2
0.3
0.4
0
5
10
15
20
25
30
Number of PLS components

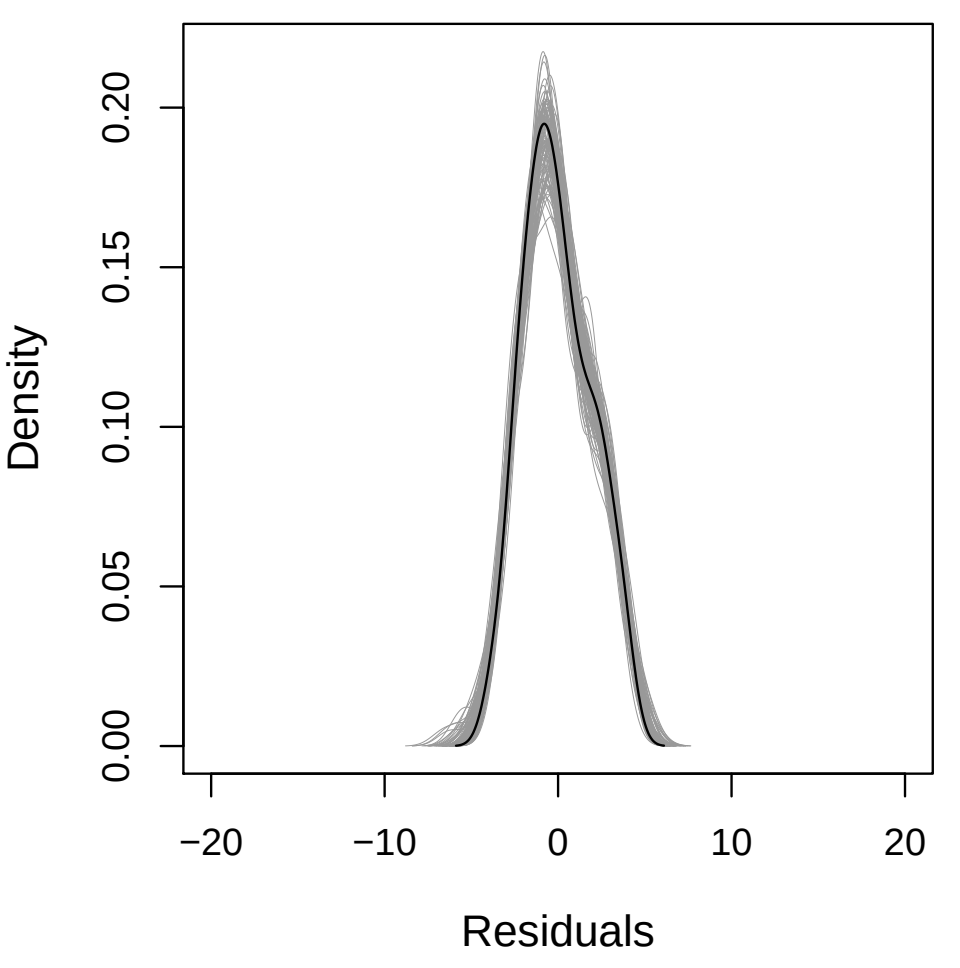

Number of inner segments
4
5
7
10
Relative frequency for optimal number
0.0
0.1
0.2
0.3
0.4
0
5
10
15
20
25
30
Number of PLS components

