A QSPR example for R
GC retention index of PACs: User Guide

Supplementary material for

MULTIVARIATE LINEAR QSPR/QSAR MODELS.
RIGOROUS EVALUATION OF VARIABLE SELECTION FOR PLS

Kurt Varmuza [a,b*], **Peter Filzmoser** [b], **Matthias Dehmer** [c]

[a] Institute of Chemical Engineering, Vienna University of Technology, Austria;
[b] Department of Statistics and Probability Theory, Vienna University of Technology, Austria;
[c] Institute for Bioinformatics and Translational Research, UMIT - The Health and Life Sciences University, Hall in Tyrol, Austria

* Corresponding author: kvarmuza@email.tuwien.ac.at

Computational and Structural Biotechnology Journal, **5** [6], e201302007, 1-10 (2013)

http://journals.sfu.ca/rncsb/index.php/csbj/article/view/csbj.201302007/224
http://dx.doi.org/10.5936/csbj.201302007

(1) Introduction

The basic procedures mentioned in this *mini review* paper are described in this User Guide and all necessary programs (R scripts) and data are provided for download. R packages utilized are 'pls' and 'chemometrics' and the packages used therein. R software has been tested with R 2.15.2 (February 2013). Most of the provided R scripts have been written by an amateur (V.K.) - sorry for bad code. However, some 'R philosophy' has been intentionally violated: "=" instead of "<-" appears in many command lines, and the begin of the sources contains hints about the goal of the function, as well as short descriptions of input (arguments) and output (value) parameters. Your response about errors in the programs or other material is welcome (also about perhaps about useful applications ;-).

For details of the example, see the paper (PDF for free download).

(2) Import of a data set with descriptors generated by Dragon 6.0

File **`PAC209_3D_all_H.SDF`** contains 209 chemical structures [1] from PACs (polycyclic aromatic compounds), with approximate 3D atom coordinates and all H-atoms explicitly given, as created by software *Corina* (generation of data not documented here) [2].

File **`PAC209_dragon_2772.zip`** contains 3 files generated by software *Dragon* 6.0 (generation of data not documented here) [3]: (*n* = 209 chemical structures (objects); *m* = 2772 molecular descriptors (x-variables); constant descriptors excluded by *Dragon*).

Descriptors	**PAC209_dragon_2772.txt** (3.5 MB)
Descriptor (variable) names	**PAC209_dragon_2772_descriptors.txt**
Structure (object) names	**PAC209_dragon_2772_molecules.txt**

R-function **Dragon60_import()** reads the 3 Dragon output files (basic filename is **descr_file = "PAC209_dragon_2772"**), replaces missing values (encoded by -9999 in *Dragon*) by NA (for R), makes a matrix **x** (*n* objects × *m* descriptors, including descriptor/variable names and structure/object names as given in the Dragon output files) and saves **x** as an RData-file (**outfile**).

```
source("Dragon60_import.R")
descr_file ="PAC209_dragon_2772"
x = Dragon60_import(dragonfile=descr_file,outfile="PAC209_X_2772")
```

```
Start:  Dragon60_import 121210
Dragon-file set basic name:  PAC209_dragon_2772
Missing values ( -9999 ) replaced by NA: TRUE
Output file with descriptor matrix: PAC209_X_2772
Descriptor names file: PAC209_dragon_2772_descriptors.TXT read with 2772
descriptor names:
  Examples [1...3]:  MW AMW Sv
Molecular names file: PAC209_dragon_2772_molecules.TXT read with 209
molecule names:
  Examples [1...2]:  001 1,2-dihydronaphthalene 002 1,4-dihydronaphthalene
Descriptor file: PAC209_dragon_2772.TXT read with X( 209 x 2772 )
 -9999 for missing values replaced by NA
 0  NAs in descriptor matrix
Descriptor file made: PAC209_X_2772
End of Dragon60_import 121210    5.57 s

dim(x)
[1]  209 2772
```

Matrix **x** can be (later) loaded by

```
load("PAC209_X_2772.RData")      # giving the matrix object x
```

