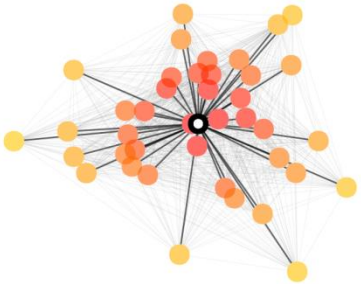
Using R for infrared spectroscopy research

Leo **Ramirez-Lopez** [1]
Alexandre **Wadoux** [2]

[1] BUCHI Labortechnik AG, Switzerland
[2] Wageningen University, Netherlands



Agenda

8:30 – 9:00	(Optional) Checking the requirements (R, Rstudio, R packages, dataset, etc)
9:00 -9:15	Introduction by Titia Mulder, Wageningen University
9:15 – 9:45	Signal processing
9:45 – 10:15	Measuring the similarity/dissimilarity between spectra and outlier detection
10:15 – 10:45	Calibration sampling (what and how many samples to choose to build accurate multivariate models)
10:45 – 11:00	Coffee break
11:00 – 11:45	Multivariate calibration (global models, local models for large and complex datasets)
11:45 – 12:30	(Optional) Transferring NIR data across different instruments

Data and codes

Available at http://wadoux.org/codes.html

Lecture

Available at http://wadoux.org/codes.html (online 27.06.17)

Software requirements:

- R 3.3.1
- R Studio 0.99.903
- Packages:
  - prospectr
  - resemble
  - clhs
  - pls
  - caret
- Data:
  - br_spectra_1.txt
  - br_spectra_2.txt
  - br_spectra_3.txt
  - br_spectra_4.txt
  - noisy_spectra.txt
  - sensorRepetitions.txt
  - spectra_soil_profile_Colombia.txt
  - syntheticCalSampling.txt
  - world_data_3644_samples.txt

C:/Users/stevens/Dropbox2/Dropbox/SPECcourse - RStudio

File Edit Code View Project Workspace Plots Tools Help

pres.Rmd | Day1.Rmd

Knit HTML | Run | Chunks

```
* The expension of __R__ is driven by the increasing number of available functions/packages.

![](http://r4stats.files.wordpress.com/2012/04/fig_10_cran.png)

source: [http://r4stats.com/articles/popularity/](http://r4stats.com/articles/popularity/)
----------------------------------------

* But, of course, __R__ has also drawbacks:
 * No GUI (altough solutions exist, such as [Rcmdr](http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/) for data import/export
and analysis or [DeduceR](http://www.deducer.org/pmwiki/index.php?n=Main.ScreenShots) for graphs)
 * Steeper learning curve than Excel, SAS or JMP (No GUI...)
 * Cryptic documentation and error messages
 * Often, __R__ makes us believe it worked as intended (see Chapter 8 of the [R Inferno](http://www.burns-stat.com/pages
/Tutor/R_inferno.pdf))

```{r}
# Exemple:
head(iris) # the iris dataset is a data.frame
head(iris[,1:2]) # the two first column of the data.frame
head(iris[,1]) # subsetting the 1st column returns a vector, not a data.frame!
head(iris[,1,drop=F]) # use [,,drop=F] to override the default behavior
```
```
111:1 (Top Level) R Markdown
```

Console C:/Users/stevens/Dropbox2/Dropbox/SPECcourse/

```
11 2005  31 263  32    799          342
12 2006  42 281  47    967          397
13 2007  49 321  60   1220          415
14 2008  45 368 106   1300          431
15 2009  54 371 153   1330          447
16 2010  75 375 187   1620          539
17 2011  74 345 199   1710          556
> save(GS_data,file="GS_hits.Rdata")
>
> ggplot(data=melt(data=GS_data,id="year"),aes(x=year,y=value,colour=variable)) + geom_point() + geom_line() + labs(x="Year",y="Num
Google Scholar") + theme_bw() + scale_color_discrete("Software")
Warning messages:
1: Removed 16 rows containing missing values (geom_point).
2: Removed 16 rows containing missing values (geom_path).
> ggsave("figures/GScholar_spectroscopy.png")
Saving 6.61 x 6.17 in image
Warning messages:
1: Removed 16 rows containing missing values (geom_point).
2: Removed 16 rows containing missing values (geom_path).
> rm() # remove objects
> save() # save in Rdata format
Error in save() : 'file' must be specified
>
```

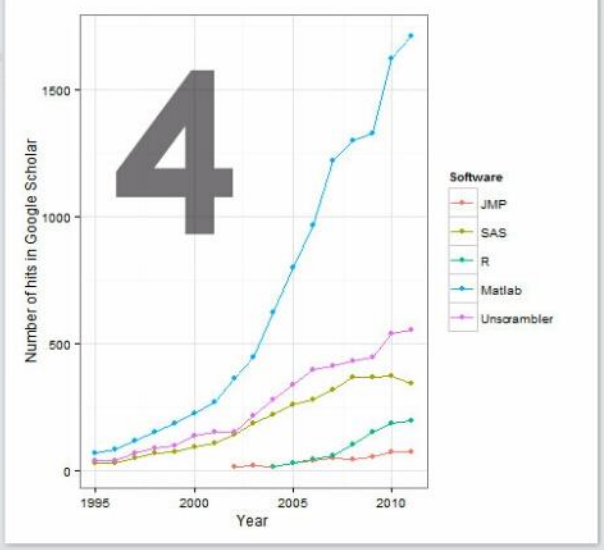
Workspace | History

To Console | To Console | To Source

```
h1 <- xpathSApply(doc, "//div[@id='gs_ab_md']", xmlValue)
h2 <- as.numeric(sub(",","",strsplit(h1, " ")[[1]][2]))
h2
}
}
colnames(hits) <- labels
data.frame(year=from:(to-1),hits)
}
GS_data <- GScholar_Scraper_hits(inputs,labels)
GS_data
save(GS_data,file="GS_hits.Rdata")
ggplot(data=melt(data=GS_data,id="year"),aes(x=year,y=value,colou
ggsave("figures/GScholar_spectroscopy.png")
rm() # remove objects
save() # save in Rdata format
```

Files | Plots | Packages | Help

Zoom | Export | Clear All


```

Introduction

In this workshop we present a tutorial on R to address the tasks that are routinely encountered when we carried out research that uses infrared spectroscopy data.

We also show some basic methods (implemented in R) that can be used to boost the efficiency of the use of infrared spectroscopy tools.

We will show how to integrate different R tools to perform infrared data analysis efficiently

This course is not:

... about multivariate statistics
... lectured by statisticians (we are not and we guess you neither)
... for R developers

Our R expererience:

We use it on a daily basis for our own scientific research purposes
We are R developers with experience in:

- Soil spectroscopy
- Spatial analysis/geostatistics
- Hyperspectral remote sensing
- Pedometrics

Our approach to learn R was based on trial and error

Step 1
- Collect your samples
- Make the scans (raw spectra)

Step 2
- Pre-processing
- Outlier detection

Step 3
- Select your calibration samples
- Multivariate calibration

Step 4
- (Optional) transfer data across different instruments

Step 1

- Collect your samples
- Make the scans (raw spectra)

Step 2

- Pre-processing
- Outlier detection

Step 3

- Select your calibration samples
- Multivariate calibration

Step 4

- (Optional) transfer data across different instruments

Signal processing

Signal processing aims to improve quality of the spectra before modeling (e.g. by removing undesired spectral variation). In this respect, it improves also the results of quantitative and qualitative spectral models.

The methods must be carefully chosen in order to avoid removing important physical information.

The most common signal processing methods are:

1. **Noise removal**
    1. Splice correction
    2. Moving average
    3. Savitzky-Golay filtering
2. **Derivative**
    1. 1st derivative/2nd derivative
3. **Scatter correction**
    1. SNV
    2. MSC
4. **Centering and scaling**
5. **Other specific transformations**
    1. Resampling
    2. Continuum removal

So what is the BEST methods ?
Which preprocessing technique should I use ?

There is no universal best method:

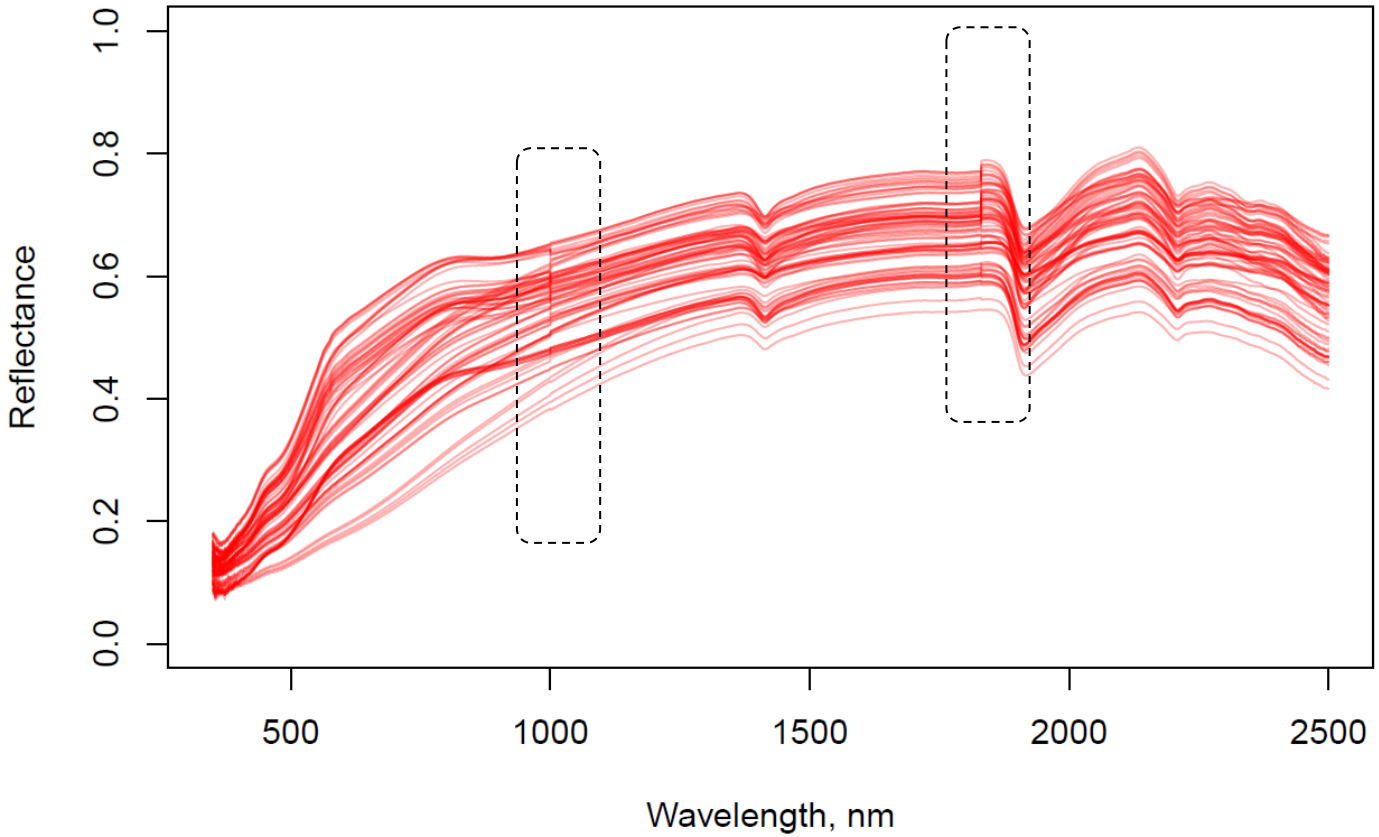
- Depends on your objective
- Depends on your NIR data

The methods must be carefully chosen in order to avoid removing important physical information and highlight the useful one.

Trial and error often performs best!

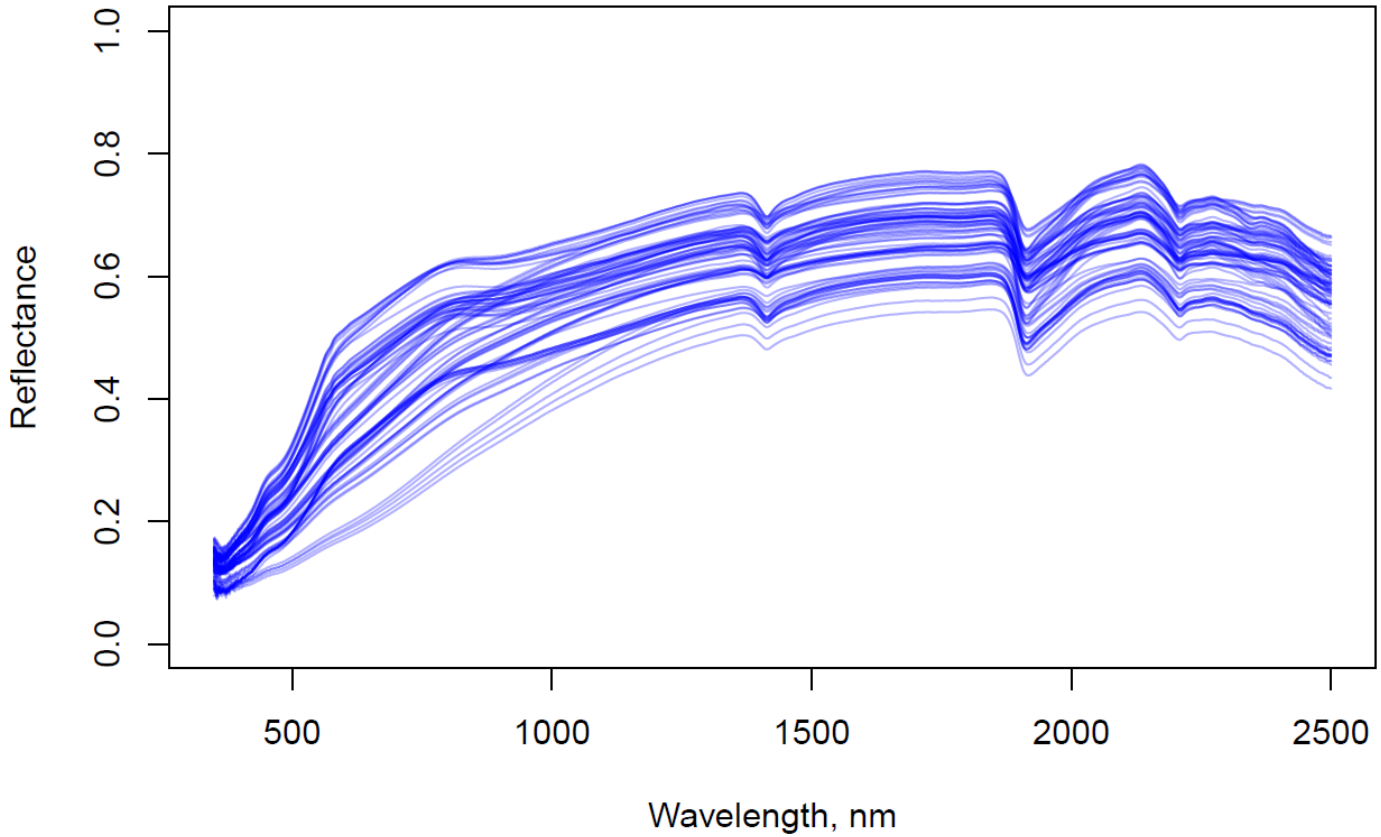
Example...

Note the "*shifts*" of these 85 soil spectra at 1000 and 1830 nm. This is due to measurement errors of the spectrometer. This can be corrected by applying the the **`spliceCorrection`** function...

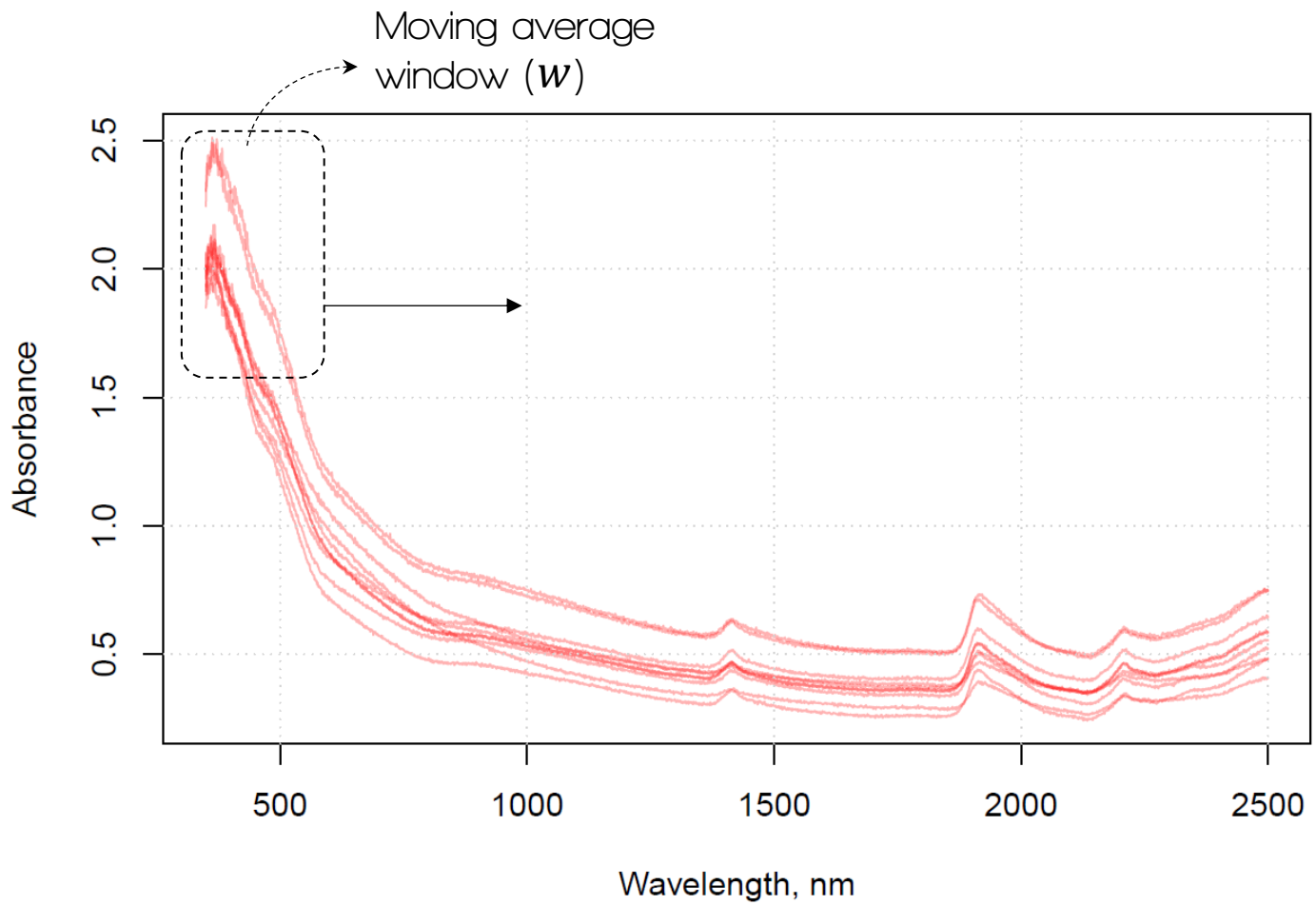


Example...

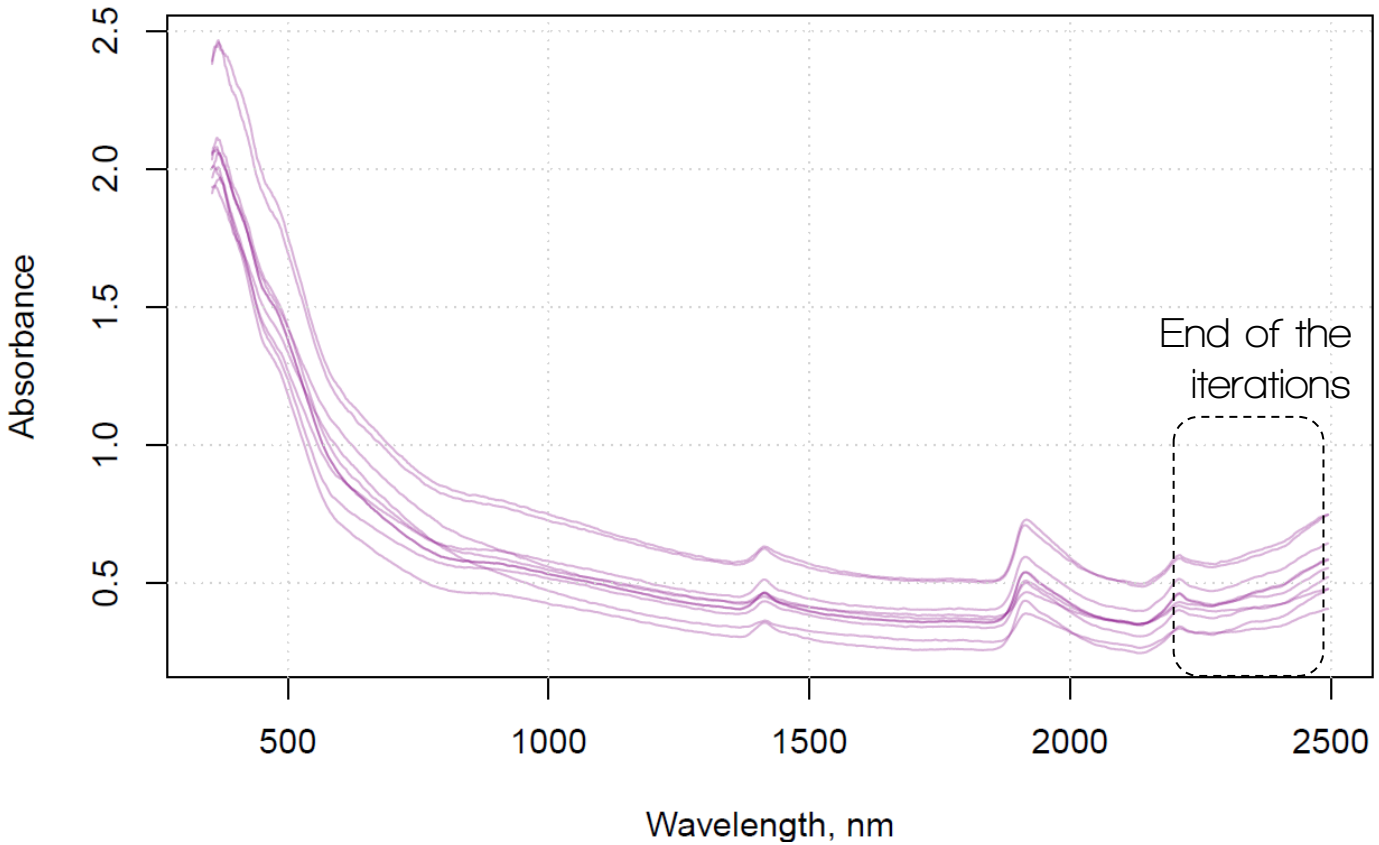
... and after applying the the **`spliceCorrection`** function...



Moving average



Moving average

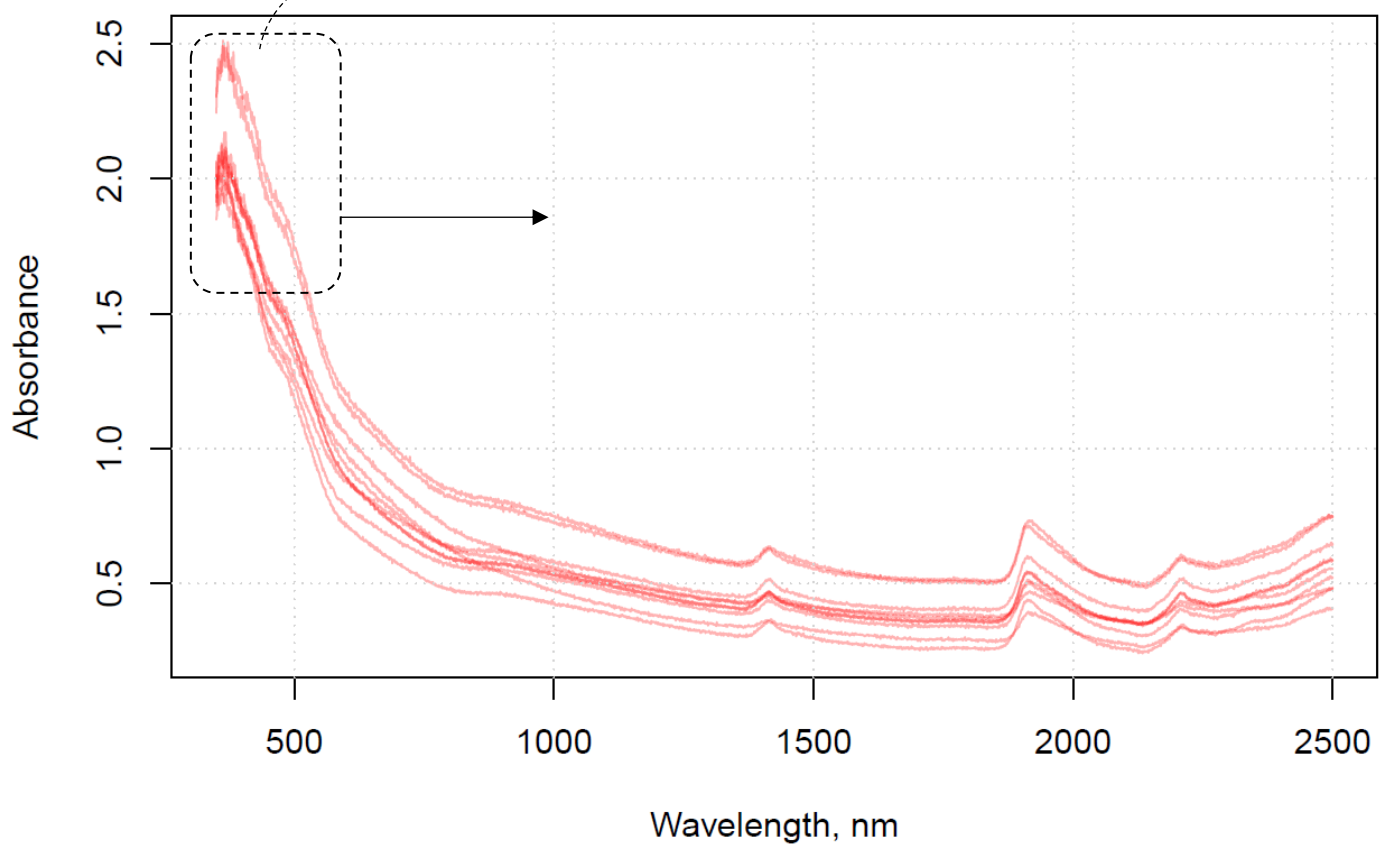


Savitzky–Golay filter

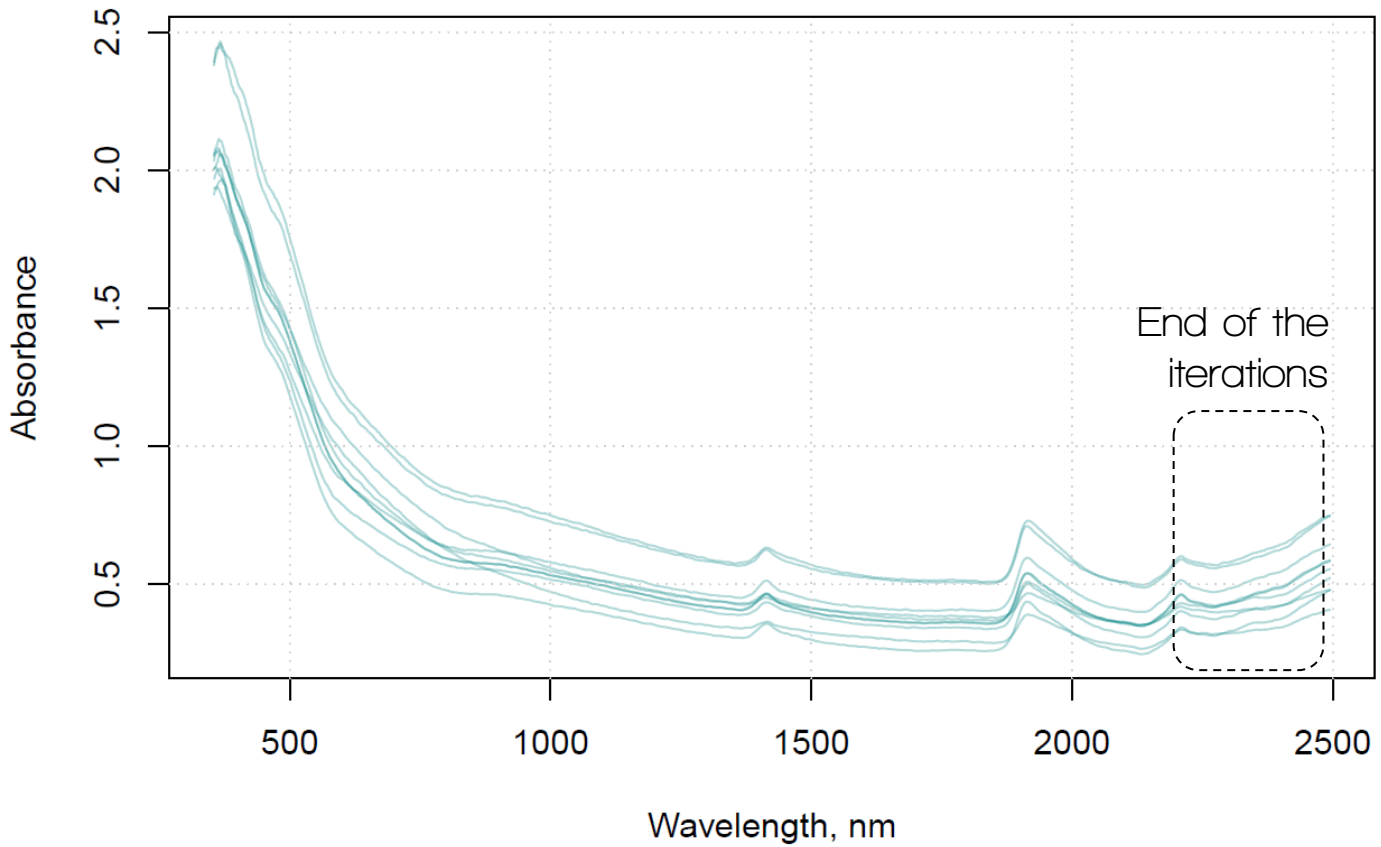
Polynomial order

Moving Polynomial fit window (w)

$$x_{sg} = b_p\, x^{p} + \cdots + b_1\, x + b_0$$

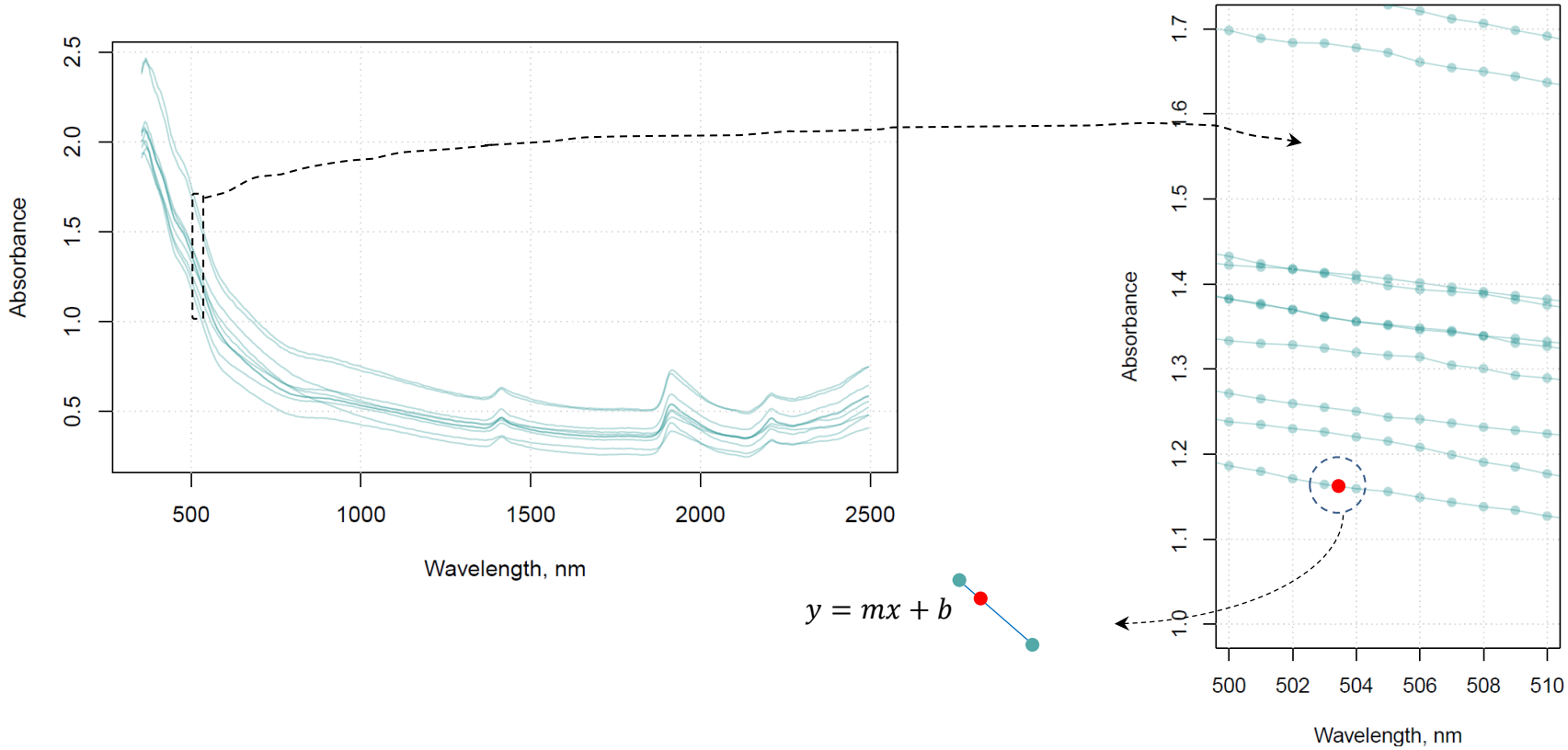


Savitzky–Golay filter



Resampling (also known as interpolation)

Example: if the first wavelength of the new set of wavelengths (at wich the spectra needs to be resampled) is 504.3 nm, the two wavelengths at the right and left sides of the new point are taken. Then a linear model is fitted between these two points and this models is then used to estimate the intensity at the new wavelength



Differentiation

Advantages

- Reduce of baseline offset
- Helps to resolve absorption overlapping
- Compensates for instrumental drift
- Enhances small spectral absorption features
- Often increases the predictive accuracy for complex datasets

Drawbacks

- Risk of overfitting
- Amplifies noise, smoothing required
- Less robust model and increase uncertainty in model coefficients
- Complicate spectral interpretation
- Remove the baseline! (albedo variation)

First and second derivatives of a spectrum can be computed with the finite difference method (difference between to subsequent data points), assuming that the band width is constant:

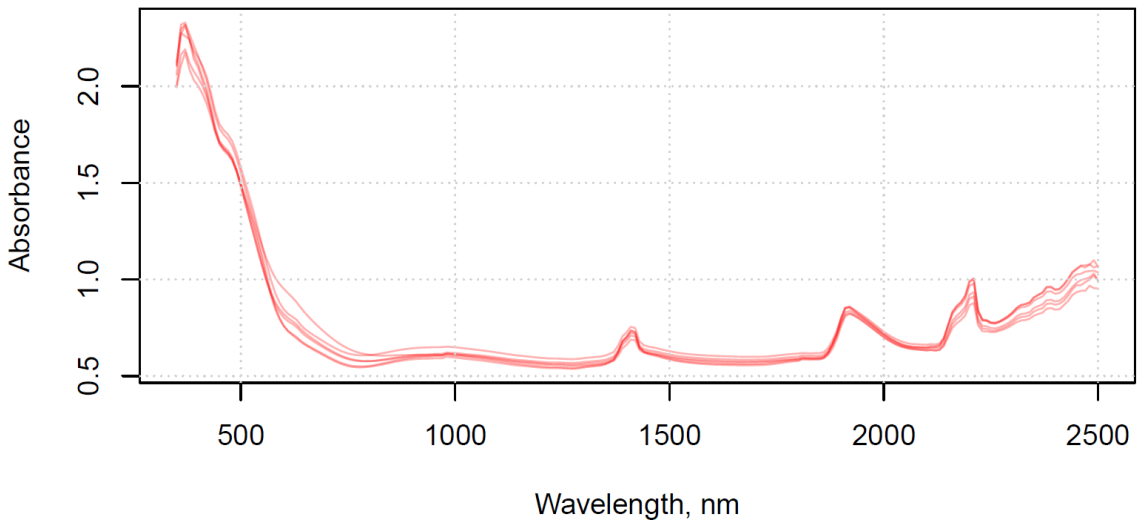
First derivative $x_i' = x_i - x_{i-1}$

Second derivative $x_i'' = x_{i-1} - 2\,x_i + x_{i+1}$

See file: 3.Preprocessing_II.R

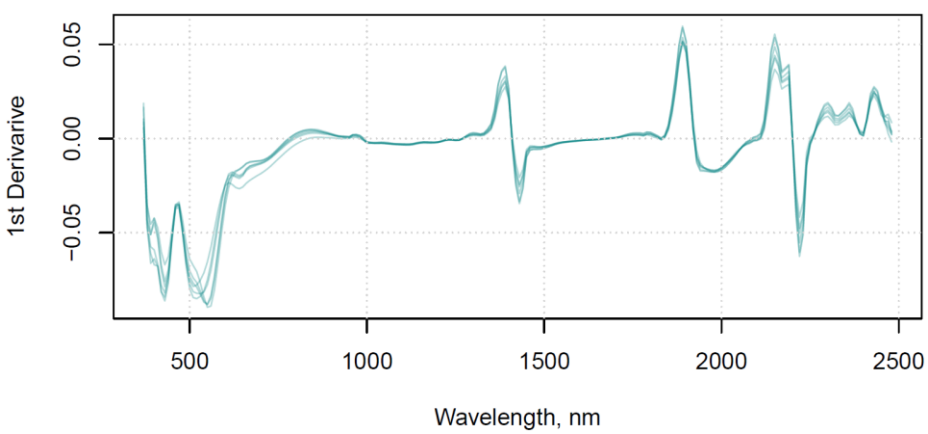
Differentiation

Original absorbance spectra of a soil profile

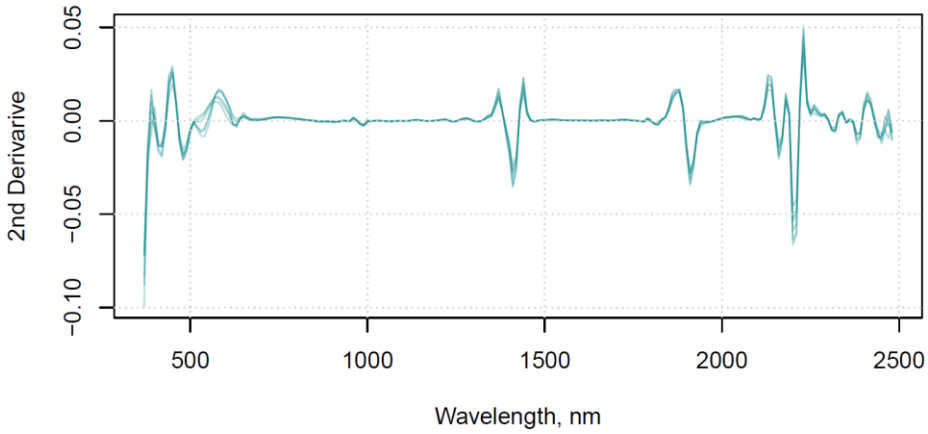


Using the **`savitzkyGolay`** function both the first and second derivative spectra, can be computed

First derivative spectra



Second derivative spectra



See file: 3.Preprocessing_II.R

Multiplicative Scatter Correction (msc)

- It mitigates the effect of scattered light (which is multiplicative) on the NIR spectra
- msc is a method that assumes for a given set of spectra, the scattering and offset effects can be removed from each spectrum.
- The msc model is determined by regressing each spectrum onto the mean spectrum. The model can be formally written as:

$$x_i = a_i + b_i\bar{x} + \varepsilon_i$$

where $\varepsilon_i$represents the chemical information of the spectrum while a_i *and* b_i represent the offset and the scatter respectively. Both a_i *and* b_i are found by using the Least Squares method. The msc corrected spectrum can be then computed as:

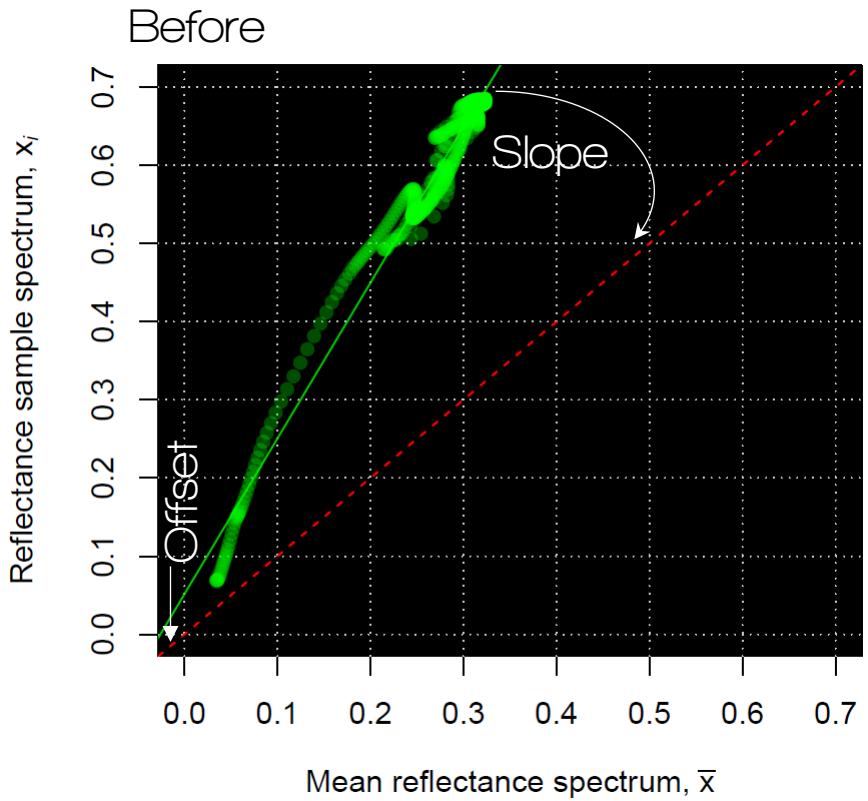
$$\mathrm{msc}(x_i) = \frac{x_i - a_i}{b_i}$$

