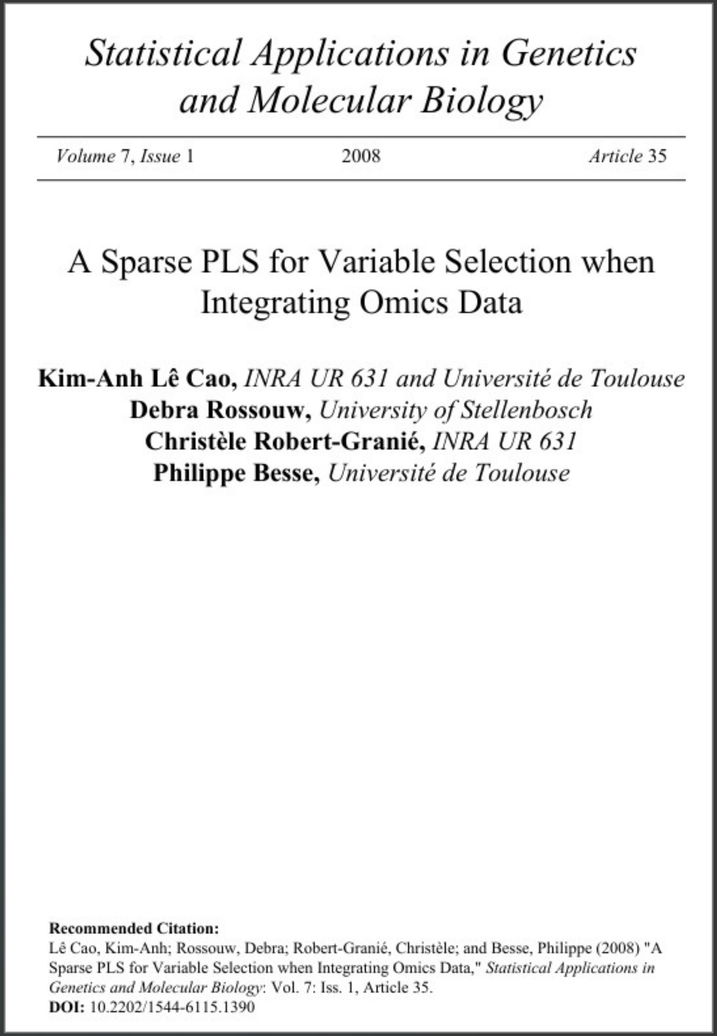
Sparse PLSR by regularized SVD

matthieu.lesnoff@cirad.fr

ChemHouse, Montpellier, 30 Sept 2025



Statistical Applications in Genetics and Molecular Biology

Volume 7, Issue 1 2008 *Article 35*

A Sparse PLS for Variable Selection when Integrating Omics Data

Kim-Anh Lê Cao, *INRA UR 631 and Université de Toulouse*
Debra Rossouw, *University of Stellenbosch*
Christèle Robert-Granié, *INRA UR 631*
Philippe Besse, *Université de Toulouse*

Recommended Citation:
Lê Cao, Kim-Anh; Rossouw, Debra; Robert-Granié, Christèle; and Besse, Philippe (2008) "A Sparse PLS for Variable Selection when Integrating Omics Data," *Statistical Applications in Genetics and Molecular Biology*: Vol. 7: Iss. 1, Article 35.
DOI: 10.2202/1544-6115.1390

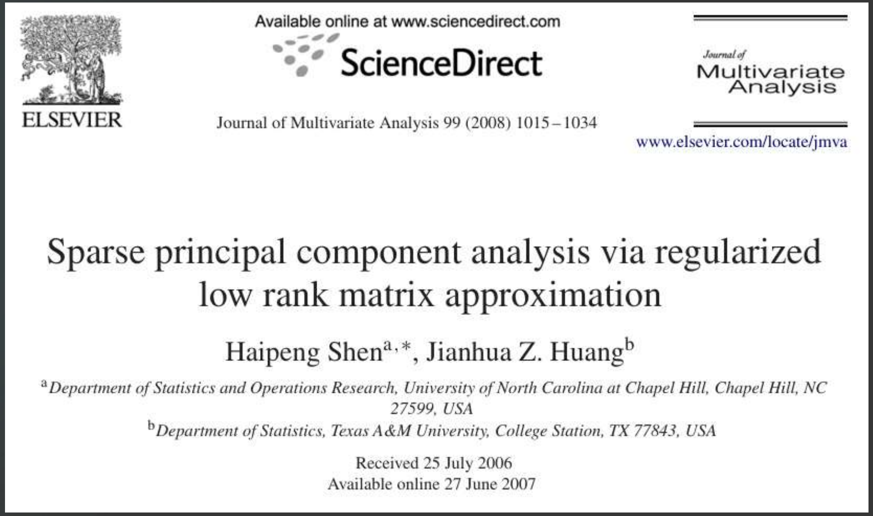
R function `mixOmics::spls`

Lê Cao, K.-A., Rohart, F., Gonzalez, I., Dejean, S., Abadi, A.J., Gautier, B., Bartolo, F., Monget, P., Coquery, J., Yao, F., Liquet, B., 2022. mixOmics: Omics Data Integration Project. https://doi.org/10.18129/B9.bioc.mixOmics

In brief about the method

- Use of a **regularized** (**instead of usual**) SVD in the PLS algorithm

rSVD-sPCA



Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Journal of Multivariate Analysis 99 (2008) 1015–1034

Journal of Multivariate Analysis

www.elsevier.com/locate/jmva

Sparse principal component analysis via regularized low rank matrix approximation

Haipeng Shen[a,*], Jianhua Z. Huang[b]

[a] Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

[b] Department of Statistics, Texas A&M University, College Station, TX 77843, USA

Received 25 July 2006
Available online 27 June 2007

PLS $\mathbf{X\ (n, p), Y\ (n, q)}$

- Scores $(n, 1)$ $t_X = X\, w_X$ $t_Y = Y\, w_Y$
- Loading weights w_X $(p, 1)$, w_Y $(q, 1)$

$$t_X = w_{X,1}\, x_1 + w_{X,2}\, x_2 + \dots + w_{X,p}\, x_p$$

$$t_Y = w_{Y,1}\, y_1 + w_{Y,2}\, y_2 + \dots + w_{Y,q}\, y_q$$

w_X, w_Y such as

$\max \operatorname{cov}^2(t_X, t_Y)$ with $\|w_X\| = \|w_Y\| = 1$

A usual PLS algorithm

Iterative NIPALS to get w_X and w_Y:

```
1. Set $t_Y$
2. Repeat until convergence
    a) $w_X = X' t_Y$
    b) $w_X = w_X / \mathrm{norm}(w_X)$
    c) $w_Y = Y' t_X / t_X' t_X$
    d) $w_Y = w_Y / \mathrm{norm}(w_Y)$
    e) $t_Y = Y w_Y$
3. $t_X = X w_X$
```

But w_X and w_Y can be computed without iteration:

- Let $K = Y'X$ (could consider $X'Y$)
- SVD of $K \Rightarrow U \Delta V'$
- $w_X = V[:, 1]$
- $w_Y = U[:, 1]$

e.g., Höskuldsson 1988 Journal of Chemometrics
https://doi.org/10.1002/cem.1180020306

(used in Jchemo: plskern, plsnipals, plscan)

A simple approach to get sparse w_x

Usual PLS

- $K = Y'X$
- SVD of $K \Rightarrow U\,\Delta V'$
- $w_X = V[:, 1]$

Sparse PLS

- $K = Y'X$
- **rSVD** (S&H 2008)of $K \Rightarrow U\,\Delta V'$
- $w_X = V[:, 1]$

This is the idea used by Lê Cao et al.