(3) Cleaning of descriptors data

The descriptors (variables) imported from the Dragon result files (see Chapter 1) contain constant or "almost constant" variables. They are deleted in this step as follows:

```
load("PAC209_X_2772.RData")       # giving the matrix object x
dim(x)
[1]  209 2772
source("varsel_almost_const.R")
sel = varsel_almost_const(X=x)   # gives logical vector 'sel'
                                 # with TRUE for selected variables
                                 # and FALSE for not selected ones

Start:  varsel_almost_const 130222 VK
Search for variables (columns in X) that are constant or almost constant.
 Almost constant means all values are constant except a maximum of k values
 (which are not NA).
 X( 209 x 2772 )   k = 3   round_to = 4 decimals
 84  variables deleted,  2688  variables remain. End.
```

```
sum(sel)
[1] 2688                    # number of selected variables
x = x[,sel]                 # select variables given in 'sel'
dim(x)
[1]  209 2688

save(x,file="PAC209_X_2688_demo.RData")   # demo save,
                                          # file PAC209_X_2688.RData
                                          # is already provided
                                          # in PAC209_X2688_y.zip
```

The provided file **PAC209_X2688_y.zip** contains:

PAC209_X_2688.RData	cleaned molecular descriptors as described above;
PAC209_y.txt	property data *y* (GC retention indices) to be modeled; input, e. g., with

```
y = scan("PAC209_y.txt",quiet=TRUE)
length(y)
[1] 209
```

PAC209_X_2688.RData	as matrix **x**	and
PAC209_y.txt	as vector **y**	are used for variable selection and model evaluation

(4) Variable selection: High correlation with *y*

Variables are selected which have highest squared correlation coefficient (Pearson, Spearman, or Kendall) with property *y*.
The (maximum) number of selected variables and/or the minimum value of the squared correlation coefficient can be defined (see comments in header of the R script for function **varsel_corr_xy()**).

```
load("PAC209_X_2688.RData")       # giving the matrix object x
dim(x)
[1]  209 2688
y = scan("PAC209_y.txt",quiet=TRUE)

source("varsel_corr_xy.R")
```

We select 50 variables with maximum squared Pearson correlation coefficient as follows:

```
sel = varsel_corr_xy(X=x,y=y,m_sel=50,corr_limit=0)
                                  # gives logical vector 'sel'
                                  # with TRUE for selected variables
                                  # and FALSE for not selected ones

Start:  varsel_corr_xy 121119 VK
Delete variables (columns in X) with too low corr. measure to y
Method: pearson
Max no. of selected variables = 50 (0=all)
Min correlation measure (squared corr. coeff.) with y = 0
X( 209 x 2688 )
 0 variables deleted with corr.measure < 0     2688 variables remaining
Final result of variable selection:
 2638 variables deleted,  50  variables remain. End.
```

```
sum(sel)
[1] 50                    # number of selected variables
```

For evaluation the performance of calibration models using a subset of variables (or all variables), see Chapter 8.

(5) Variable elimination: High correlation with another x-variable

Variables are eliminated which have a higher correlation to another variable than the given limit.

Method: All variable pairs are checked; if the correlation measure is above **`r2limit`**, then one of the variables is marked FALSE (it is the variable with the higher sum of the correlation measures to all variables). Criterion is a squared correlation coefficient (Pearson, Spearman, or Kendall) *y*.

Optionally a histogram of the distribution of the correlation measure for all variable pairs can be produced.

```
load("PAC209_X_2688.RData")       # giving the matrix object x
dim(x)
[1]  209 2688
y = scan("PAC209_y.txt",quiet=TRUE)

source("varsel_corr_xx.R")
```

We delete variables that have a squared Pearson correlation to another x-variable > 0.9 as follows:

```
sel = varsel_corr_xx(X=x,r2limit=0.9)
                                  # gives logical vector 'sel'
                                  # with TRUE for selected variables
                                  # and FALSE for not selected ones

Start:  varsel_corr_xx 130222 VK
Delete variables (columns in X) that have a squared corr. measure > 0.9
 with any other variable. Method: pearson
 X( 209 x 2688 )
 1984  variables deleted,  704  variables remain. End.

sum(sel)
[1] 704                    # number of selected variables
```

Note that computation time for this example may be almost 1 minute.
For evaluation the performance of calibration models using a subset of variables (or all variables), see Chapter 8.