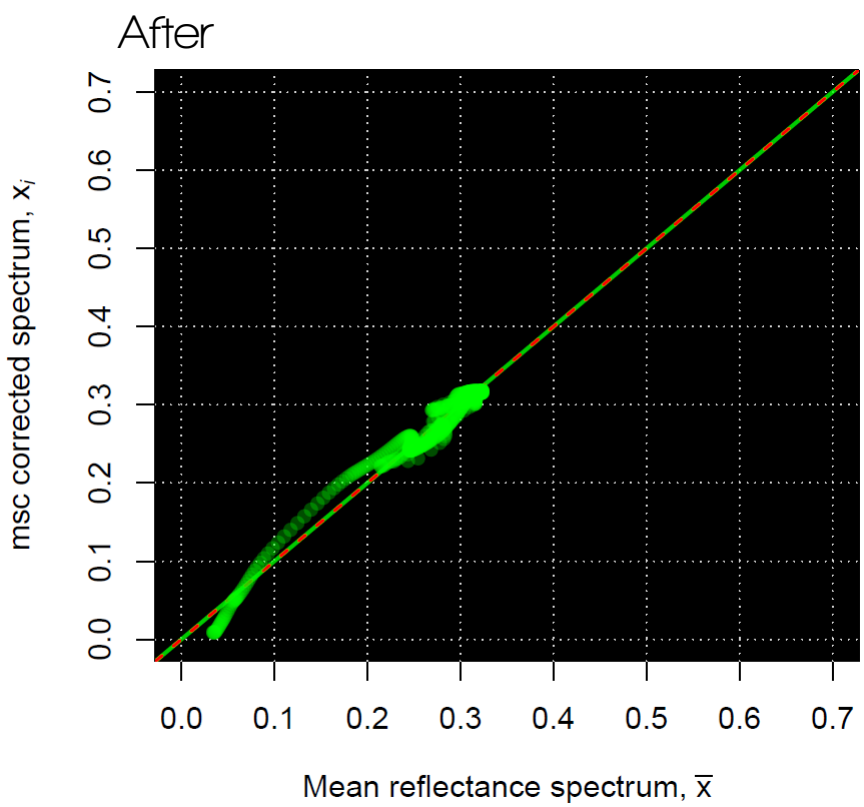
The `msc` function of the `pls` package can be used to apply the Multiplicative Scatter Correction

Multiplicative Scatter Correction (msc)

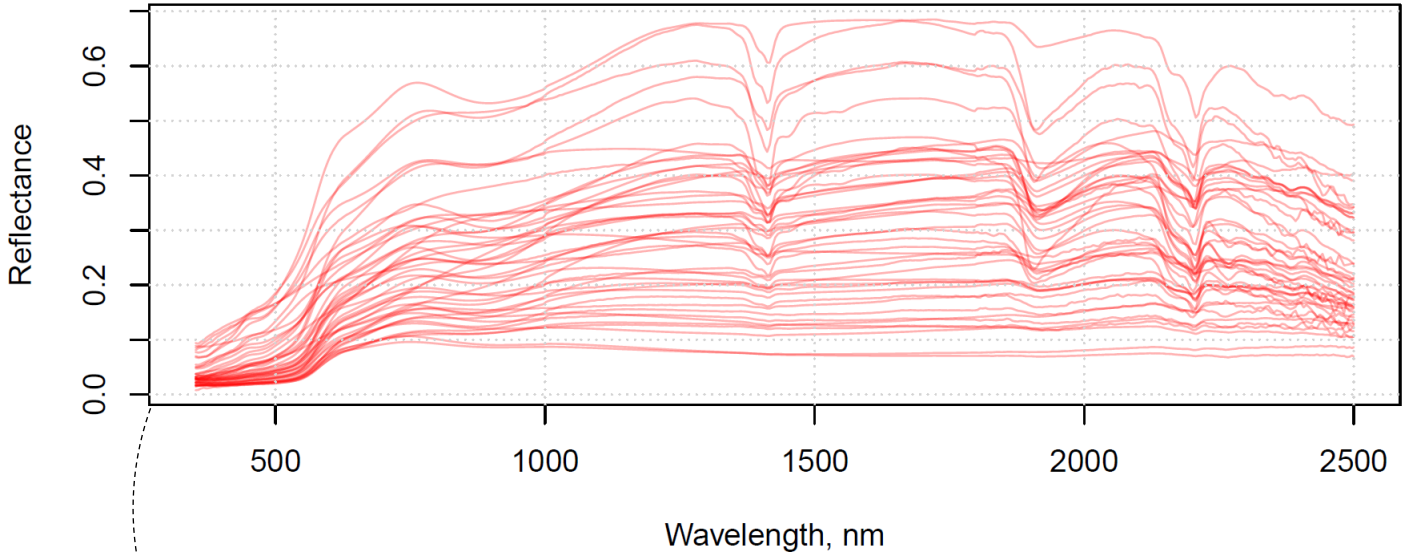
It can also be seen as an offset and slope adjustment of the spectra based on the average spectrum of the set of spectra

Example

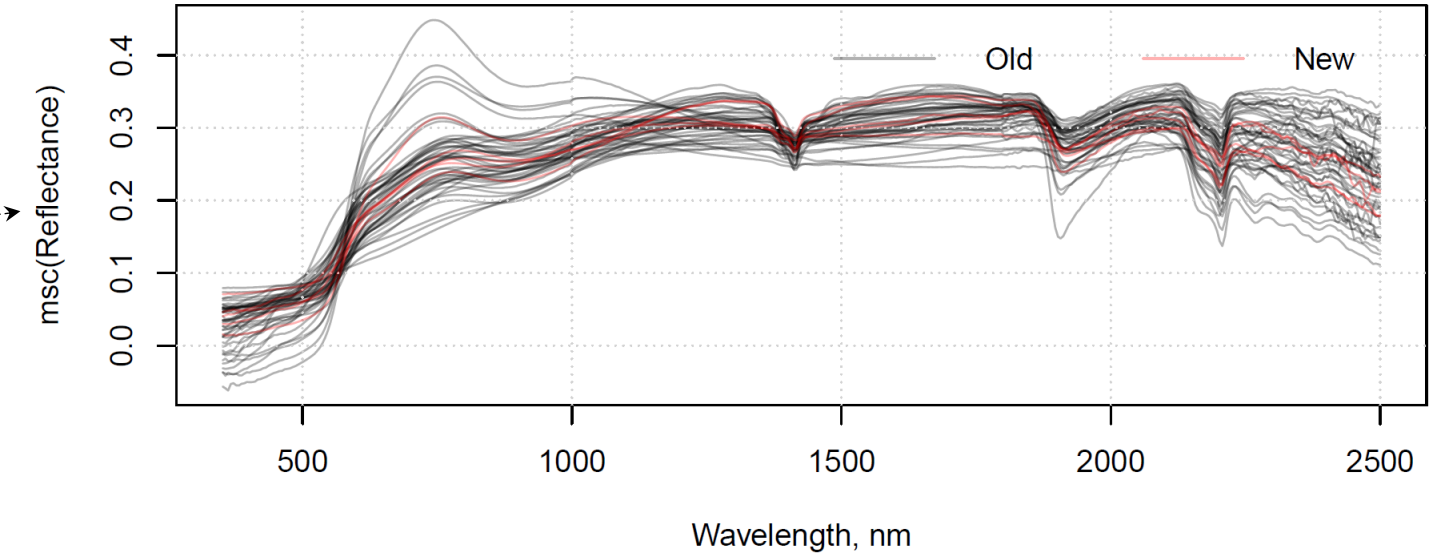




Multiplicative Scatter Correction (msc)



msc



Standard Normal Variate (snv)

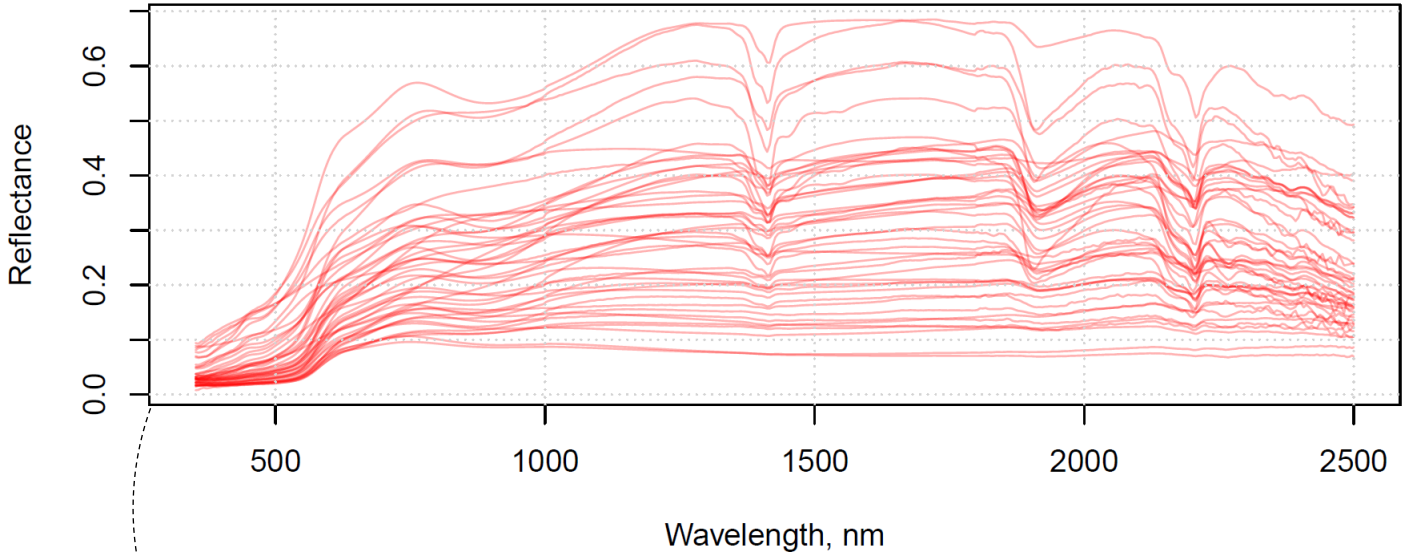
snv is another simple way for normalizing spectra that intends to correct for light *scatter*. But compared to msc, it operates row-wise:

$$\mathrm{snv}(x_i) = \frac{x_i - \bar{x}_i}{sd(x_i)}$$

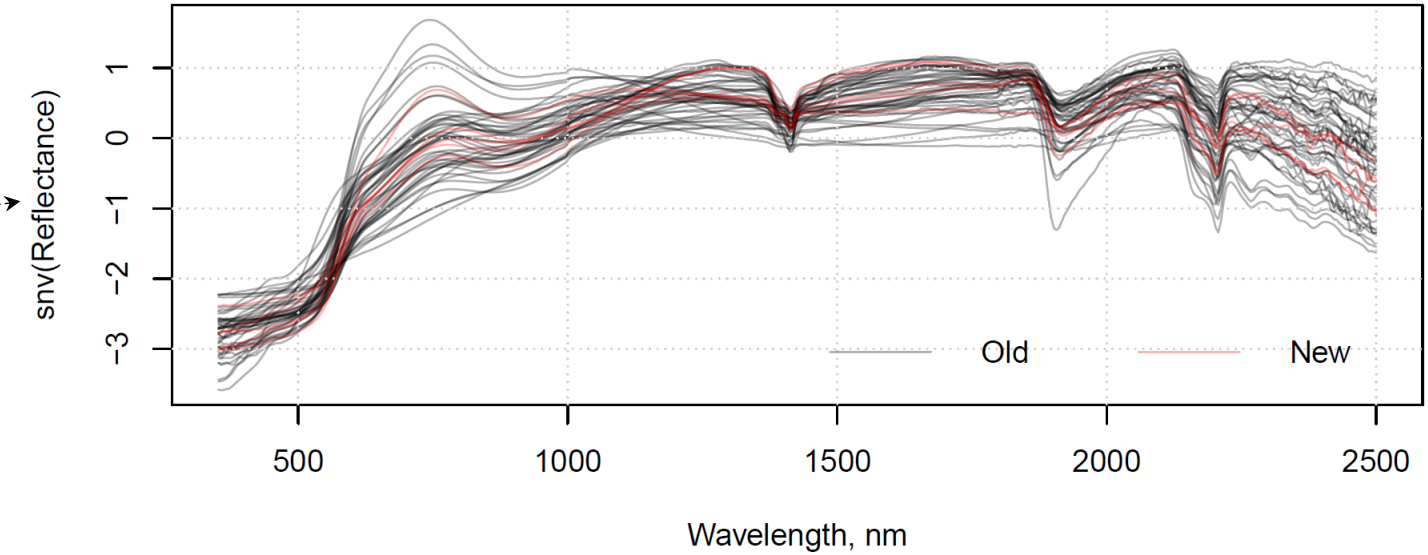
where $\bar{x}_i$ and $sd(x_i)$ represent the single mean and standard deviation values of the spectrum.

The `standardNormalVariate` function of the `prospectr` package can be used for applying this type of correction

Standard Normal Variate (snv)



snv



Centering and scaling

- *Centering:* a given matrix is transform into a matrix with columns that have zero mean. This is achieved by subtracting the mean vector of the matrix columns from each row of the original matrix. From a statistical point of view, *centering* eliminates the need for an intercept in the model and makes the model more simple and interpretable. This operation is **strongly** advised for most regression approaches, although it does not imply that in all cases the model performance will improve

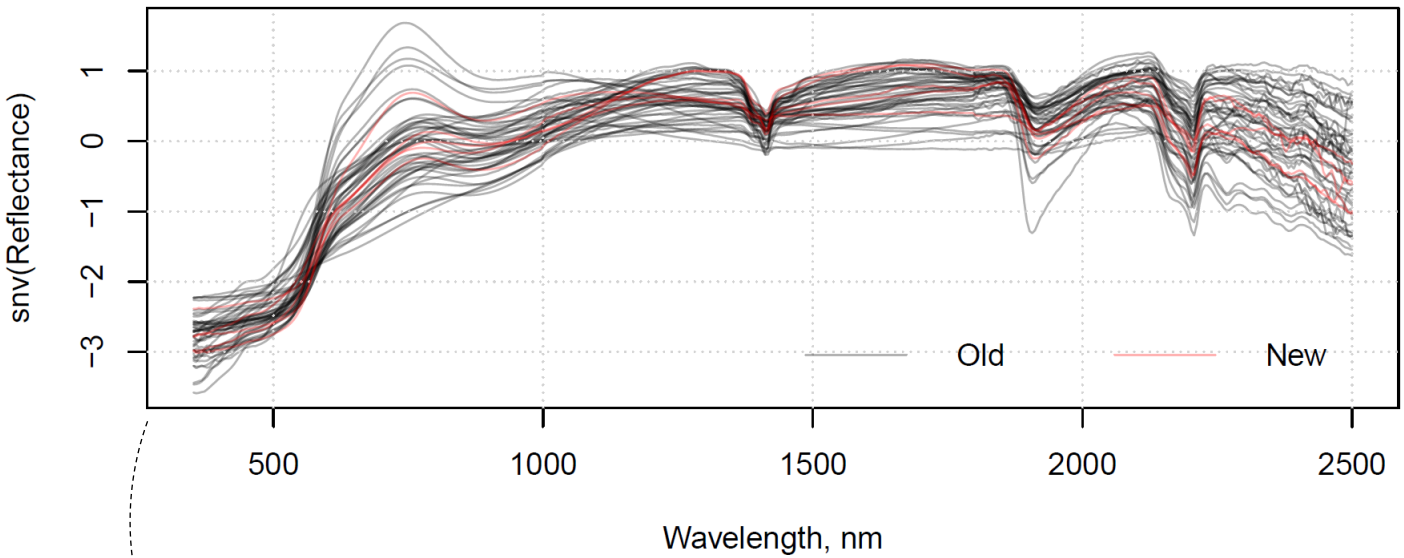
- *Scaling:* a given matrix is transform into a matrix with columns that have the same variance. This is achieved by dividing each row of the original matrix by the vector of standard deviations of the matrix columns. The purpose is to get all the variables . . . on the same scale.

- Both centering and scaling are usually applied after data transformation.

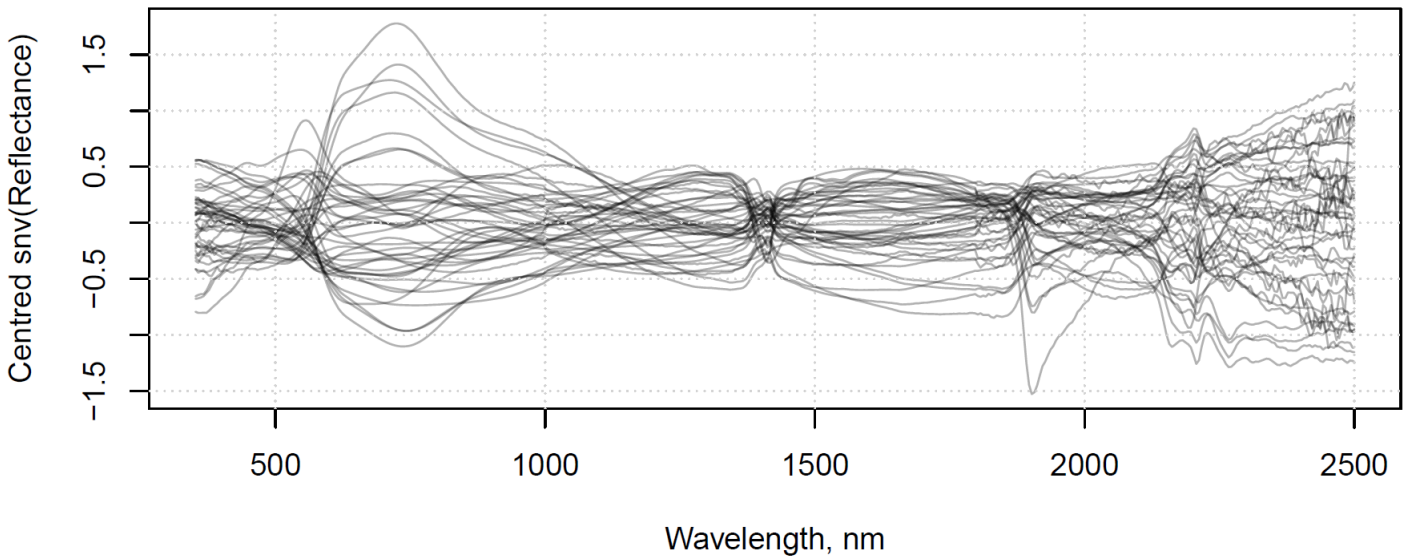
The `scale` function can be used for applying either centering and scaling or both

See file: 4.Preprocessing_III.R

Centering and scaling

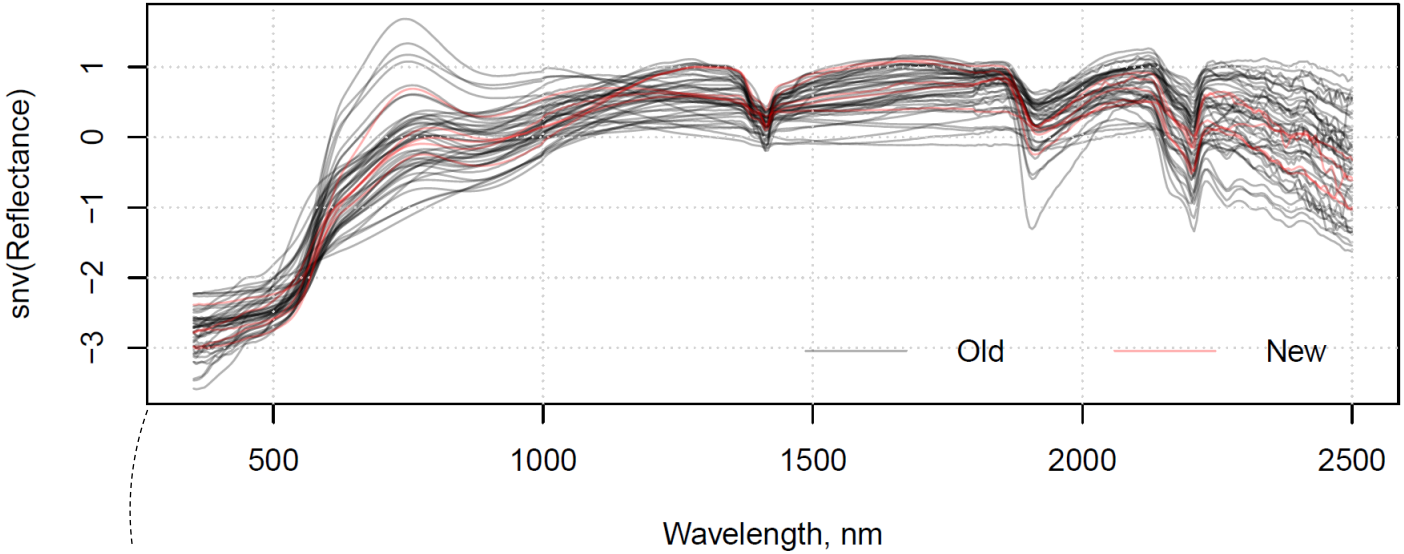


Centering

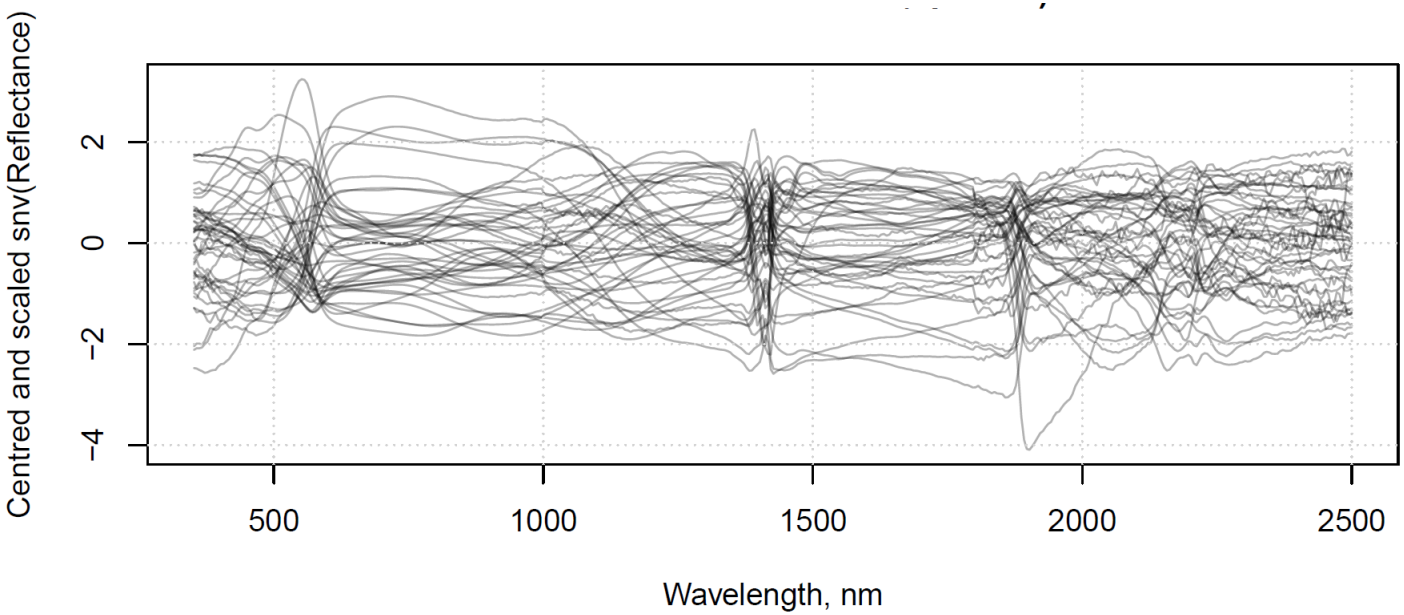


See file: 4.Preprocessing_III.R

Centering and scaling



Centering and scaling



See file: 4.Preprocessing_III.R

Centering and scaling

To *scale* or not to *scale*?

- Scaling will down-scale variables with large variation, but can inflate exaggeratedly variables with small variance (e.g. variables with no variation but small noise).
- Decision depends on the data pre-treatment also: is it meaningful?
- For instance, variation in some variables is greater than in other for good reasons (e.g. absorption features in continuum removed spectra, 1st derivative), so that *scaling* is not appropriate and will reduce model interpretability
- *Centering and scaling* is sensitive to outliers. A robust scaling can be computed using the median instead of the mean and the median absolute deviation (MAD) instead of the standard deviation

See file: 4.Preprocessing_III.R

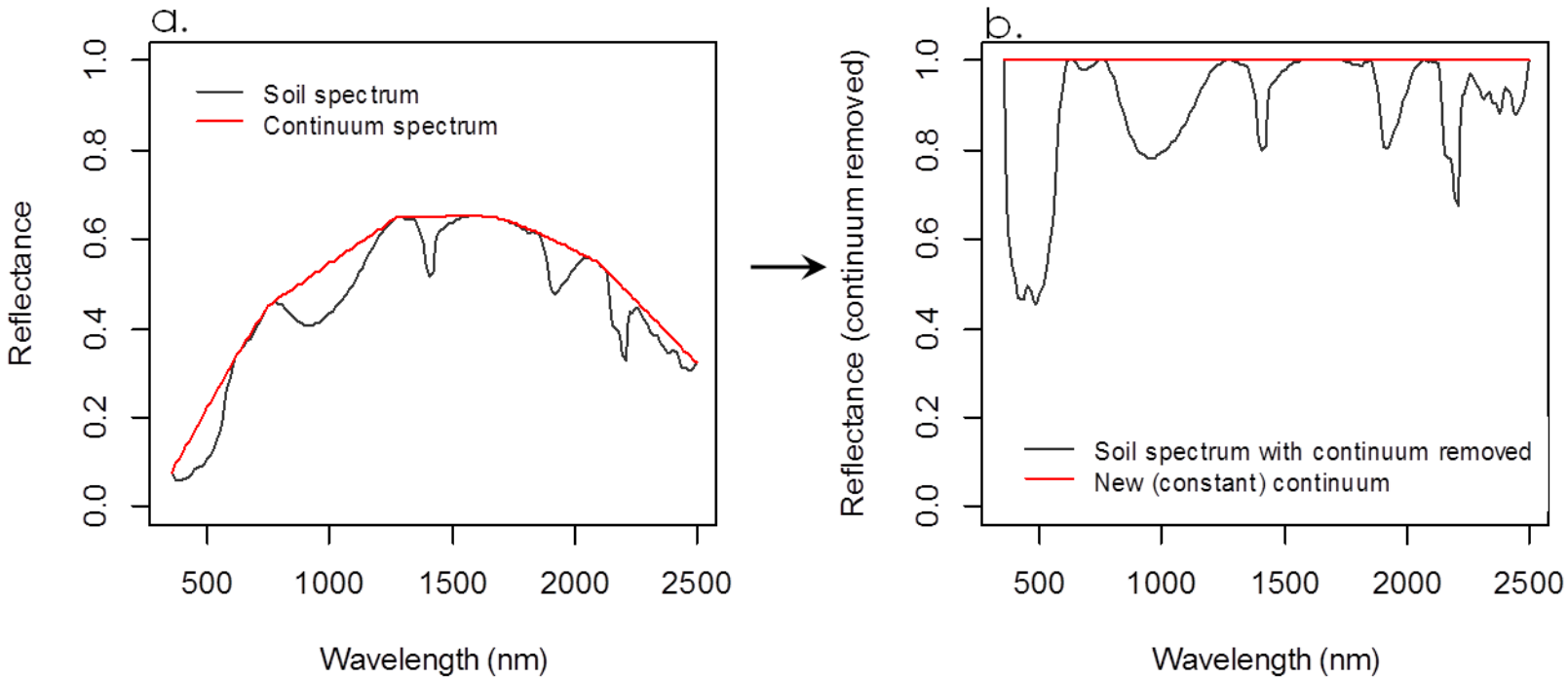
Continuum removal

- The continuum removal technique was introduced by Clark and Roush (1984) an effective method to highlight energy absorption features of minerals.
- It can be viewed as an albedo normalization technique. This technique is based on the computation of the continuum (spectrum) of a given spectrum.
- The continuum spectrum (or envelope) of a given spectrum (original spectrum) is defined as follows:
  - The local reflectance spectrum maxima points (in the case of absorbance local minima points) are identified.
  - Then, those points are concatenated (using linear interpolation at each original wavelength) to replicate the original spectrum but without absorption features.
  - The final continuum-removed spectrum is given by $\phi_i = \frac{x_i}{c_i}$ where x_i and c_i are the original and the continuum reflectance (or absorbance) values respectively at the ith wavelength, and ϕ_i is the respective final reflectance (or absorbance) value after continuum removal.

Clark, R.N., Roush, T.L. 1984. Reflectance spectroscopy: Quantitative analysis techniques for remote sensing applications, Journal of Geophysical Research 89, 6329- 6340.

See file: 5.Preprocessing_IV.R

Continuum removal



In soil vis-NIR spectroscopy this technique has been successfully employed in several works for describing soil mineralogical characteristics.

Most of the minerals commonly found in soils, present specific absorption features (spectral signature) in the vis-NIR regions of the electromagnetic spectrum

The **`continuumRemoval`** function of the **`prospectr`** package for applying this method

See file: 5.Preprocessing_IV.R

Step 1
- Collect your samples
- Make the scans (raw spectra)

Step 2
- Pre-processing
- Outlier detection

Step 3
- Select your calibration samples
- Multivariate calibration

Step 4
- (Optional) transfer data across different instruments

Measuring the similarity/dissimilarity between spectra

In NIR spectroscopy, similarity/dissimilarity estimates are important since they usually are the basis of several important methods for NIR spectra analysis (e.g. outlier detection, *k*-means clustering, Kennard-Stone sampling, local regression, etc).

- Similarity/dissimilarity measures between objects are often estimated by means of distance measurements.
- The closer two objects are to one another, the higher the similarity degree between them.
- There is a wide variety of distance metric algorithms which have been applied in NIR spectroscopy and remote sensing.

There are several methods to compute the similarity/dissimilarity between spectra. The most common ones are:

- Mahalanobis distance computed on the scores of the Principal Component (PC) space of the spectral data
- Correlation similarity
- Cosine distance (*a.k.a.* Spectral Angle Mapper)

See file: 6.Dissimilarity_between_spectra.R

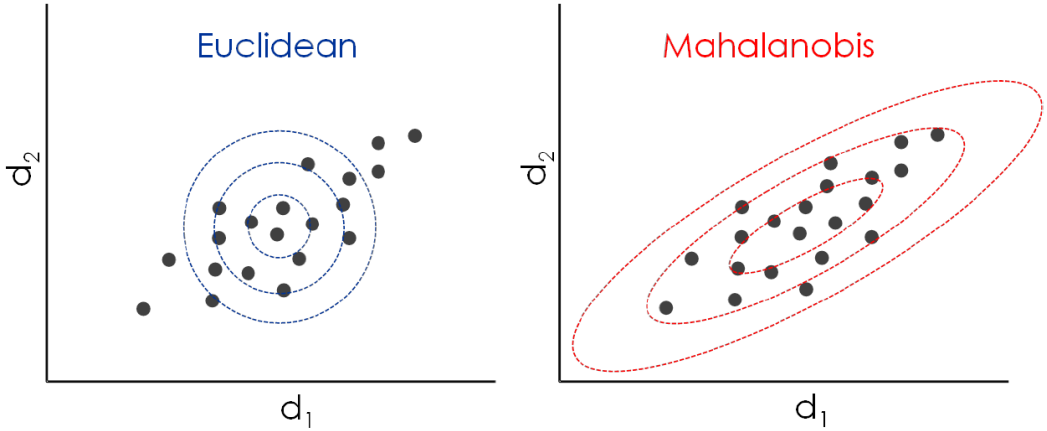
Mahalanobis distance computed on the PC space

For computing the pairwise Mahalanobis distance (M) between samples of a given dataset the following equation is applied to the scores (S) of the PCs:

$$M(S) = \frac{1}{p}\sqrt{(s_i - s_j)C^{-1}(s_i - s_j)}$$

where C^{-1} is the inverse of the covariance matrix of the matrix of scores (S) and p is the number of dimensions (number of PCs).

In contrast to the Euclidean distance, the Mahalanobis distance includes the term C^{-1} which accounts for correlation between variables. However, since the score variables of the PCs are orthogonal (i.e. uncorrelated), the Mahalanobis distance is equivalent to the Euclidean distance (after the score variables are centered and standardized/scaled to unit variance).



See file: 6.Dissimilarity_between_spectra.R

Correlation dissimilarity

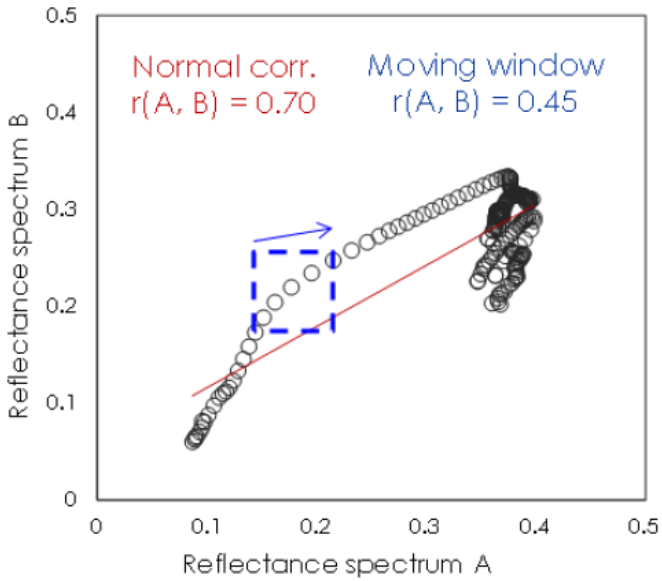
- The correlation dissimilarity (CD) is based on the correlation coefficient (r) between samples
- This is a dissimilarity measure and not a distance metric. For example, two spectral samples perfectly correlated (correlation = 1) can present different reflectance values. Despite this the CD is a good indicator of the spectral dissimilarity.

For computing thethe pairwise CD between samples of a given dataset (X) the following equation is applied directly to the spectra:

$$CD(X) = \frac{1 - r(x_i, x_j)}{2}$$

See file: 6.Dissimilarity_between_spectra.R

The CD can also be applied by using a moving window approach



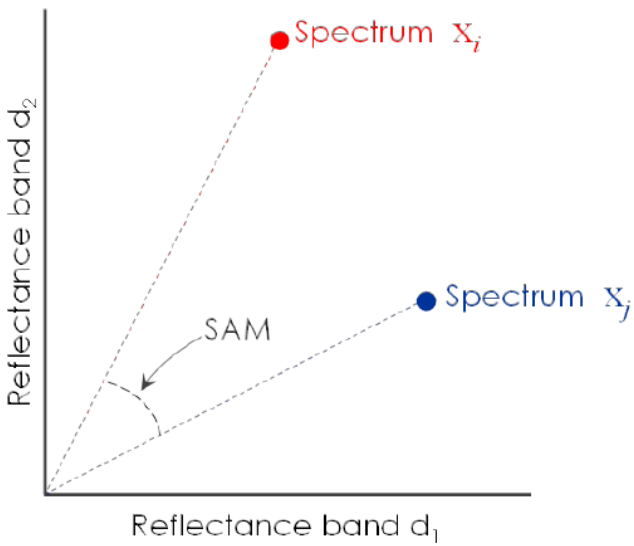
Here the dashed box indicates a moving window where the correlation similarity is calculated for that specific region. The final correlation similarity is computed by averaging all the moving window correlation similarity measures. In this example the global correlation between the spectra A and B is 0.70 while the final correlation measured by using the moving window approach is 0.45.

Cosine distance (*a.k.a.* Spectral Angle Mapper)

The spectral angle mapper (SAM, Yuhas et al., 1992) distance has been extensively applied in remote sensing as a tool for unsupervised classification and spectral similarity analysis. The SAM distance between two spectra (x_i and x_j) is calculated as:

$$SAM(\mathrm{x}_i, \mathrm{x}_j) = cos^{-1} \frac{\sum_{k=1}^{d} \mathrm{x}_{i,k}\mathrm{x}_{j,k}}{\sum_{k=1}^{d} (\mathrm{x}_{i,k}^2)^{1/2} \sum_{k=1}^{d} (\mathrm{x}_{j,k}^2)^{1/2}}$$

where d is the number of variables or spectral bands



An example of the SAM distance between two spectra with 2 bands

Yuhas, R. H., Goetz, A. F., & Boardman, J. W. (1992). Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm.

See file: 6.Dissimilarity_between_spectra.R

How reliable are the similarity/dissimilarity measures that I get from my spectra?

Methods for assessing the reliability of the similarity/dissimilarity measures between soil spectra are scare. In this workshop a "semi-supervised" method for doing so is also described along with a **R** example.

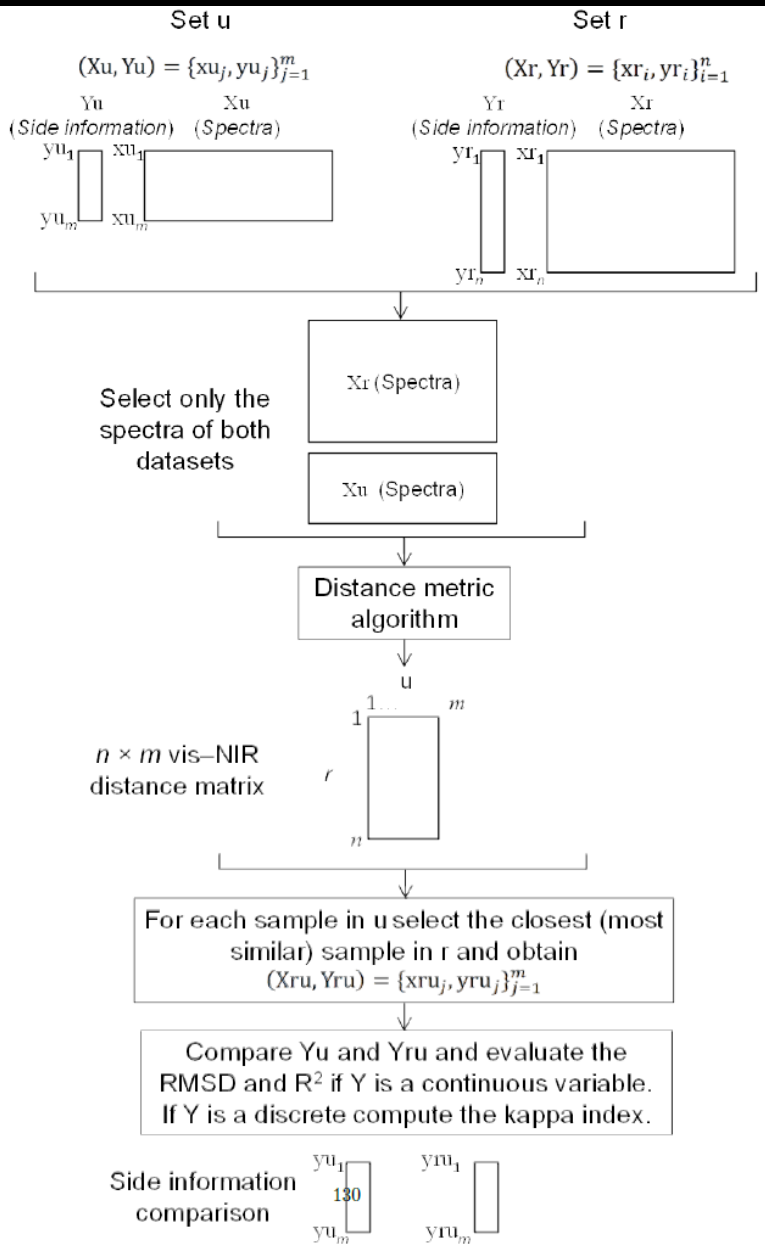
This method is based on a side information approach (Ramirez-Lopez et al., 2013). The side information means information about one or more variables available for a group of samples in a given dataset.

This approach is based on the assumption that the dissimilarity measures between the spectra should be able to reflect their dissimilarity also in terms of their side information (e.g. compositional similarity).

Ramirez-Lopez, L., Behrens, T., Schmidt, K., Rossel, R. V., Demattê, J. A. M., & Scholten, T. (2013). Distance and similarity-search metrics for use with soil vis-NIR spectra. *Geoderma*, *199*, 43-53

Measuring the similarity/dissimilarity between spectra

Asessing the dissimilarity measures by using side information



The side information approach works as follows:

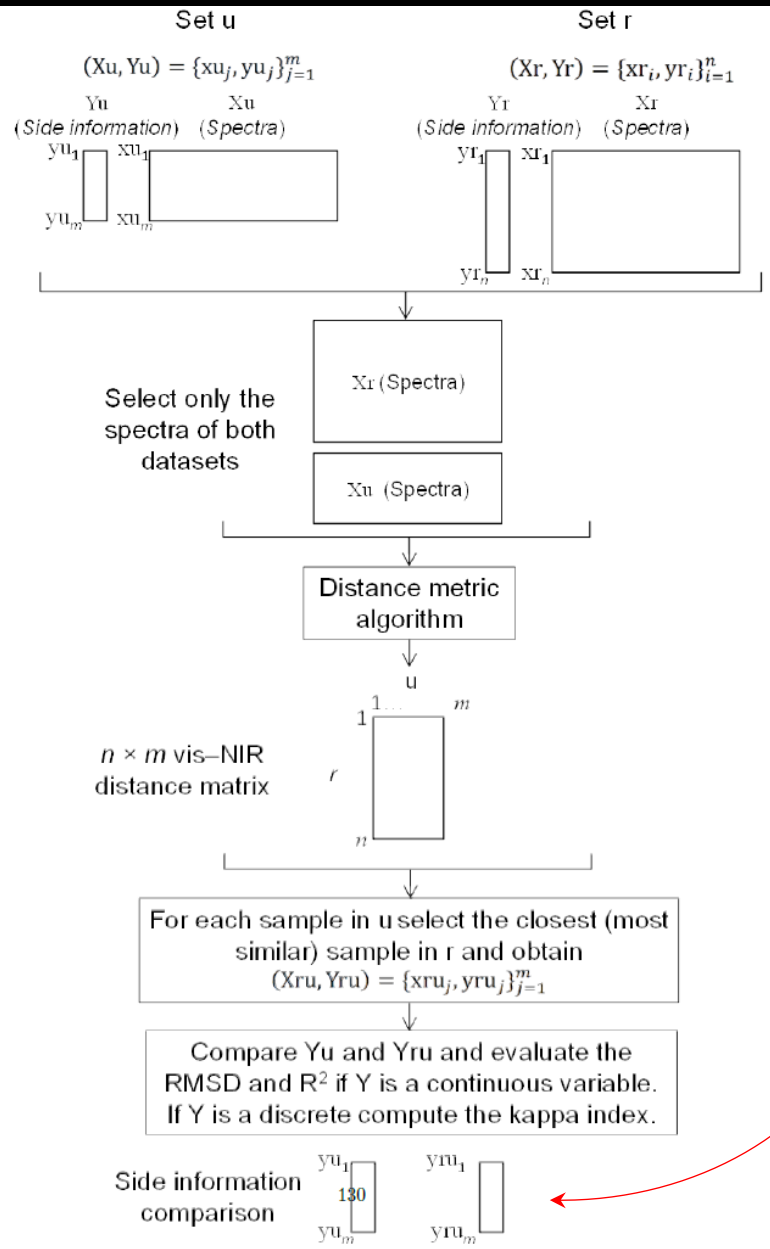
1. A distance matrix is derived from the spectral matrix X. This spectral matrix has a side information Y.
2. By using the distance matrix, for each sample in X select its closet (most spectrally similar) sample.
3. Compare the side information of each sample in X to the side information of its corresponding closest sample. Evaluate the statistics of these comparisons for assessing the reliability of the distance metric algorithm

Here we describe this semi-supervised method based on the Mahalanobis distance computed on the PC space, as it is one of the most 'robust' methods for computing the dissimilarity between spectra.