(authors also compute sparse w_Y, not detailed in this presentation)

Summary of the rSVD of Shen & Huang 2008

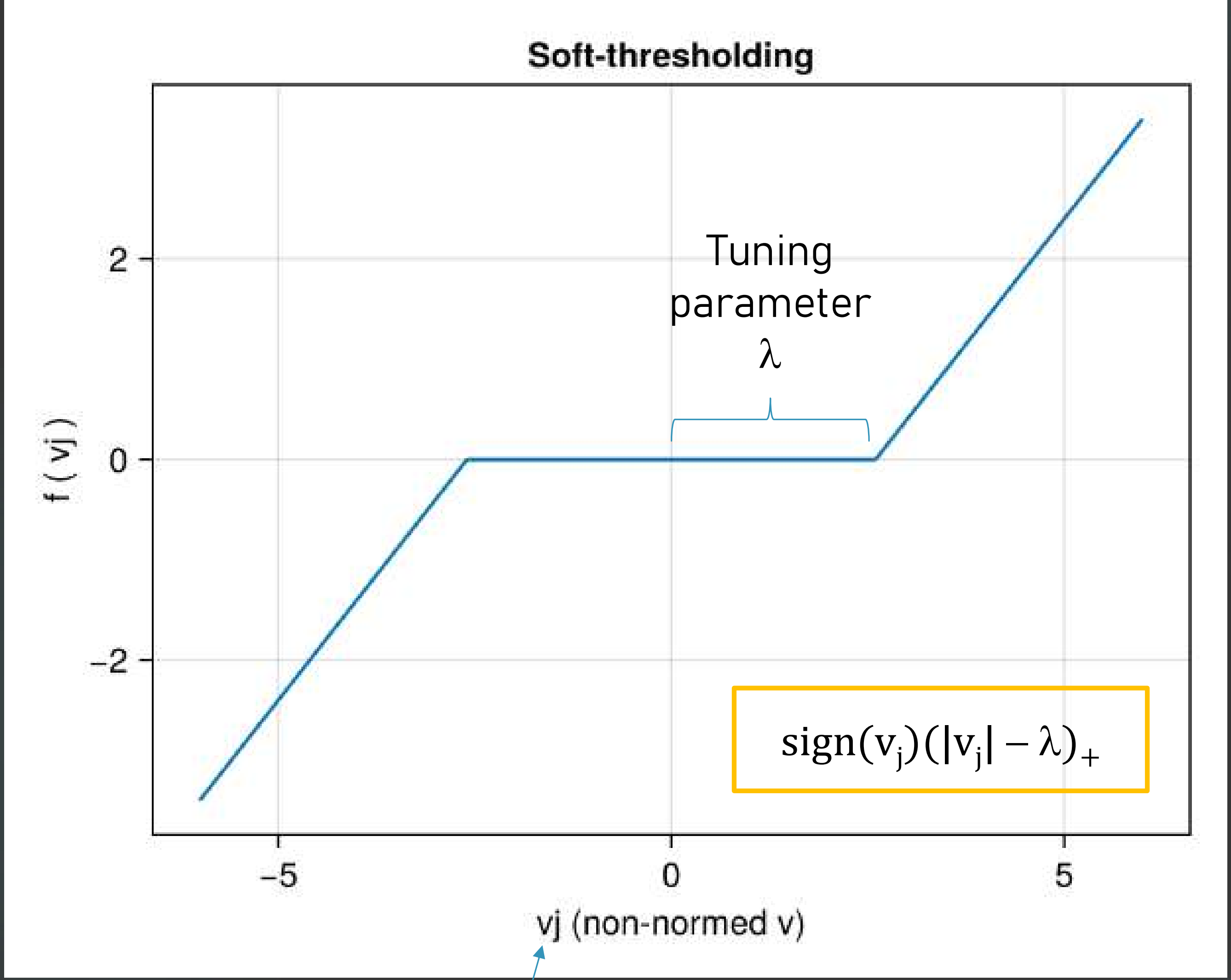
$\hat{X} = u\,\delta\,v'$ $\|u\| = \|v\| = 1$ 1-rank SVD of a given matrix X

A usual iterative NIPALS algorithm

```
1. Set $u$
2. Repeat until convergence
   a) $v = X'u$
   b) $u = X v / \mathrm{norm}(X v)$
3. $v = v / \mathrm{norm}(v)$
```

Regularization of Shen & Huang

```
1. Set $u$
2. Repeat until convergence
   a) $v = f_{\lambda}(X'u)$    soft thresholding
   b) $u = X v / \mathrm{norm}(X v)$
3. $v = v / \mathrm{norm}(v)$
```

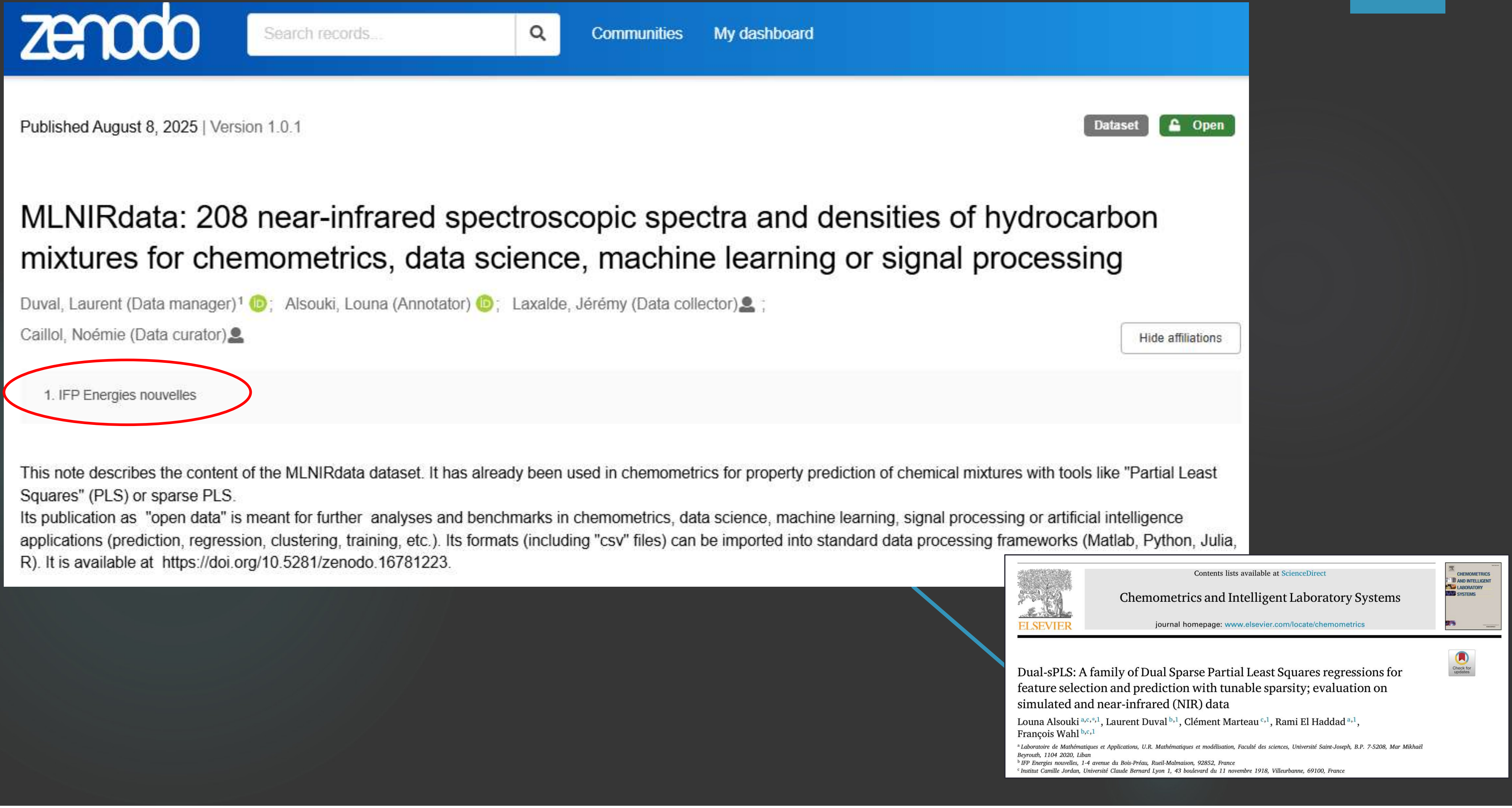


Tuning parameter λ

$$\text{sign}(v_j)(|v_j| - \lambda)_+$$

w_j for PLS

Illustration: MLNIR dataset



Published August 8, 2025 | Version 1.0.1

MLNIRdata: 208 near-infrared spectroscopic spectra and densities of hydrocarbon mixtures for chemometrics, data science, machine learning or signal processing

Duval, Laurent (Data manager)[1]; Alsouki, Louna (Annotator); Laxalde, Jérémy (Data collector); Caillol, Noémie (Data curator)

1. IFP Energies nouvelles

This note describes the content of the MLNIRdata dataset. It has already been used in chemometrics for property prediction of chemical mixtures with tools like "Partial Least Squares" (PLS) or sparse PLS.
Its publication as "open data" is meant for further analyses and benchmarks in chemometrics, data science, machine learning, signal processing or artificial intelligence applications (prediction, regression, clustering, training, etc.). Its formats (including "csv" files) can be imported into standard data processing frameworks (Matlab, Python, Julia, R). It is available at https://doi.org/10.5281/zenodo.16781223.