(6) Variable selection: High absolute regression coefficient in PLS model

Variables are selected which have highest absolute standardized regression coefficients (*b*) in a PLS model from all objects. The number of PLS components is optimized by rdCV (repeated double cross validation), or can be defined.

The (maximum) number of selected variables and/or the minimum value of *b* can be defined (see examples below).

The limit for *b* can be separately defined for negative an positive values (vector **`regr_coeff_limit[1:2]`**) as follows:

`regr_coeff_limit[1]` lower limit, for negative *b*, variable is deleted if *b* is between 0 and **`regr_coeff_limit[1]`**

`regr_coeff_limit[2]` higher limit, for positive *b*, variable is deleted if *b* is between 0 and **`regr_coeff_limit[2]`**

In other words: variables with *b* in the interval (exclusive) **`regr_coeff_limit[1]`** to **`regr_coeff_limit[2]`** are deleted.

Values 0 and 0 are used for 'no limits'.

See examples in comments of header in R script of function **`varsel_pls_regr_coeff()`**.

For rdCV default parameters can be used or can be defined by the user.

Optionally, a PDF with rdCV results (diagnostic plots) can be produced.

R packages 'chemometrics' and 'pls' are necessary (available via CRAN).

```
load("PAC209_X_2688.RData")       # giving the matrix object x
dim(x)
[1]  209 2688
y = scan("PAC209_y.txt",quiet=TRUE)

source("varsel_pls_regr_coeff.R")
```

We select 50 variables which have maximum |b|, using default parameters, as follows:

```
sel = varsel_pls_regr_coeff(X=x,y=y,m_sel=50)
                                  # gives logical vector 'sel'
                                  # with TRUE for selected variables
                                  # and FALSE for not selected ones

Start:  varsel_pls_regr_coeff 130222 VK
Delete variables (columns in X) with too low absolute regr. coeff. (b)
   in rdCV-optimized PLS model
Max no. of selected variables = 50 (0=all)
Limits for b: for a negative b the variable is deleted if b > 0
              for a positive b the variable is deleted if b < 0
rdCV parameter: amax = 10  repetitions = 50
                seg_test = 3  seg_calib = 5   parsimonity = 1
X( 209 x 2688 )
rdCV results: a_final = 6  with SEP = 7.226086
PLS with all autoscaled X-data and 6 components yields: SEC = 5.041192
Calibration prediction errors: -17.65422 to 17.38644
```

```
           mean (bias_calib): 1.783044e-14
Regression coeff. (from autoscaled X): -0.2192 to 0.3401  mean = 0.03142
 Quantiles 0.1, 0.2, 0.8, 0.9 = -0.04759412 -0.02050018 0.07832596
0.09636999
 0 variables deleted with b outside defined ranges, 2688 variables
remaining
Final result of variable selection:
 2638 variables deleted,  50  variables remain. End.
End of  varsel_pls_regr_coeff 130222 VK     179.33 s

sum(sel)
[1] 50                    # number of selected variables
```

Note that computation time for this example is ca 3 minutes.

Remarks to rdCV: By default parameter, a maximum of 10 PLS components has been considered; actually 6 (**a_final**) have been found to be optimal; SEP is 7.2 (for test set objects, mean of 50 repetitions); SEC is 5.0 (final model with 6 PLS components from all objects applied to the same objects).

For evaluation the performance of calibration models using a subset of variables (or all variables), see Chapter 8.

(7) Variable selection: Stepwise (forward or forward/backward)

Variables are selected by the traditional stepwise strategy, either in forward manner or by a combination of forward and backward ("both"). The criterion used is BIC. A new R function has been developed allowing stepwise variable selection with more than 2000 variables (and/or more variables than objects) in reasonable computing time.

Optionally, results for all steps can be saved in an RData-file.
A plot with BIC versus step number is produced.