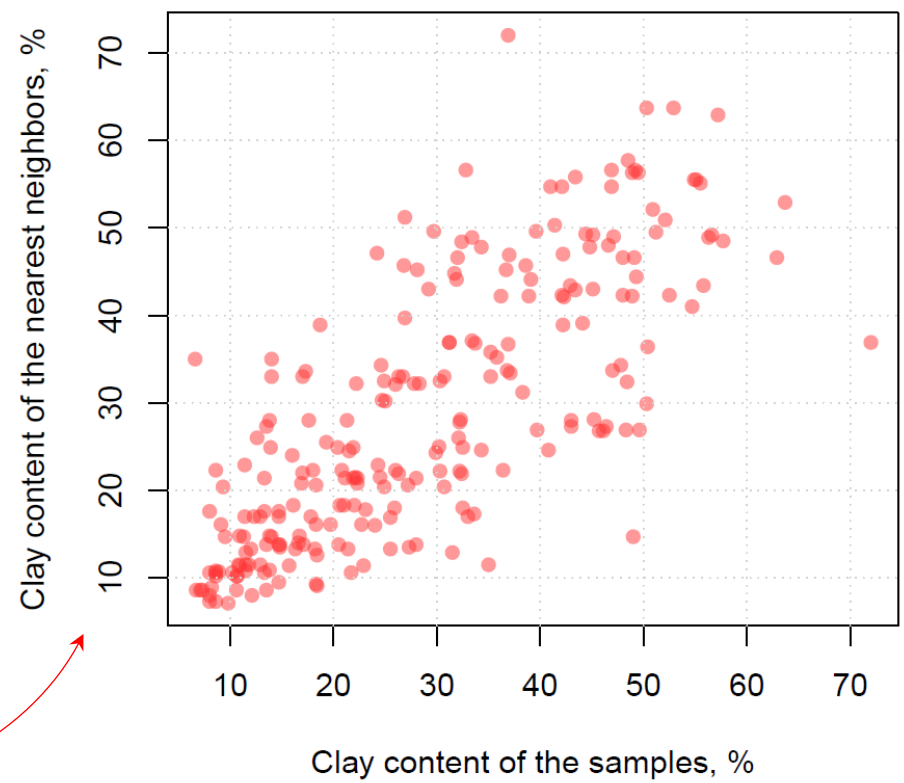
See file: 7.Assessing_the_dissimilarity_between_spectra.R

Measuring the similarity/dissimilarity between spectra

Asessing the dissimilarity measures by using side information



Example of a graphical comparison between the samples in a NIR dataset and its nearest neighbors in terms of their side information (clay content in this case)



See file: 7.Assessing_the_dissimilarity_between_spectra.R

Outlier detection

Outlier: An observation which is significantly distant (dissimilar) from the population.

Spectral outliers may be due to variations related to:

- Sample inhomogeneity
- Spectral measurement errors (instrument noise, wrong optical set up)
- Chemical/physical variation in sample (e.g. sample belongs to a different population)

Types of spectral outliers:

- Global outliers: samples lying far away from the center of the population
- Local outliers: samples lying far away from given local area
- Model outliers: samples that do not fit in the relationship between spectra (**X**) and the response variable (**Y**)

In NIR spectroscopy, outlier detection is usually carried out by using spectral dissimilarity estimates

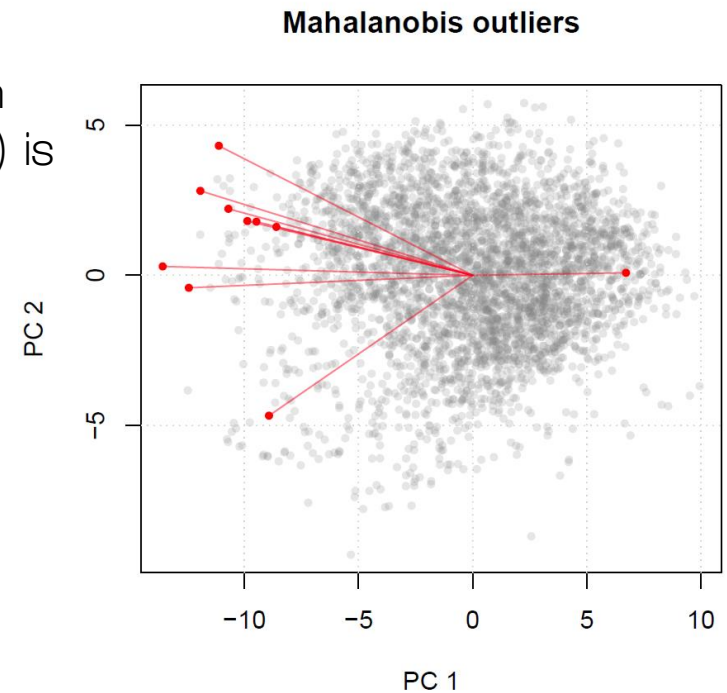
See file: 8.Outlier_detection.R

Outlier detection

In NIR spectroscopy, the Mahalanobis distance (in the PC space) to the center of the spectral set ($\bar{s}$) is the most widely used method for outlier detection

$$M(S) = \frac{1}{p}\sqrt{(s_i - \bar{s})C^{-1}(s_i - \bar{s})}$$

Samples with M values larger than 1 (arbitrary threshold) are usually considered as outliers



Alternatively the H distance can also be used for outlier detection, it can be derived from M as follows:

$$H(S) = \sqrt{p\,M^2}$$

Samples with H values larger than 3 (arbitrary threshold) are usually considered as outliers

There is no consensus on outlier detection methods and limits… The best advice is: apply your expert knowledge to judge such samples that are far apart from the center and set your own (statistically sound) limits

See file: 8.Outlier_detection.R

How many observations (samples) should be included in a calibration set in order **to efficiently develop the best NIR models**?

How many samples should be used for calibration?

→ Calibration set size

How to select the calibration samples?

→ Sampling design

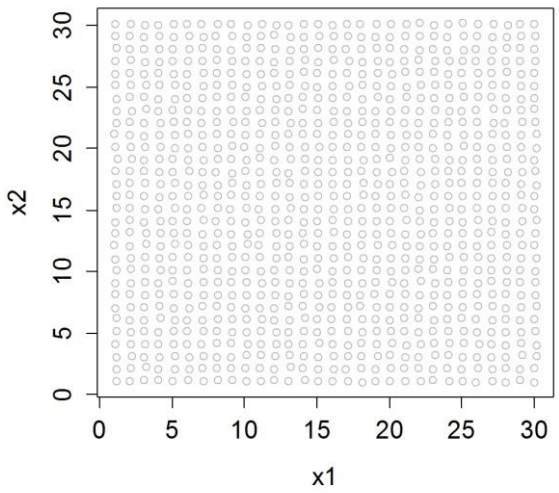
Step 1
- Collect your samples
- Make the scans (raw spectra)

Step 2
- Pre-processing
- Outlier detection

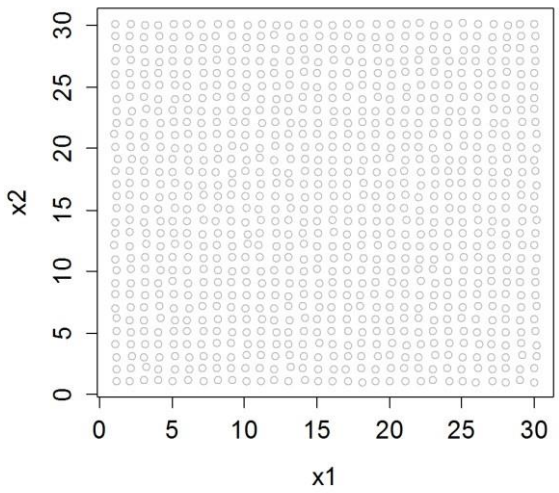
Step 3
- Select your calibration samples
- Multivariate calibration

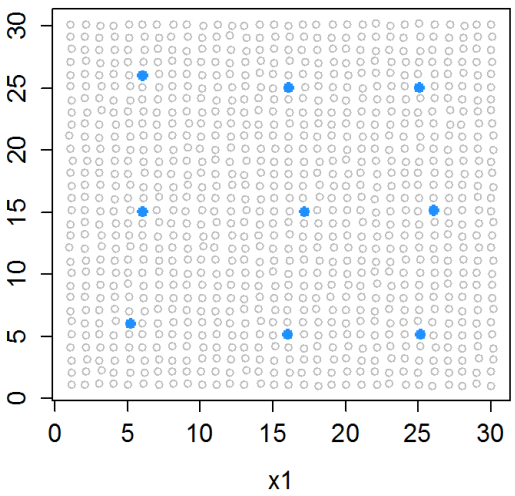
Step 4
- (Optional) transfer data across different instruments

Sample representativeness



Sample representativeness





Kennard-Stone sampling (KSS)

The KSS algorithm (Kennard and Stone, 1969) to select a (calibration) subset of m samples (S_{tr}) from a given set of N samples (S) works as follows:

1. Compute the distance matrix between all the samples in X
2. Find in S the two samples s_{tr1} and s_{tr2}, which are farthest apart, allocate them in S_{tr} and remove them from S.
3. Find in S the sample s_{tr3}, which is the most dissimilar to the ones already allocated in S_{tr}. Allocate s_{tr3} in S_{tr} and then remove it from S. Note that the dissimilarity between S_{tr} and each s_i is given by the minimum distance of any sample allocated in S_{tr} to each s_i.
4. Repeat the step $3m - 4$ times in order to select the remaining samples.

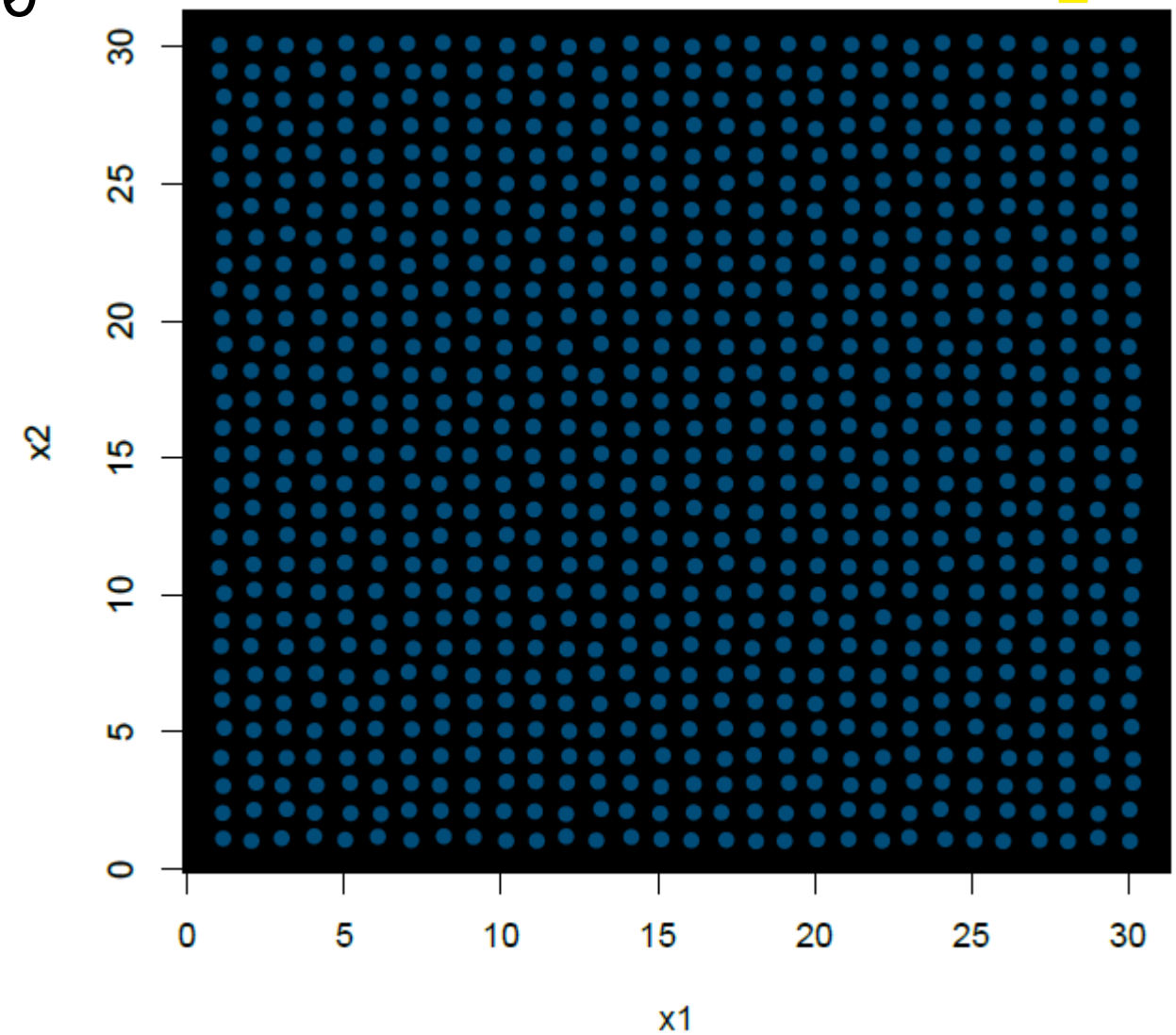
The KSS is usually applied to the PC socres of the spectral data

The **`kenStone`** function of the **`prospectr`** package can be used for KSS

Kennard, R.W., Stone, L., 1969. Computer aided design of experiments. Technometrics 11, 137-148.

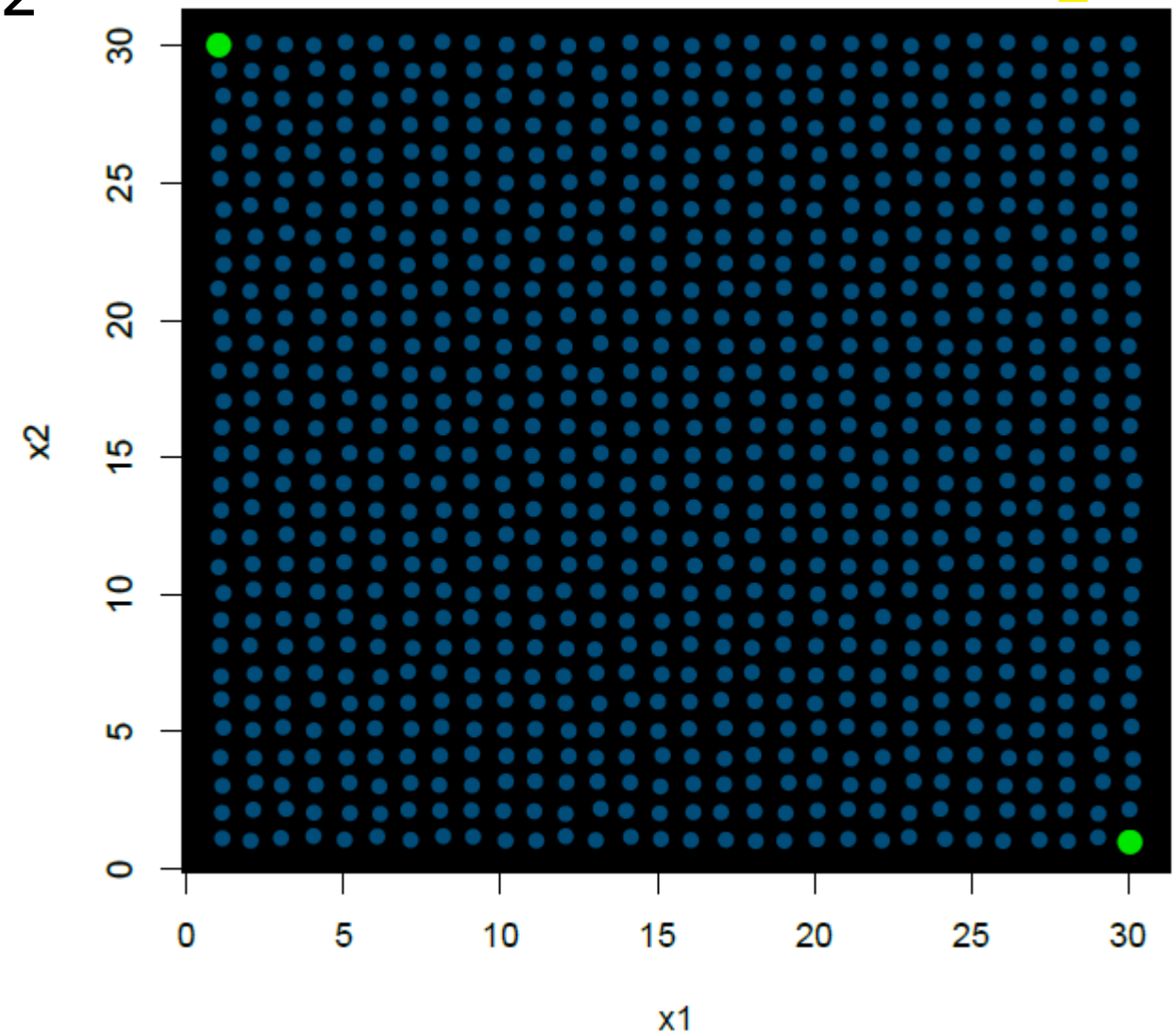
n = 0

Kennard-Stone sampling (KSS)



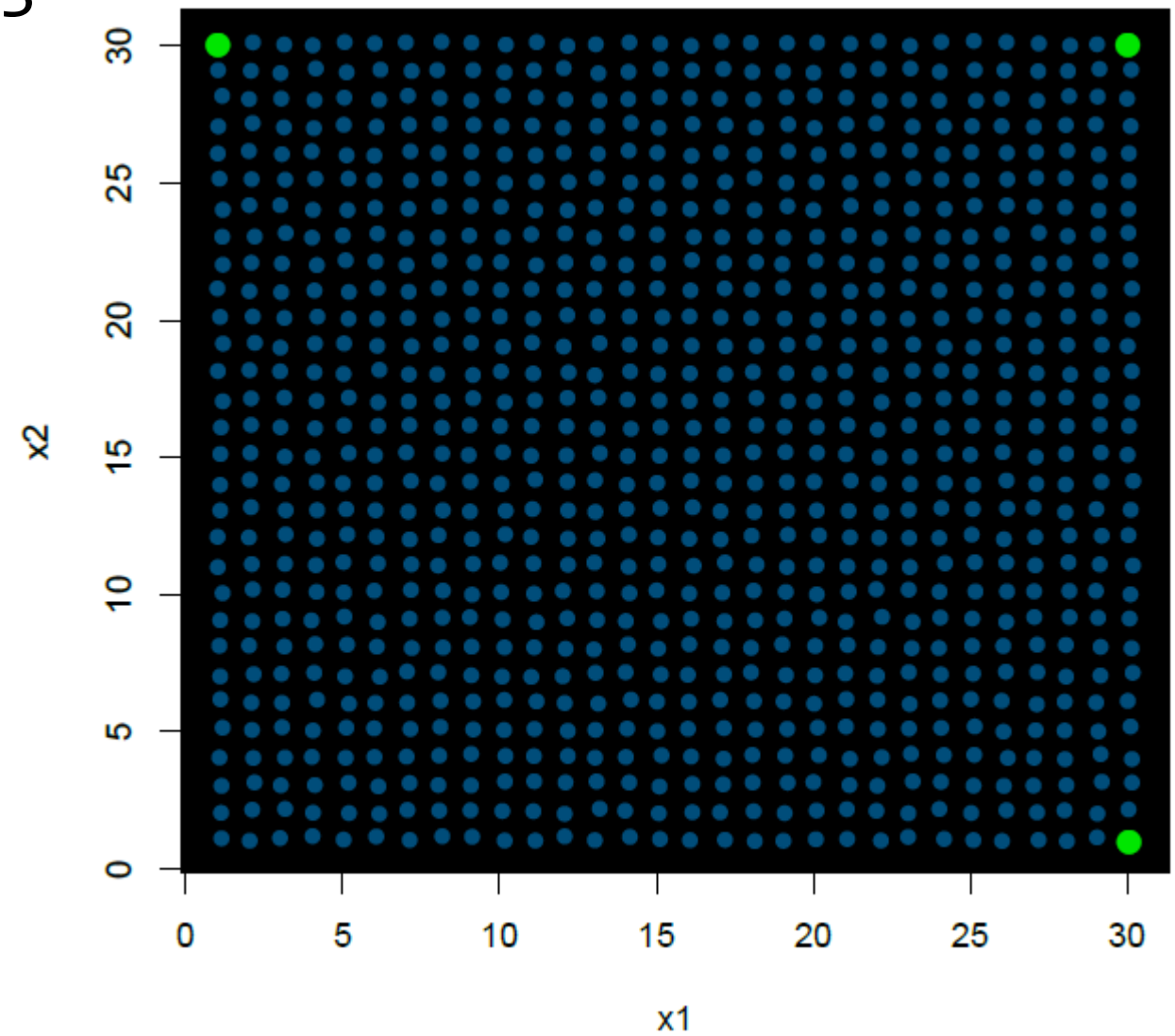
n = 2

Kennard-Stone sampling (KSS)



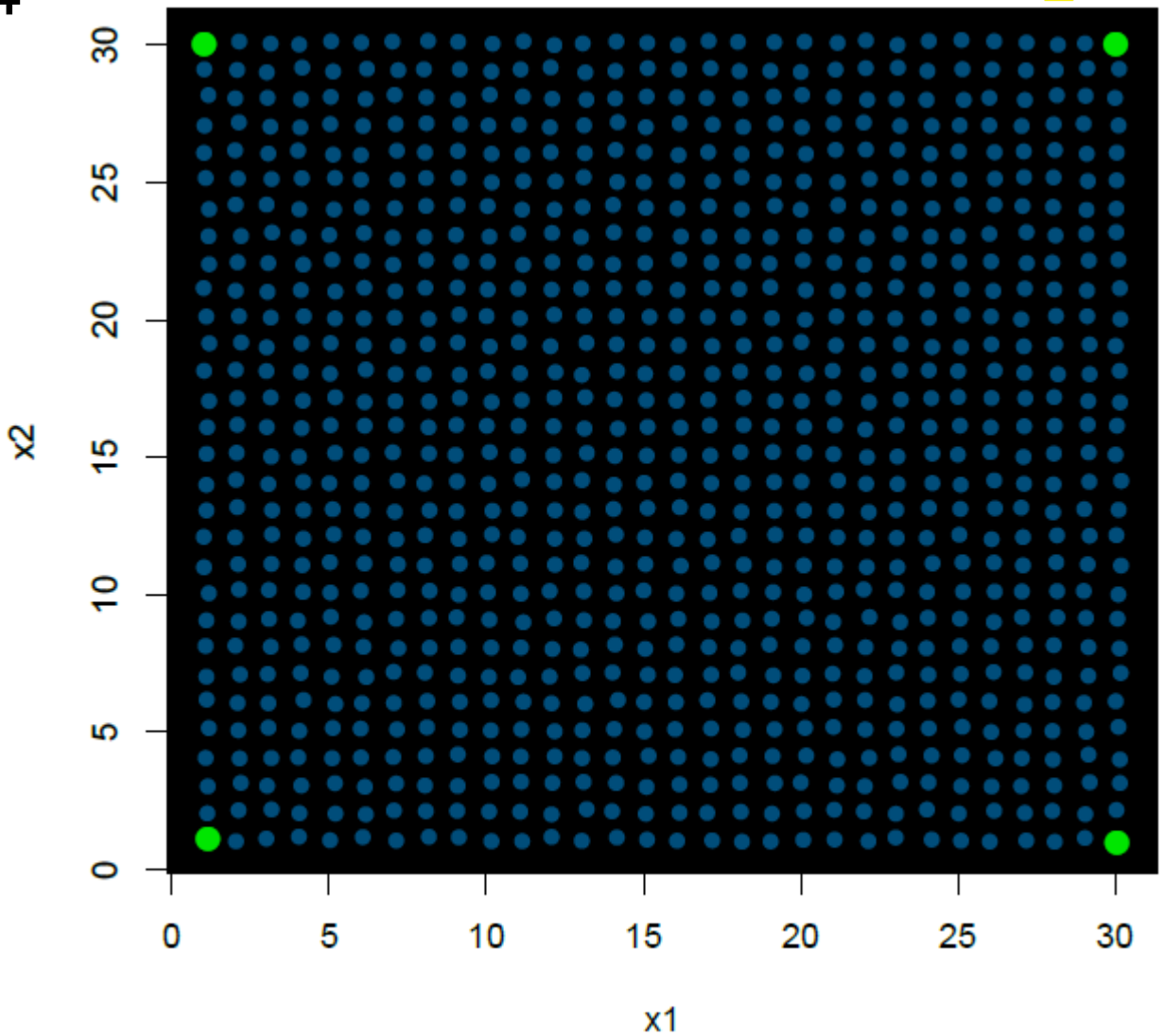
n = 3

Kennard-Stone sampling (KSS)



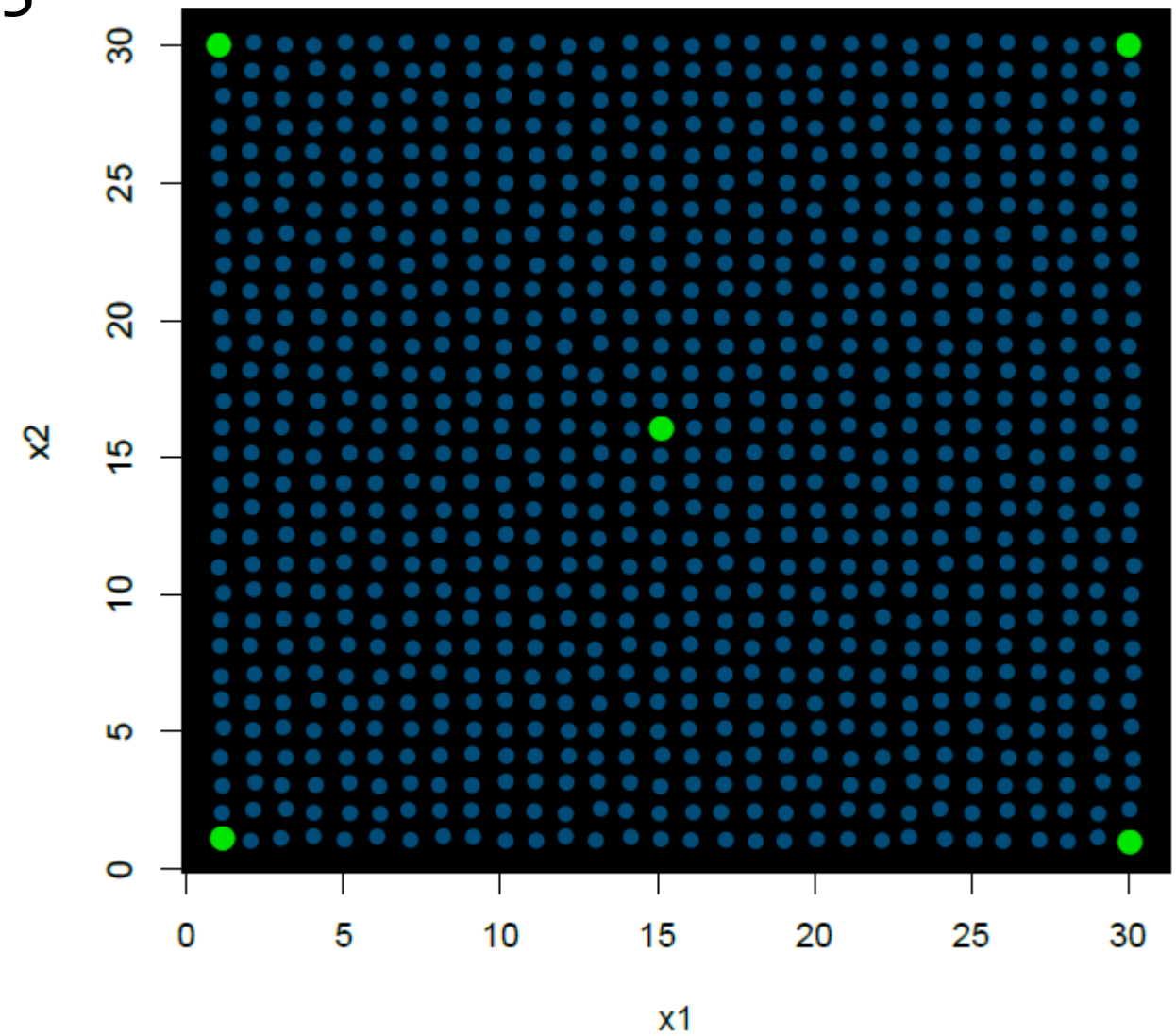
n = 4

Kennard-Stone sampling (KSS)



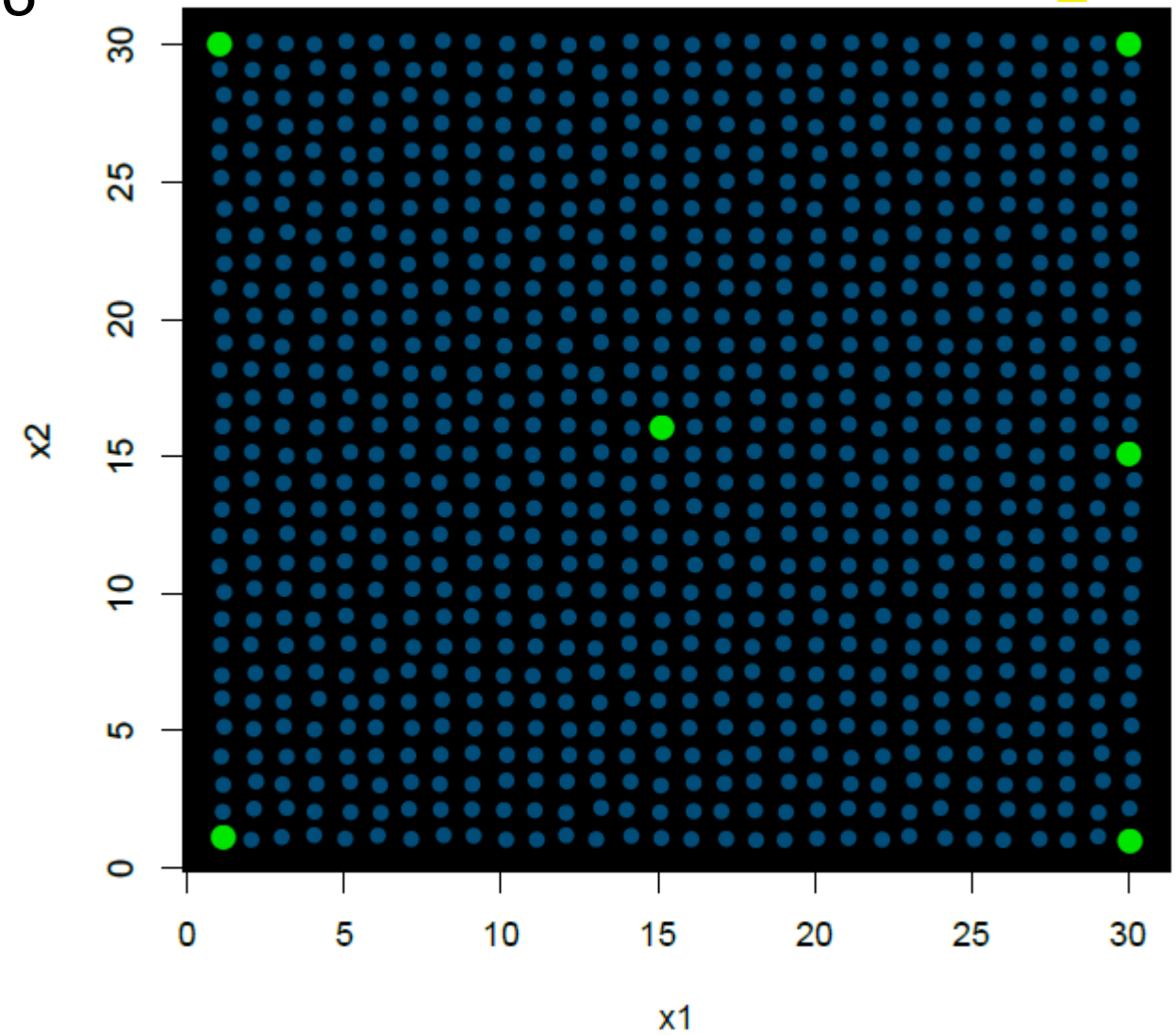
n = 5

Kennard-Stone sampling (KSS)



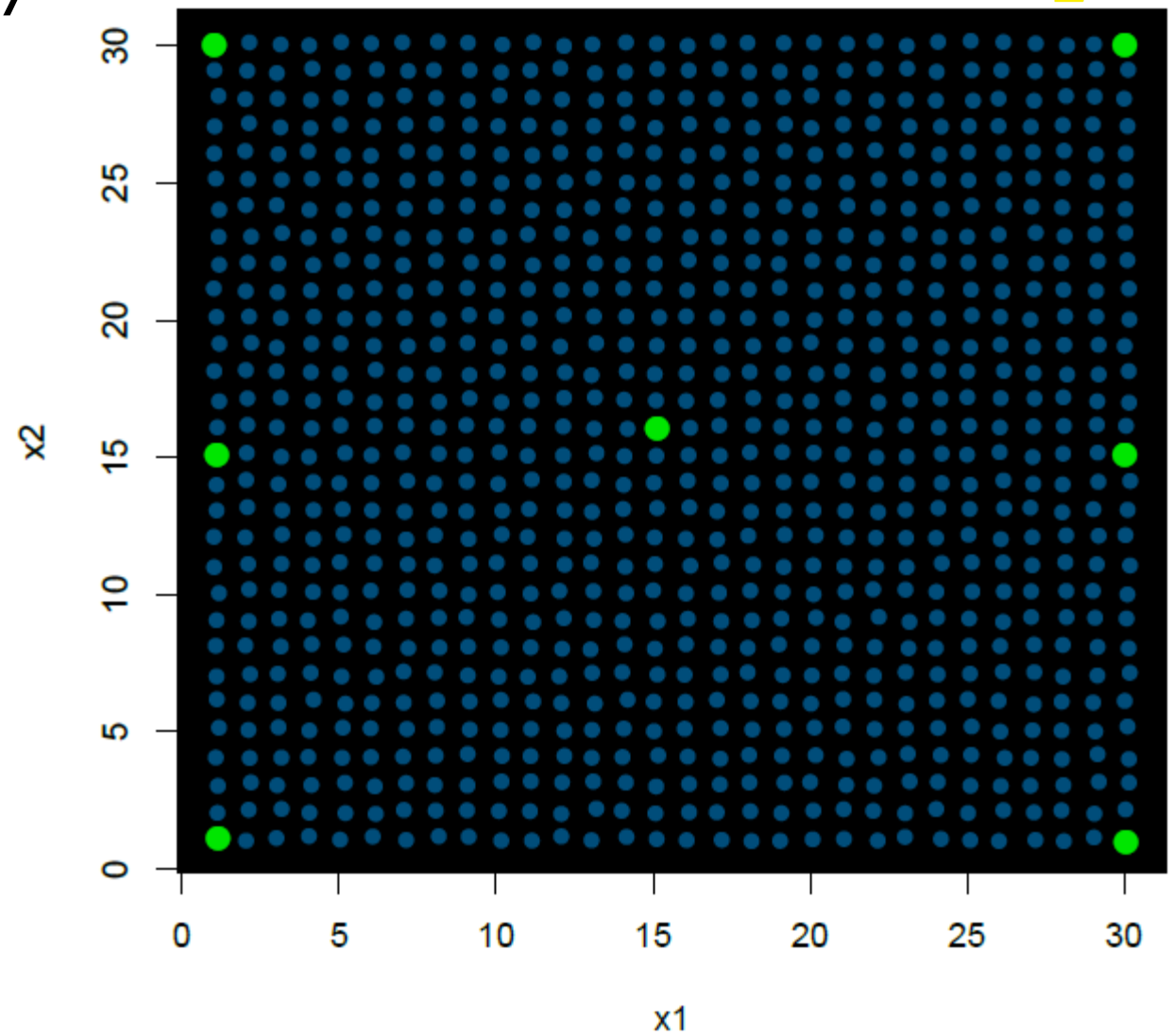
n = 6

Kennard-Stone sampling (KSS)



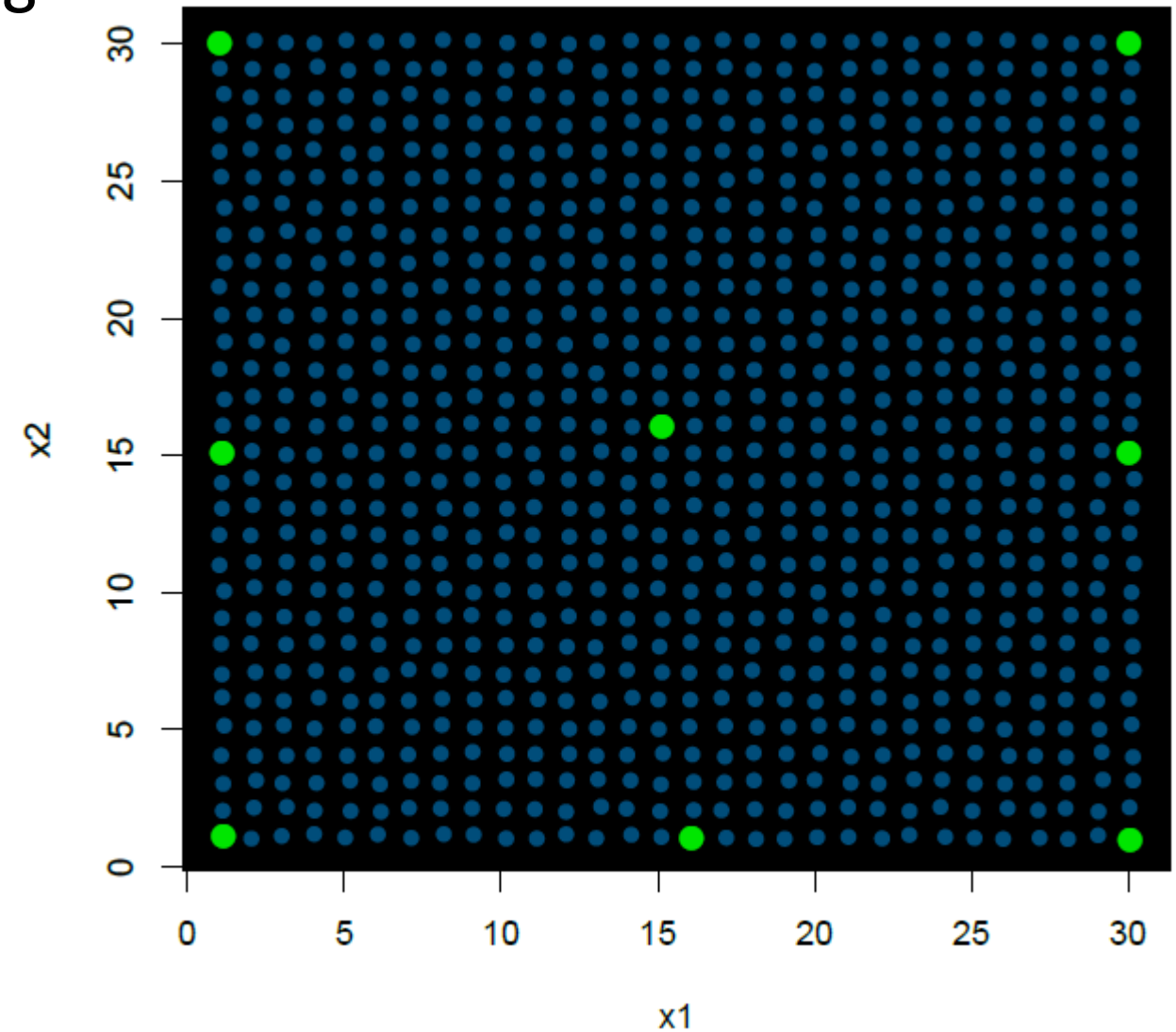
n = 7

Kennard-Stone sampling (KSS)