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Dual-sPLS: A family of Dual Sparse Partial Least Squares regressions for feature selection and prediction with tunable sparsity; evaluation on simulated and near-infrared (NIR) data

Louna Alsouki [a,c,*,1], Laurent Duval [b,1], Clément Marteau [c,1], Rami El Haddad [a,1], François Wahl [b,c,1]

[a] Laboratoire de Mathématiques et Applications, U.R. Mathématiques et modélisation, Faculté des sciences, Université Saint-Joseph, B.P. 7-5208, Mar Mikhaël Beyrouth, 1104 2020, Liban
[b] IFP Energies nouvelles, 1-4 avenue du Bois-Préau, Rueil-Malmaison, 92852, France
[c] Institut Camille Jordan, Université Claude Bernard Lyon 1, 43 boulevard du 11 novembre 1918, Villeurbanne, 69100, France

All the analyses in this presentation: done with **Jchemo.jl**
https://github.com/mlesnoff/Jchemo.jl

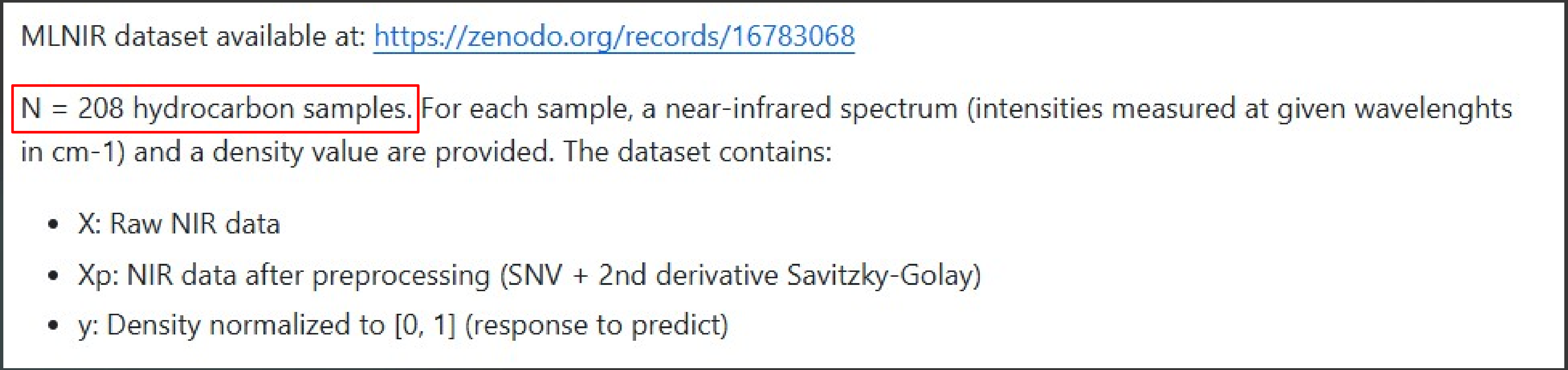
In particular,

function **`splsr`**

– Sames results as **`mixOmics::spls`** but

a) Sparsity only on w_x (predictive approach)

b) Nipals PLS algorithm replaced by the "***improved kernel algorithm #1***" of Dayal & McGegor 1997 ⇒ faster



MLNIR dataset available at: https://zenodo.org/records/16783068

N = 208 hydrocarbon samples. For each sample, a near-infrared spectrum (intensities measured at given wavelenghts in cm-1) and a density value are provided. The dataset contains:

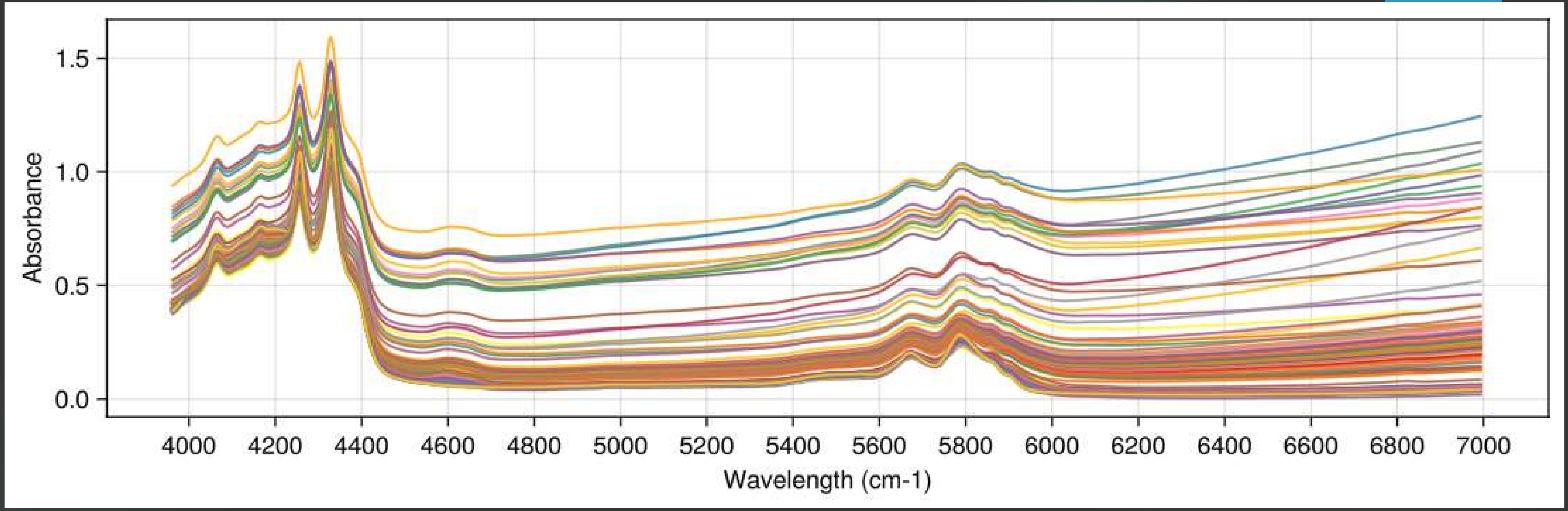
- X: Raw NIR data
- Xp: NIR data after preprocessing (SNV + 2nd derivative Savitzky-Golay)
- y: Density normalized to [0, 1] (response to predict)

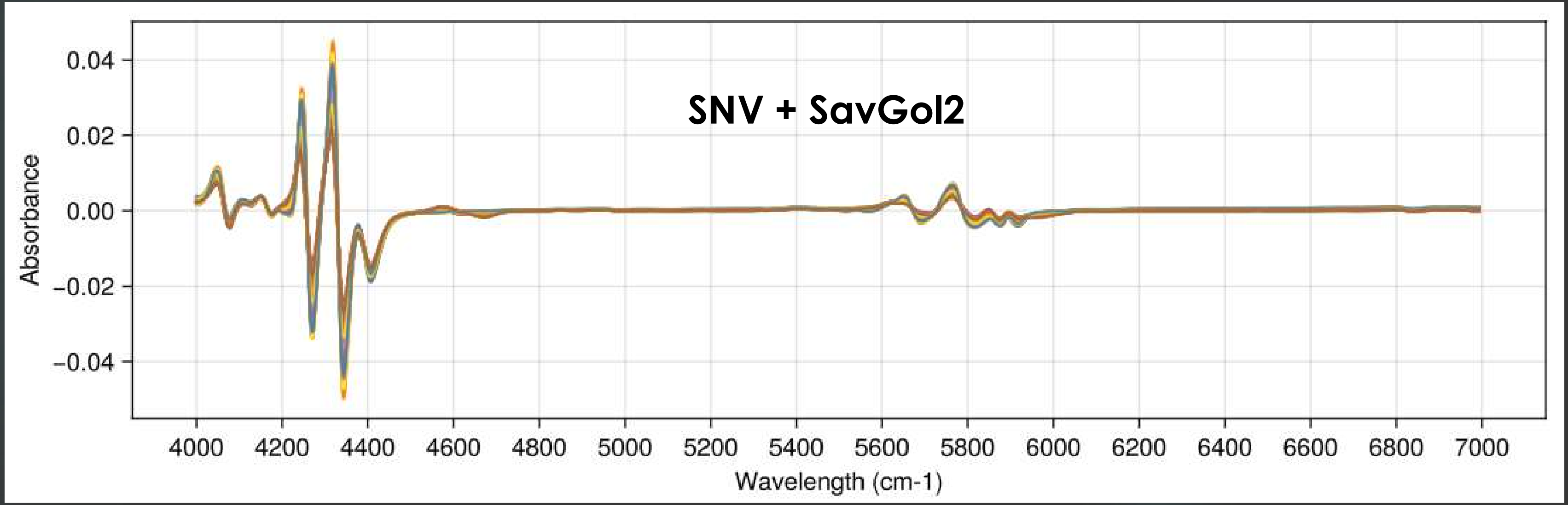
Available (JLD2) at: https://github.com/mlesnoff/**JchemoData.jl**?tab=readme-ov-file#mnist20pcts