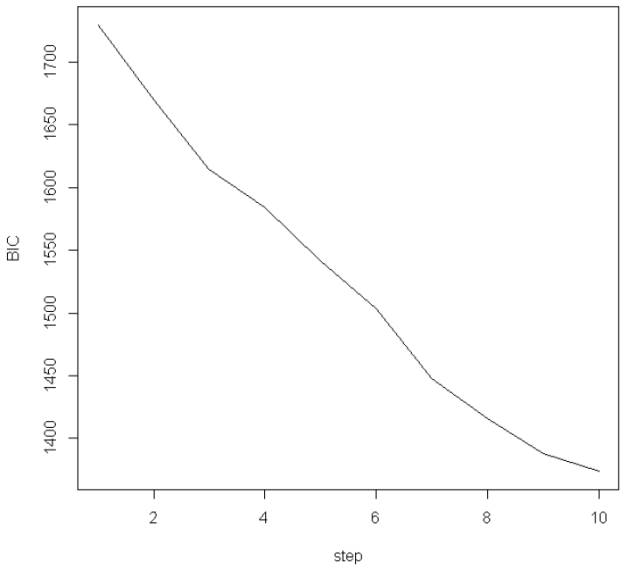
```
load("PAC209_X_2688.RData") # giving the matrix object x
dim(x)
[1] 209 2688
y = scan("PAC209_y.txt",quiet=TRUE)

source("varsel_stepwise_BIC.R")
```

We use default parameters (mode is "both", maximum computing time is 200 s, maximum number of steps is 20):

```
sel = varsel_stepwise_BIC (X=x,y=y)
                                  # gives logical vector 'sel'
                                  # with TRUE for selected variables
                                  # and FALSE for not selected ones

Start: varsel_stepwise_BIC3 130222 VK
Stepwise (forward/backward) selection of variables (columns in X).
Method as implemented in 'stepforward' and 'stepboth' criterion = BIC
Mode = both max comp. time = 200 max no. of steps = 20
X( 209 x 2688 ) y: 209
*** Start stepboth. Step no. (time):
```

```
1 ( 0 ) 2 ( 17 ) 3 ( 35 ) 4 ( 54 ) 5 ( 74 )
6 ( 94 ) 7 ( 116 ) 8 ( 139 ) 9 ( 162 ) 10 ( 186 )
Result of stepwise variable selection: 10 steps net, 212 s
2678 variables deleted, 10 variables selected.
Results from stepwise selection written to file: r_step_BIC.RData
End of varsel_stepwise_BIC3 130222 VK 214.88 s
>
```

The job was terminated by the limit for the computation time, and only 10 steps have been performed. The plot "BIC versus step number" shows that the minimum of BIC has not been reached and a longer computation time (and possible also more steps than 20) should be allowed.



For evaluation the performance of calibration models using a subset of variables (or all variables), see Chapter 8.

(8) Performance of a calibration model

Modeling power of the original variable set and the variable subsets obtained by variable selection has been evaluated by repeated double cross validation (rdCV) [4,5]; see the review paper. R software and a short description for rdCV is available via http://www.lcm.tuwien.ac.at/R/rdCV.zip

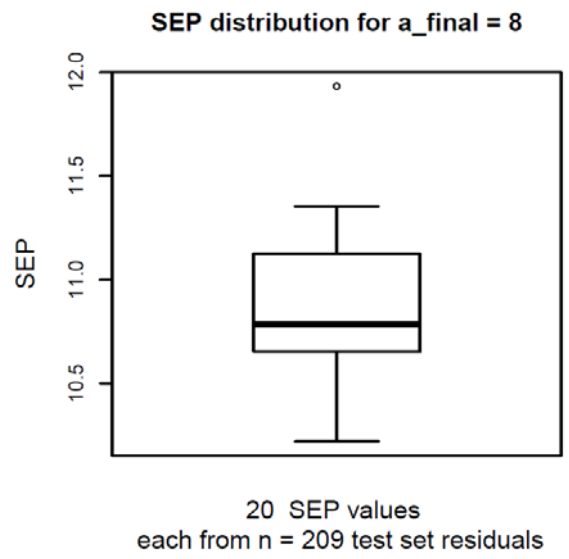
Here is presented a comparison of models with (a) all m = 2688 variables, and (b) models with m = 10 variables obtained by stepwise selection (see Chapter 7; note this is not the best variables subset obtainable by this method) is shown.