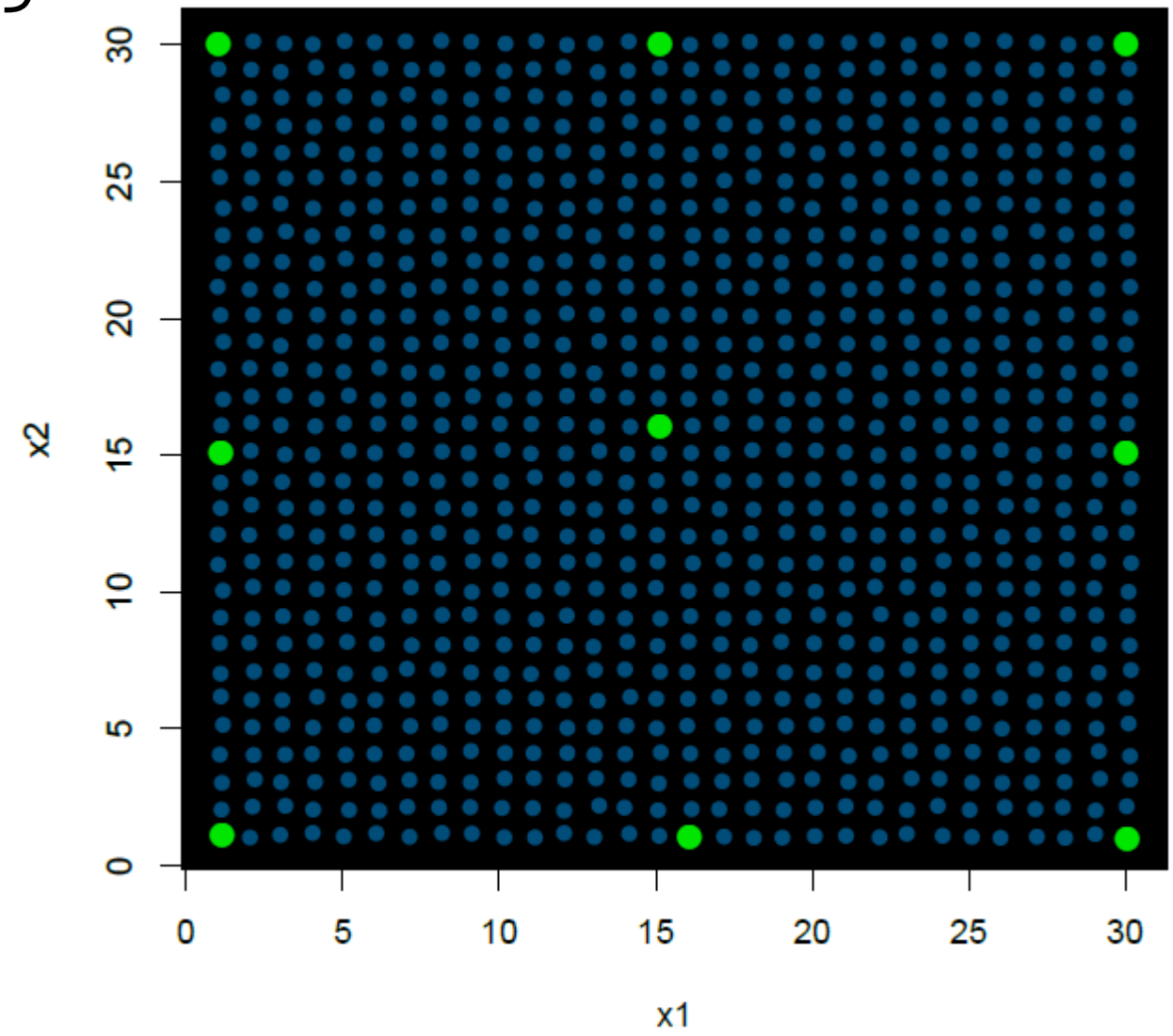
n = 8

Kennard-Stone sampling (KSS)



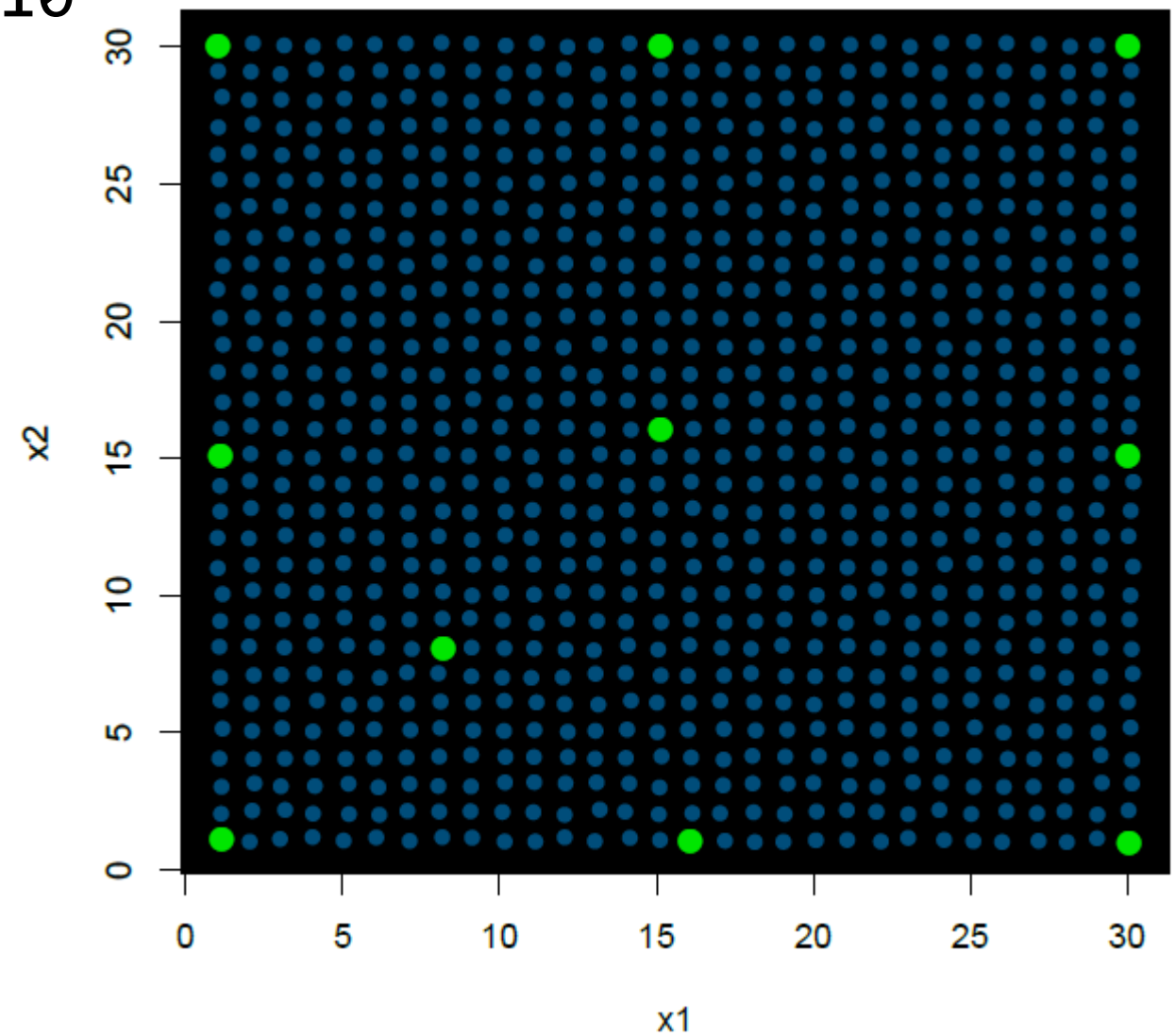
n = 9

Kennard-Stone sampling (KSS)



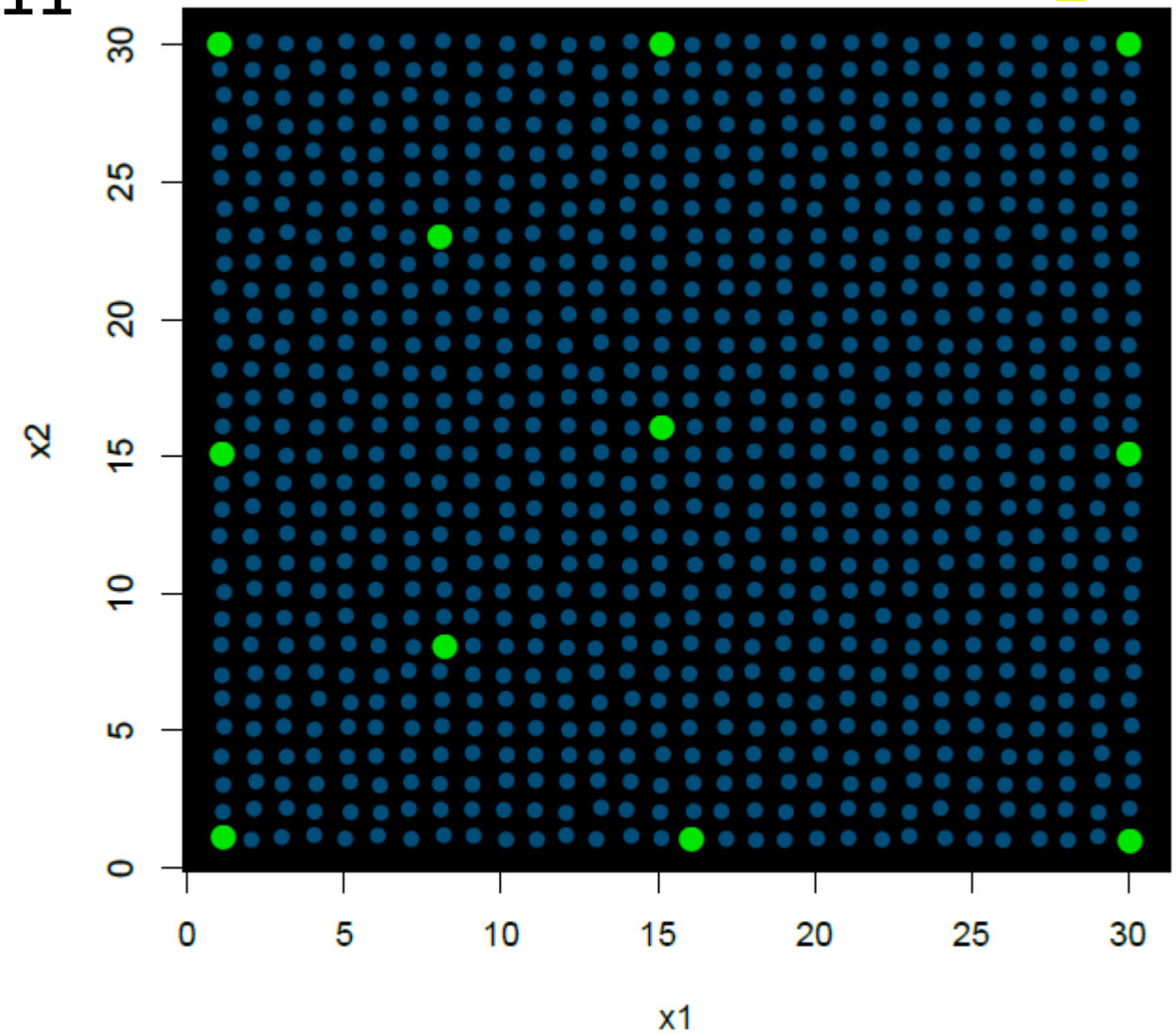
n = 10

Kennard-Stone sampling (KSS)



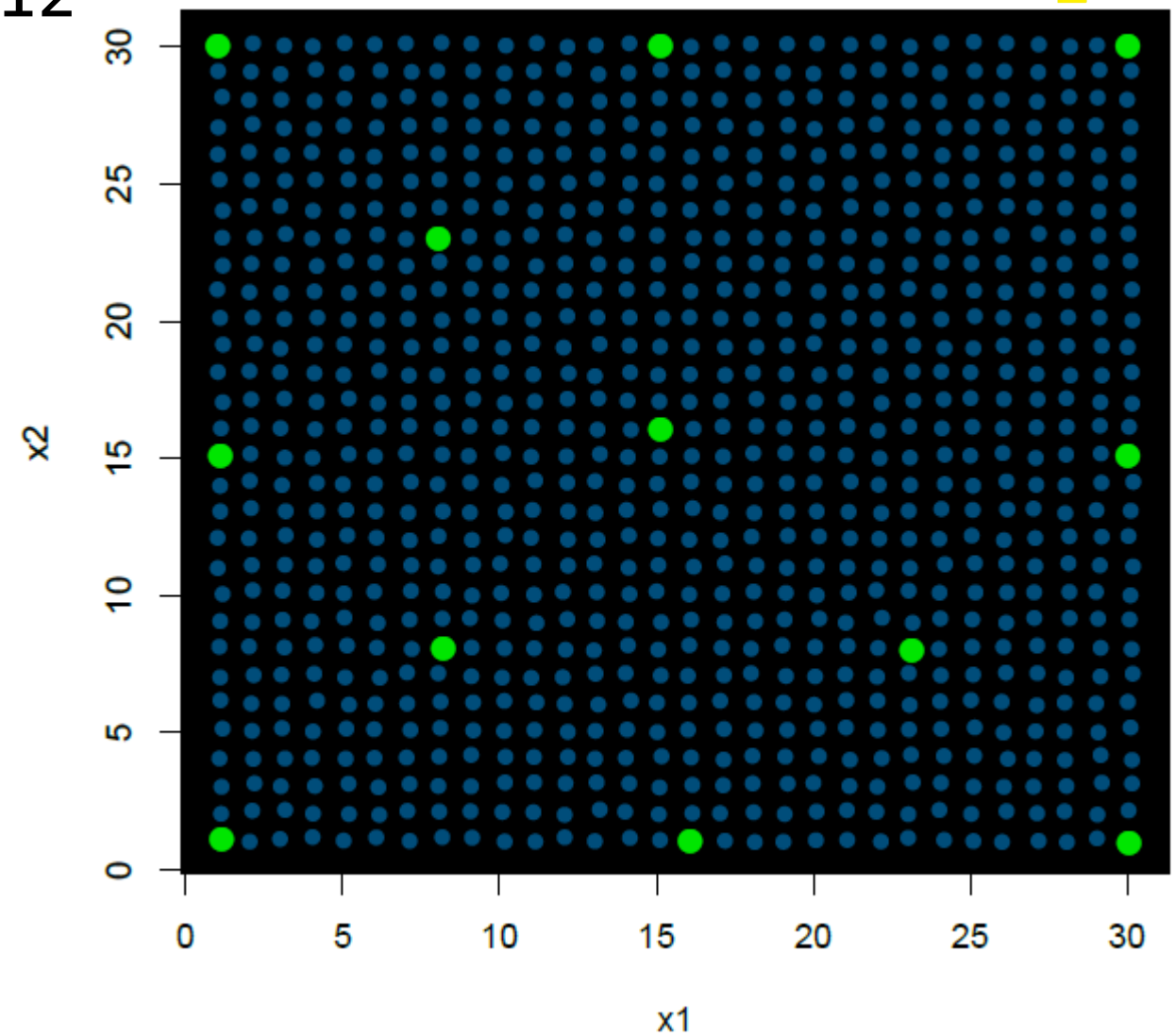
n = 11

Kennard-Stone sampling (KSS)



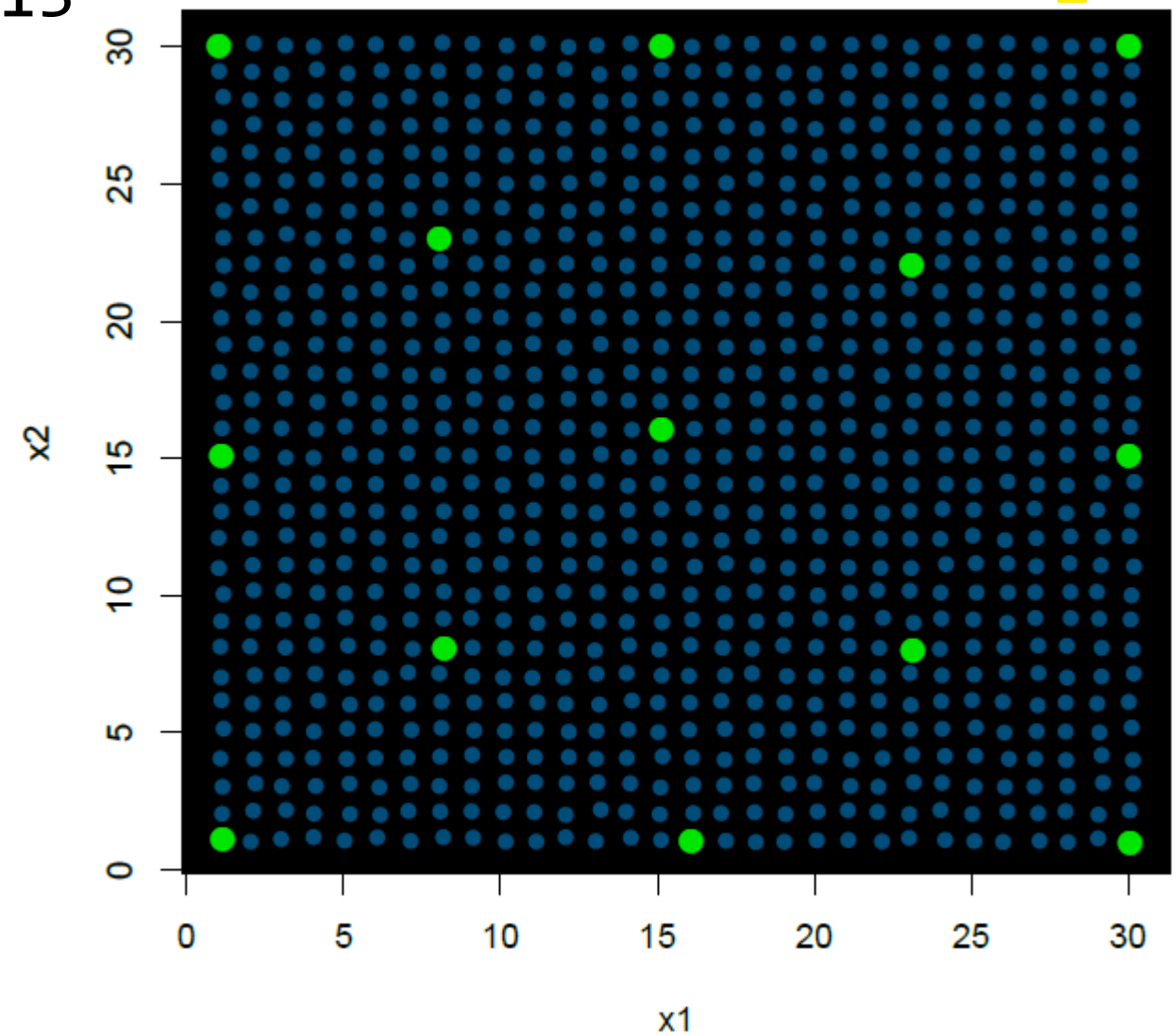
n = 12

Kennard-Stone sampling (KSS)



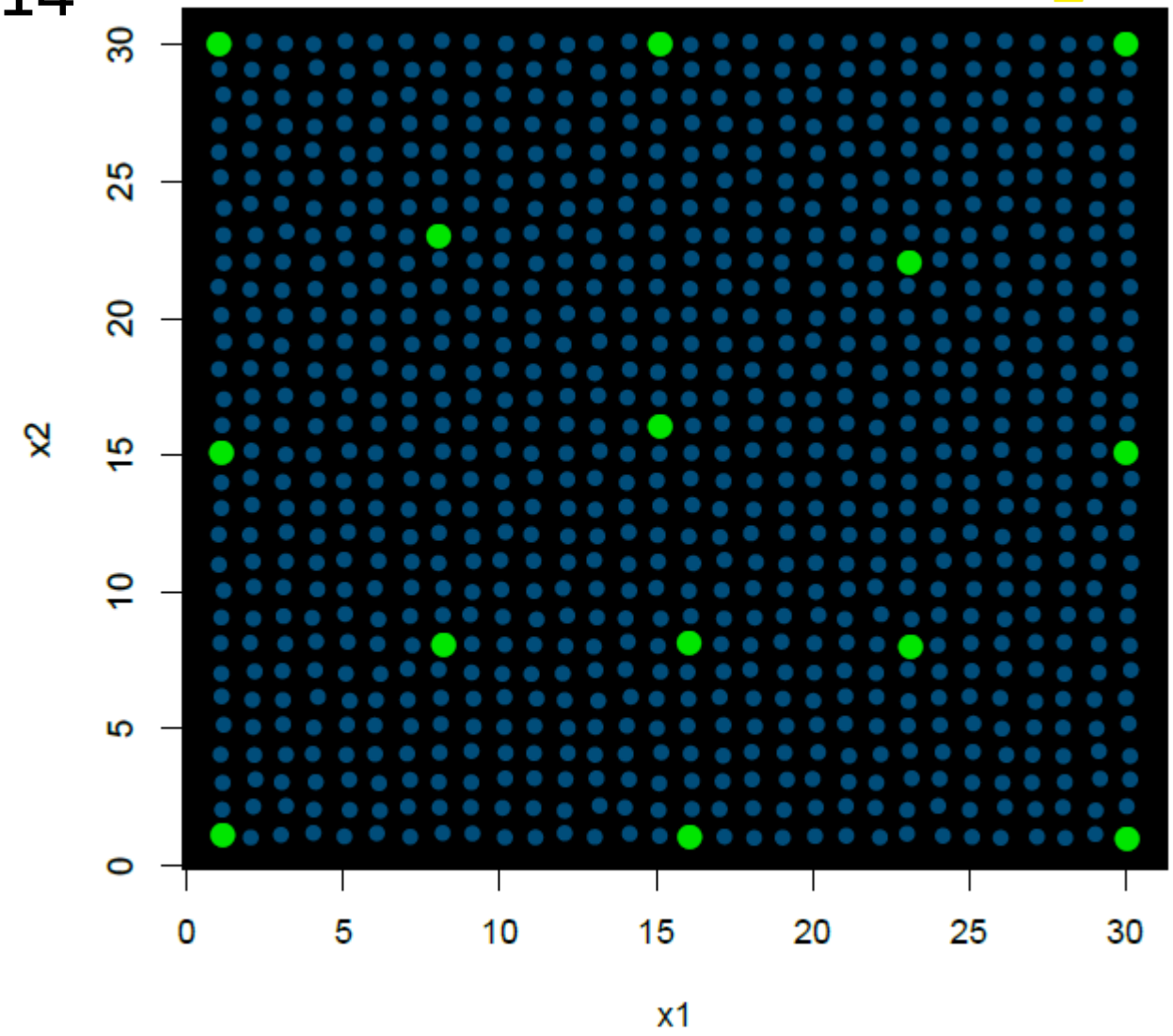
n = 13

Kennard-Stone sampling (KSS)



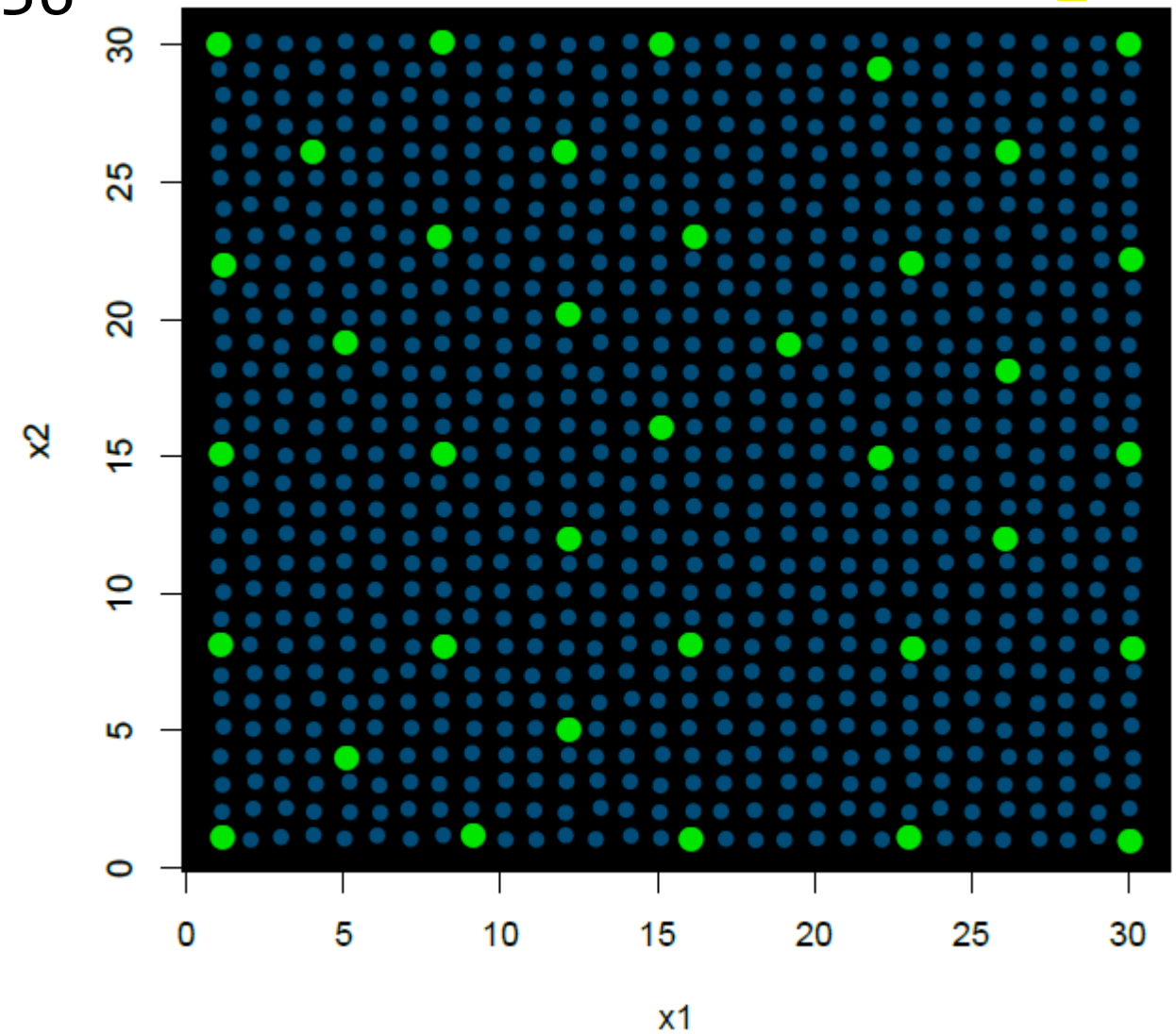
n = 14

Kennard-Stone sampling (KSS)



n = 36

Kennard-Stone sampling (KSS)



If there are outliers KSS will select some of them

K-means sampling (KMS)

The KMS algorithm (Naes, 1987) to select a (calibration) subset of m samples (S_{tr}) from a given set of N samples (S) works as follows:

1. Carry out a k-means clustering on the principal component scores and choose the number of resulting clusters to be equal to the number of desired calibration samples (m),

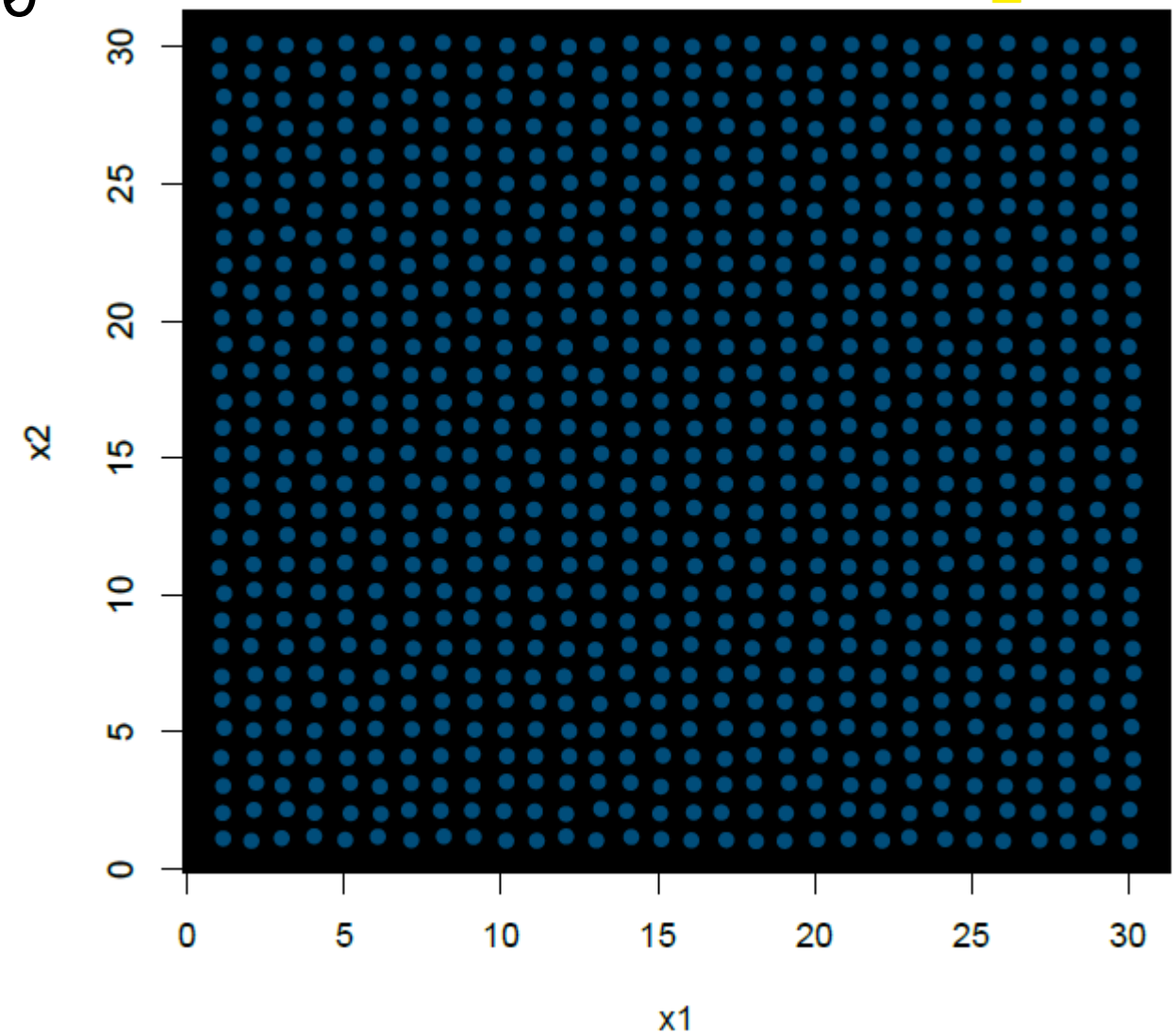
2. Select one sample from each cluster (usually the closest to the center of the cluster).

The **naes** function of the **prospectr** package can be used for KMS

Naes, T., 1987. The design of calibration in near infra-red reflectance analysis by clustering. Journal of Chemometrics 1, 121-134.

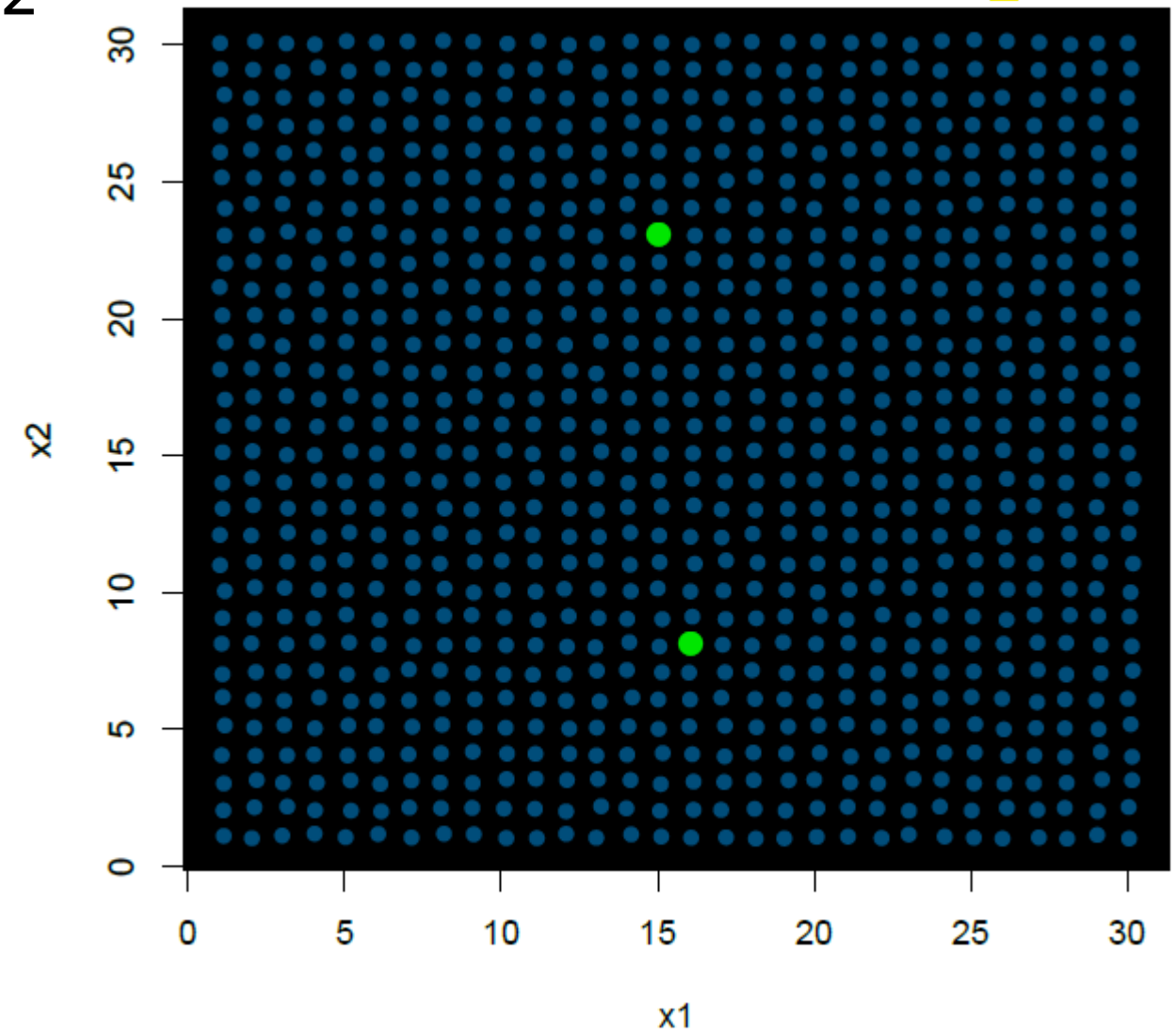
n = 0

K-means sampling (KMS)



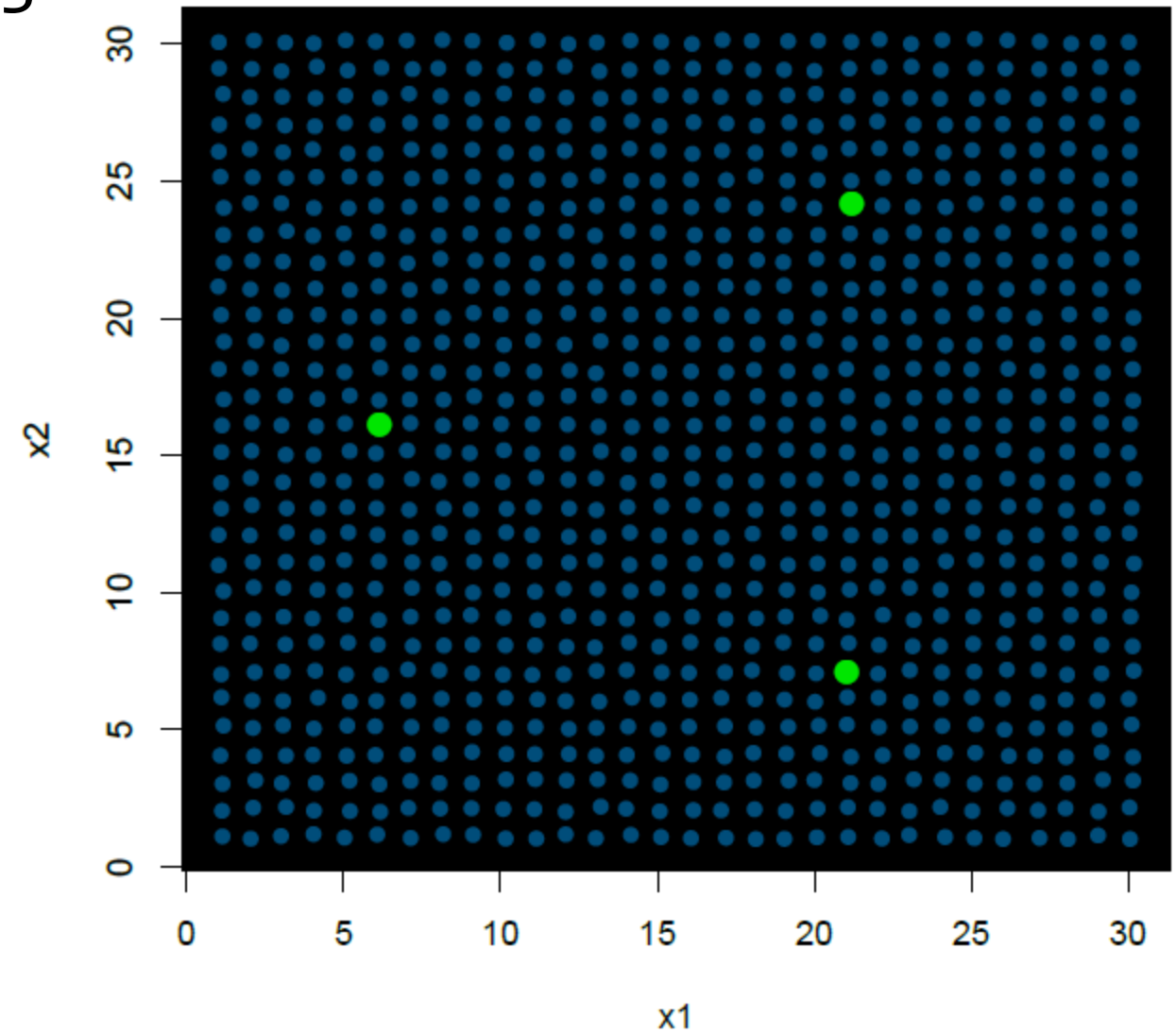
n = 2

K-means sampling (KMS)



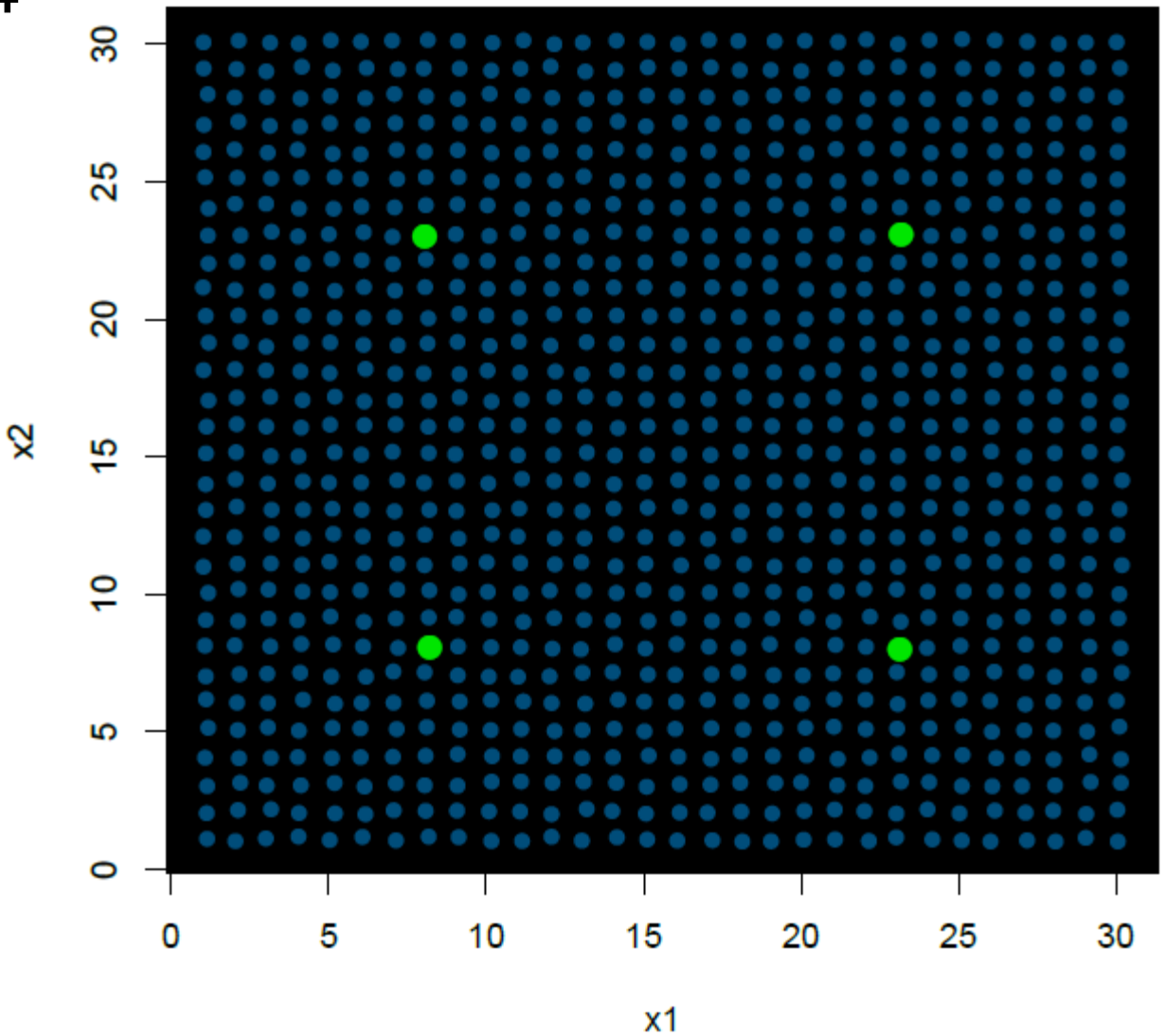
n = 3

K-means sampling (KMS)