MLNIR data

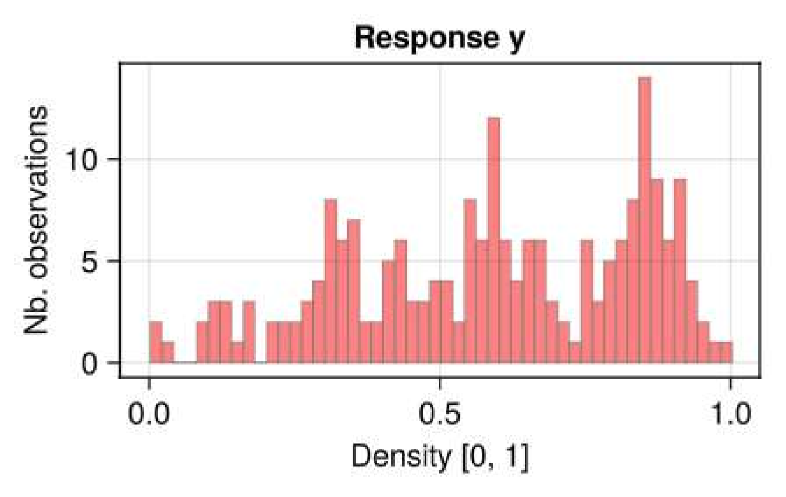
with pre-selection of range 4000-7000 cm-1

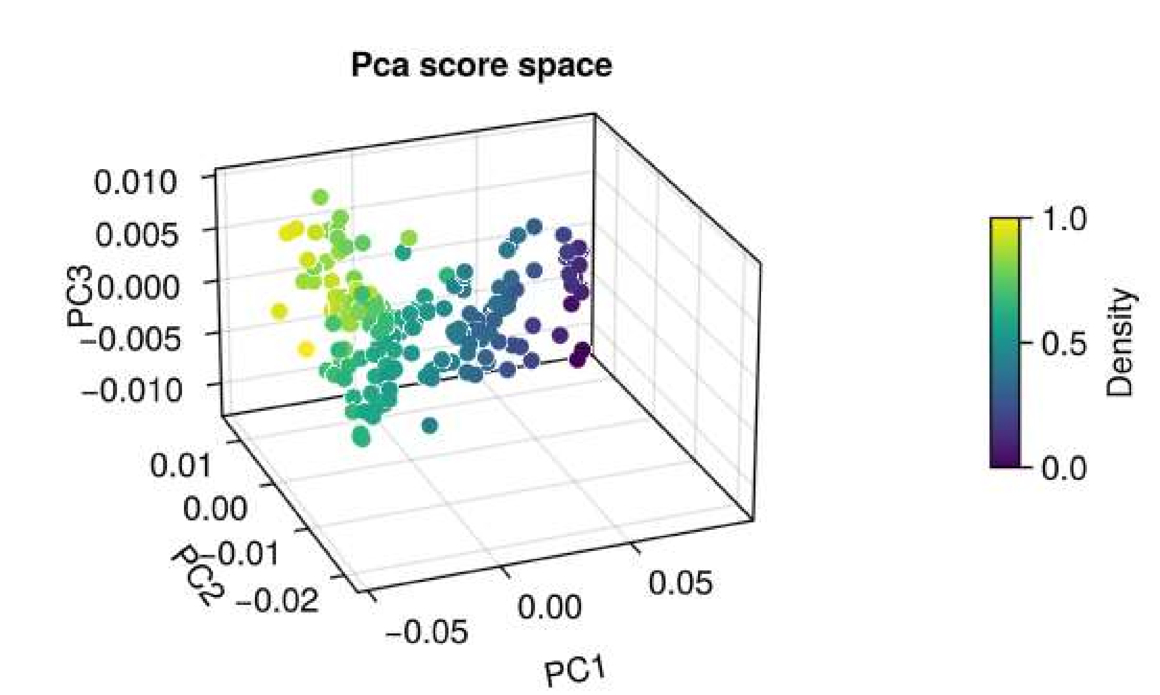
1556 variables (wavelengths)




SNV + SavGol2

Response y
Hydrocarbon density





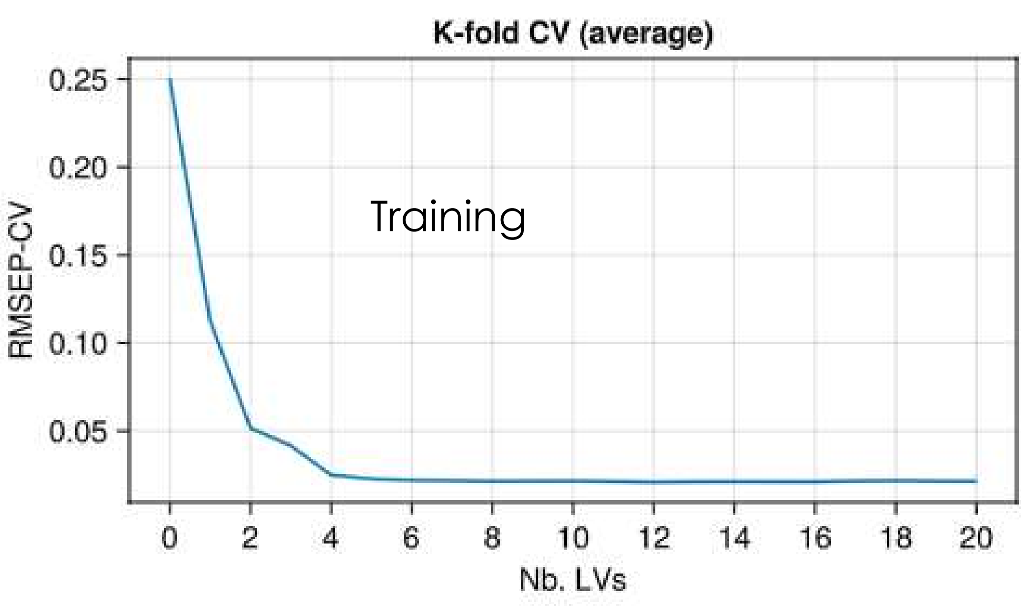
Model calibration and validation

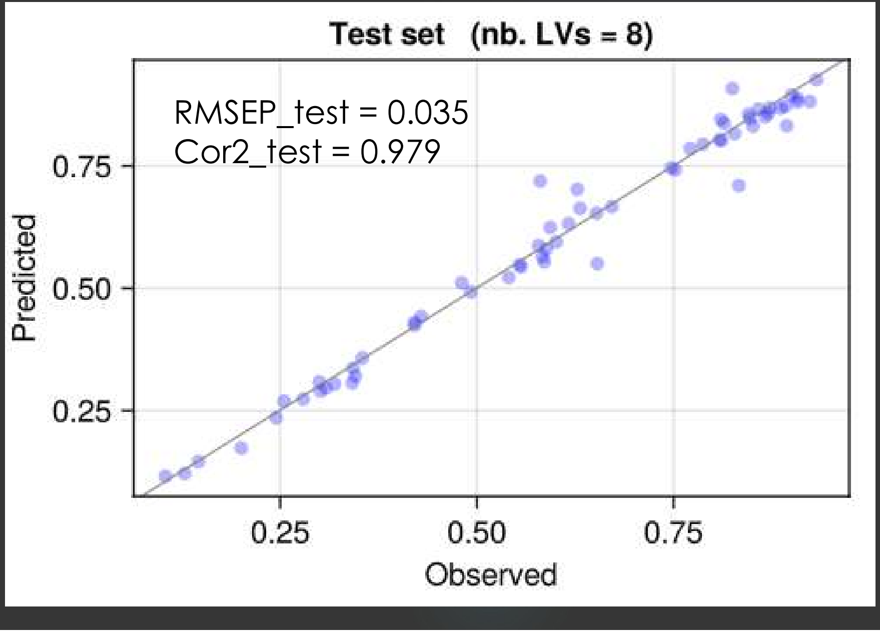
Total n = 208 obs.