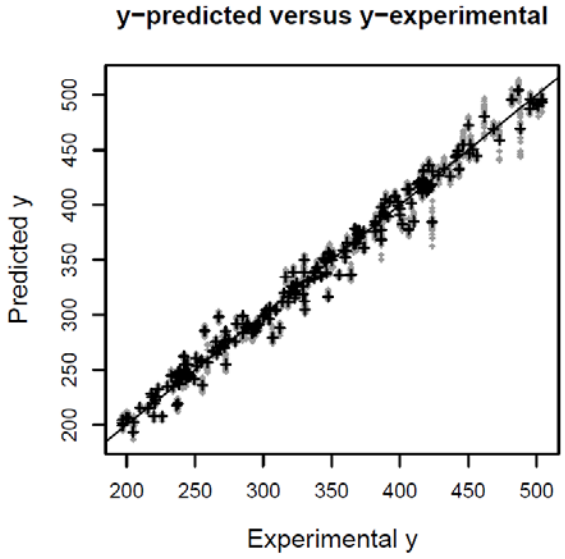
First the stepwise selection is performed (mode is "both", maximum computing time is 200 s, maximum number of steps is 20) resulting in a variable set **`x_stepwise`** (209 × 10).

Then rdCV is applied to the data set with all m = 2688 variables, and then to the data set with m = 10 selected variables. For parameters used in rdCV see the rdCV documentation. Each rdCV run produces a PDFs with diagnostic plots. Selected plots and results are shown here.

```
load("PAC209_X_2688.RData")           # giving the matrix object x
y = scan("PAC209_y.txt",quiet=TRUE)   # dependent variable y

source("varsel_stepwise_BIC.R")
sel_stepwise = varsel_stepwise_BIC (X=x,y=y)  # stepwise selection
```

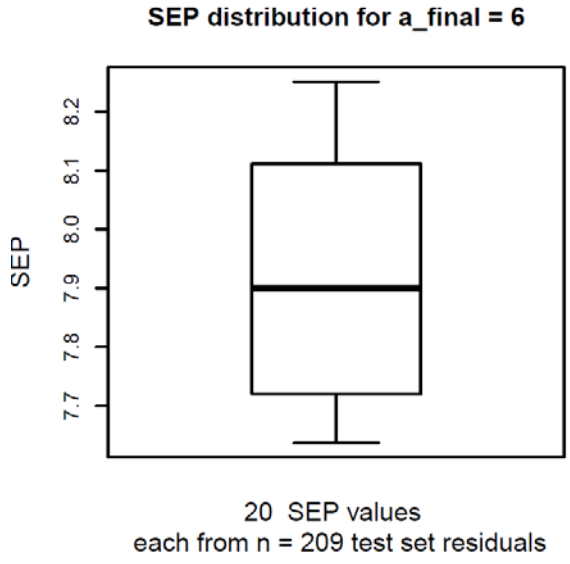
```
Start:  varsel_stepwise_BIC3 130222 VK
Stepwise (forward/backward) selection of variables (columns in X).
Method as implemented in 'stepforward' and 'stepboth' criterion = BIC
  Mode = both   max comp. time = 200   max no. of steps = 20
X( 209 x 2688 )  y: 209
*** Start stepboth. Step no. (time):
1 ( 0 ) 2 ( 16 ) 3 ( 33 ) 4 ( 51 ) 5 ( 70 )
6 ( 90 ) 7 ( 111 ) 8 ( 132 ) 9 ( 155 ) 10 ( 178 )
Result of stepwise variable selection: 10 steps net,  203 s
 2678 variables deleted,  10 variables selected.
Results from stepwise selection written to file: r_step_BIC.RData
End of  varsel_stepwise_BIC3 130222 VK     205.34 s

x_stepwise = x[,sel_stepwise]                  # reduced x-data
dim(x_stepwise)
[1] 209  10

source("go_rdcv.R")

res_all =
go_rdcv(title="all",X=x,y=y,amax=8,repetitions=20,PDFfile="rdCV_plot
s_all.PDF")

===  go_rdcv 100907a start    Mon Feb 25 10:36:59 2013
 Title used:  all
 X:  209  rows    2688 cols
 y:  209  values    197.01 to 503.91    sd= 80.7652
 no. of PLS comp.: 8  desired    8  computed
PDFfile  rdCV_plots_all.PDF  opened
PDFfile closed
=== End of go_rdcv  go_rdcv 100907a     121.04 s
```