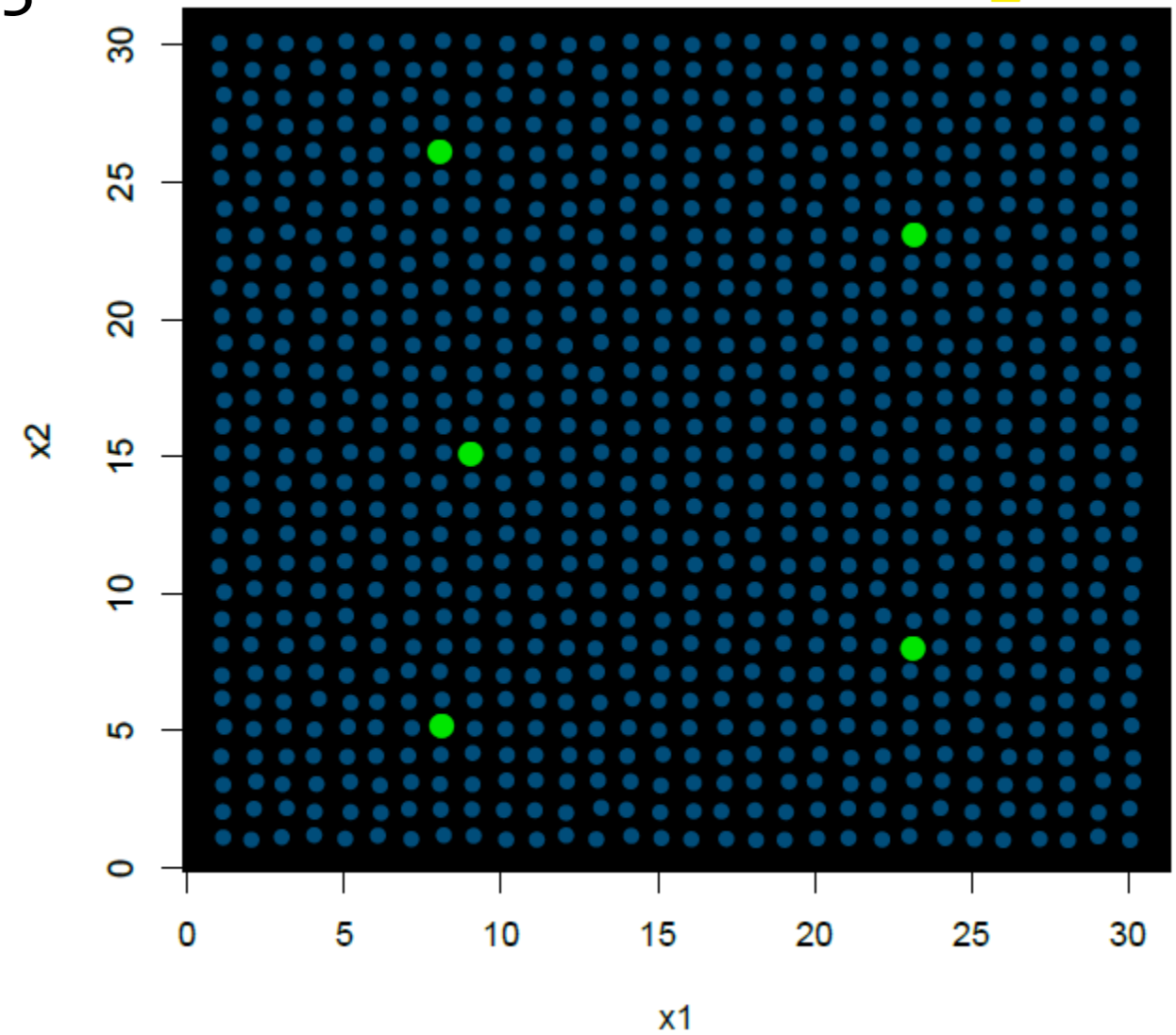
n = 4

K-means sampling (KMS)



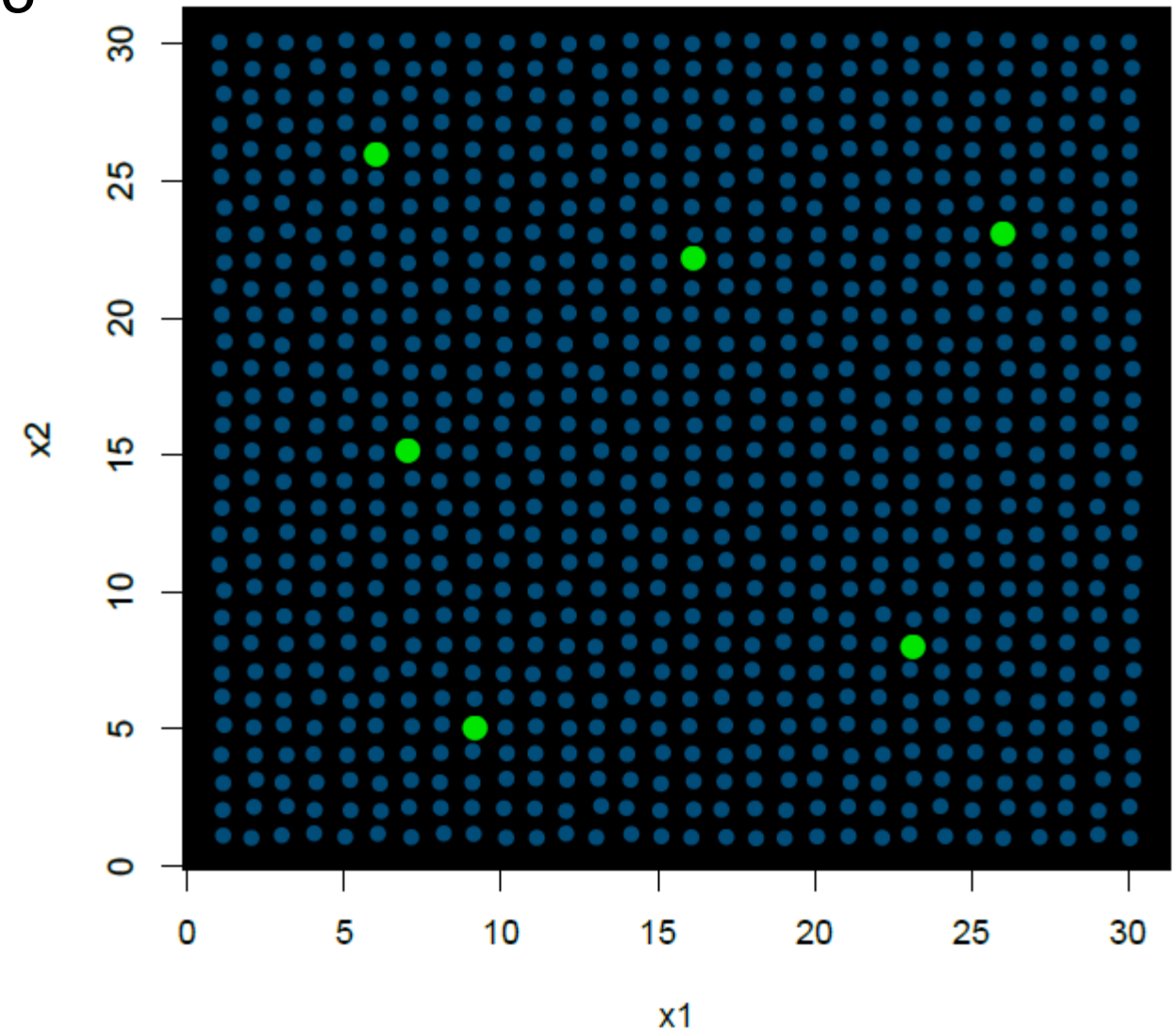
n = 5

K-means sampling (KMS)



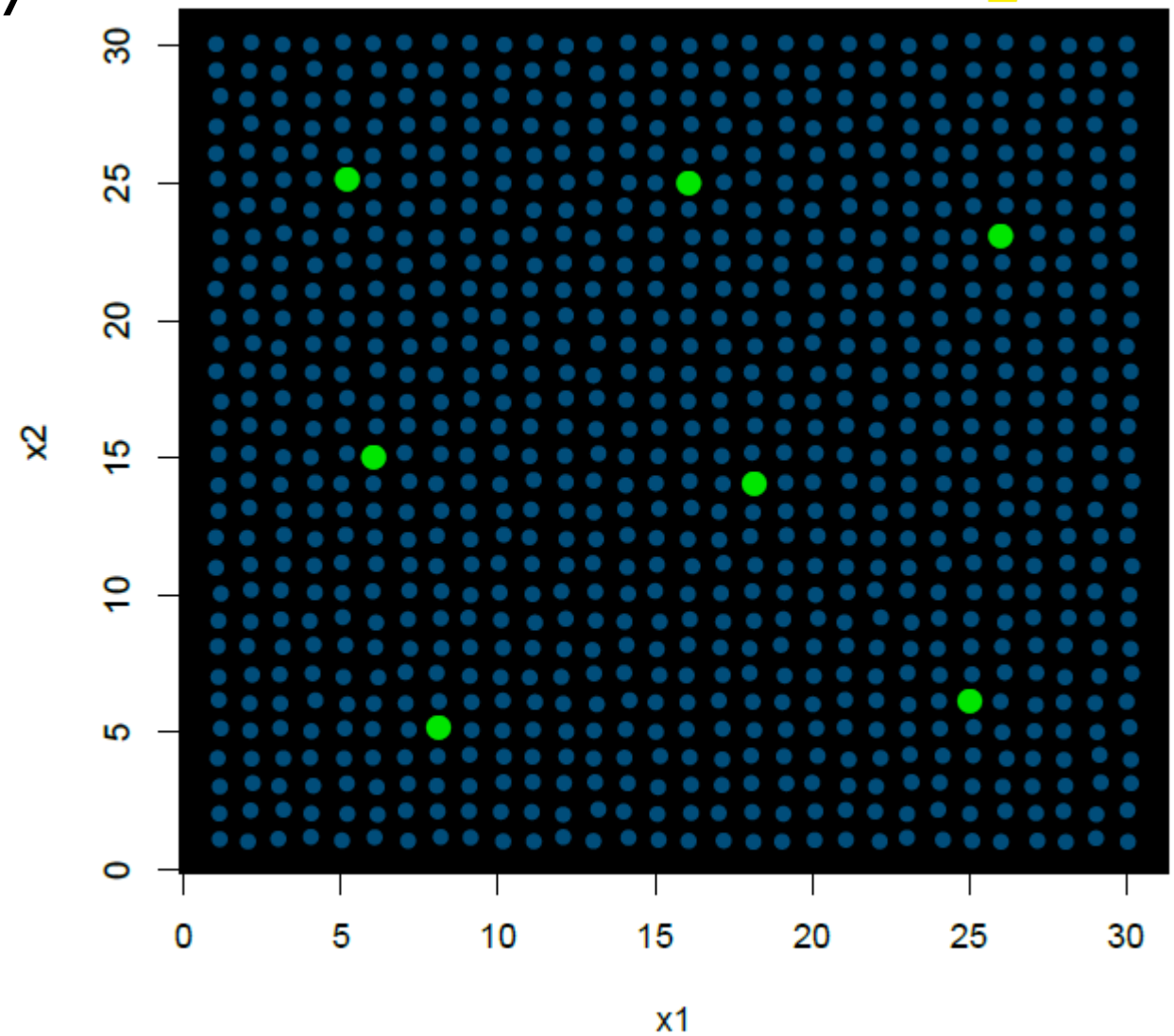
n = 6

K-means sampling (KMS)



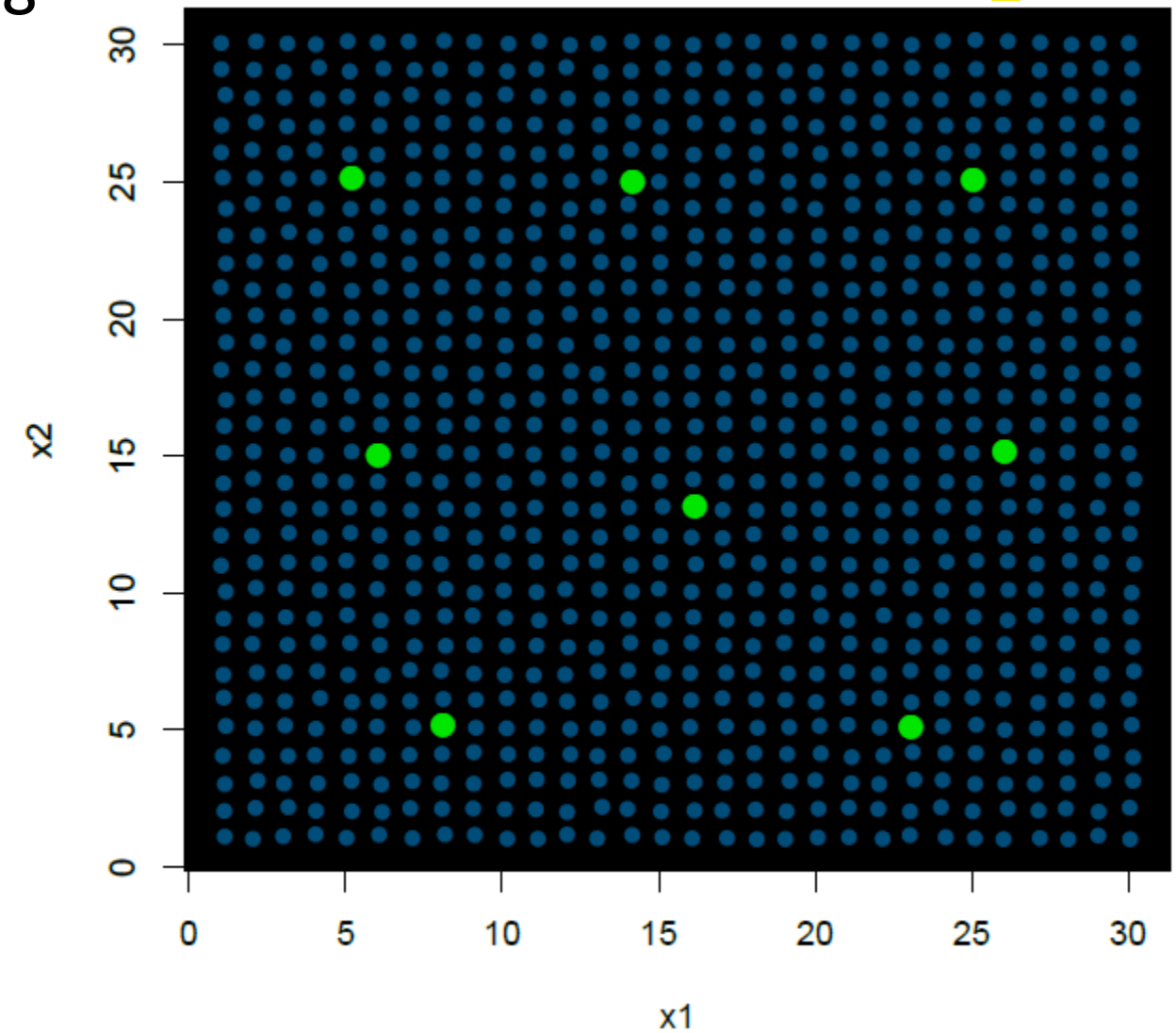
n = 7

K-means sampling (KMS)



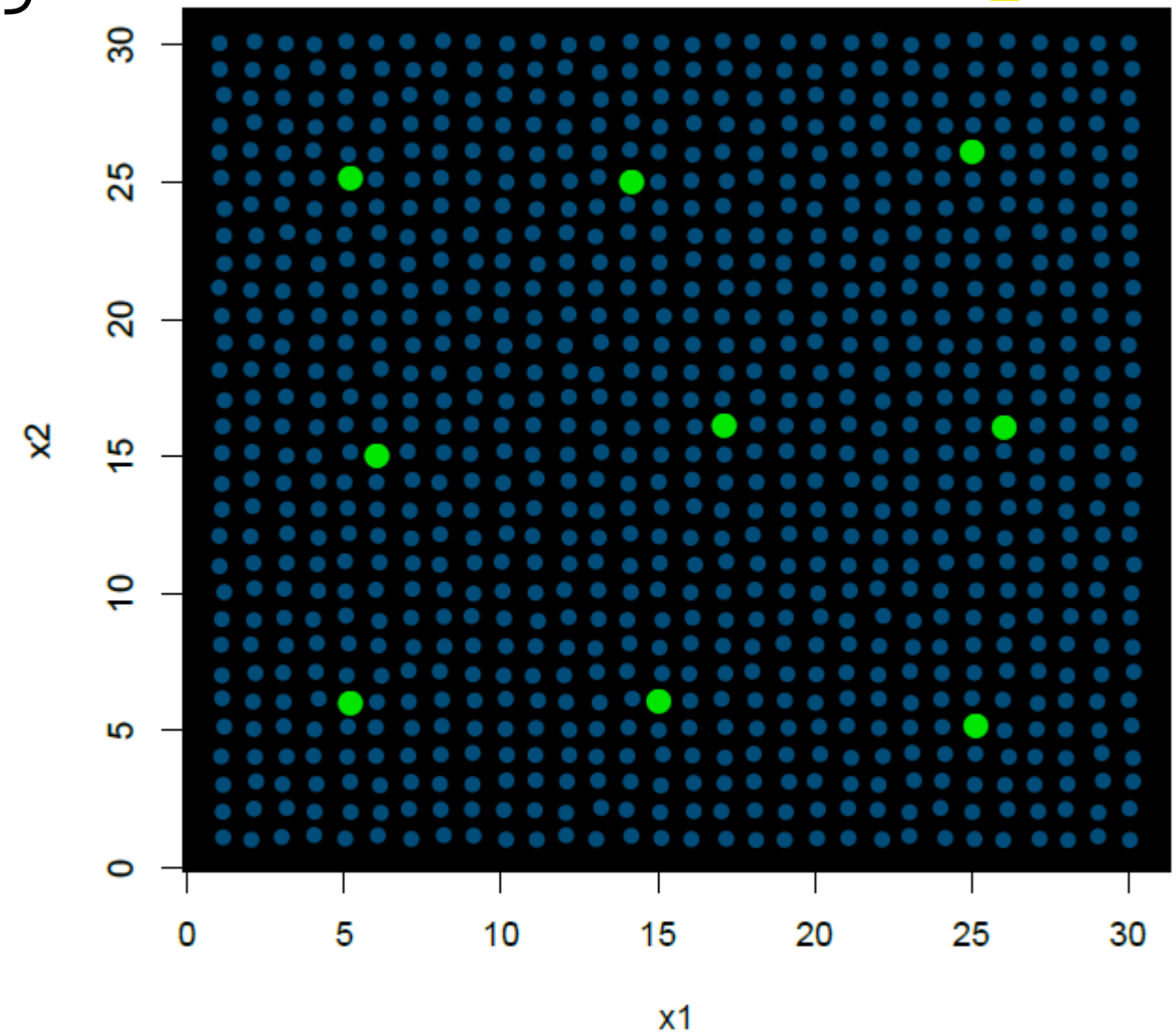
n = 8

K-means sampling (KMS)



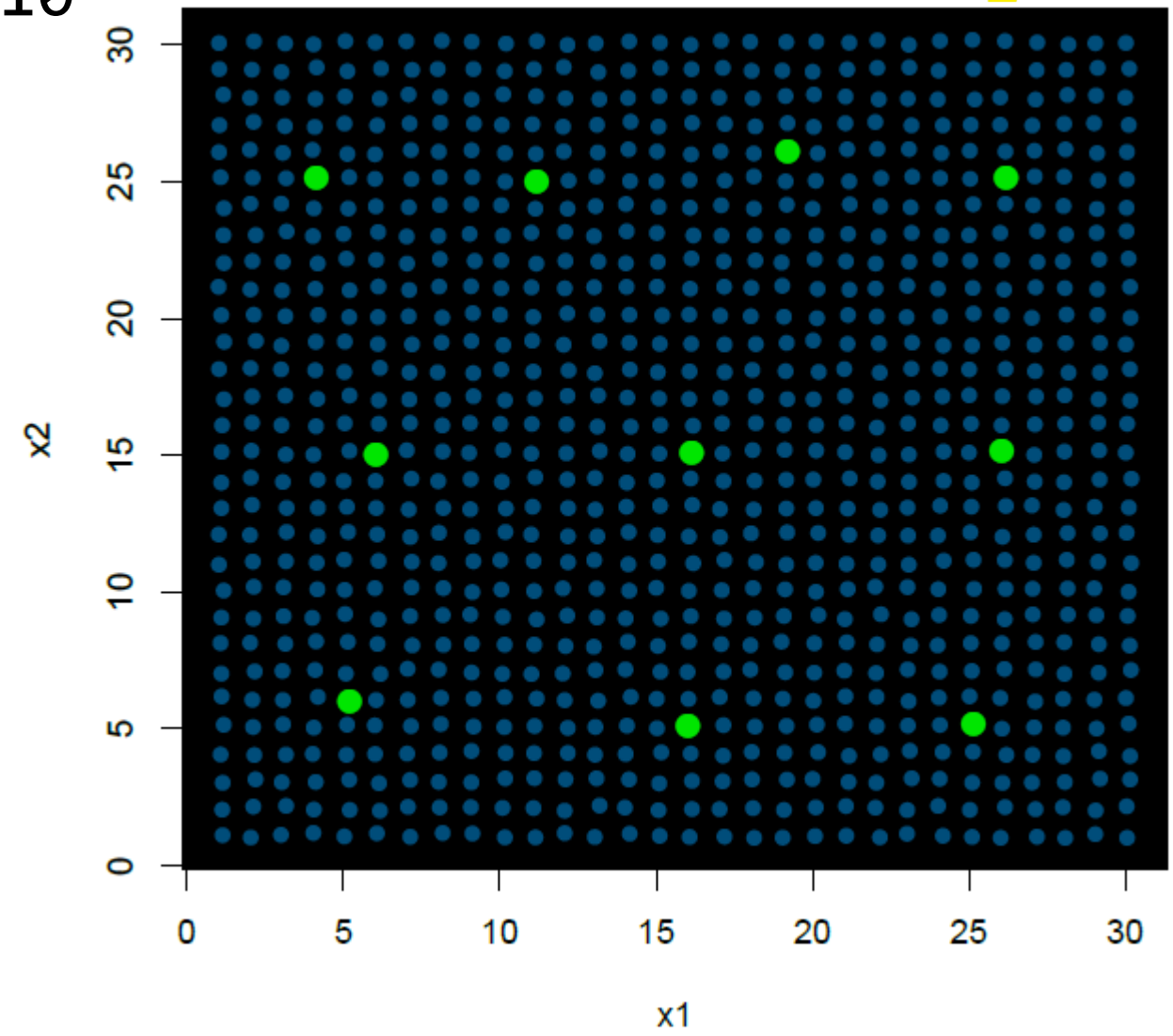
n = 9

K-means sampling (KMS)



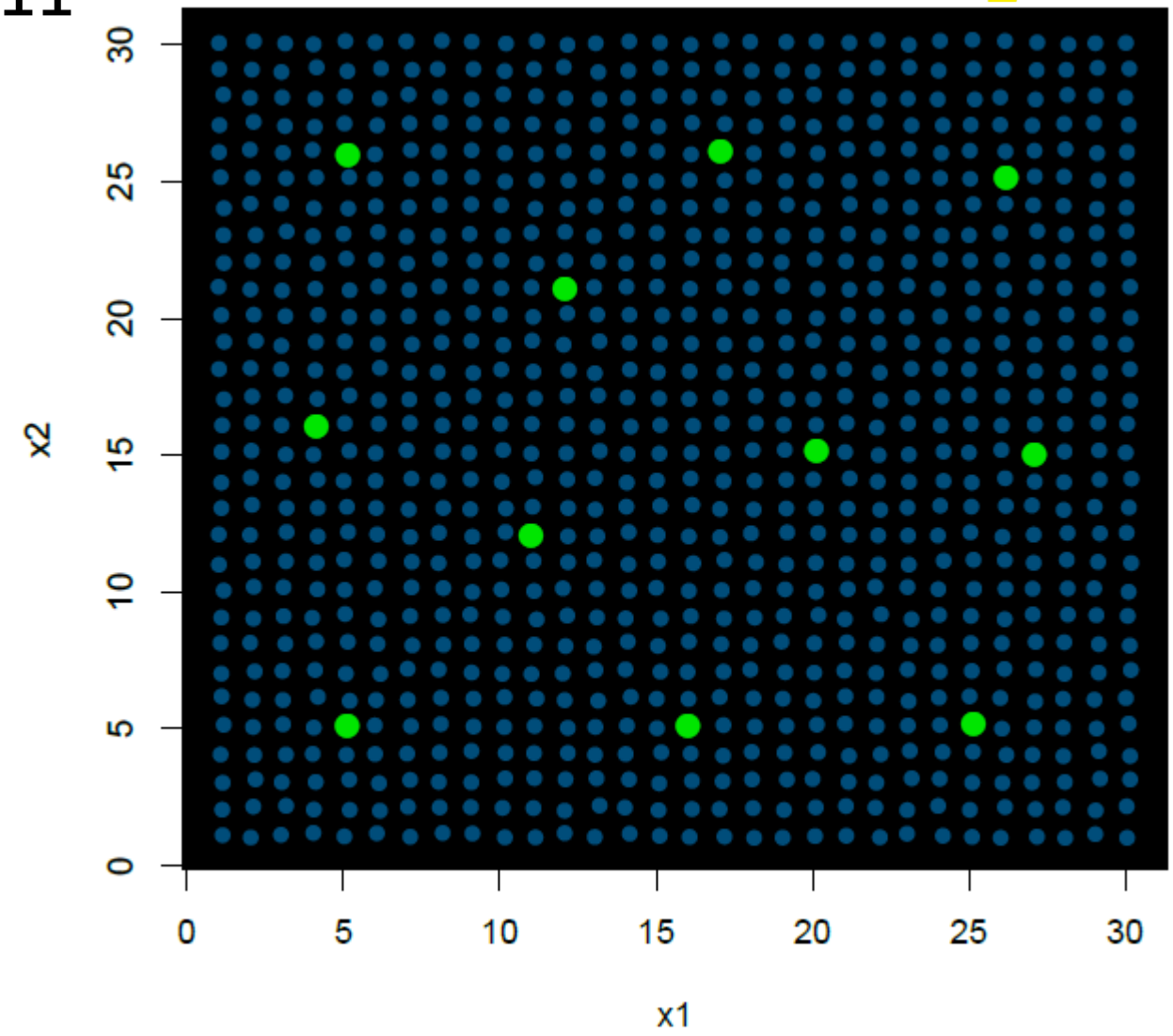
n = 10

K-means sampling (KMS)



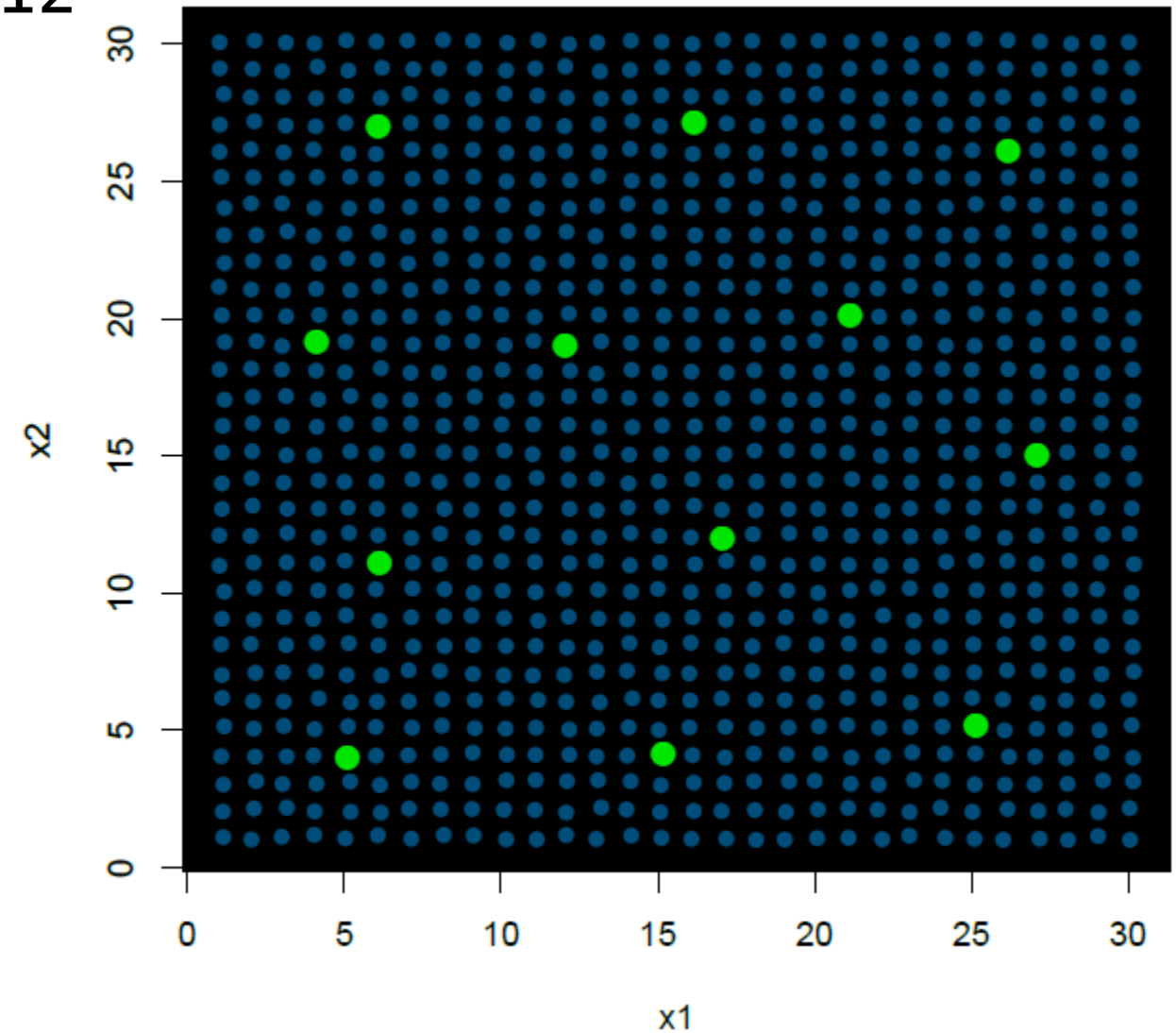
n = 11

K-means sampling (KMS)



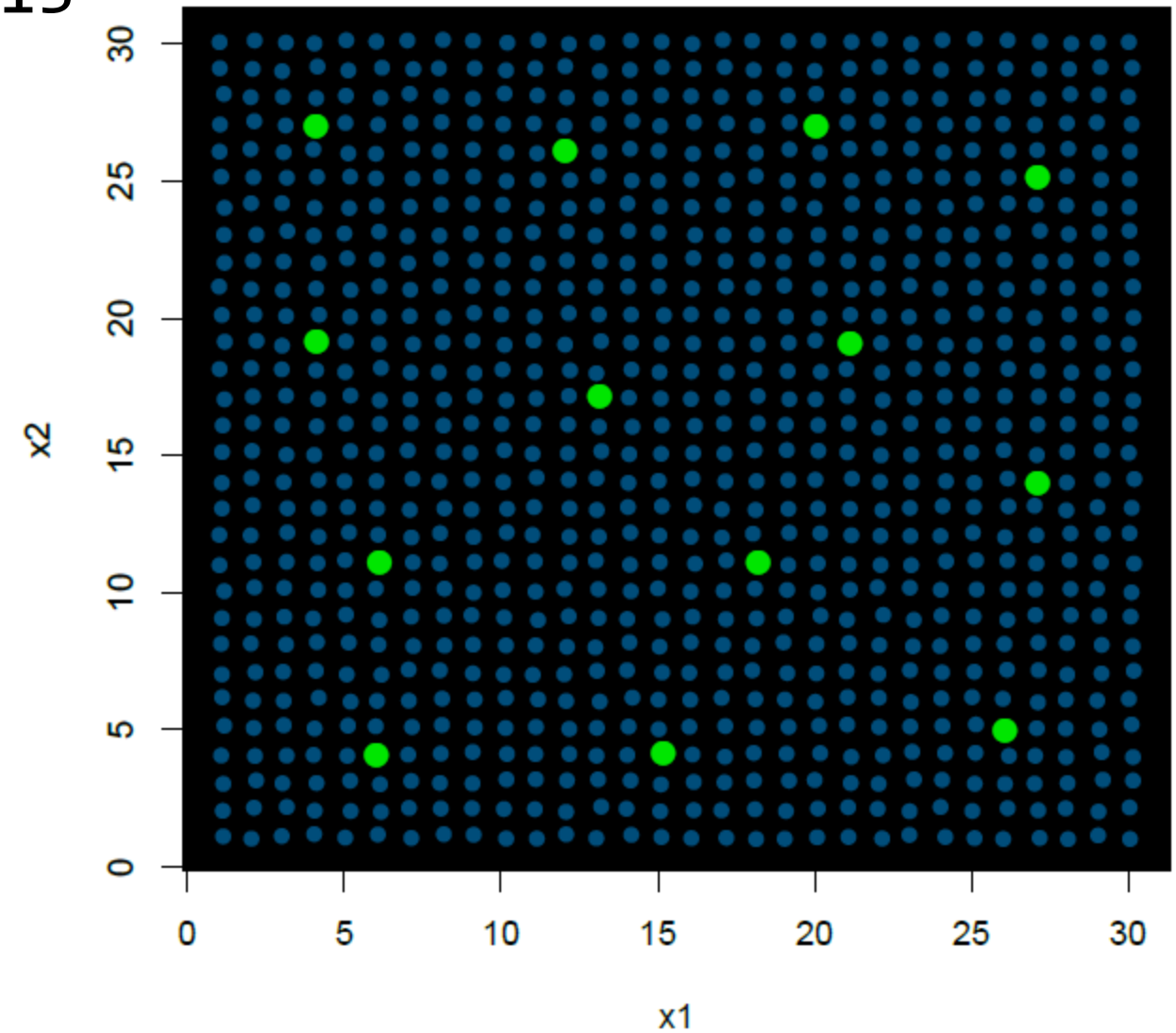
n = 12

K-means sampling (KMS)



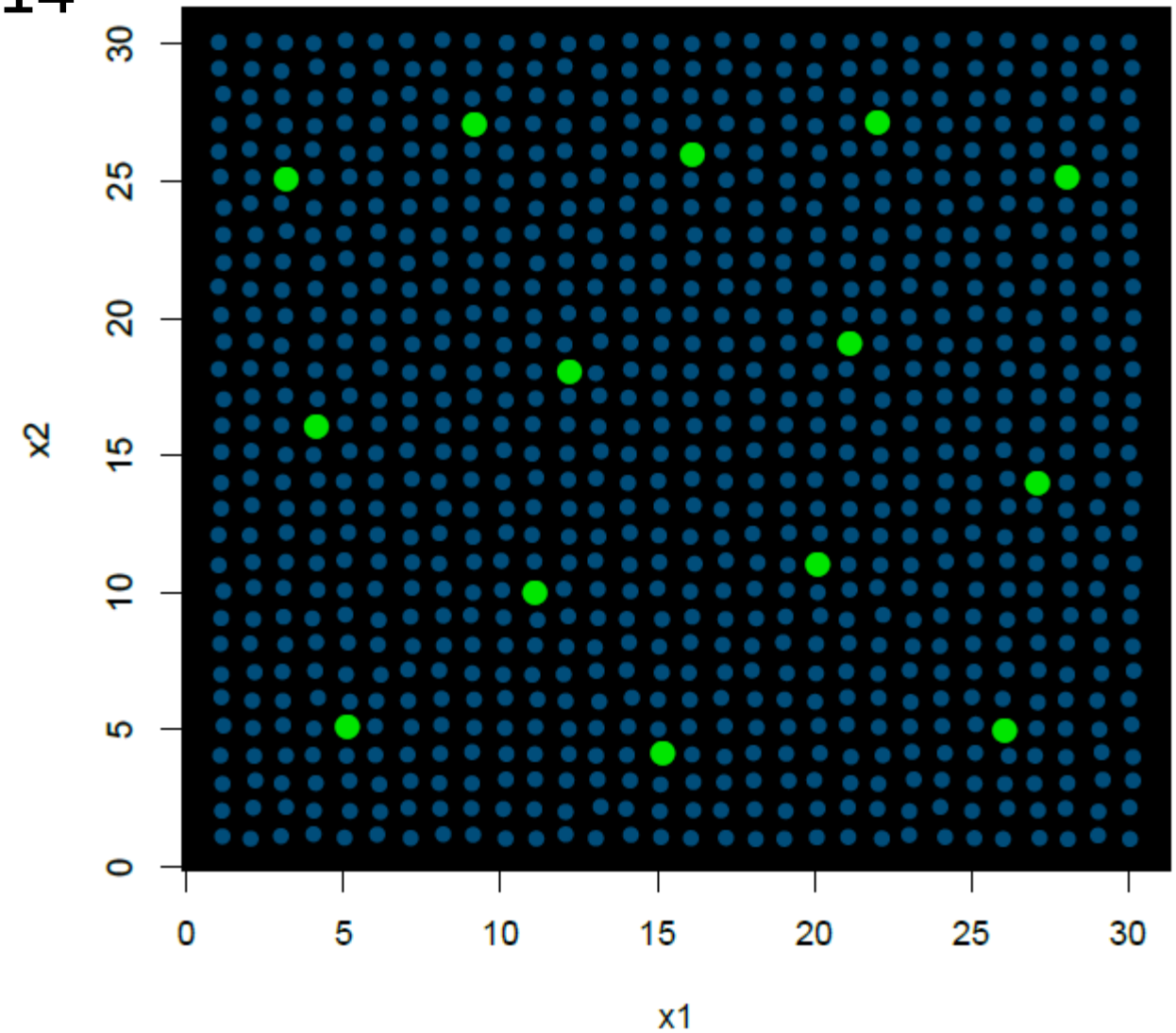
n = 13

K-means sampling (KMS)



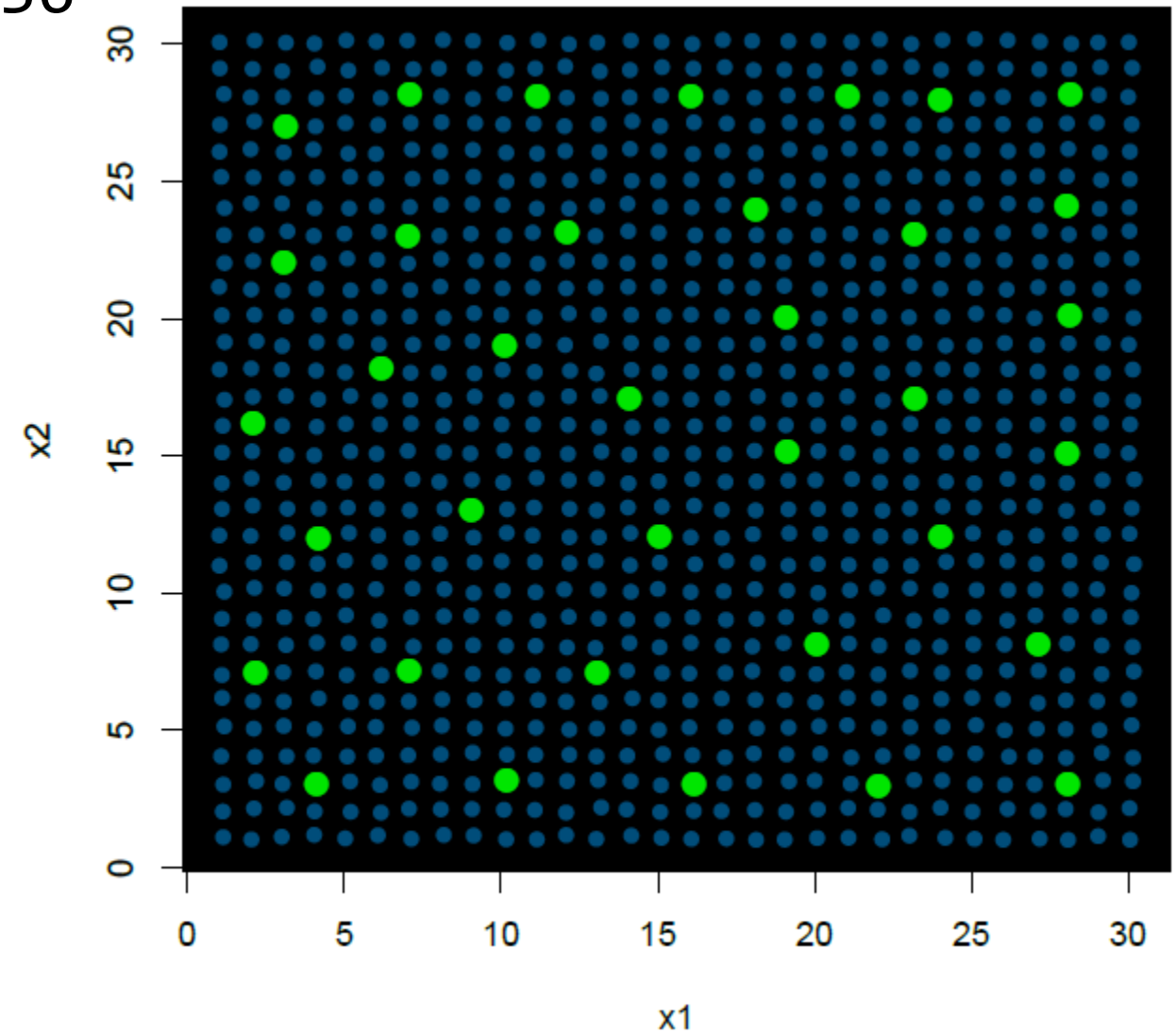
n = 14

K-means sampling (KMS)

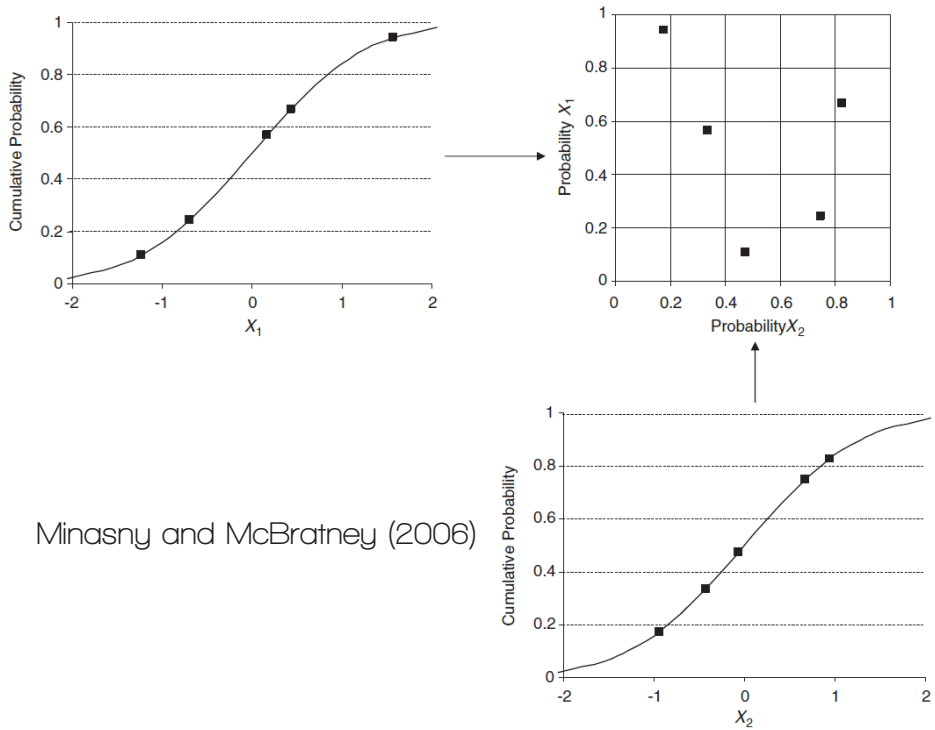


n = 36

K-means sampling (KMS)



Conditioned Latin hypercube sampling (cLHS)



Minasny and McBratney (2006)

The cLHS attempts to cover the multidimensional distribution corresponding to a set of variables by using a stratified random sampling.