Train n = 146 ⇒ Replicated K-fold CV K = 3, nrep = 30

test n = 62 30% by random sampling

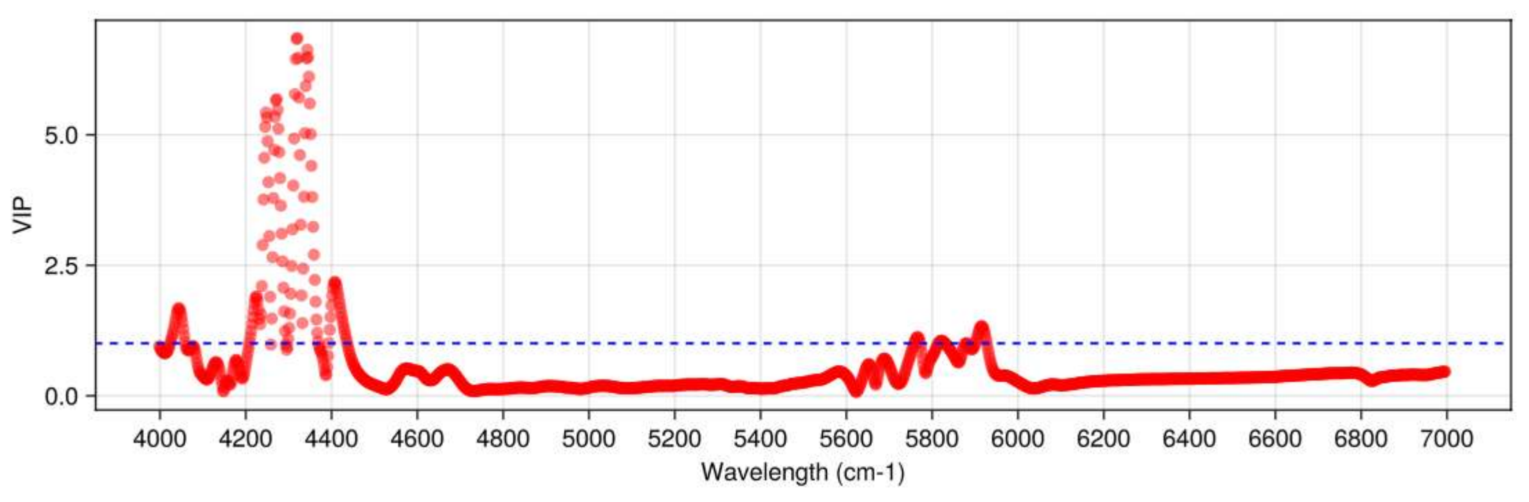
PLSR

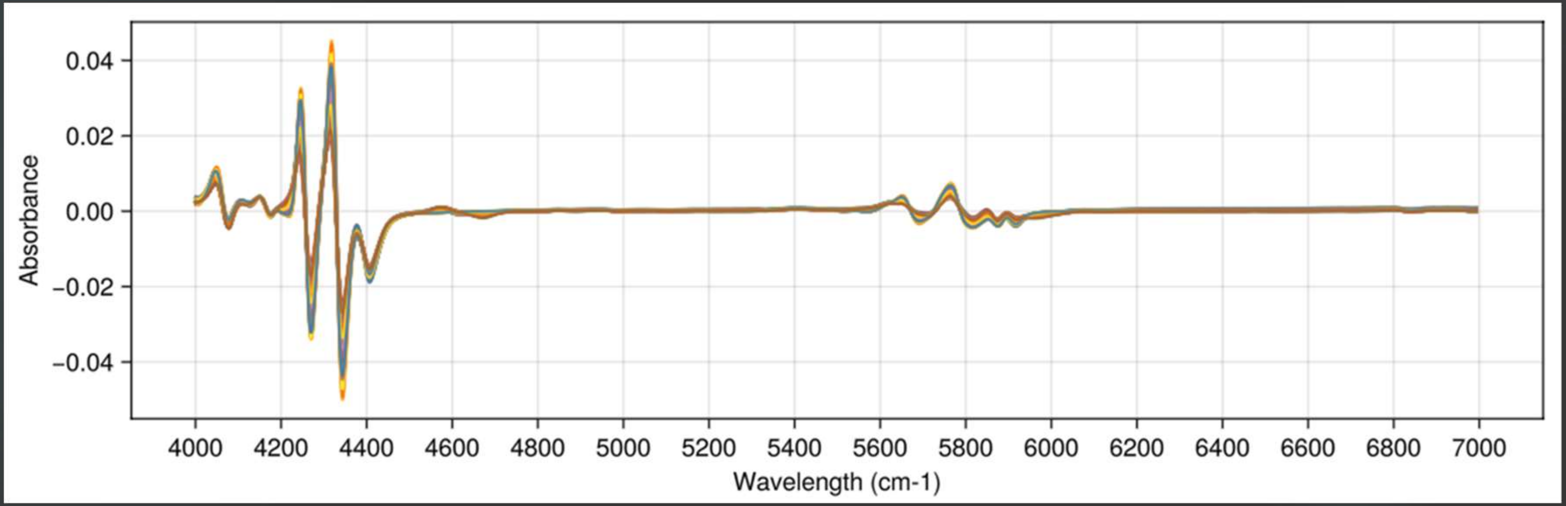

K-fold CV (average)
Training
RMSEP-CV
0.25
0.20
0.15
0.10
0.05
0
2
4
6
8
10
12
14
16
18
20
Nb. LVs


Test set (nb. LVs = 8)
RMSEP_test = 0.035
Cor2_test = 0.979
Predicted
0.75
0.50
0.25
0.25
0.50
0.75
Observed

VIPs

PLS 8 LVs





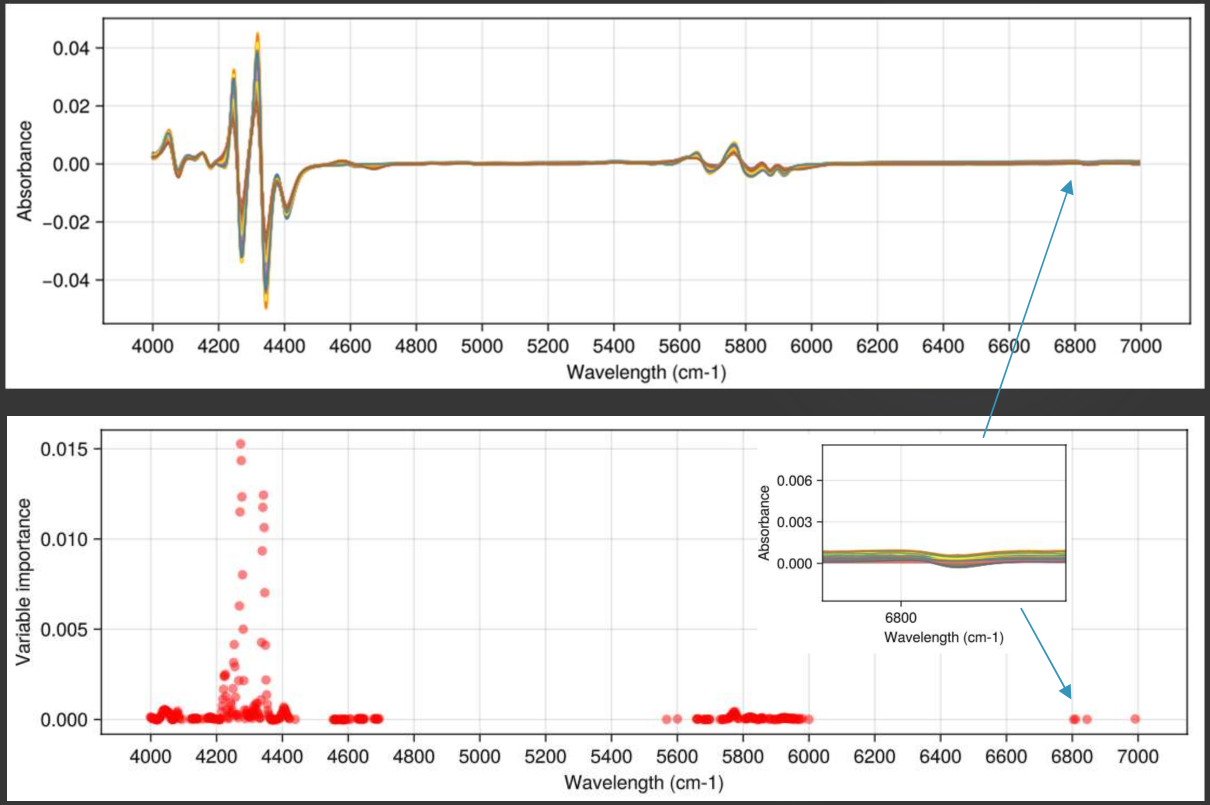
Alternative to VIPs (very simple and efficient): Variable importance by permutation