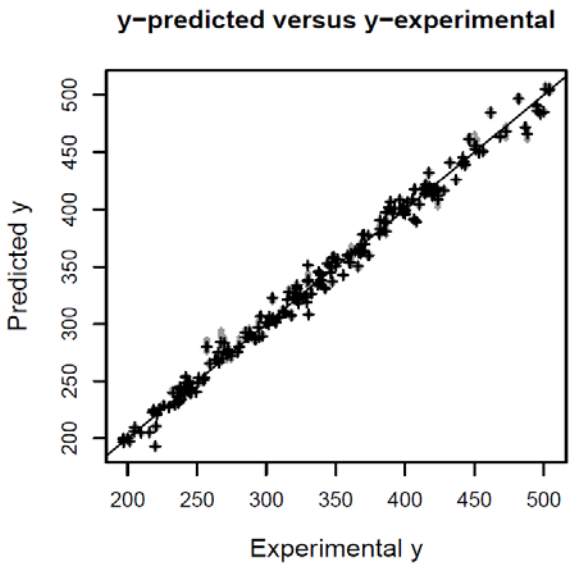





```
res_stepwise =
go_rdcv(title="stepwise",X=x_stepwise,y=y,amax=8,repetitions=20,PDFf
ile="rdCV_plots_stepwise.PDF")

===  go_rdcv 100907a start    Mon Feb 25 10:45:20 2013
 Title used:  stepwise
 X:  209  rows    10 cols
 y:  209  values    197.01 to 503.91    sd= 80.7652
 no. of PLS comp.: 8  desired    8  computed
PDFfile  rdCV_plots_stepwise.PDF  opened
```

```
PDFfile closed
=== End of go_rdcv  go_rdcv 100907a     2.75 s
```





Remark: In the paper a function **`rdcv_pls()`** is mentioned that simply combines the extraction of selected variables and rdCV; however, this function is not included here.

Summary of this (demo) comparison

all variables	m = 2688	SEP_{FINAL} = 10.9	a_{FINAL} = 8
stepwise variable selection	m = 10	SEP_{FINAL} = 7.9	a_{FINAL} = 6

SEP_{FINAL} (standard deviation of prediction errors for test set objects within rdCV) is the mean of 20 repetitions, each from n = 207 values. Distributions of the 20 SEP values are shown by box plots. a_{FINAL} is the estimated optimum number of PLS components.

Variable selection was successful: models with selected 10 variables are considerably better than models with 2688 variables.

(9) Final PLS model

In Chapter 8 was estimated:
- optimum number of PLS components (a_{FINAL} = 6 for the 10 variables obtained by stepwise selection),
- performance for prediction of new objects (SEP_{FINAL} = 7.9 for the 10 variables obtained by stepwise selection).

We now make a 'final PLS model' from all available (n = 209) objects with a_{FINAL} = 6 PLS components for this data X(209 × 10). In this step no further optimization of the number of PLS components must be done, and the obtained performance, SEC, standard error of calibration (for fit rather than prediction), has only informative character.