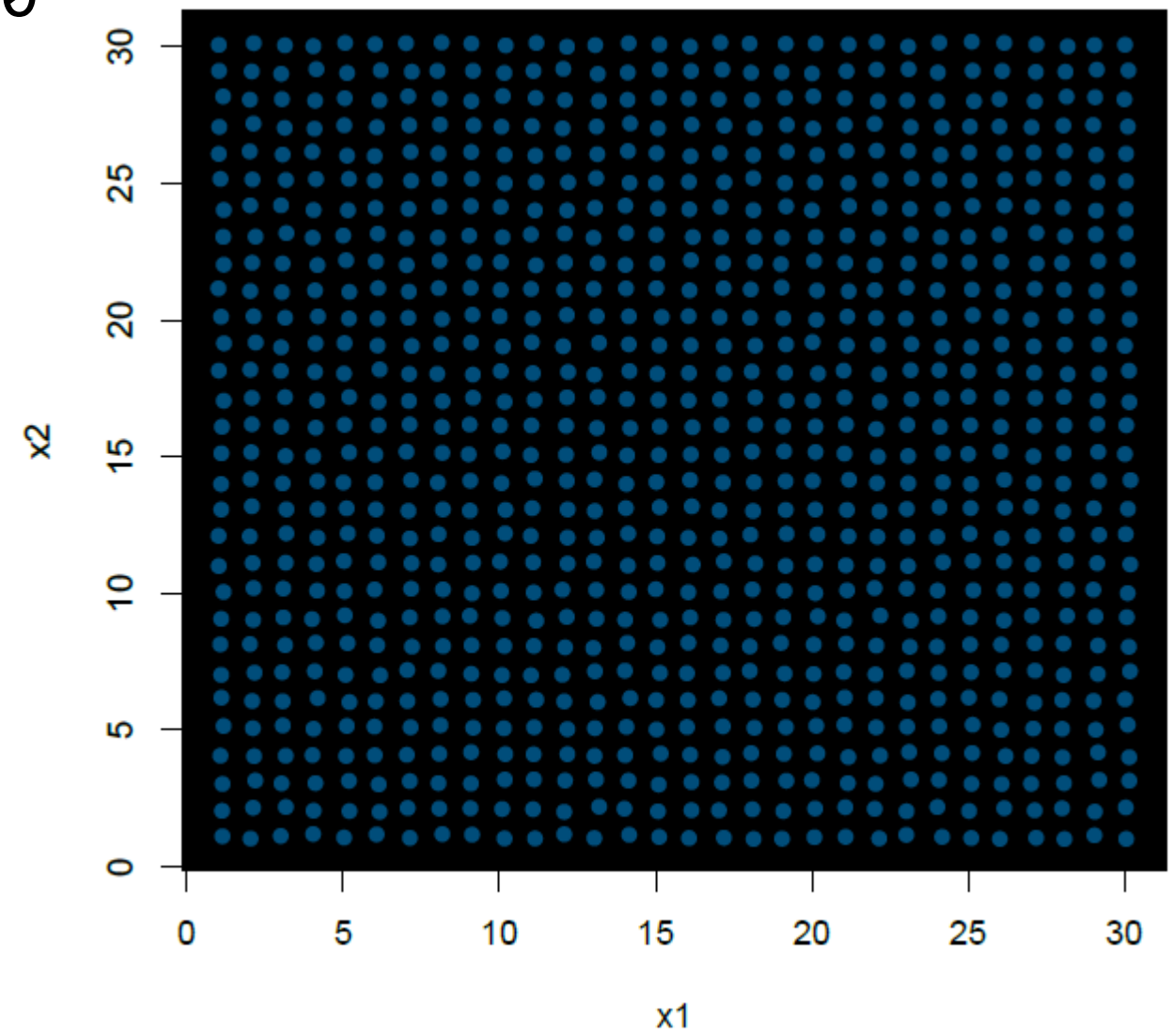
The **clhs** function of the **clhs** package can be used for cLHS

cLHS does not maximize the dissimilarity between samples

Minasny, B., & McBratney, A. B. (2006). A conditioned Latin hypercube method for sampling in the presence of ancillary information. Computers & Geosciences, 32(9), 1378-1388.

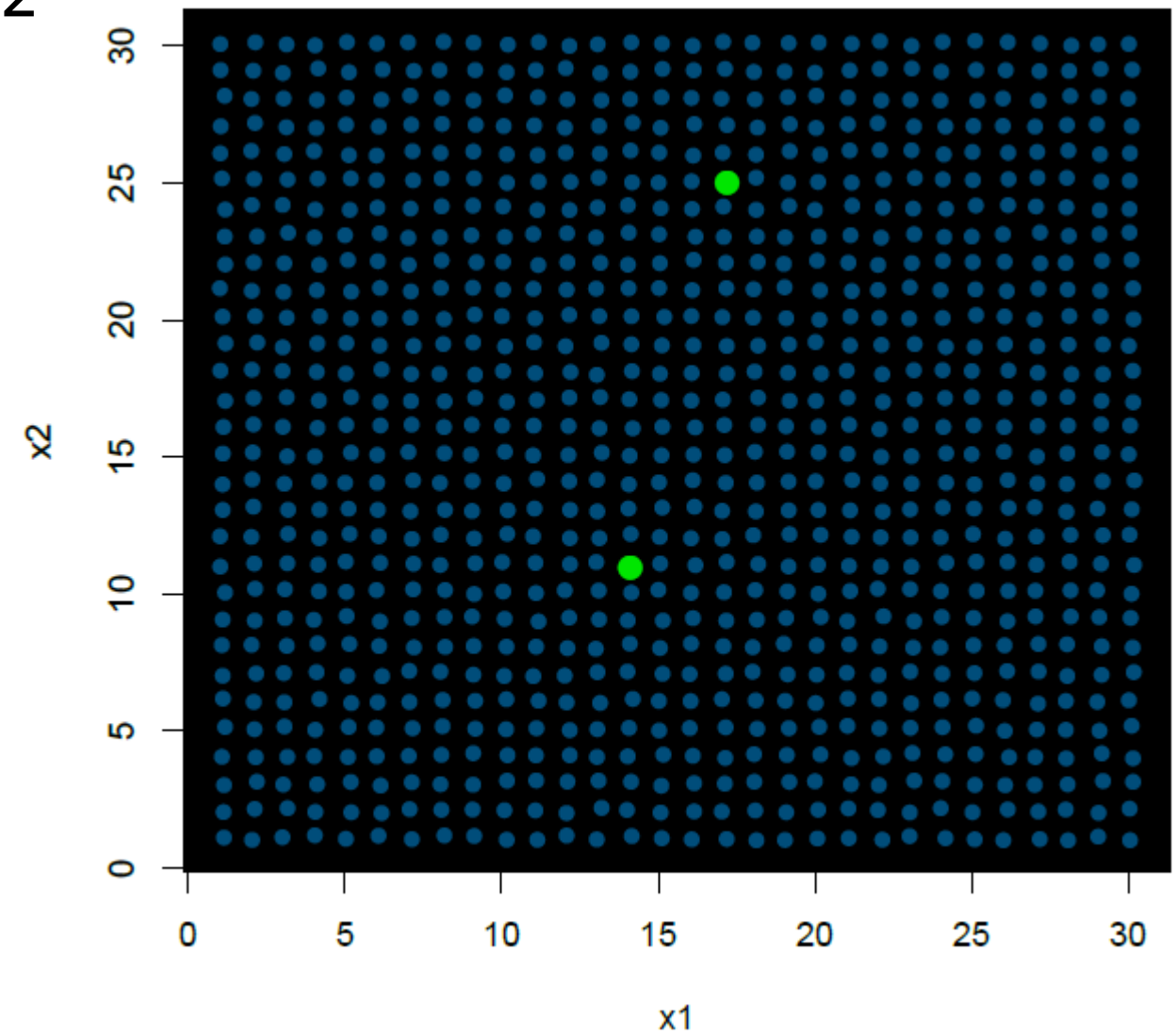
n = 0

contioned Latin Hypercube sampling (cLHS)



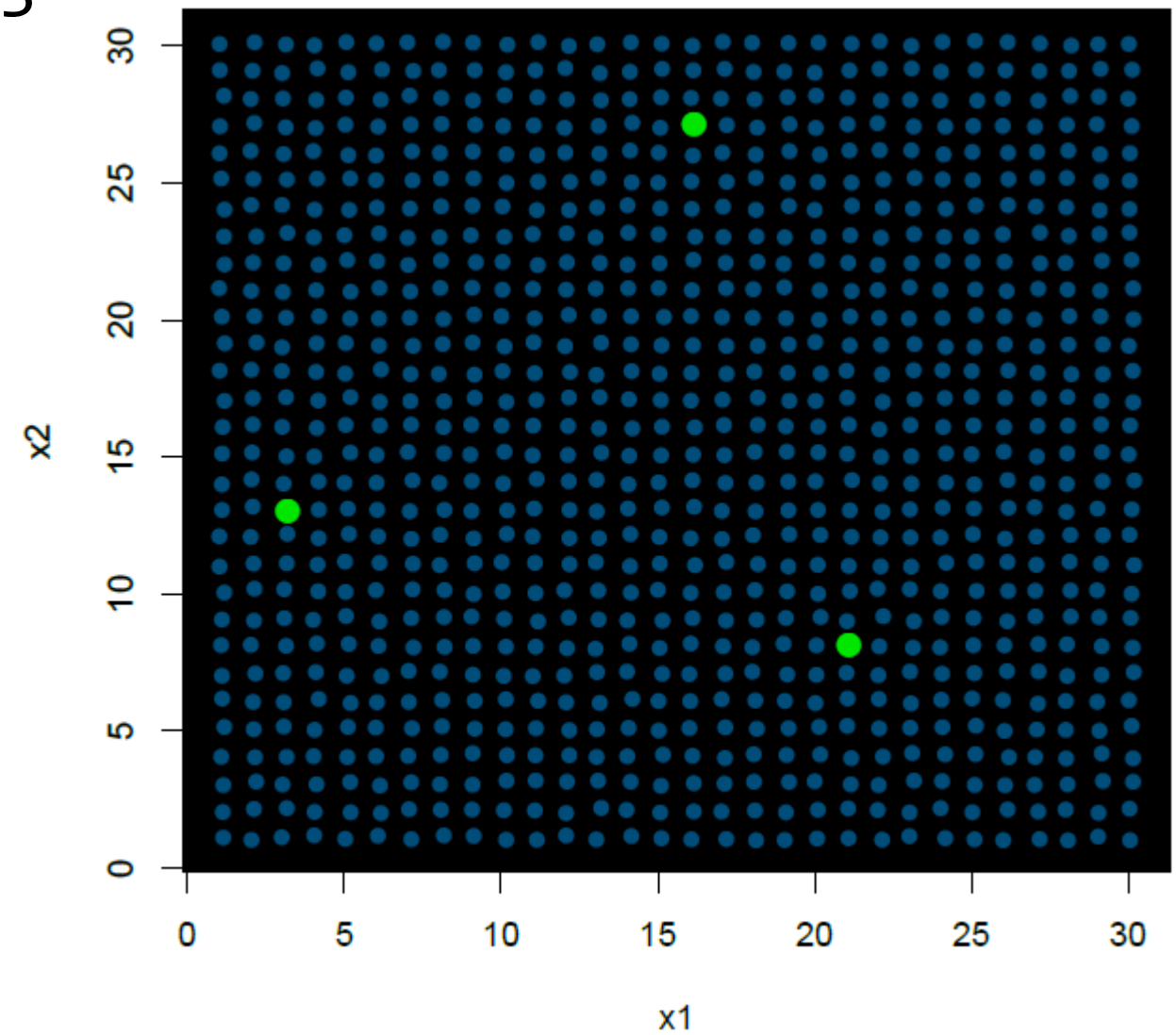
n = 2

contioned Latin Hypercube sampling (cLHS)



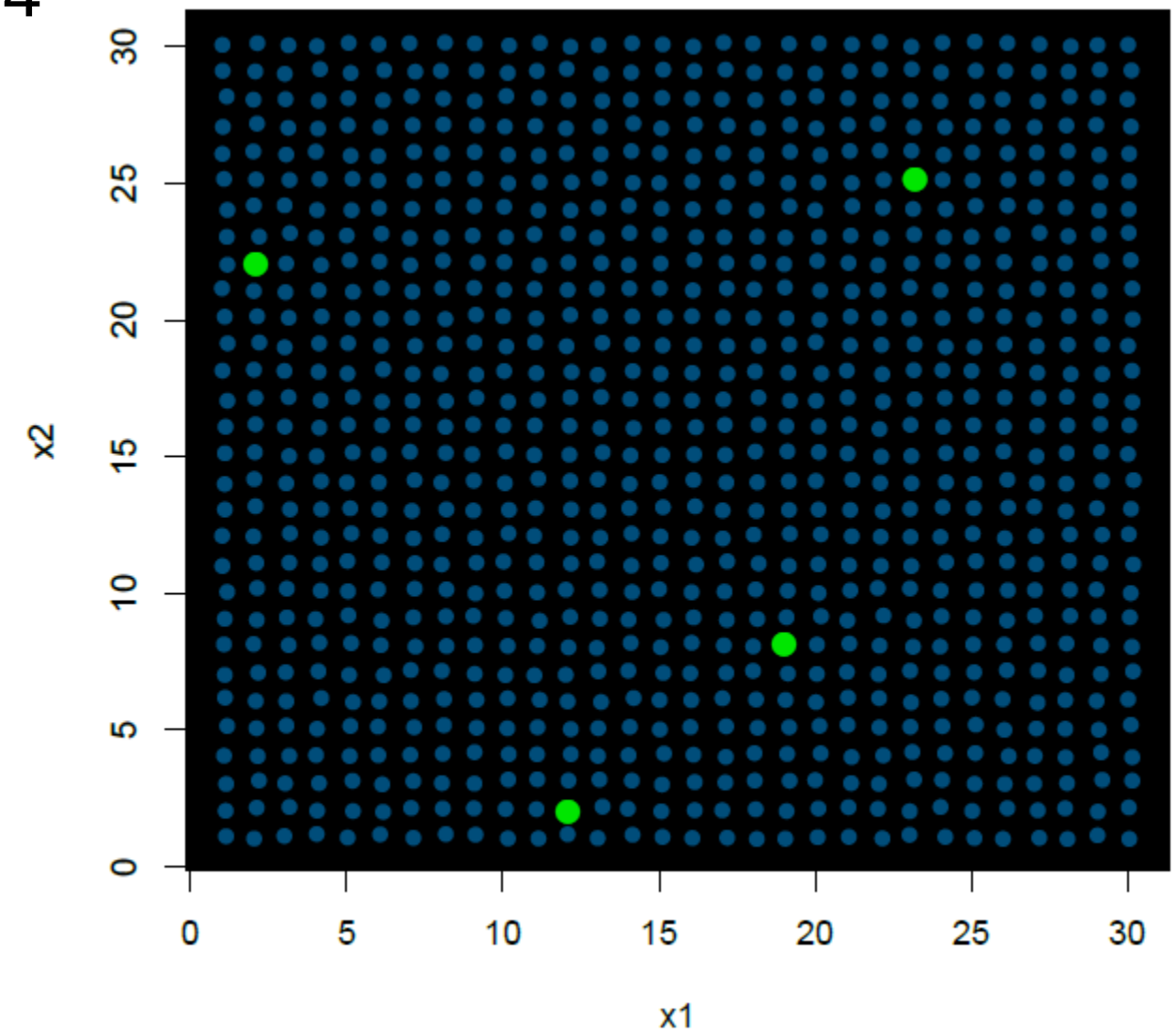
n = 3

contioned Latin Hypercube sampling (cLHS)



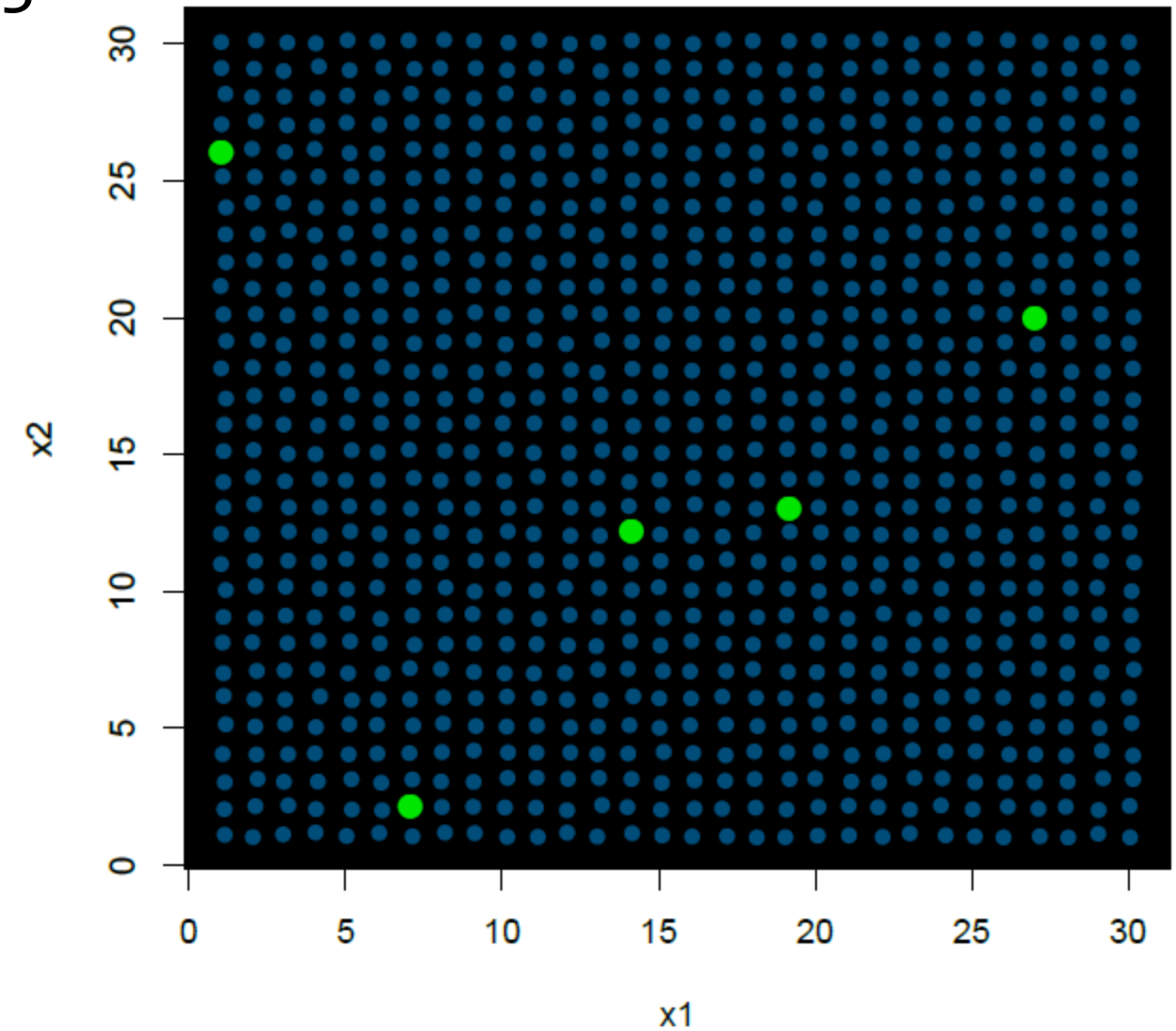
n = 4

contioned Latin Hypercube sampling (cLHS)



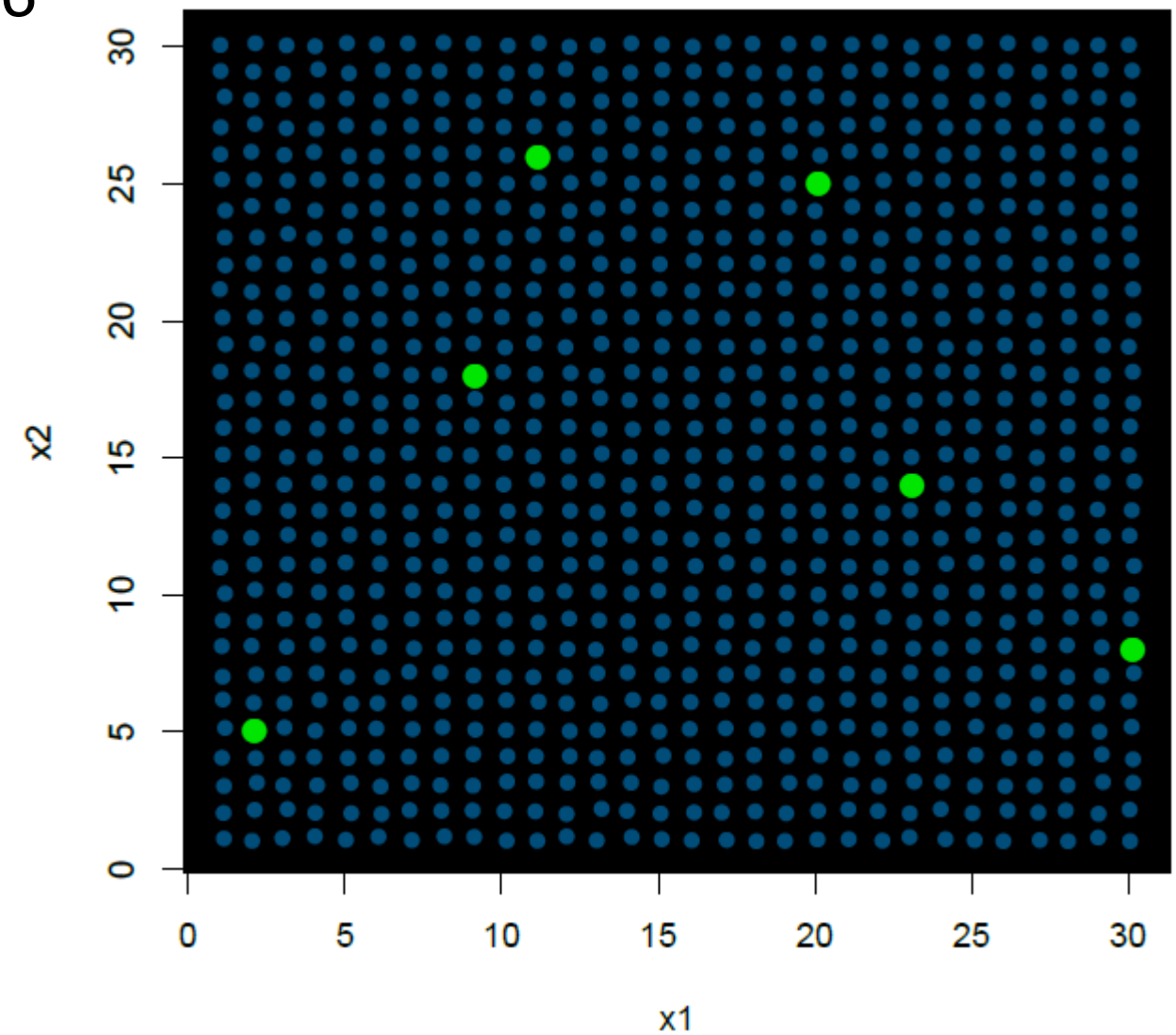
n = 5

contioned Latin Hypercube sampling (cLHS)



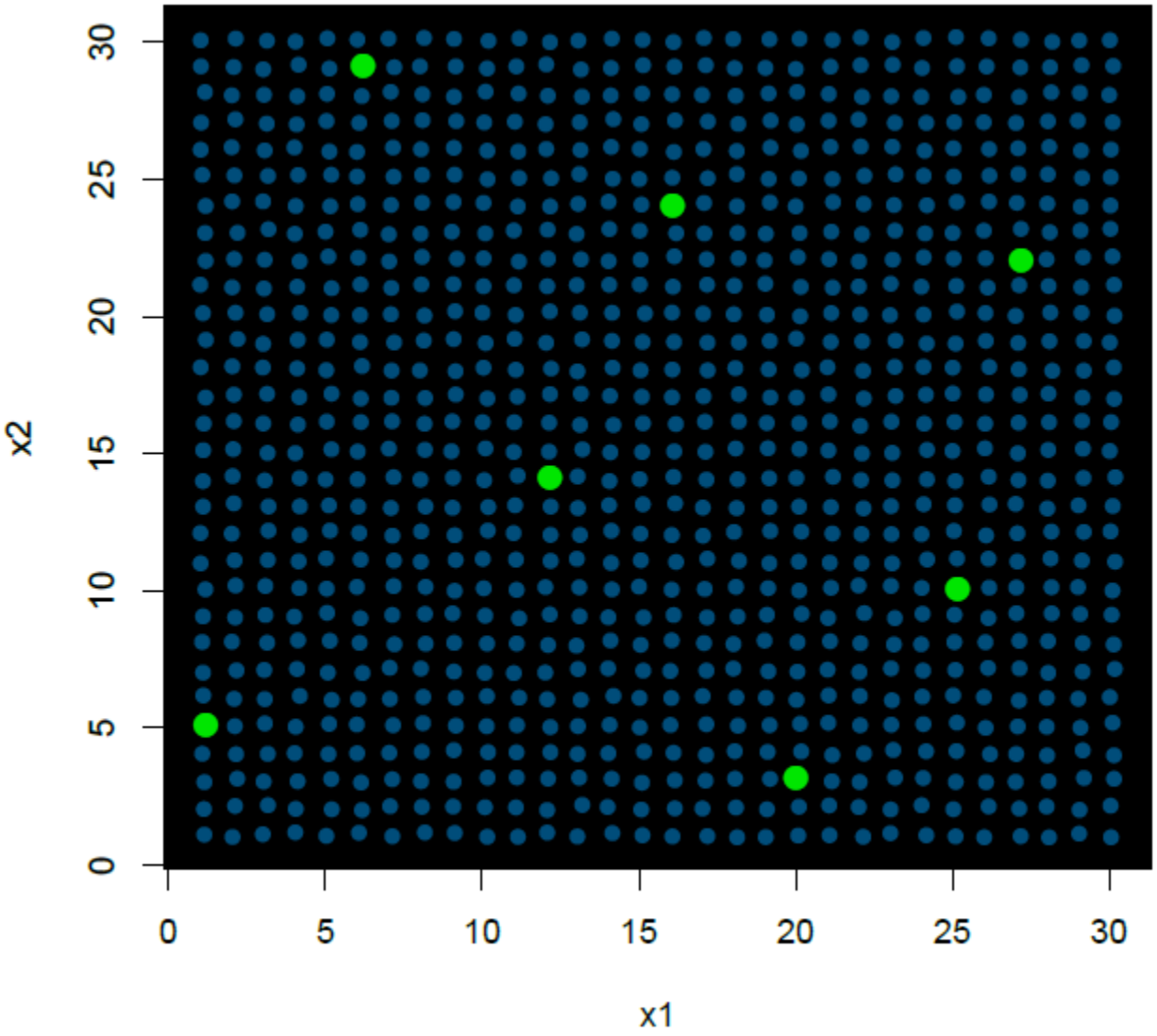
n = 6

contioned Latin Hypercube sampling (cLHS)



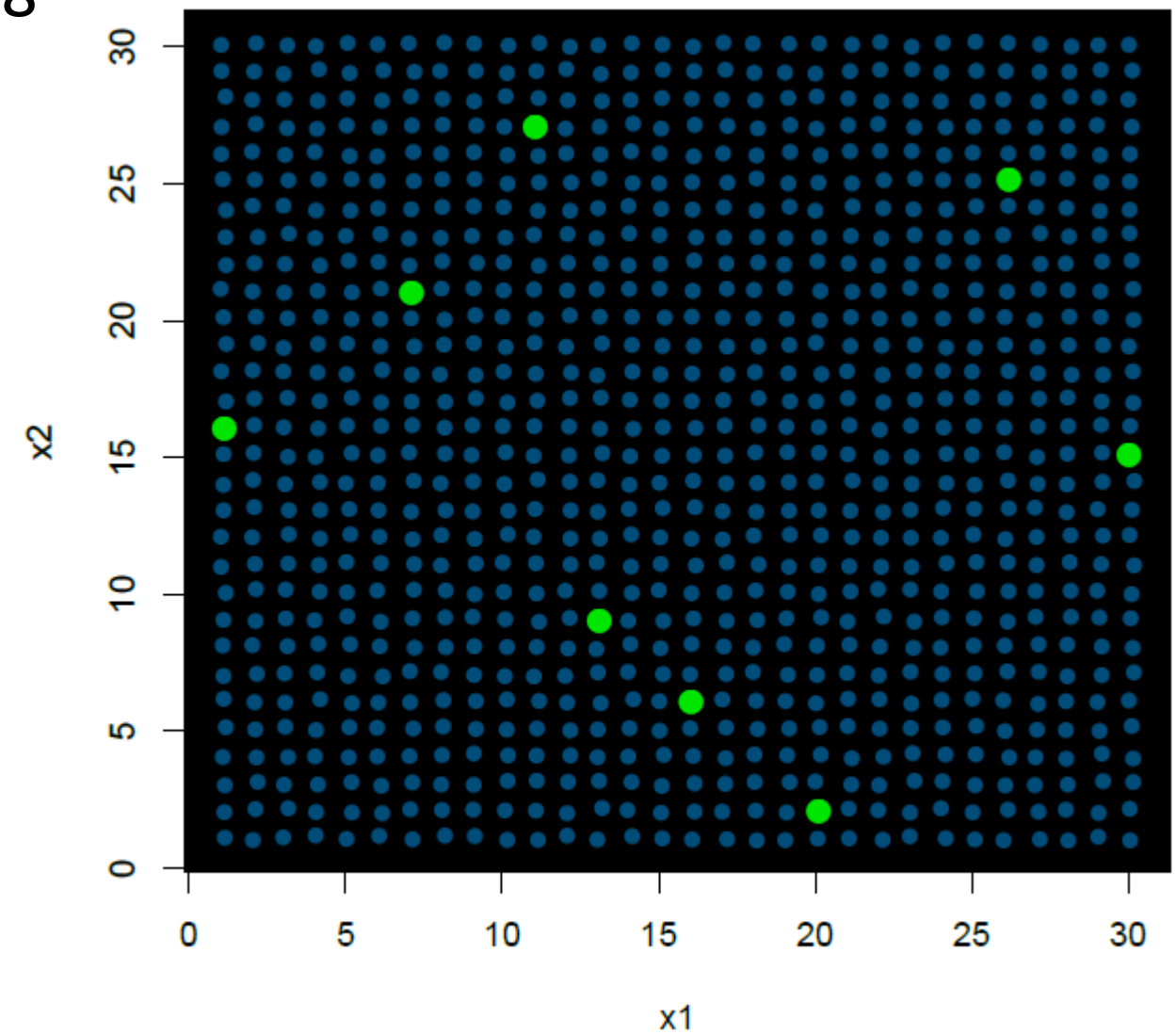
n = 7

contioned Latin Hypercube sampling (cLHS)



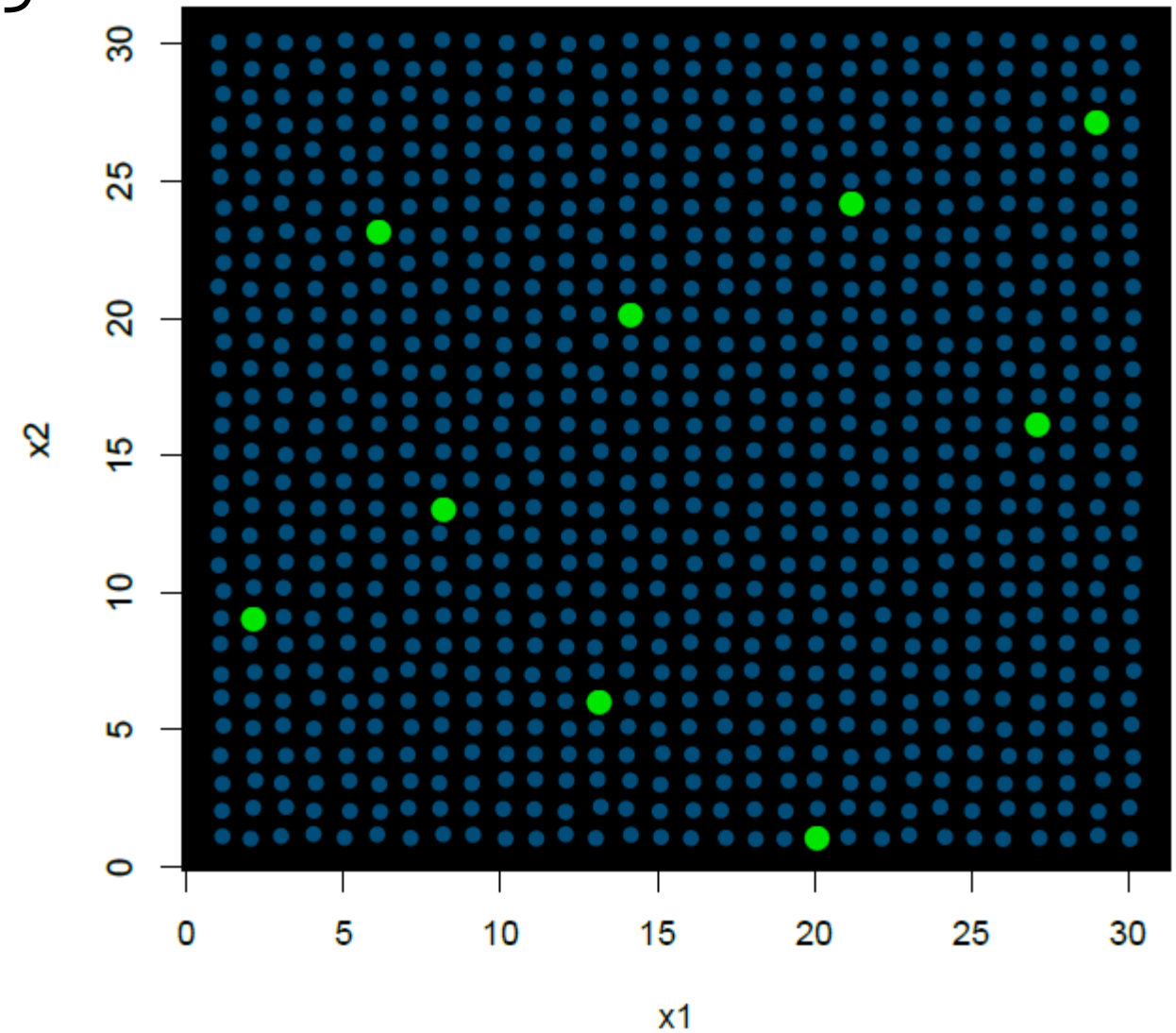
n = 8

contioned Latin Hypercube sampling (cLHS)



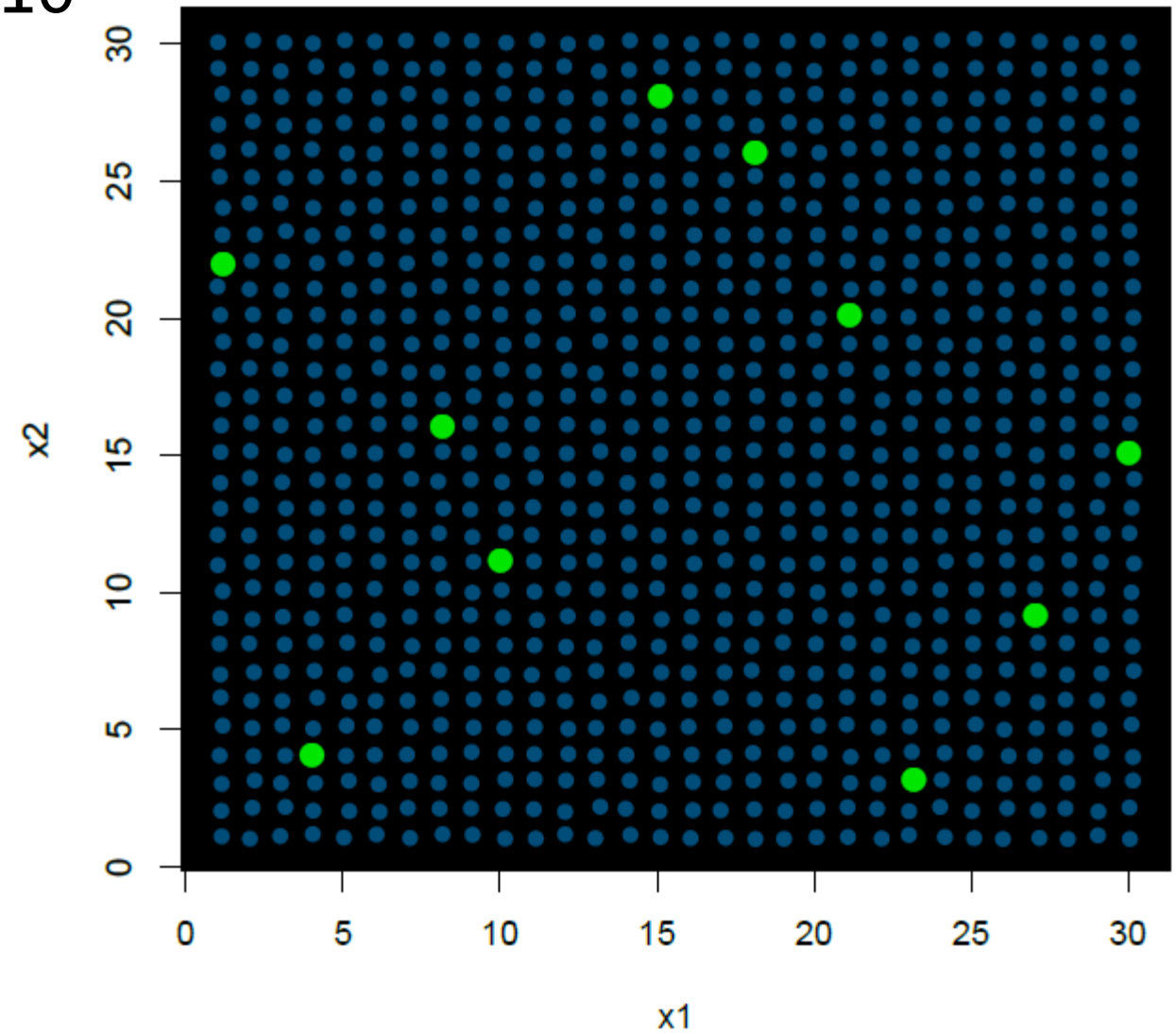
n = 9

contioned Latin Hypercube sampling (cLHS)



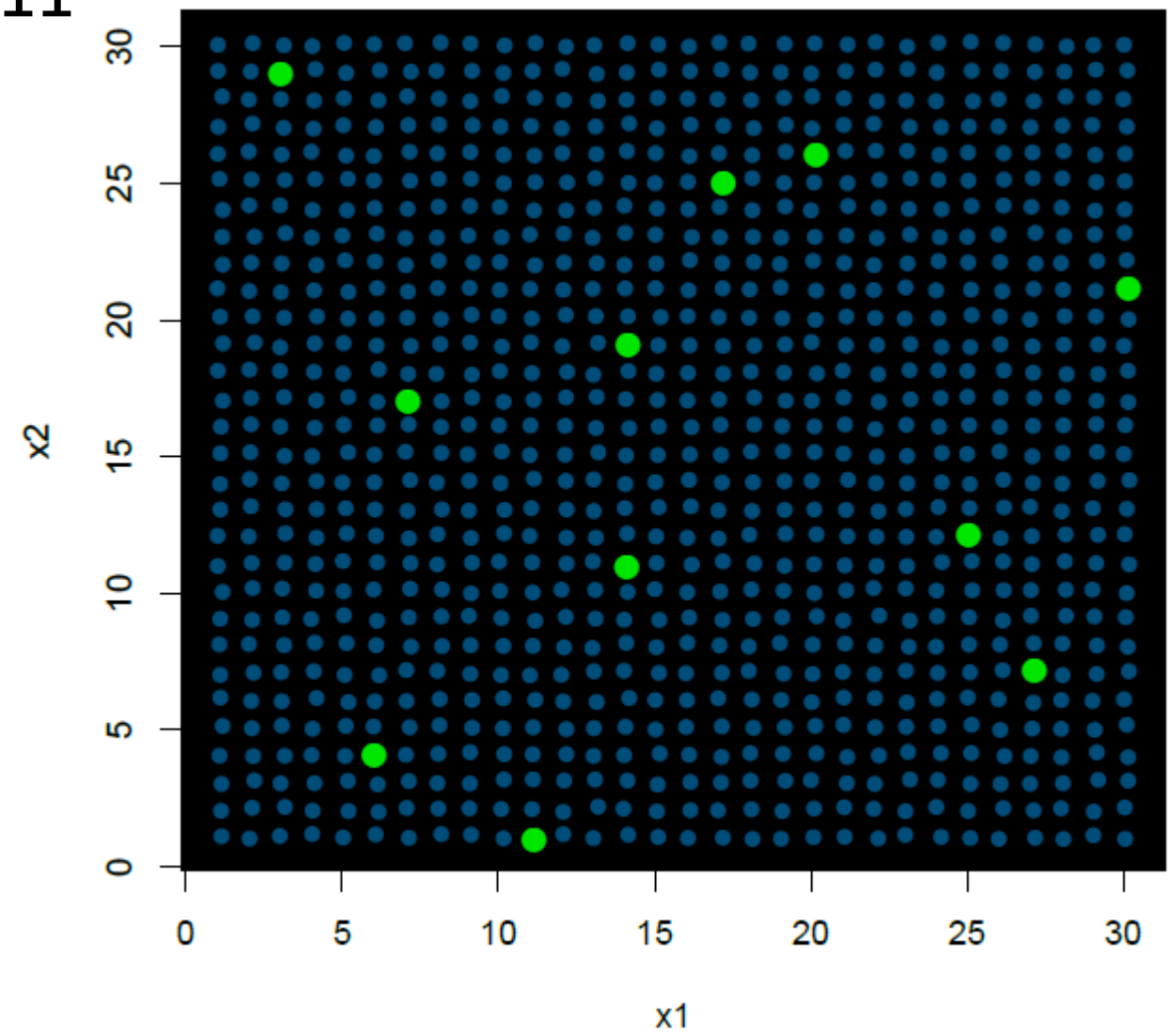
n = 10

contioned Latin Hypercube sampling (cLHS)