- Successively for each variable:

  a) Obs. are **randomly permuted**

  b) The model (here PLSR 8 LVs) is fitted on Xtrain and used to predict Xtest

  c) Rmsep_test (or other indicator) is computed and compared to the original without permutation

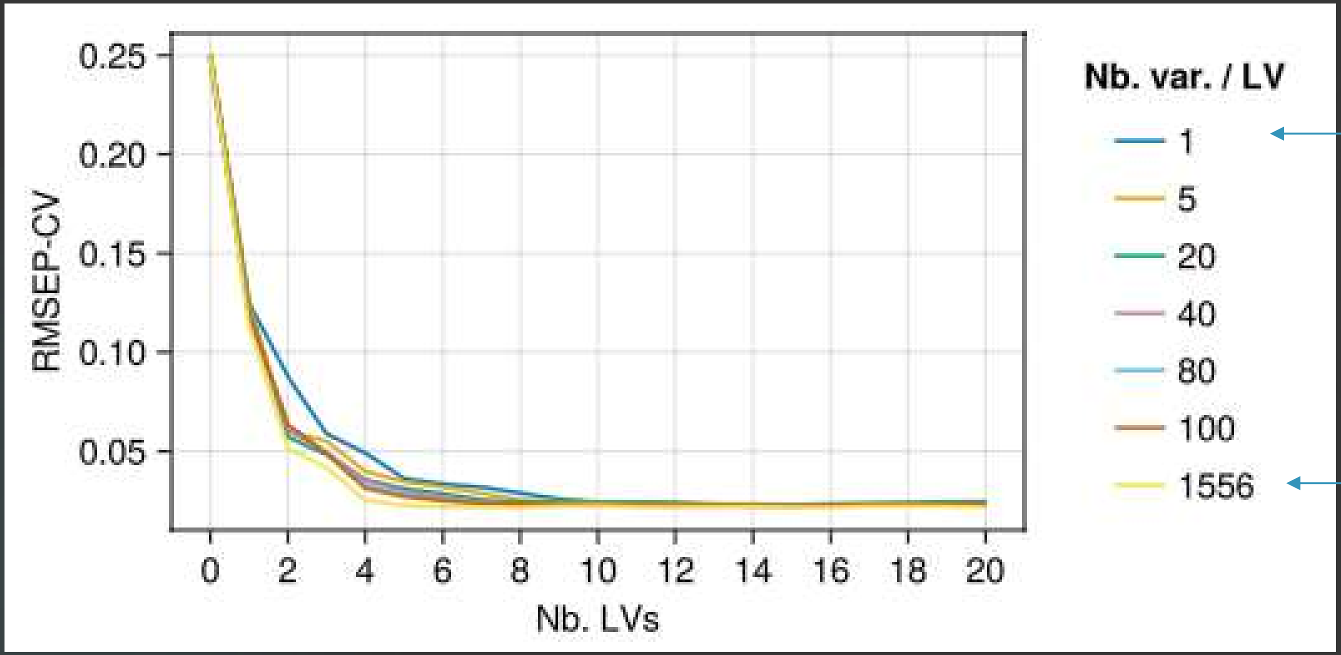


- Grouped-approaches are very easy to implement (simultaneous permutation of sets of variables)

- Better than Interval-PLS

Sparse PLSR

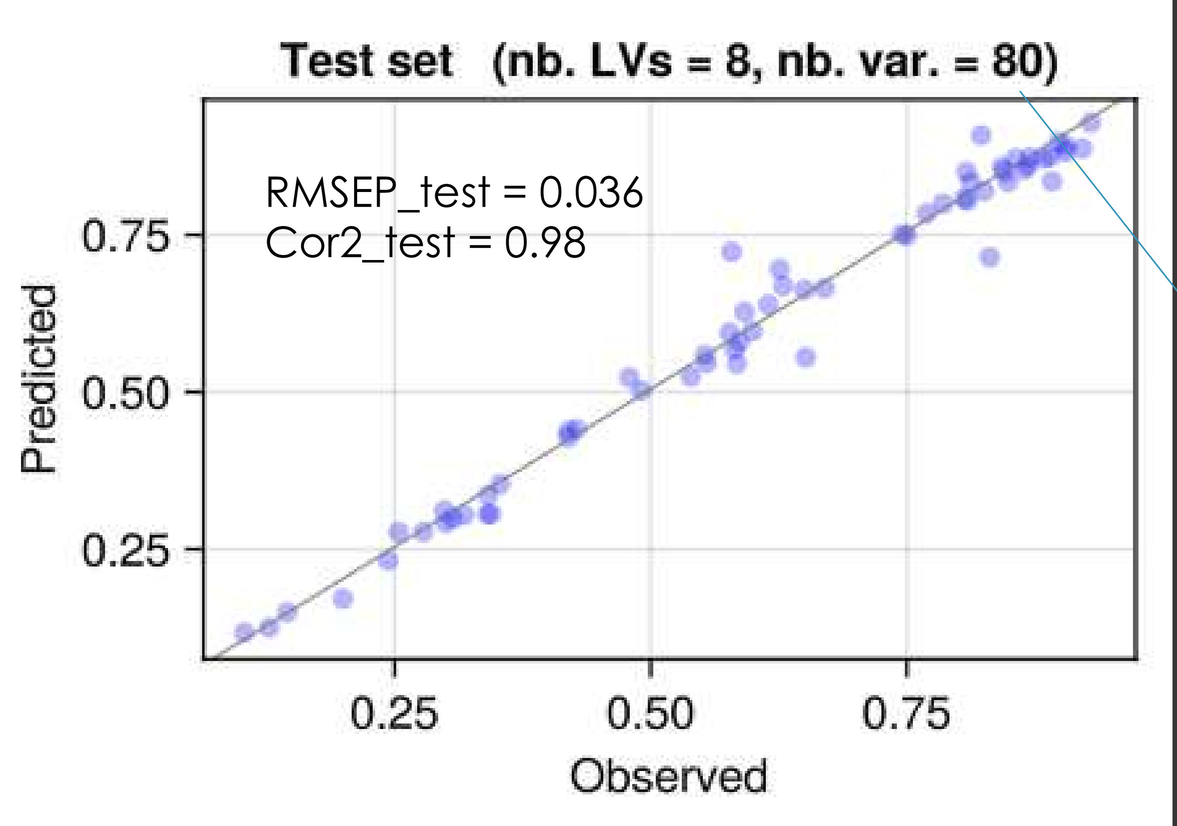
⇒ CV to tune the nb. LVs and the nb. variables that are selected for each LV