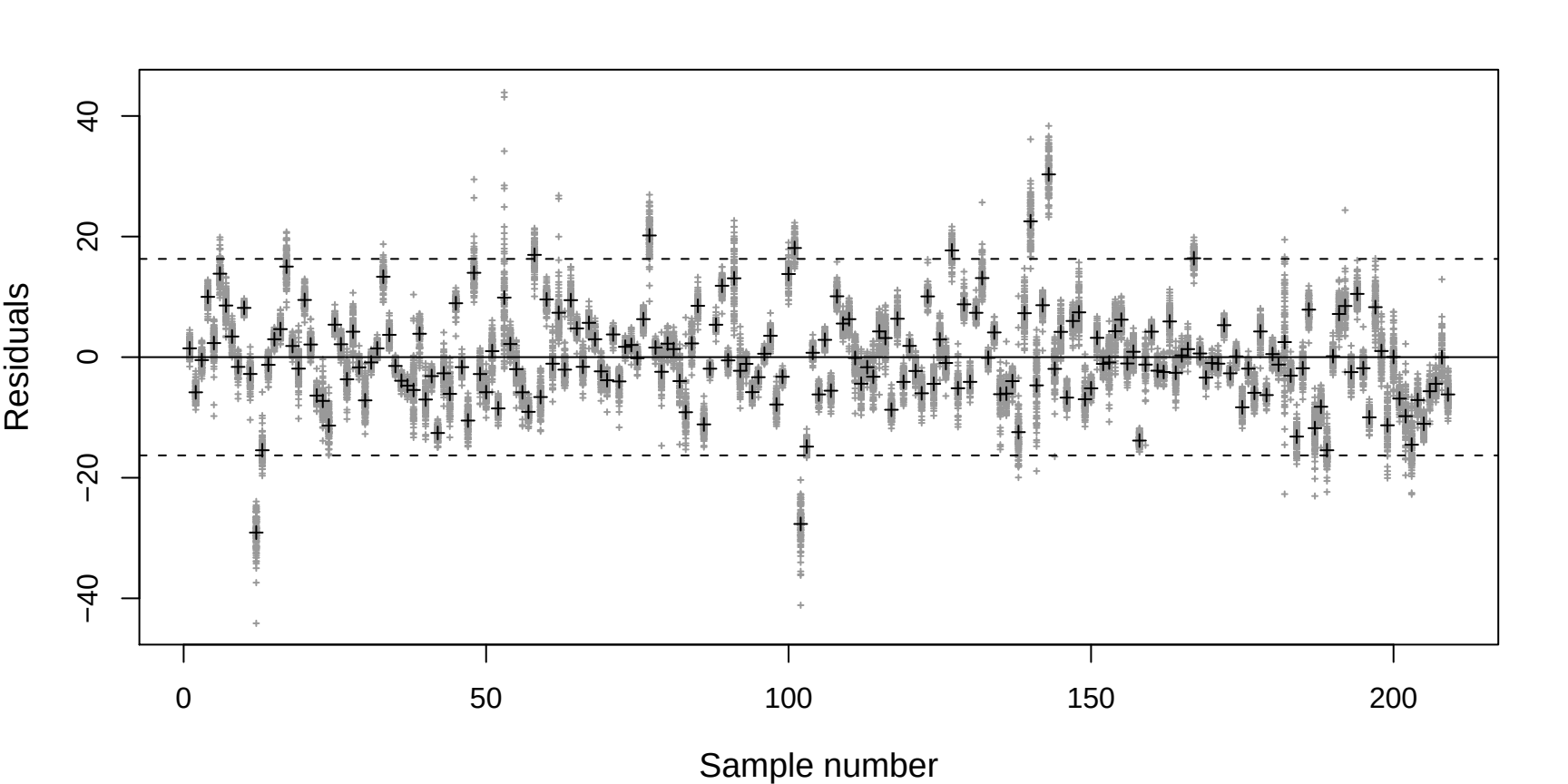

GLC: all variables
CV
Bootstrap
SEP_FINAL
3.0
3.5
4.0
SEP
PAC: all variables
CV
Bootstrap
SEP_FINAL
8
10
12
14
16
SEP
GLC: GA−selected variables
CV
Bootstrap
SEP_FINAL
1.6
1.8
2.0
2.2
2.4
2.6
SEP
PAC: GA−selected variables
CV
Bootstrap
SEP_FINAL
7.0
7.5
8.0
8.5
9.0
9.5
SEP
GLC: 15 randomly selected variables
CV
Bootstrap
SEP_FINAL
6.0
6.5
7.0
SEP
PAC: 15 randomly selected variables
CV
Bootstrap
SEP_FINAL
24
26
28
30
32
34
36
SEP


SEP
10
8
6
4
0
10
20
30
40
50
Number of components


Predicted y
60
40
20
0
−20
0
10
20
30
40
50
Experimental y


Density
0.00
0.05
0.10
0.15
−20
−10
0
10
20
Residuals


Density
0.00
0.05
0.10
0.15
0.20
−20
−10
0
10
20
Residuals


Residuals
40
20
0
−20
−40
0
50
100
150
200
Sample number