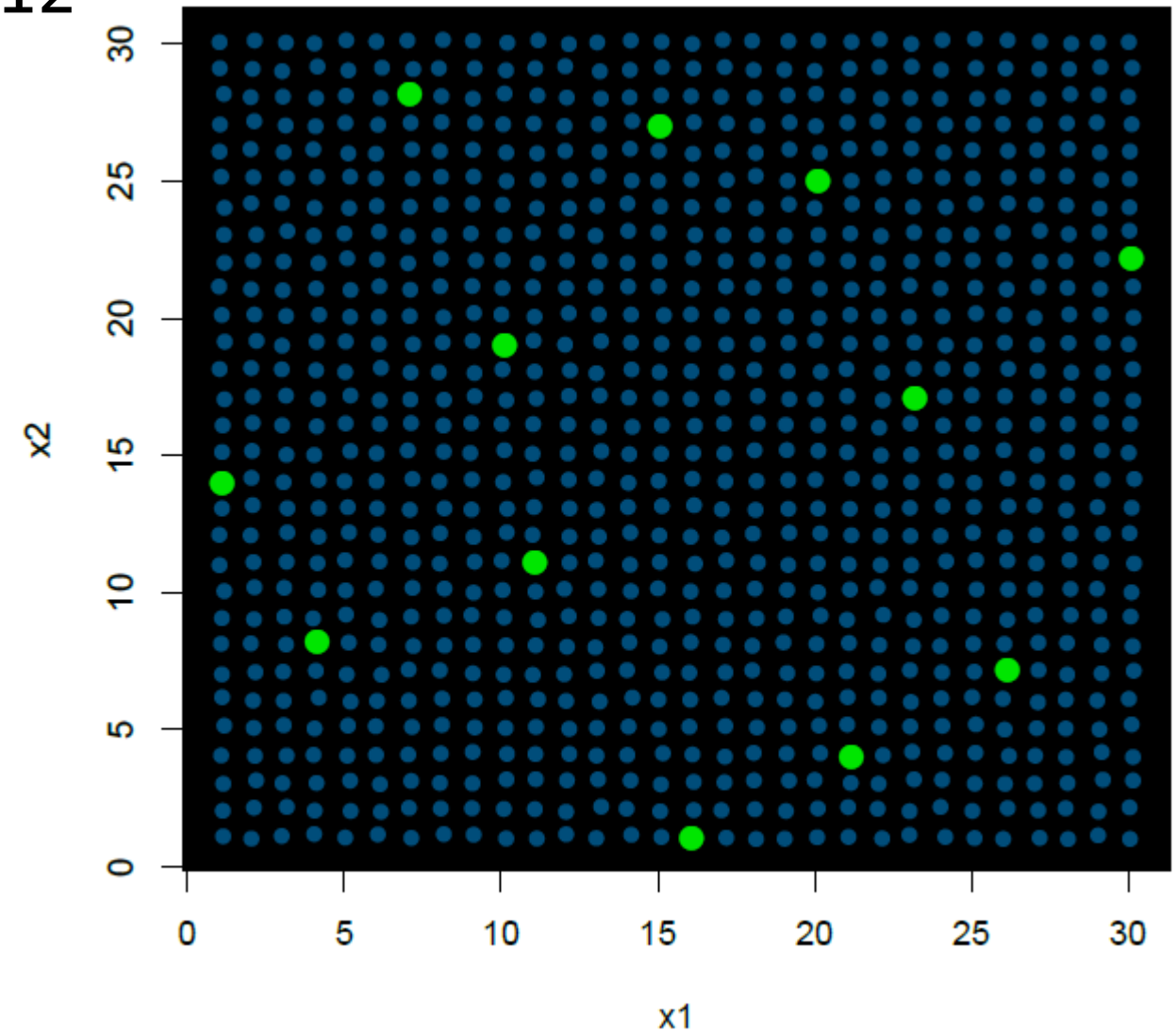
n = 11

contioned Latin Hypercube sampling (cLHS)



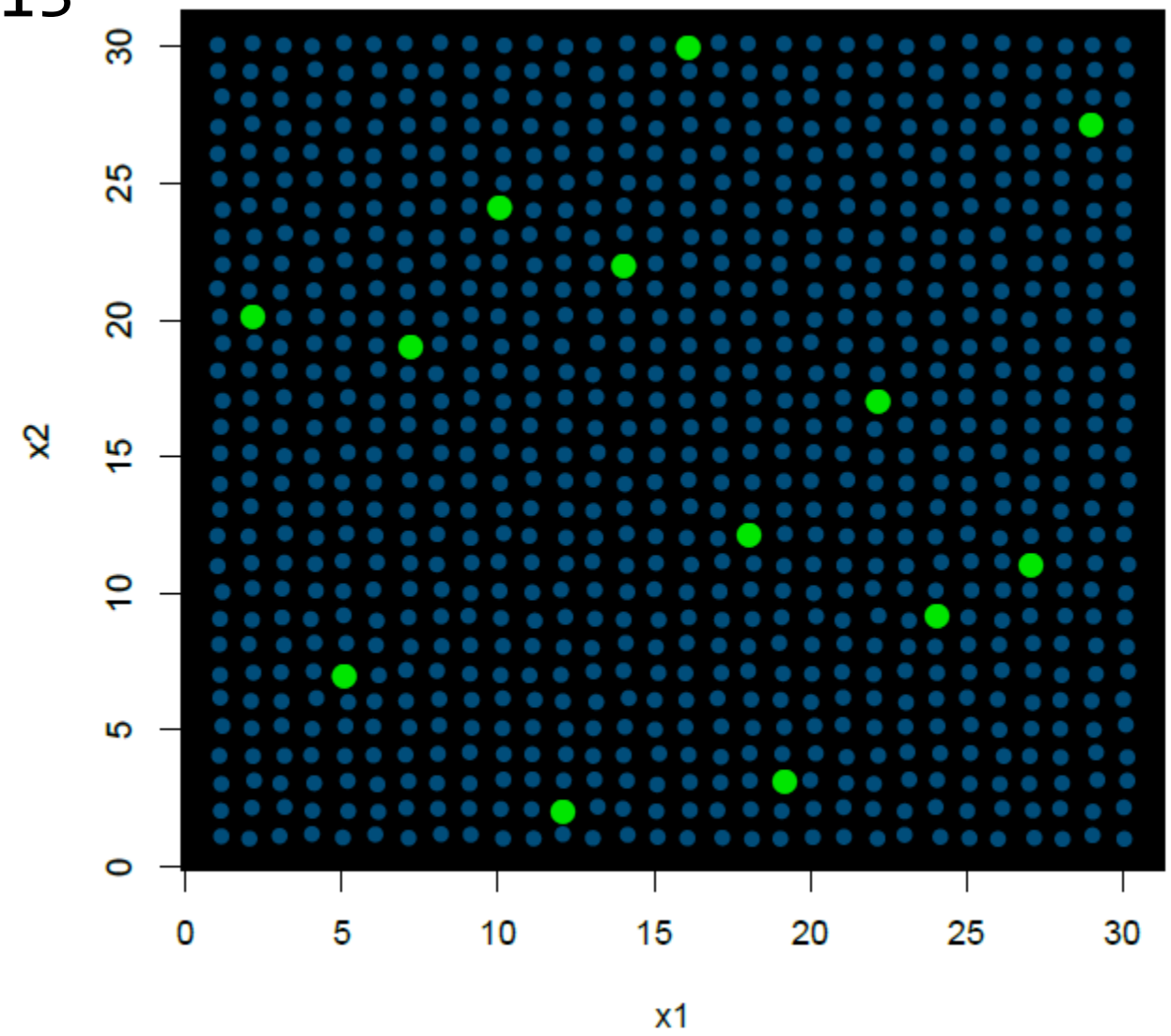
n = 12

contioned Latin Hypercube sampling (cLHS)



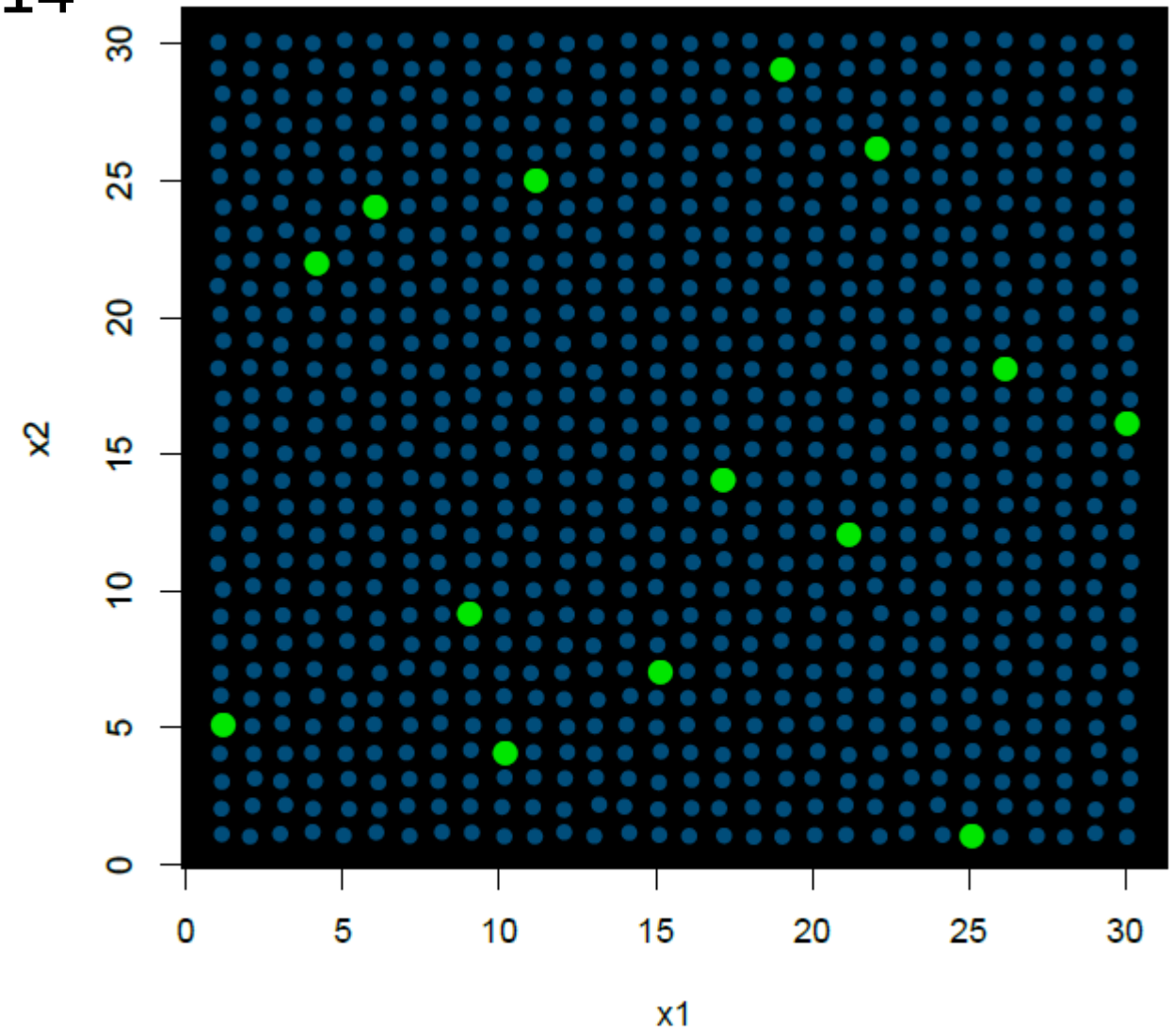
n = 13

contioned Latin Hypercube sampling (cLHS)



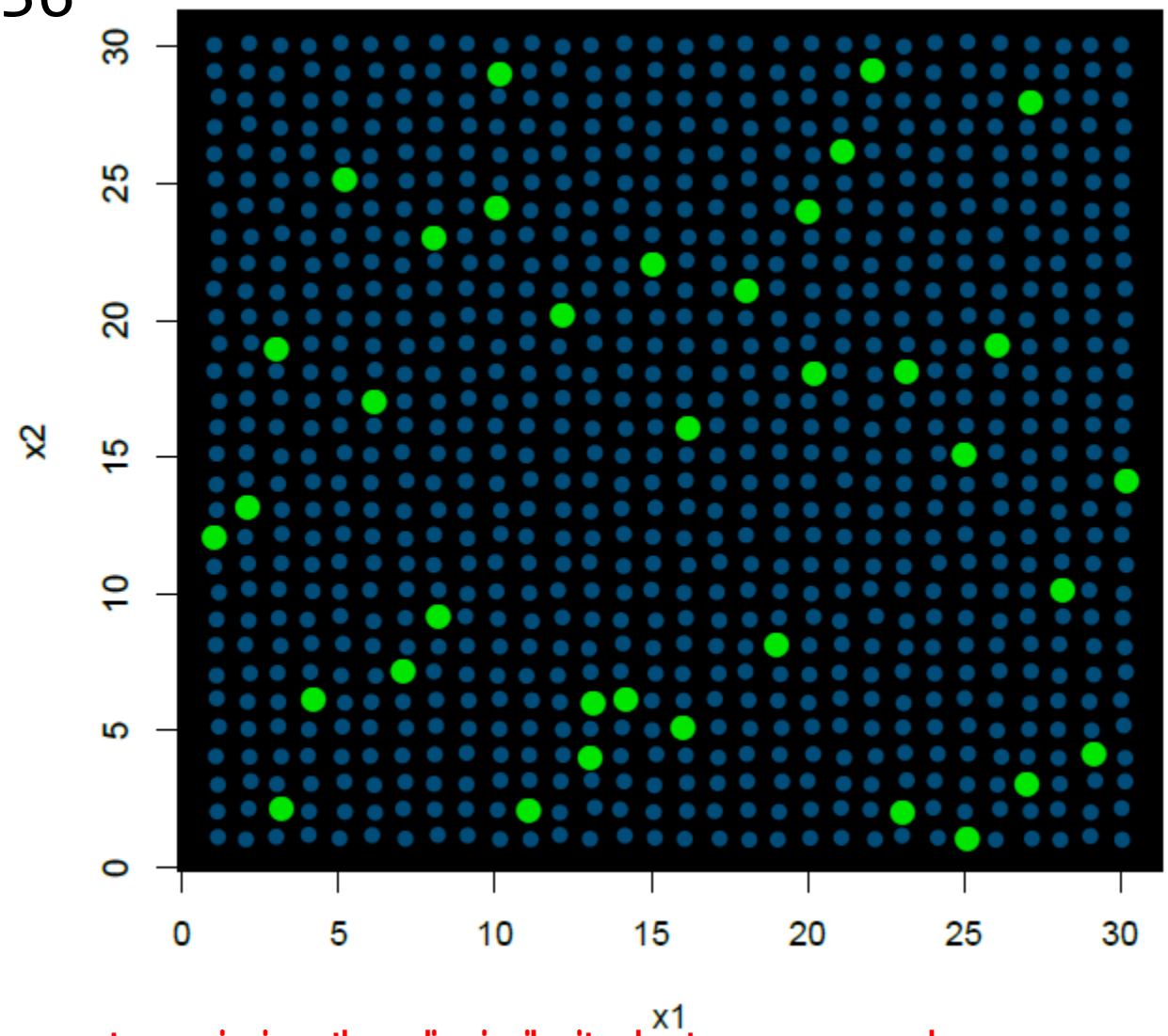
n = 14

contioned Latin Hypercube sampling (cLHS)



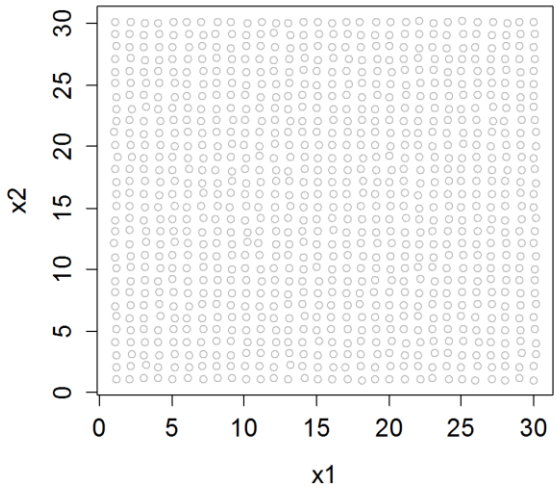
n = 36

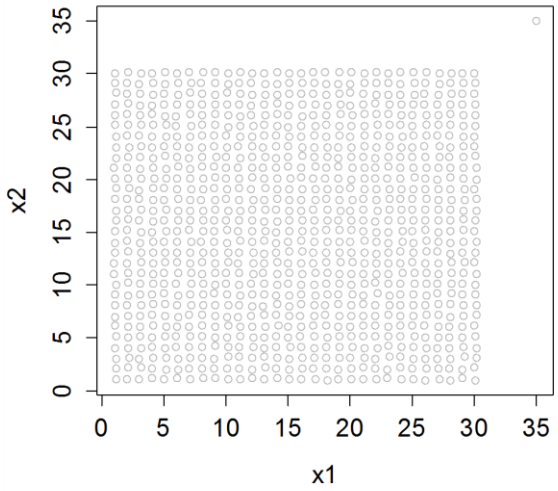
contioned Latin Hypercube sampling (cLHS)



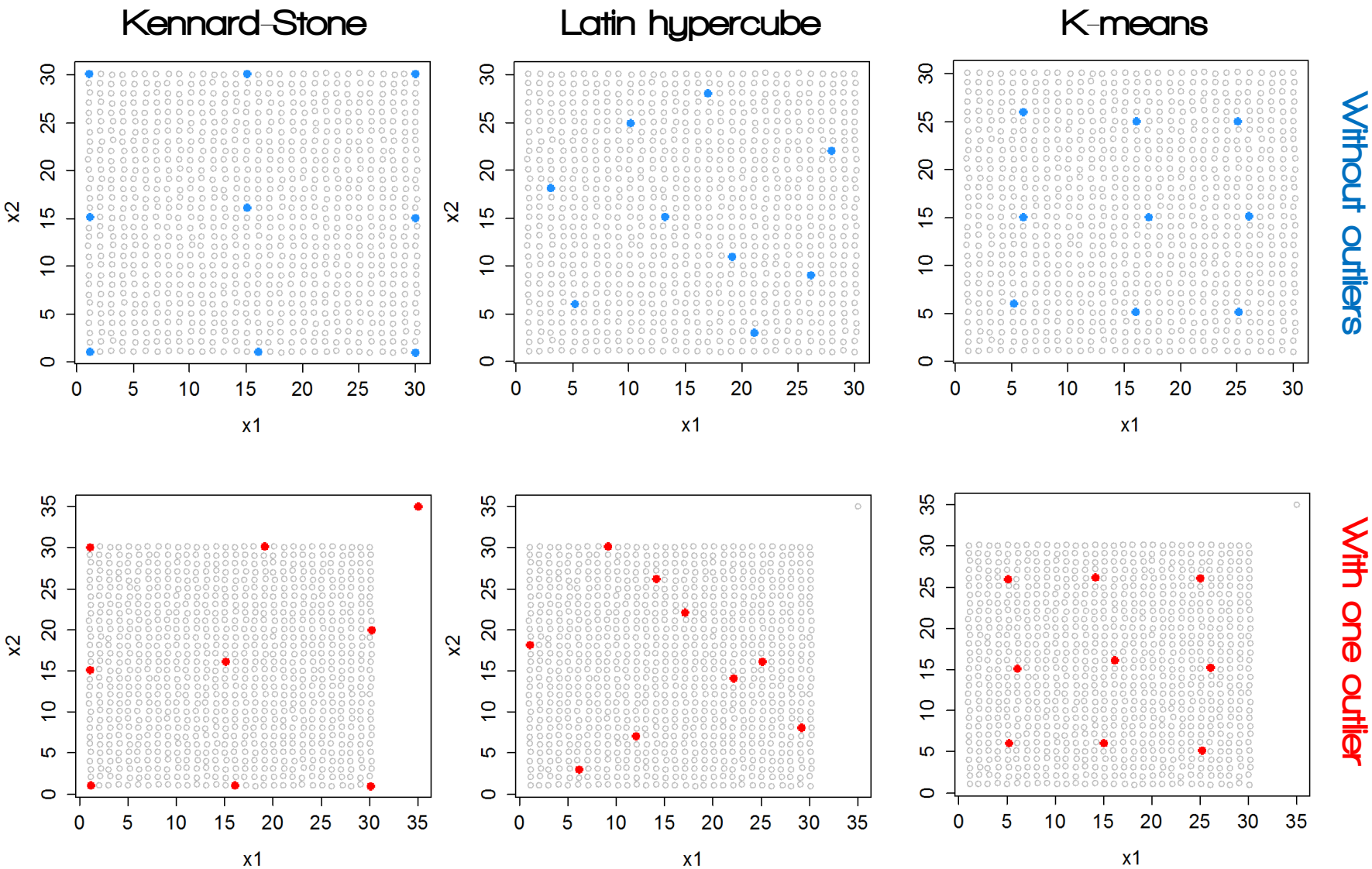
cLHS does not maximize the dissimilarity between samples

What if we have outliers in our data?





What if we have outliers in our data?

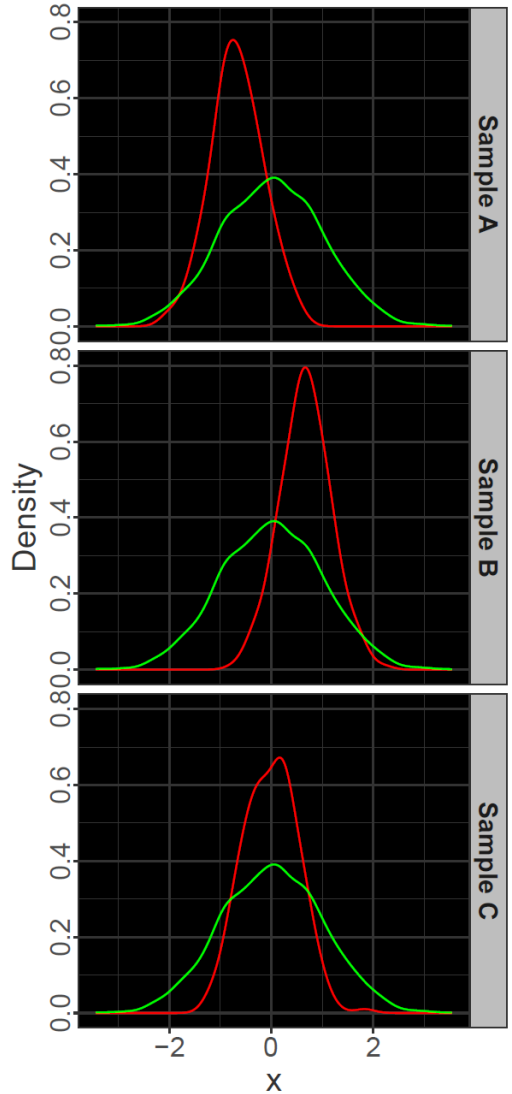


Size of the calibration set

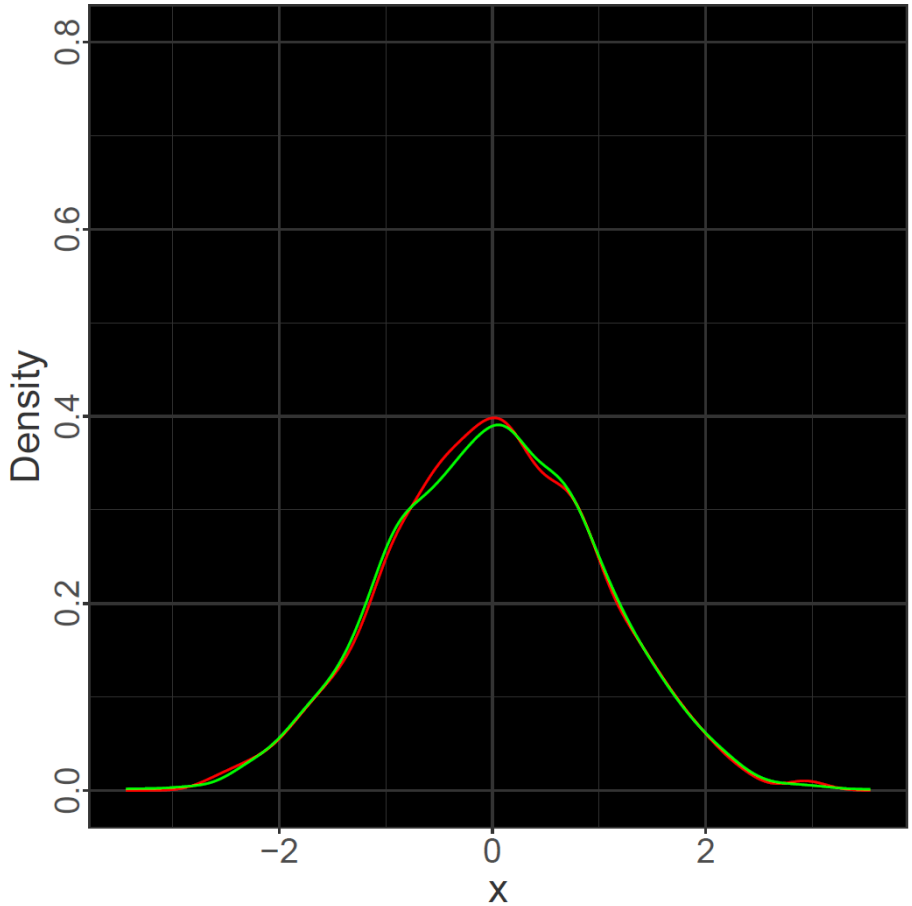
Distribution of the whole population

Distribution of the sample

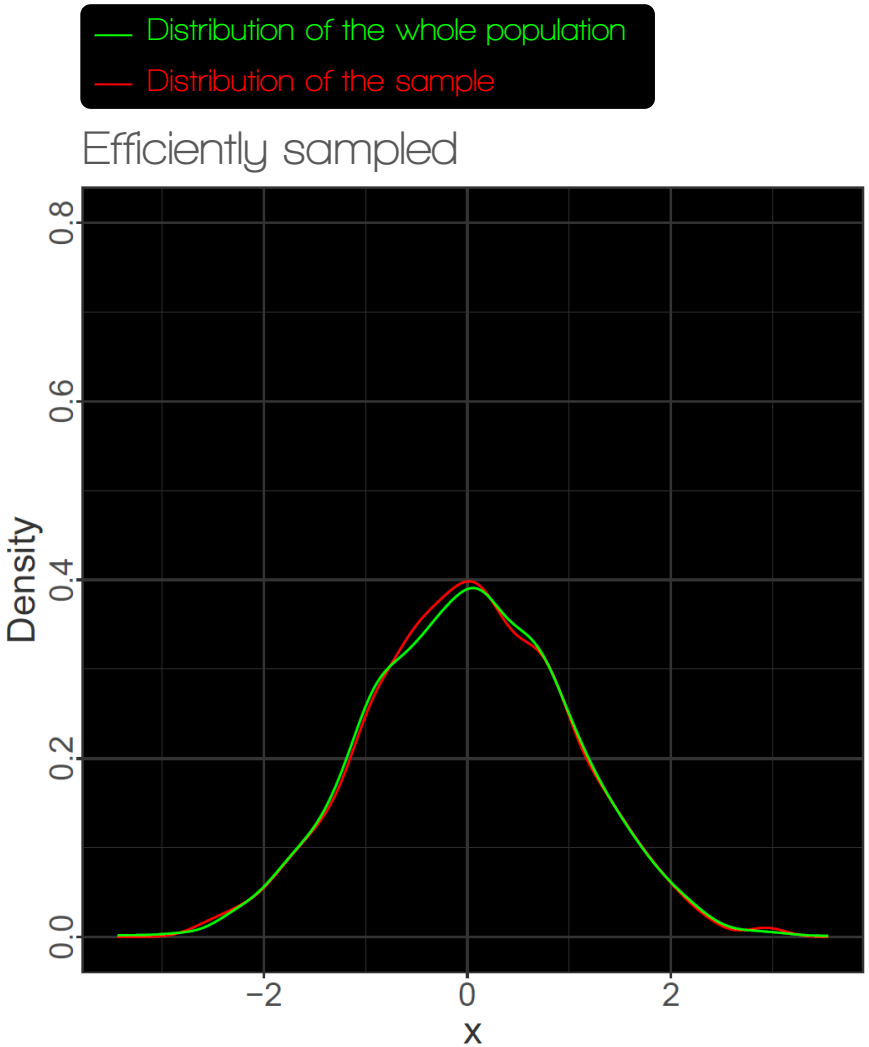
Low sampling efficiency



Efficiently sampled



Size of the calibration set



Comparing the probability density function (pdf) of the population and the pdf of the sample set can be useful for assesing the representativeness of the sample set.

These pdfs can be computed for the PCs of the vis–NIR data.

Size of the calibration set

The mean squared Euclidean distance (msd) can be used as a dissimilarity measure between the pdfs. It can be computed as follows:

$$msd = \frac{1}{k}\sum_{j=1}^{k} d^2\left[P_s(x_j \in cs), P_p(x_j)\right]$$

where:

$$d^2\left[P_s(x_j \in cs), P_p(x_j)\right] = \int_a^b \left[P_p(x_j) - P_s(x_j \in cs)\right]^2 dx_j$$

and where:

cs: a given subset of samples

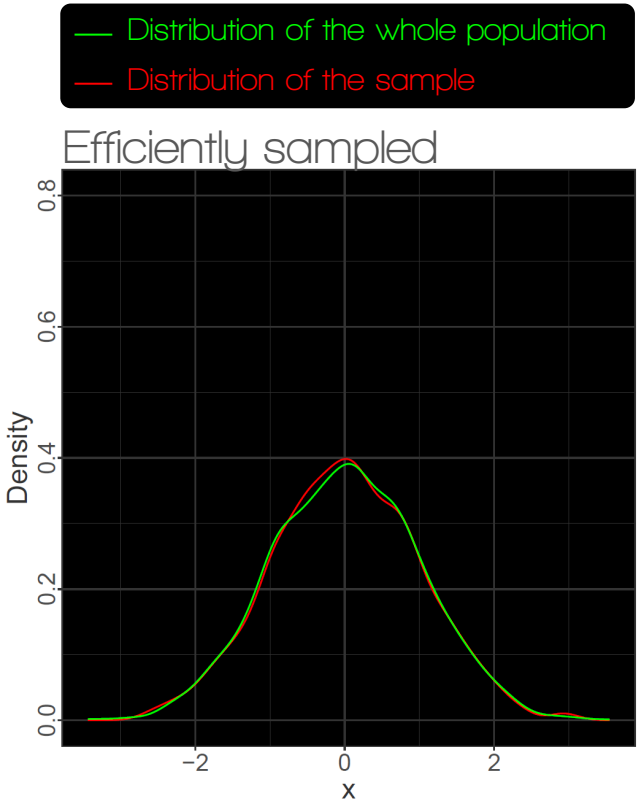
$P_s(x_j \in cs)$: the estimated pdf of the jth PC of cs,

$P_p(x_j)$: is the pdf of the jth PC for the whole population,

a and b: The range of the jth PC,

d^2 : represents the squared Euclidean distance between the two single distributions being compared,

k: the total number of PCs retained,



Example...

Soil vis-NIR library of Europe (n ~ 19.000)

Calibration sampling algorithms

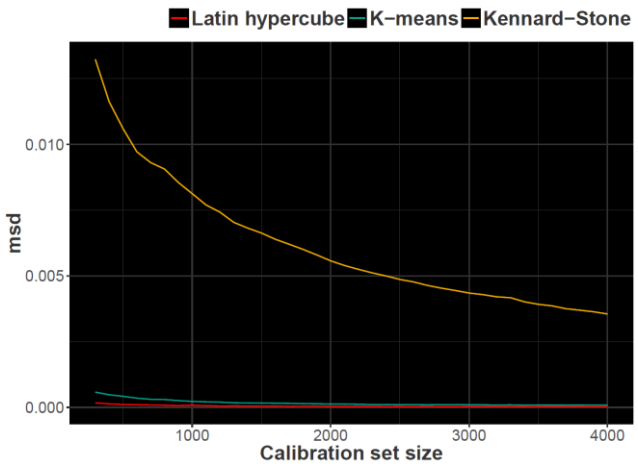
- Kennard-Stone sampling (KSS)
- Conditioned Latin hypercube sampling (cLHS)
- K-means sampling (KMS)

Calibration set size (CSS)

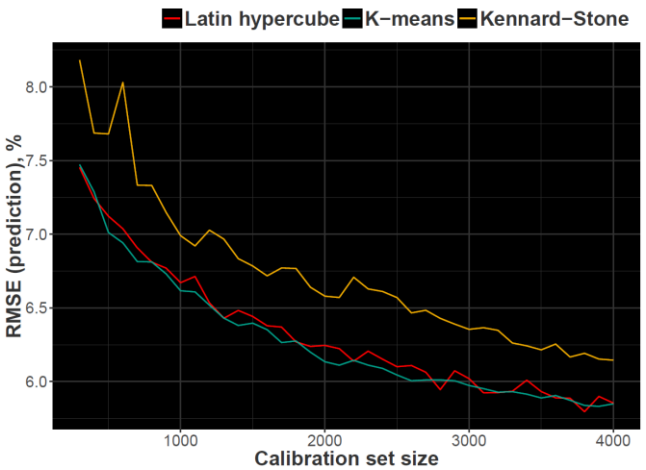
CSS = (300, 200, 400, ..., 4000)

The analysis is based on the comparison between the probability density function (pdf) of the population and the pdf of the sample set

Comparisons of the pdfs



Prediction error



Step 1
- Collect your samples
- Make the scans (raw spectra)

Step 2
- Pre-processing
- Outlier detection

Step 3
- Select your calibration samples
- Multivariate calibration

Step 4
- (Optional) transfer data across different instruments

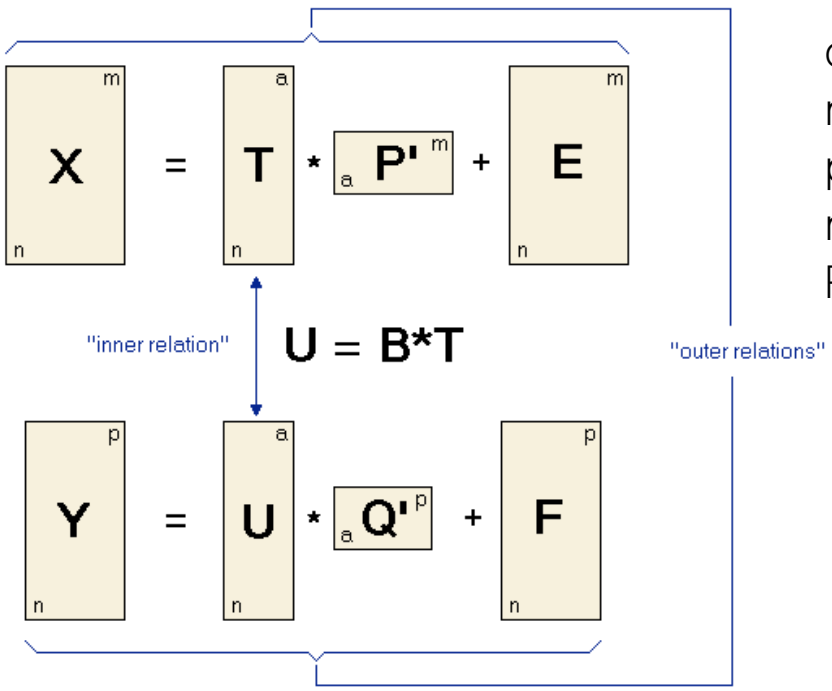
Partial Least Square Regression

- Easy to compute
- Stable solutions
- Provides scores and loadings for interpretation
- Often less number of components needed than PCR
- Robust (good) results over various case study
- It is the most common calibration method for Infrared spectroscopy

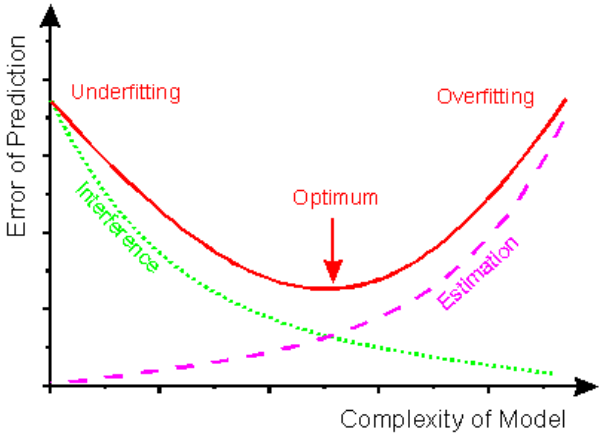
Partial Least Squares (PLS) regression builds a linear modell between

$\underline{X}$ source space (independent variables) and
$\underline{Y}$ target space (dependent, predicted variables)

Inside, it makes a PCA on $\underline{X}$ space, a PCA on $\underline{Y}$ space, then builds a linear regression between the first p dim (latent variables, factors) of two PCA spaces.



The optimal number of latent variables are determined by cross validation (building model on calibration data set, then checking prediction on validation set) on the base of minimal Root Mean Squared Error of Prediction (RMSEP)



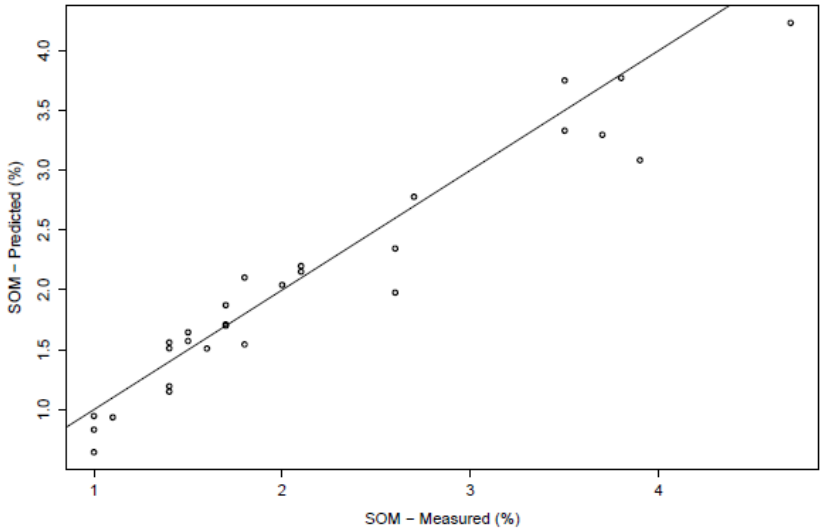
Validation is important

- Measure quality of the predictor
- Determine number of components
- Compare methods
- Which samples to validate?

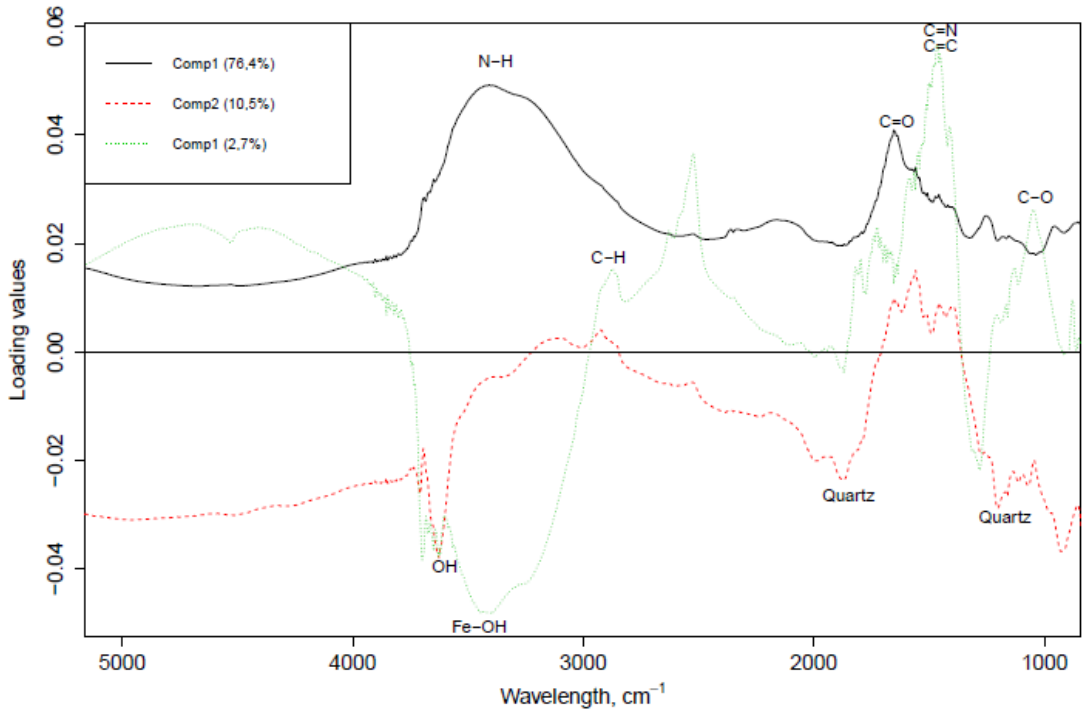
Prediction vs. Cross-validation

Prediction testing: Prediction ability of the predictor at hand. Requires much data.

Cross-validation: Property of the method. Better for smaller data set.



The coefficient of determination (R^2) characterizes the efficiency of PLSR model.



Interpretation

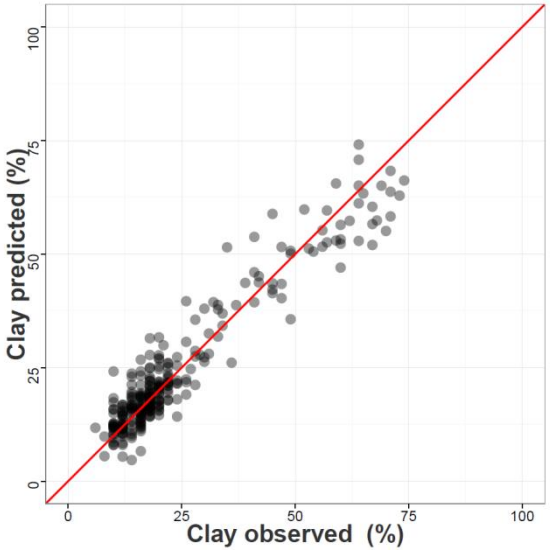
The significant wavelengths can be assigned by the loading values of regression.

Example: SOM band assignment based on the 3 first PCA loading values.