CovselR

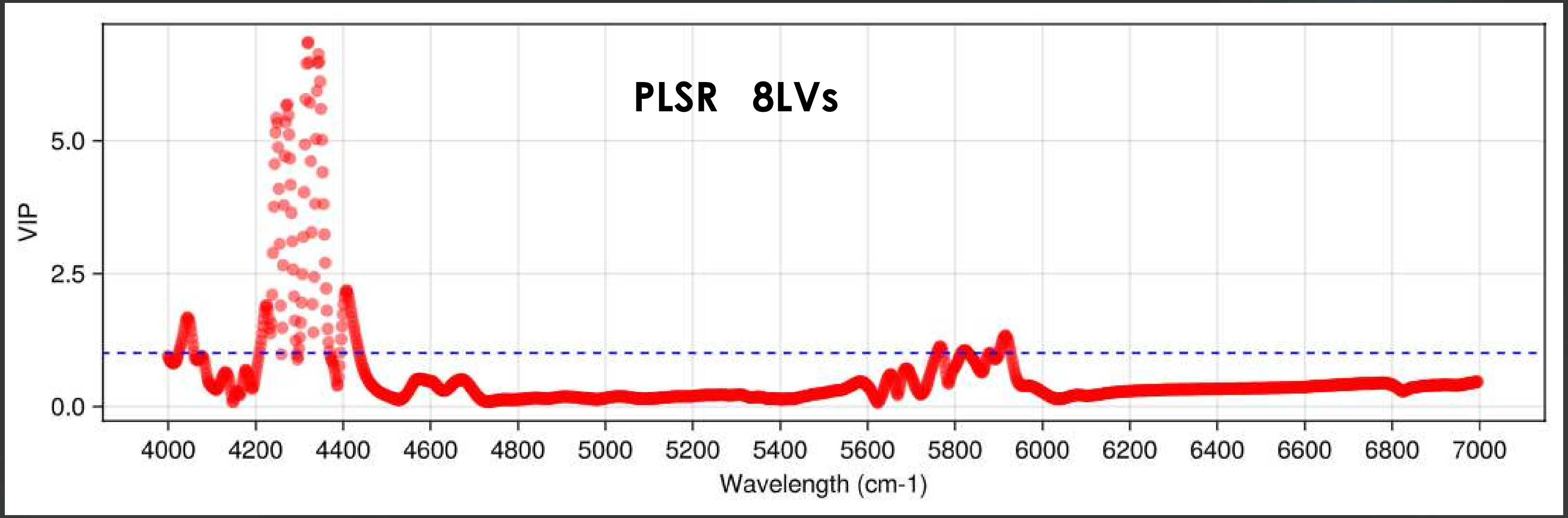
PLSR

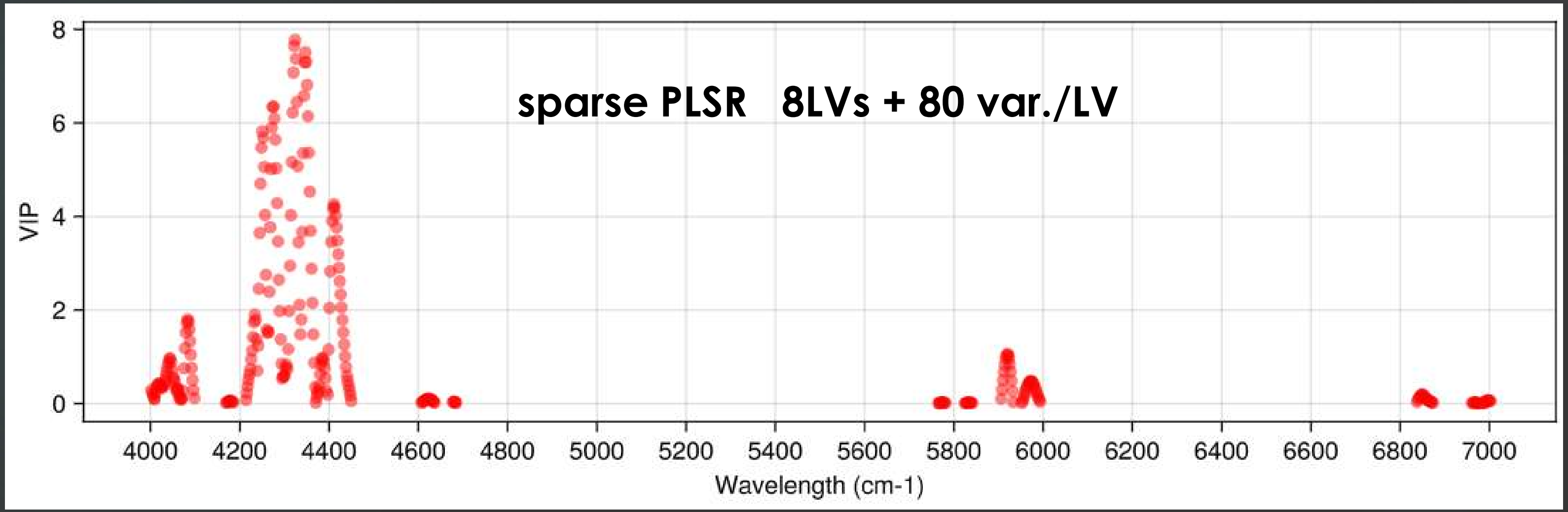
In this example, the same nb. variables are selected for all the LVs, but this can be relaxed



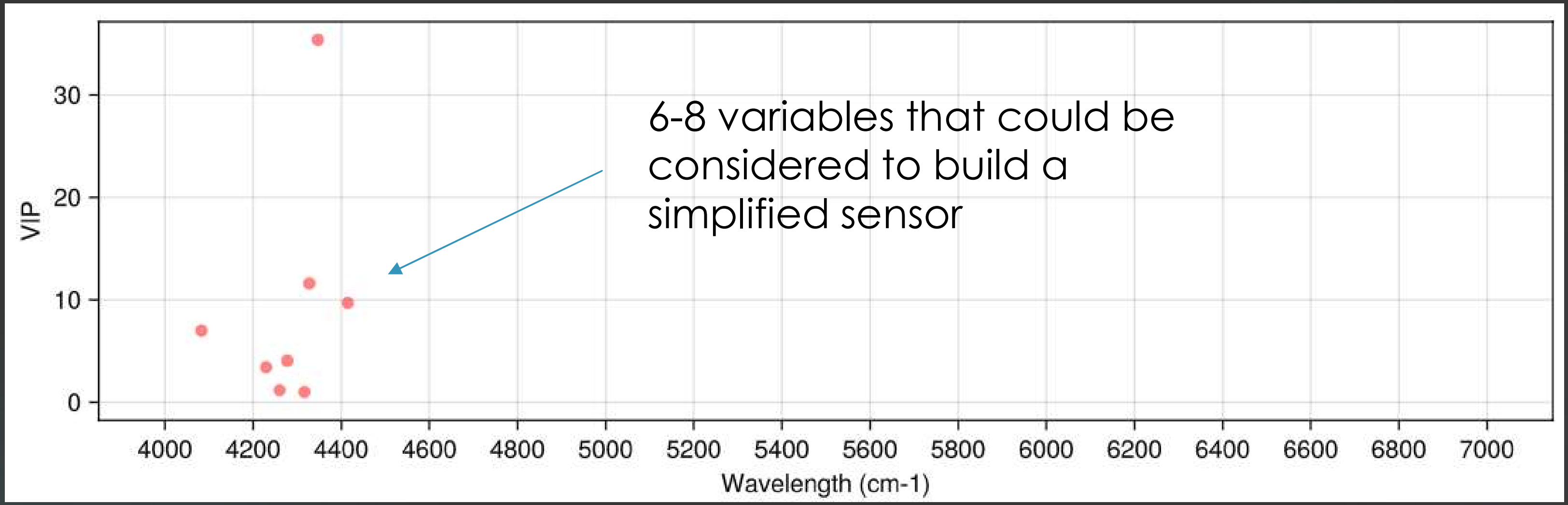
RMSEP_test = 0.036
Cor2_test = 0.98

⇒ Total nb. selected variables = 256 (over 1556)

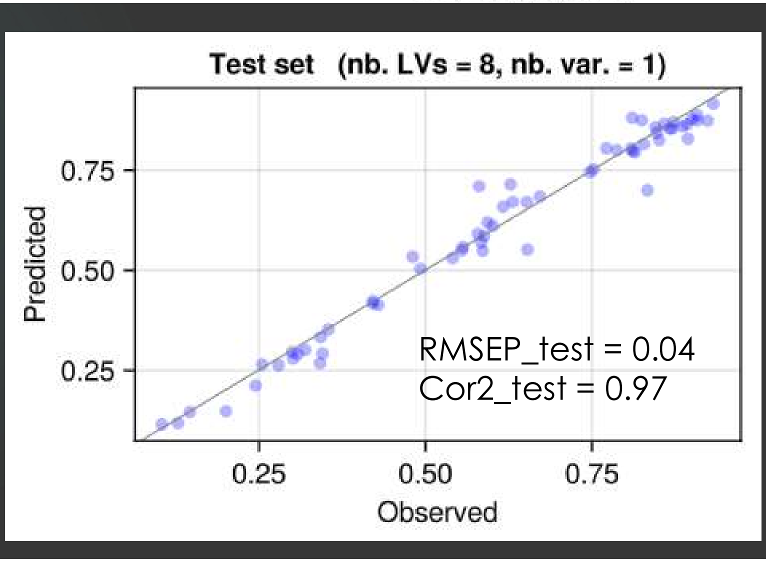

PLSR 8LVs
VIP
0.0
2.5
5.0
4000
4200
4400
4600
4800
5000
5200
5400
5600
5800
6000
6200
6400
6600
6800
7000
Wavelength (cm-1)


sparse PLSR 8LVs + 80 var./LV
VIP
0
2
4
6
8
4000
4200
4400
4600
4800
5000
5200
5400
5600
5800
6000
6200
6400
6600
6800
7000
Wavelength (cm-1)