We start from the original data set with *m* = 2688 variables.

```
load("PAC209_X_2688.RData")           # giving the matrix object x
y = scan("PAC209_y.txt",quiet=TRUE)   # dependent variable y
```

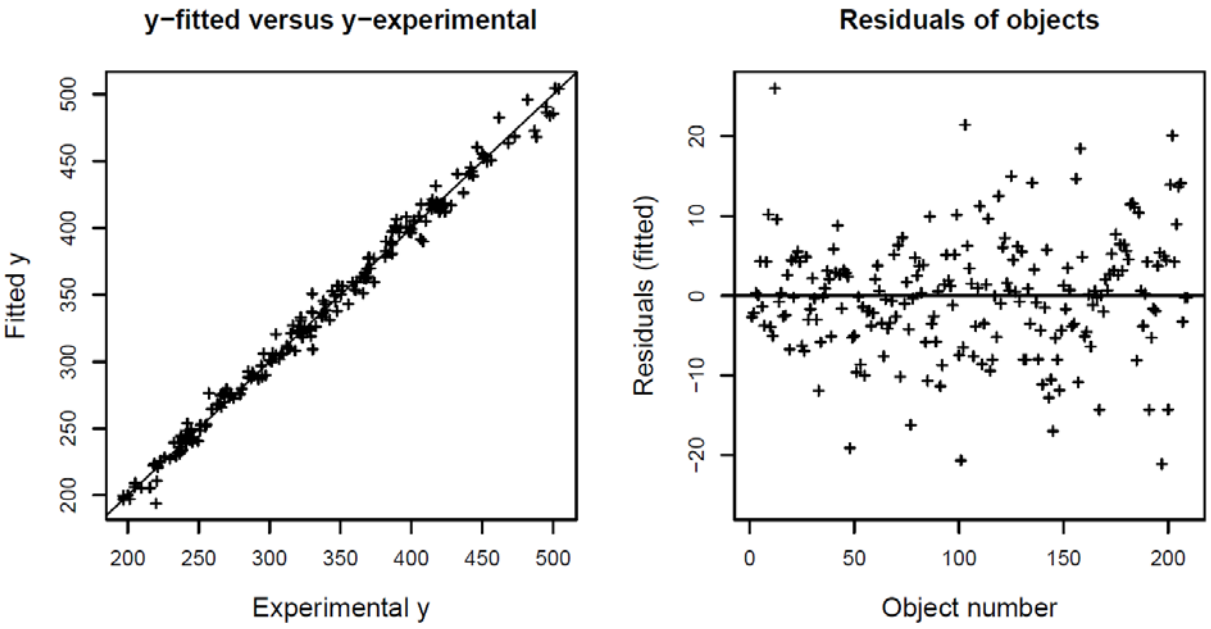
Next, we make stepwise variable selection as described above.

```
source("varsel_stepwise_BIC.R")
sel_stepwise = varsel_stepwise_BIC (X=x,y=y)  # stepwise selection
x_stepwise = x[,sel_stepwise]                 # reduced x-data
dim(x_stepwise)
[1] 209  10
```

Finally, we make the 'final PLS model' (see source code for parameter description).

```
source("pls_one_model.R")
pls_model=pls_one_model(title="PAC_stepwise_m10",X=x_stepwise,y=y,a_
final=6,scale=FALSE,PDFfile="PAC_stepwise_m10_final_model.PDF")

Start:  pls_one_model 130226 VK
Title: PAC_stepwise_m10
X: 209 x 10    y: 209 values
 No scaling of X before PLS
PLS with all objects and 6 components made:
 SEC = 7.3585    R2calib = 0.991699
 Calibration prediction errors: -21.11361 to 26.03672
             mean (bias_calib): 1.990516e-14
 Intercept (b0) = 338.0862
 Regression coeff.: -9.364 to 20.3  mean = 2.782
PDFfile  PAC_stepwise_m10_final_model.PDF  opened, closed.
End of  pls_one_model 130226 VK     0.05 s
```

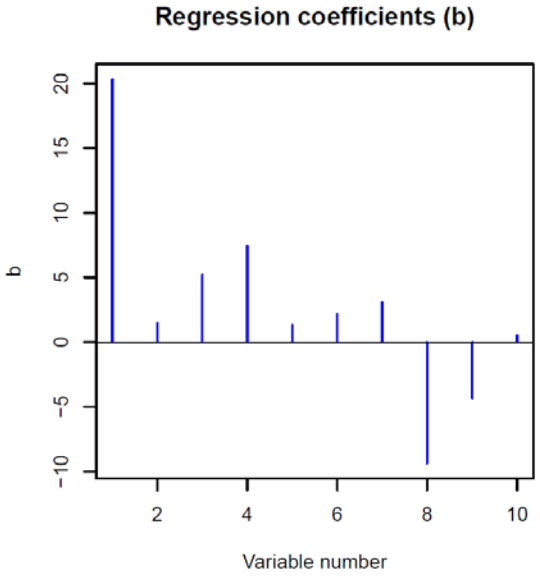
As expected, SEC = 7.4 is somewhat smaller (too optimistic) than SEP_{FINAL} = 7.9 from rdCV. The squared Pearson correlation coefficient between experimental *y* and fitted *y* is 0.9917. Here are plots from the resulting PDF (**PAC_stepwise_m10_final_model.PDF**):



The names (from Dragon software) of the selected 10 descriptors are:

```
colnames(x_stepwise)
[1] "HyWi_Dt"     "P_VSA_s_5"   "SpAD_EA(bo)" "Mor06m"      "Mor02e"
 [6] "E1p"         "H4m"         "O-060"       "NsssCH"      "TPSA(Tot)"
```

The corresponding regression coefficients (not standardized!) are plotted in the PDF as



The output of **`pls_one_model()`** is a list (**`pls_model`**) with the model parameters for further use of the model (see comments in the R script):

```
str(pls_model)
List of 17
 $ title     : chr "PAC_stepwise_m10"
 $ b0        : num 338
 $ b         : Named num [1:10] 20.3 1.47 5.19 7.46 1.32 ...
  ..- attr(*, "names")= chr [1:10] "HyWi_Dt" "P_VSA_s_5" "SpAD_EA(bo)"
"Mor06m" ...
 $ Xscale    : logi FALSE
 $ Xmean     : num [1:10] 0 0 0 0 0 0 0 0 0 0
 $ Xsd       : num [1:10] 1 1 1 1 1 1 1 1 1 1
 $ SEC       : num 7.36
 $ bias_calib: num 1.99e-14
 $ R2calib   : num 0.992
 $ n         : int 209
 $ m         : int 10
 $ X         : num [1:209, 1:10] 7.33 7.33 7.33 7.33 6.9 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:209] "001-corr.mol" "002-corr.mol" "003.mol" "004.mol"
...
  .. ..$ : chr [1:10] "HyWi_Dt" "P_VSA_s_5" "SpAD_EA(bo)" "Mor06m" ...
 $ y         : num [1:209] 197 197 197 200 201 ...
 $ y_fit     : Named num [1:209] 200 199 197 200 197 ...
  ..- attr(*, "names")= chr [1:209] "1" "2" "3" "4" ...
 $ a_final   : num 6
 $ PLSmode   : chr "simpls"
 $ origin    : chr "pls_one_model 130226 VK Tue Feb 26 11:39:03 2013"
```

1. Lee ML, Vassilaros DL, White CM, Novotny M (1979) Retention indices for programmed-temperature capillary-column gas chromatography of polycyclic aromatic hydrocarbons. Anal Chem 51: 768-773.
2. Corina (2004) Software for the generation of high-quality three-dimensional molecular models, by Sadowski J, Schwab CH, Gasteiger J. Erlangen, Germany: Molecular Networks GmbH Computerchemie, www.molecular-networks.com.
3. Dragon (2010) Software for molecular descriptor calculation, version 6.0, by Todeschini R, Consonni V, Mauri A, Pavan M. Milan, Italy: Talete srl, www.talete.mi.it.
4. Filzmoser P, Liebmann B, Varmuza K (2009) Repeated double cross validation. J Chemometrics 23: 160-171.
5. Varmuza K, Filzmoser P (2009) Introduction to multivariate statistical analysis in chemometrics. Boca Raton, FL, USA: CRC Press.

--- End of User Guide ---