With only 1 variable selected / LV (CovSelR)



6-8 variables that could be considered to build a simplified sensor

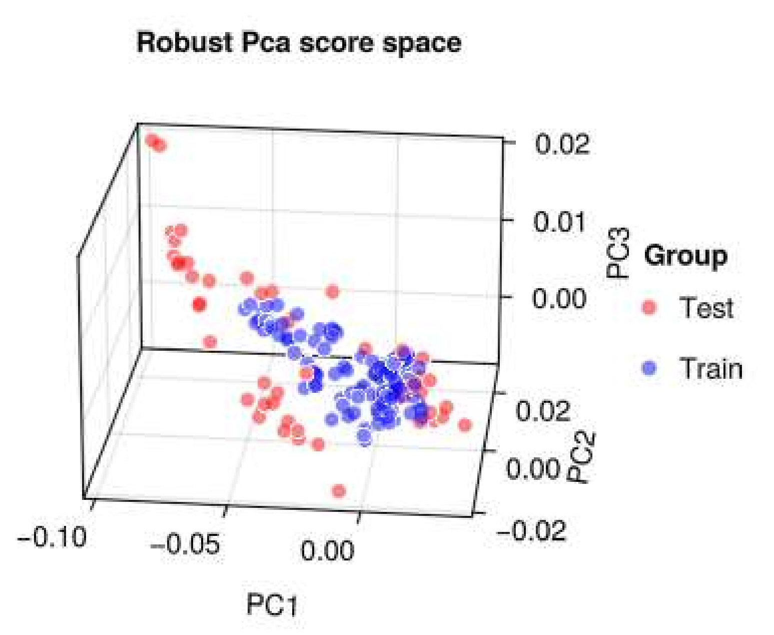


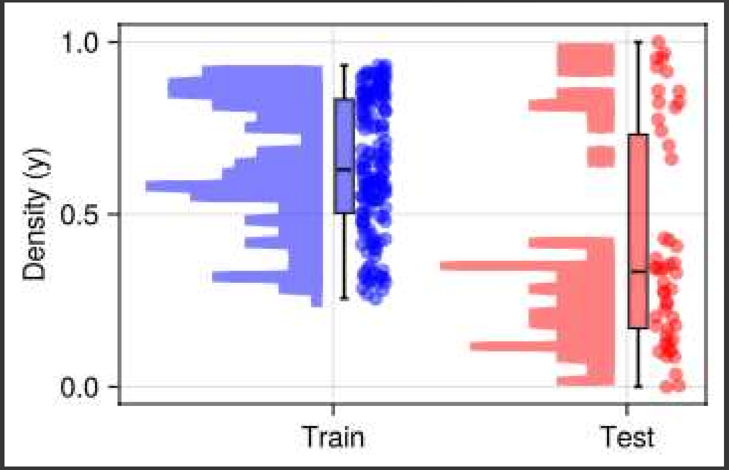
RMSEP_test = 0.04
Cor2_test = 0.97

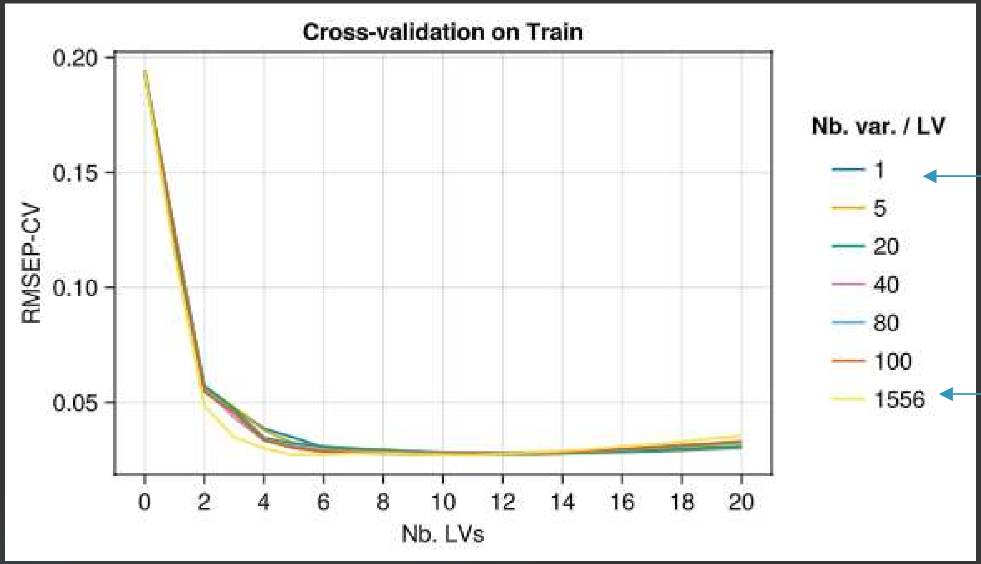
And what about predictive robustness?

New split Train/Test

- Test (n = 50): Mainly in extrapolation

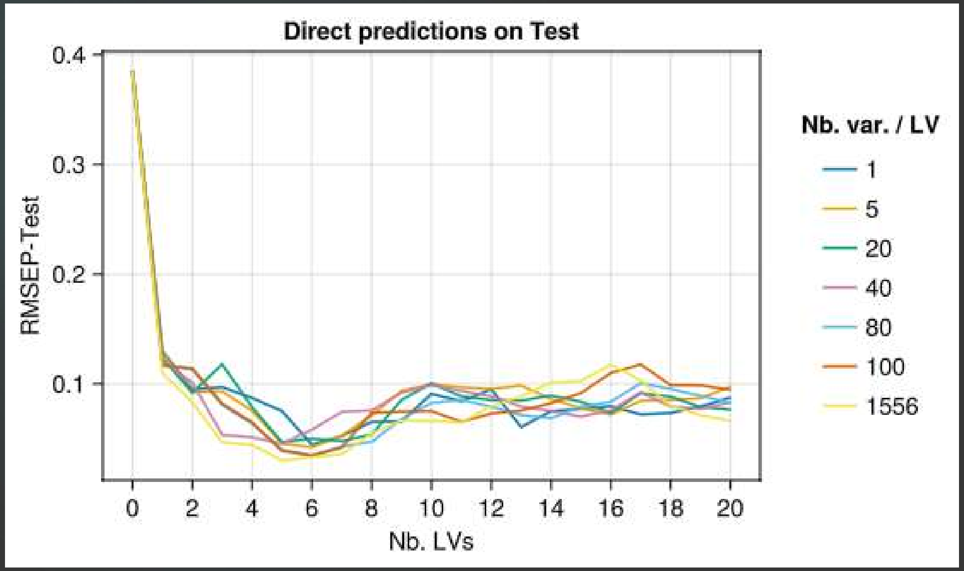




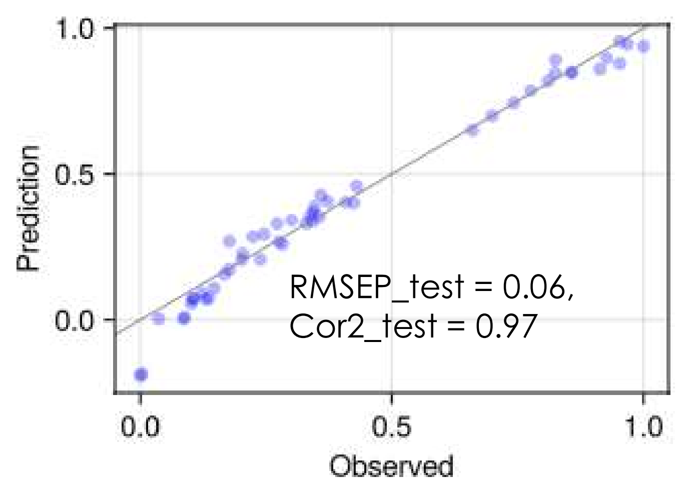


ComSelR

PLSR



sparse PLSR 8LVs 80 var. / LV



RMSEP_test = 0.06,
Cor2_test = 0.97