Soil vis–NIR libraries

Field scale

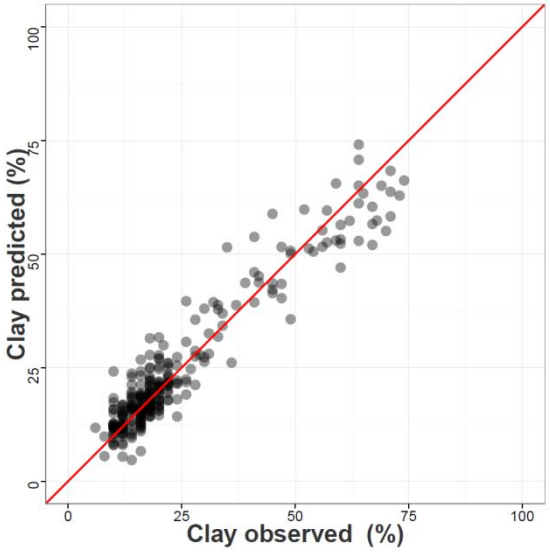
R^2 = 0.91; RMSE=5.02%



Soil vis–NIR libraries

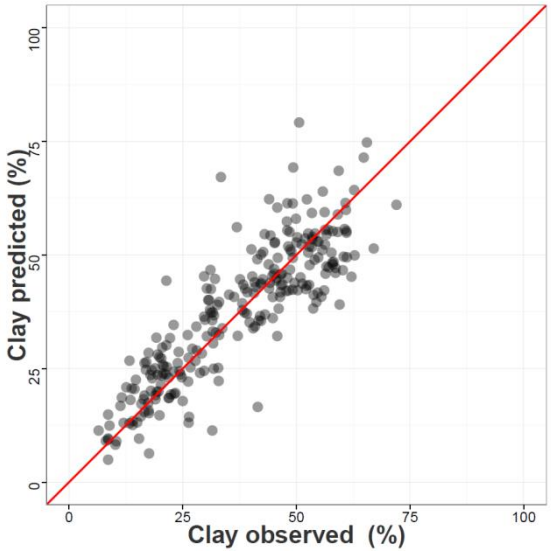
Field scale

R^2 = 0.91; RMSE=5.02%



Regional scale

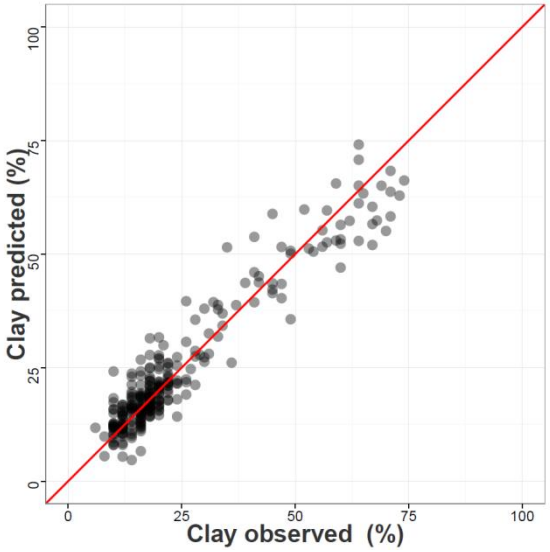
R^2 = 0.75; RMSE=8.03%



Soil vis–NIR libraries

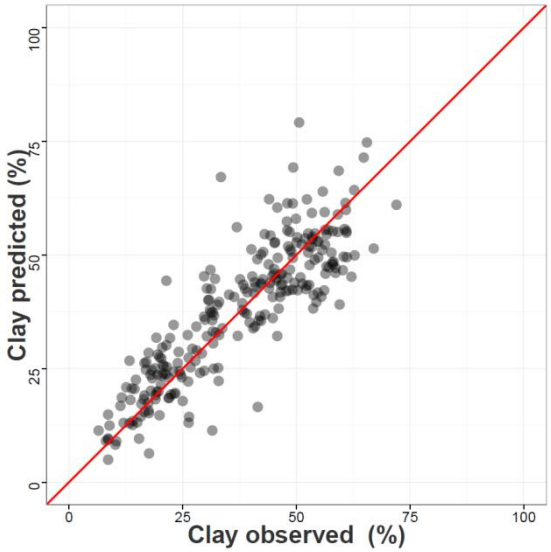
Field scale

R^2 = 0.91; RMSE=5.02%



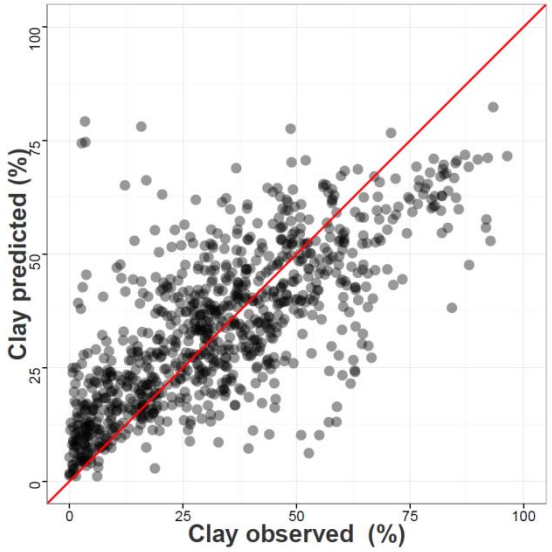
Regional scale

R^2 = 0.75; RMSE=8.03%



Global scale

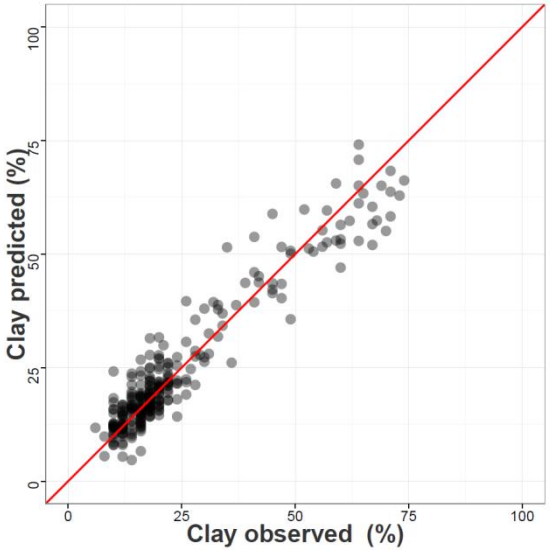
R^2 = 0.50; RMSE=15.53%



Soil vis–NIR libraries

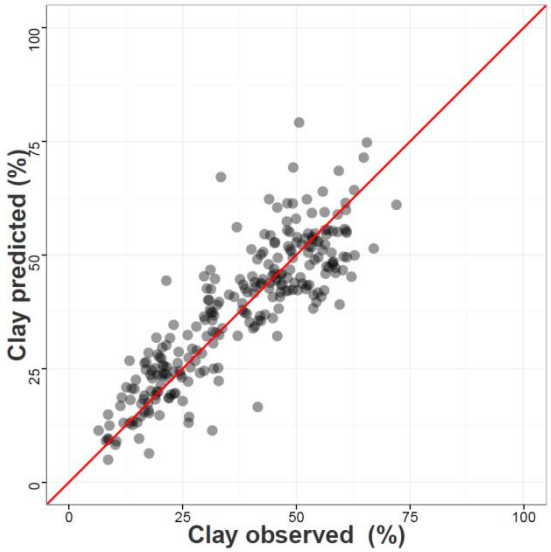
Field scale

R^2 = 0.91; RMSE=5.02%



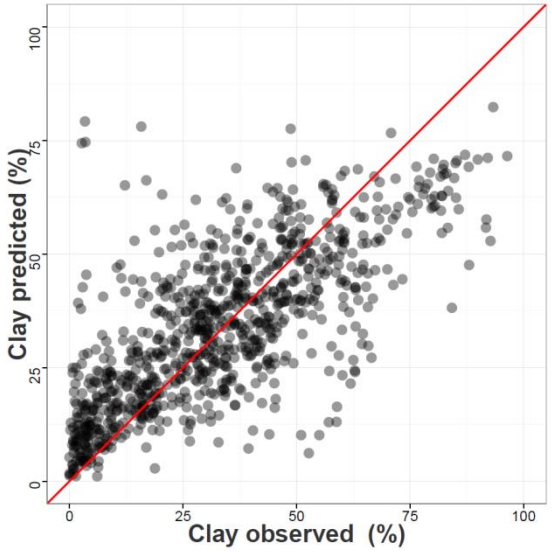
Regional scale

R^2 = 0.75; RMSE=8.03%



Global scale

R^2 = 0.50; RMSE=15.53%

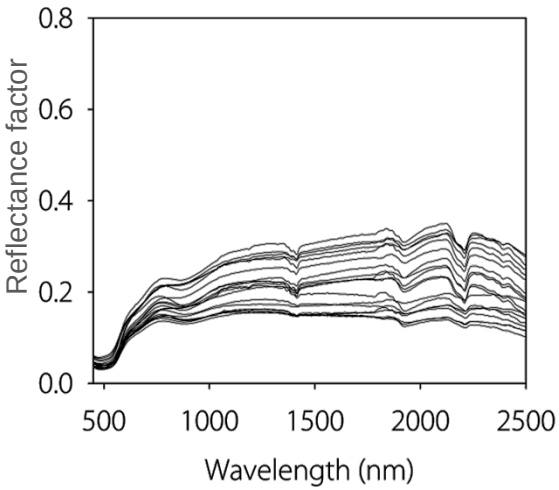


– Data complexity +

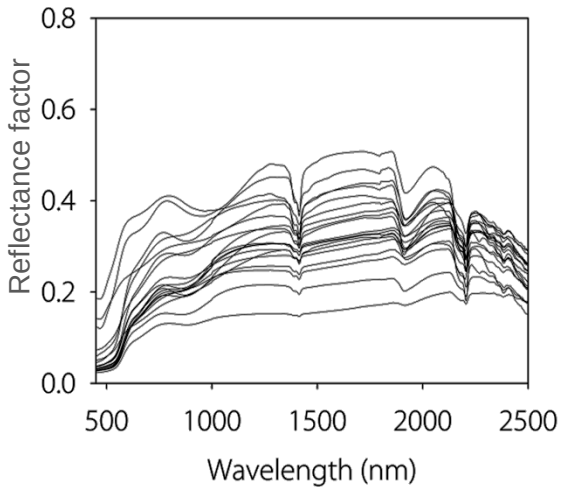
Degradation of the accuracy

Why are big soil vis–NIR libraries so complex?

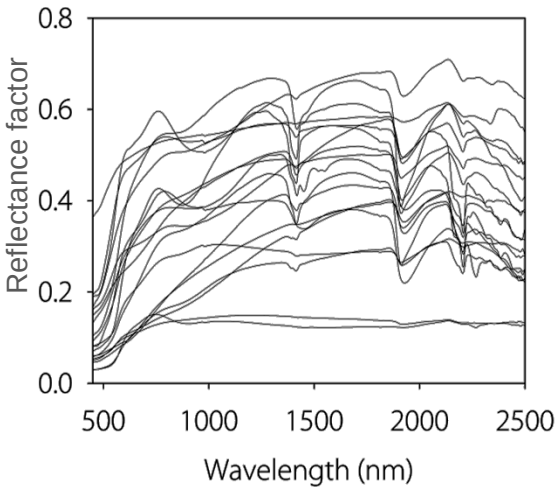
Field scale



Regional scale



Global scale



20 spectra sampled at random

Memory–based learning (MBL)

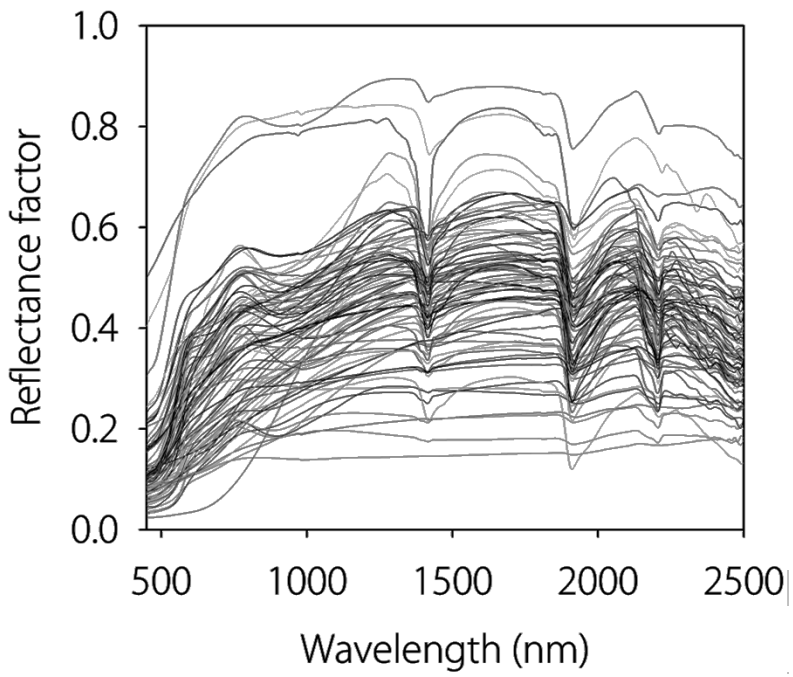
MBL is closely related to case based–reasoning (CBR) which emulates the human reasoning process:

1. Remember previous situations
2. Adapt them for solving the current problem
3. Examine the probability to solve the problem with the new solution
4. Memorize the experience for improving knowledge

MBL for soil spectral libraries

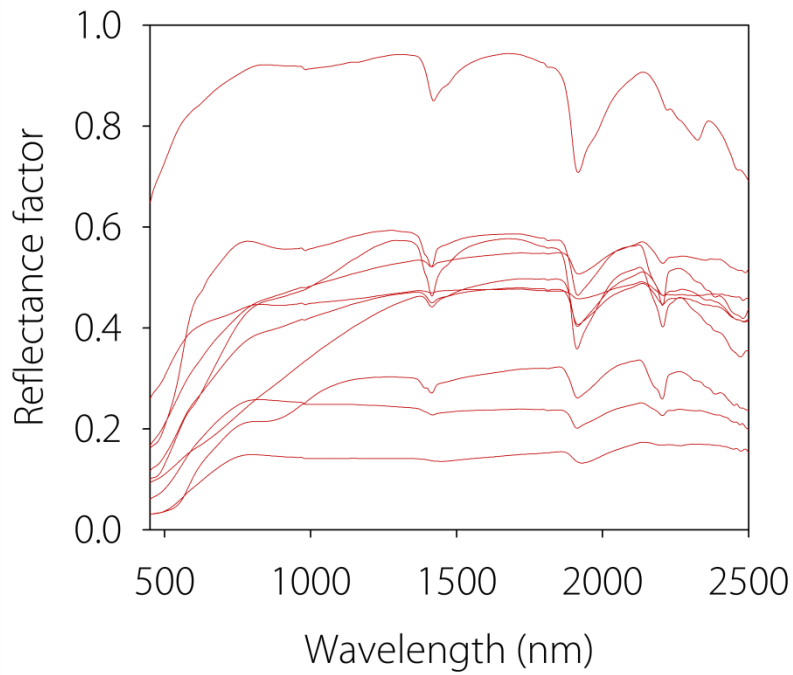
Soil spectral library

$$(\mathrm{Xr}, \mathrm{Yr}) = \{\mathrm{xr}_j, \mathrm{yr}_j\}_{j=1}^{n}$$



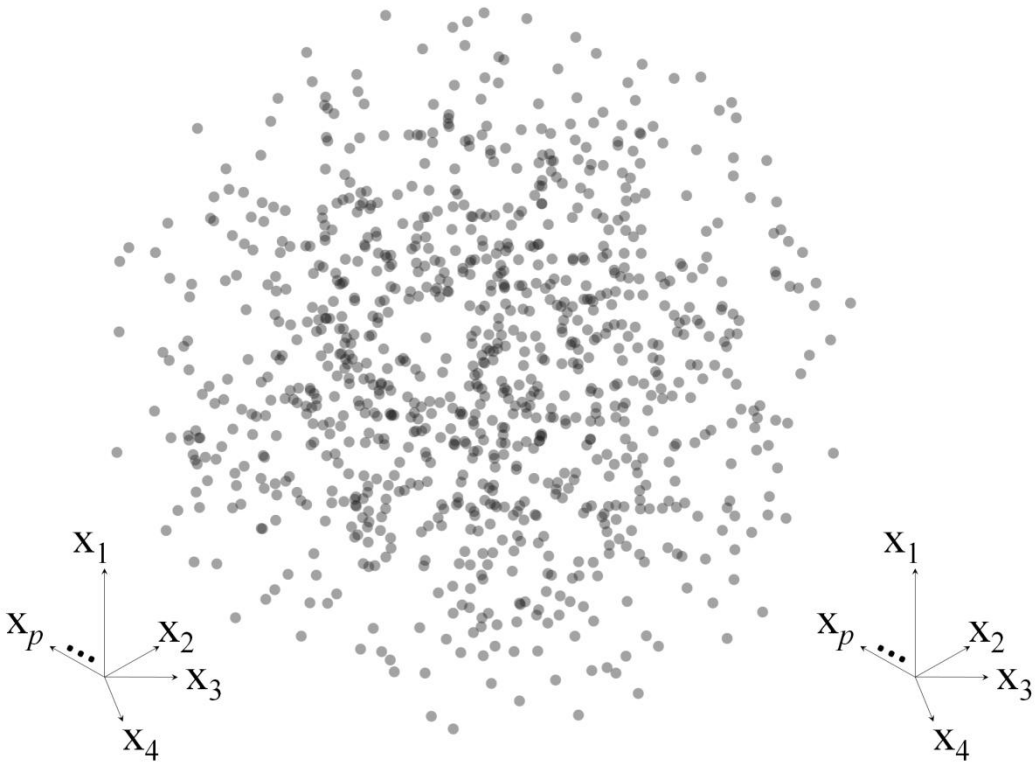
Soil attribute with unknown values

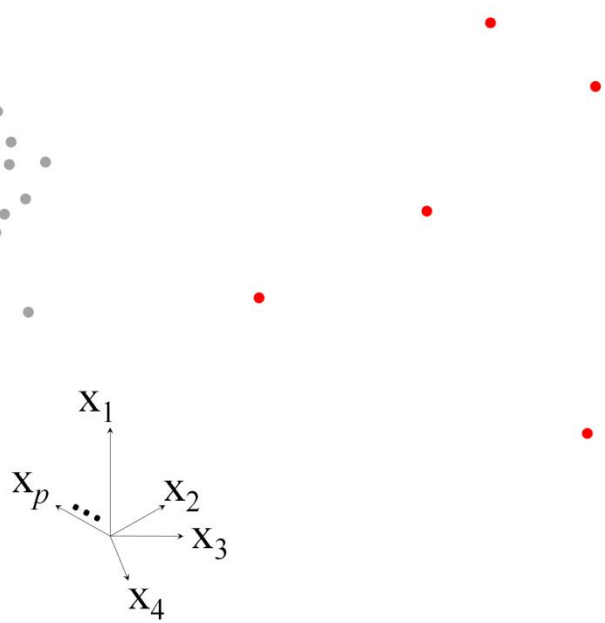
$$(\mathrm{Xu}, \mathrm{Yu}) = \{\mathrm{xu}_i, \mathrm{yu}_i\}_{i=1}^{m}$$



$$(\mathrm{Xr}, \mathrm{Yr}) = \{\mathrm{xr}_j, \mathrm{yr}_j\}_{j=1}^{n}$$

$$(\mathrm{Xu}, \mathrm{Yu}) = \{\mathrm{xu}_i, \mathrm{yu}_i\}_{i=1}^{m}$$

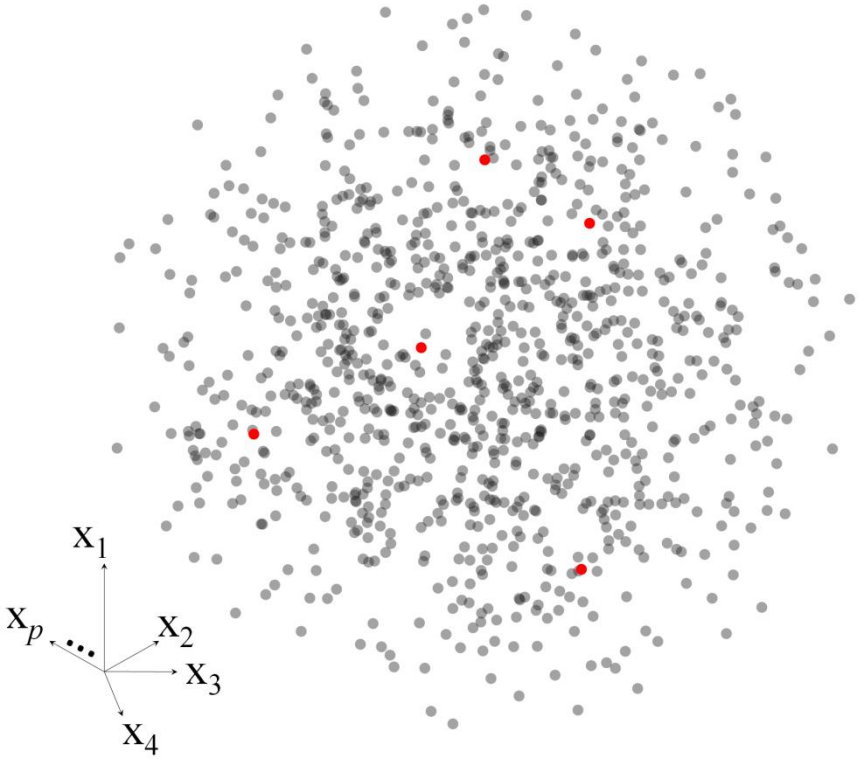




p–dimensional spectral feature space

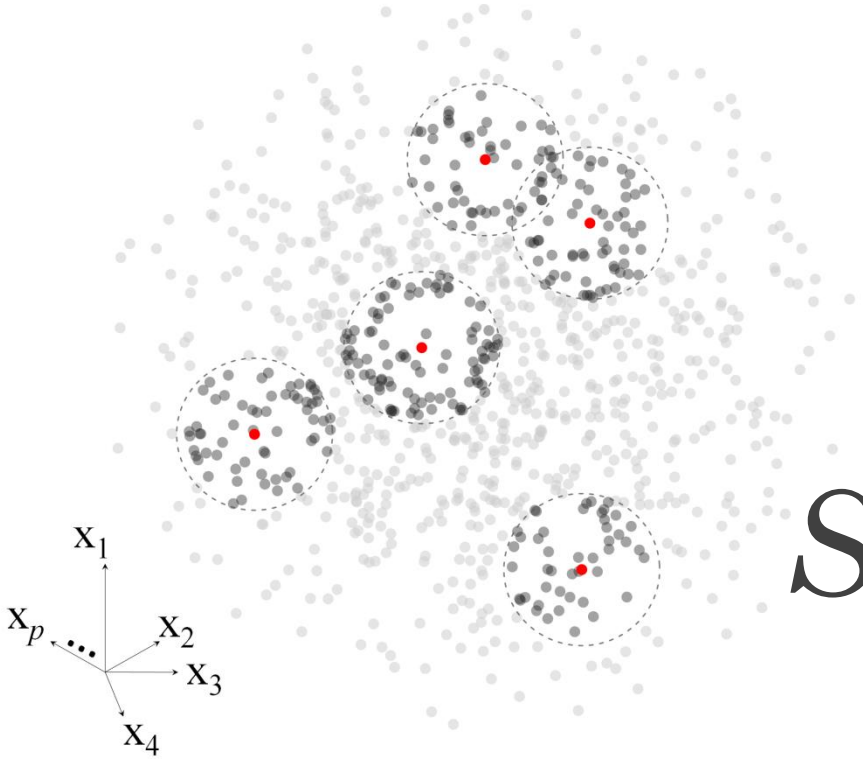
$$(\mathrm{Xr}, \mathrm{Yr}) = \{\mathrm{xr}_j, \mathrm{yr}_j\}_{j=1}^{n}$$

$$(\mathrm{Xu}, \mathrm{Yu}) = \{\mathrm{xu}_i, \mathrm{yu}_i\}_{i=1}^{m}$$



$$S(\mathrm{xu}_i, \mathrm{xr}_j)$$

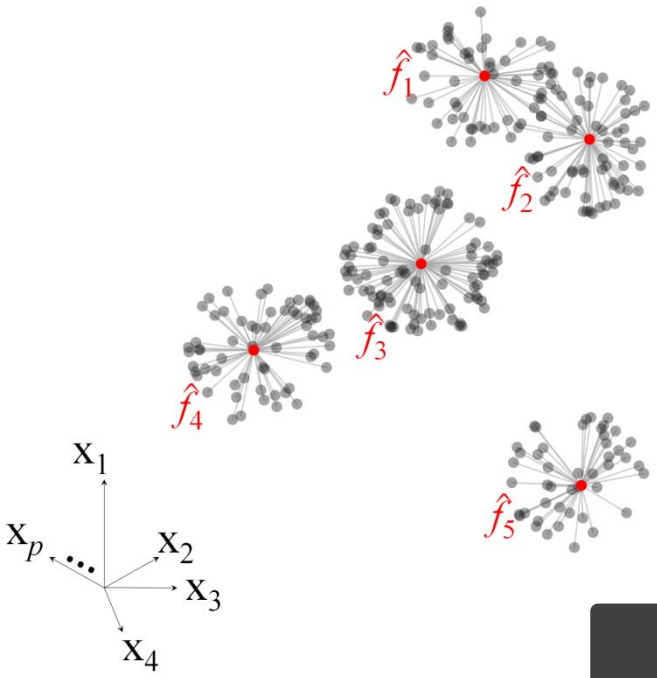
Sample neighbors



$$S(\mathrm{xu}_i, \mathrm{xr}_j)$$

Memory based learning (MBL)

The solution to a complex task is constructed by a collection of simple local functions



$$y_i = \boxed{\hat{f}_i}(x;\theta)+\varepsilon$$

MBL

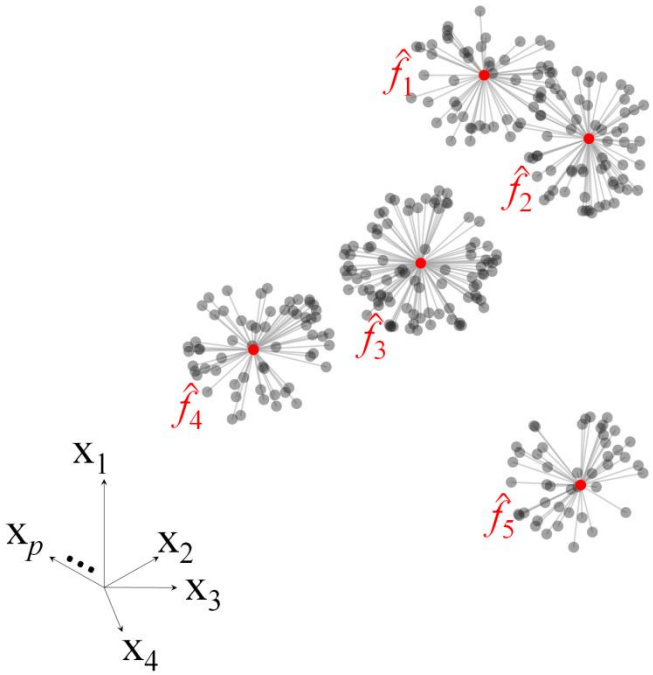
$$\{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_m\}$$

Noisy and/or non-informative samples for f are ignored

Memory based learning (MBL)

There are four basic aspects that must be defined for any MBL algorithm:

1. A similarity/dissimilarity metric
2. How many nearby neighbors to look at?
3. How to use the similarity/dissimilarity information
4. How to fit with the local points?



Memory based learning (MBL)

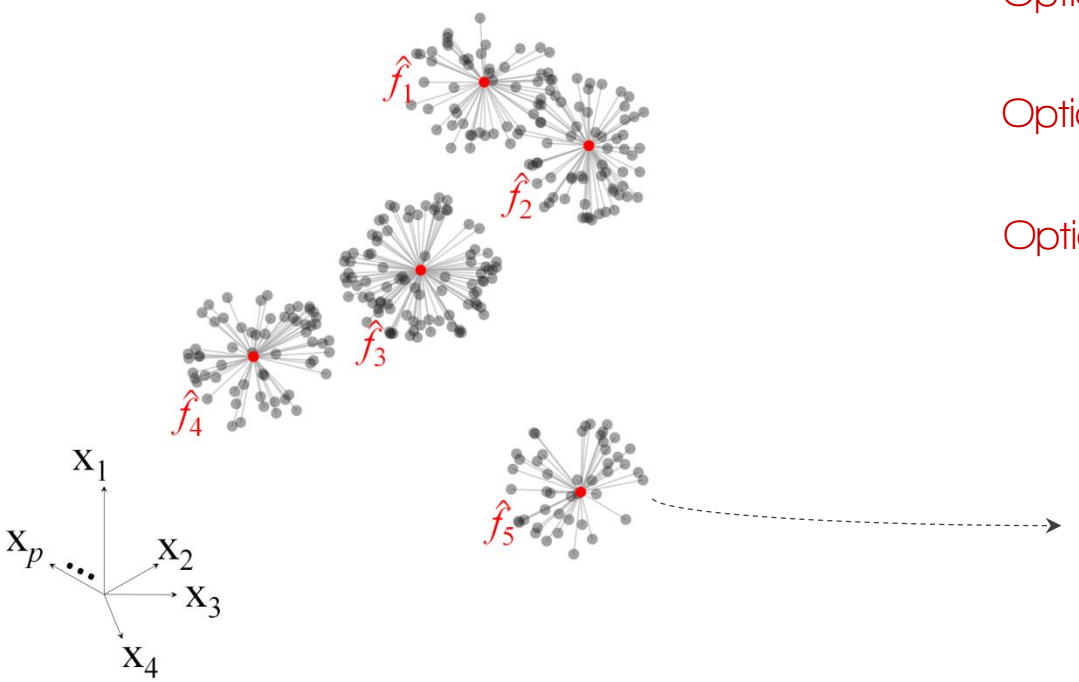
There are four basic aspects that must be defined for any MBL algorithm:

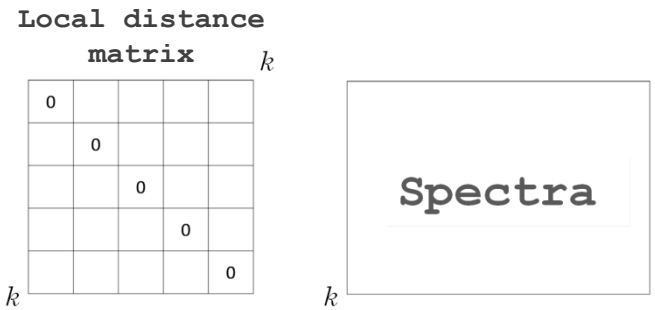
1. A similarity/dissimilarity metric
2. How many nearby neighbors to look at?
3. How to use the similarity/dissimilarity information
4. How to fit with the local points?

Option 1: Ignore it

Option 2: Use it for assigning weights

Option 3: Source of additional predictors...





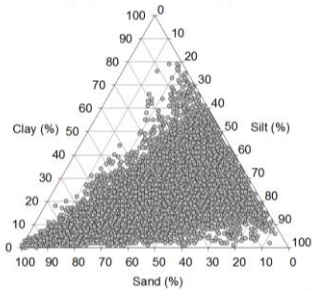
The datasets…

Two (complex) soil vis-NIR libraries were used in order to test the performance of the MBL algorithms

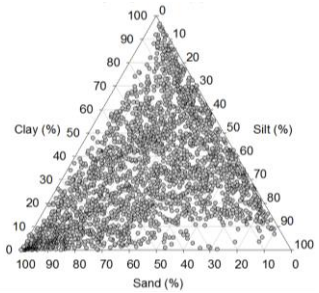
Library	'Unknown'set	Reference set
Continental (Europe)*	4644	14392
Global**	736	2907

*23 Countries
**55 Countries

Continental SSL



Global SSL



The algorithms…

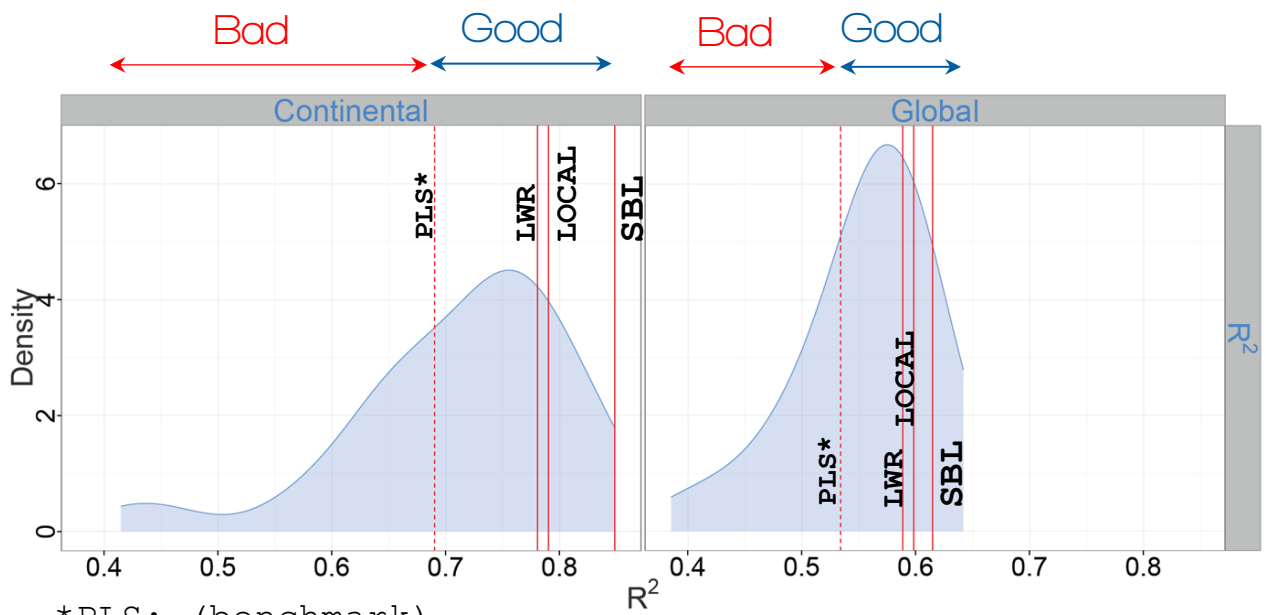
90 combinations of Dissimilarity + Regression algorithm + Dissimilarity usage

Dissimilarity	Regression algorithm	Use of dissimilarity scores in local regressions
Euclidean		
Cosine		
Spectral information divergence 1		
Spectral information divergence 2		
Correlation	PLS	None
Rolling correlation	Weighted average PLS	Weights
Principal components	Gaussian process	Predictors
Optimized principal components		
PLS		
Optimized PLS		

* Tóth, G., Jones, A., Montanarella, L. (eds.) 2013. LUCAS Topsoil Survey. Methodology, data and results. JRC Technical Reports. Luxembourg. Publications Office of the European Union, EUR 26102 – Scientific and Technical Research series – ISSN 1831-9424 (online); ISBN 978-92-79-32542-7; doi: 10.2788/979224

** World Agroforestry Centre (ICRAF) and ISRIC – World Soil Information. 2010. ICRAF-ISRIC Soil vis-NIR spectral Library. Nairobi, Kenya: World Agroforestry Centre (ICRAF)

Memory based learning (MBL)



the RESULTS...

Probability distribution of R^2 when local models are used

*PLS: (benchmark)

Name	Dissimilarity	Regression	Dissimilarity usage
LOCAL**	Correlation	Weighted average PLS	None
LWR	Principal components (PC)	PLS	Weights
SBL***	Optimized PC***	Gaussian process	Predictors

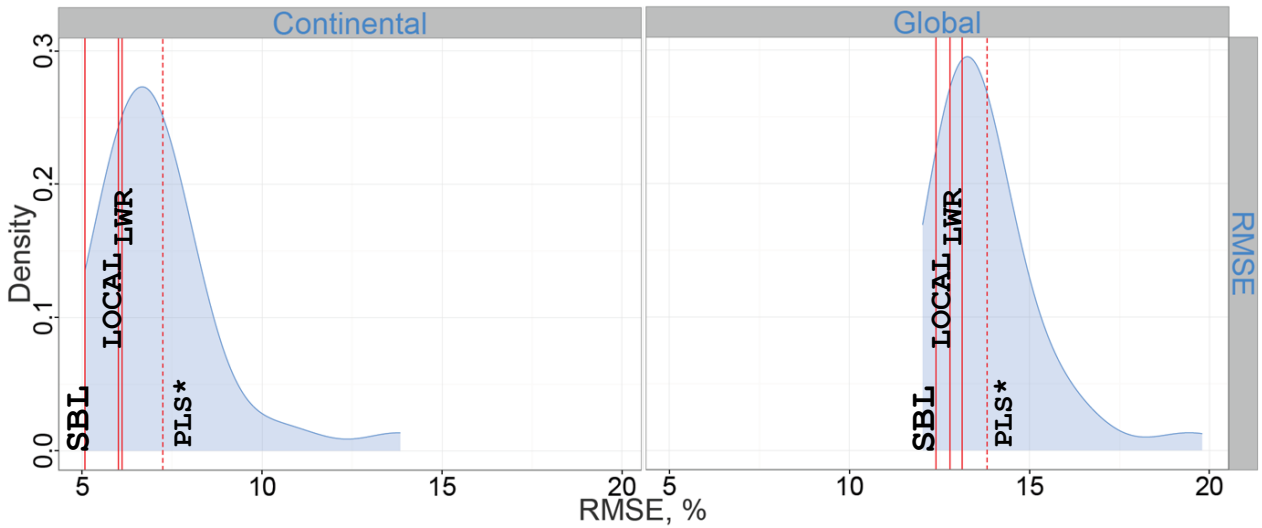
The 'bad' predictions were dominated by those methods that applied weighting

The very good predictions were dominated by the methods that used the o-pc and o-pls dissimilarity measures

**Shenk, J.S., Westerhaus, M.O., Berzaghi, P. 1997. Investigation of a LOCAL calibration procedure for near infrared instruments. Journal of Near Infrared Spectroscopy 5, 223-232.

Naes, T., Isaksson, T., Kowalski, B. 1990. Locally weighted regression and scatter correction for nearinfrared reflectance data. Analytical Chemistry 62, 664-673

***Ramirez-Lopez, L, et al. The spectrum-based learner: A new local approach for modeling soil vis-NIR spectra of complex datasets. *Geoderma* 195 (2013): 268-279.



the RESULTS...

Probability distribution of **RMSE** when local models are used

*PLS: (benchmark)

Name	Dissimilarity	Regression	Dissimilarity usage
LOCAL**	Correlation	Weighted average PLS	None
LWR	Principal components (PC)	PLS	Weights
SBL***	Optimized PC***	Gaussian process	Predictors

The 'bad' predictions were dominated by those methods that applied weighting

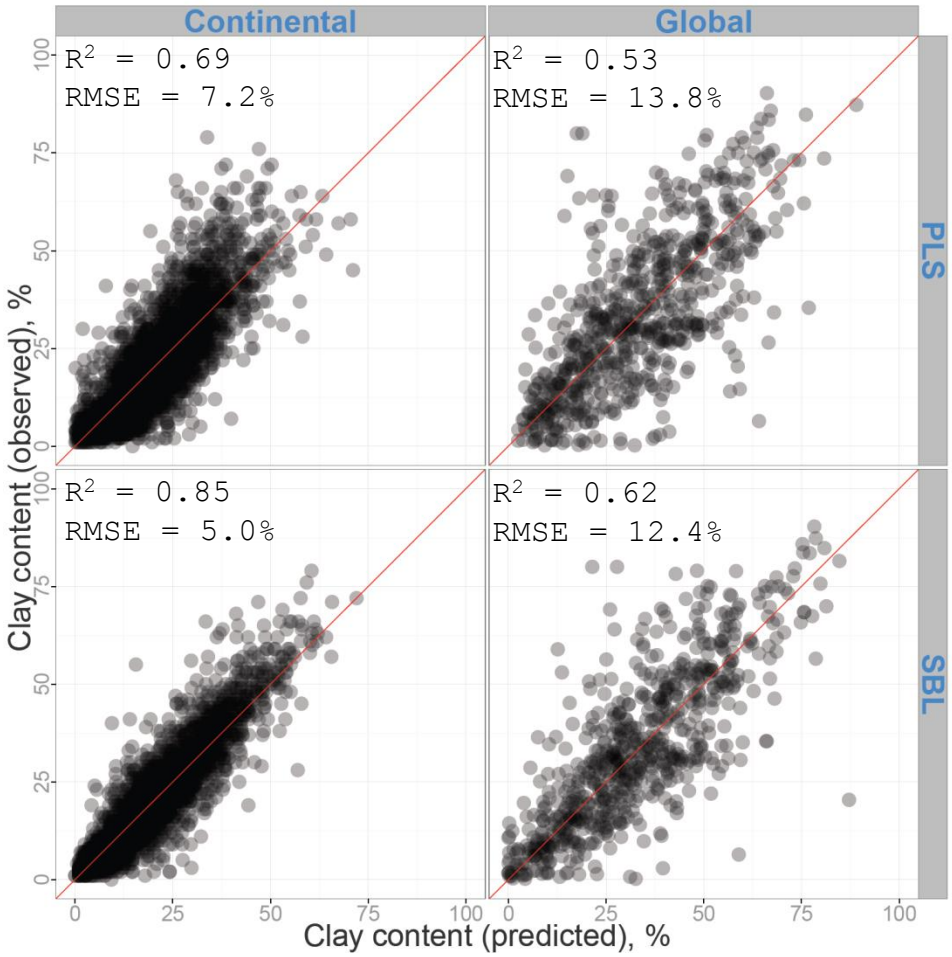
The very good predictions were dominated by the methods that used the o-pc and o-pls dissimilarity measures

**Shenk, J.S., Westerhaus, M.O., Berzaghi, P. 1997. Investigation of a LOCAL calibration procedure for near infrared instruments. Journal of Near Infrared Spectroscopy 5, 223-232.

Naes, T., Isaksson, T., Kowalski, B. 1990. Locally weighted regression and scatter correction for nearinfrared reflectance data. Analytical Chemistry 62, 664-673

***Ramirez-Lopez, L, et al. The spectrum-based learner: A new local approach for modeling soil vis-NIR spectra of complex datasets. *Geoderma* 195 (2013): 268-279.

Memory based learning (MBL)



the RESULTS...

If (still_time?==YES){continue below}
else{STOP}

Step 1
- Collect your samples
- Make the scans (raw spectra)

Step 2
- Pre-processing
- Outlier detection

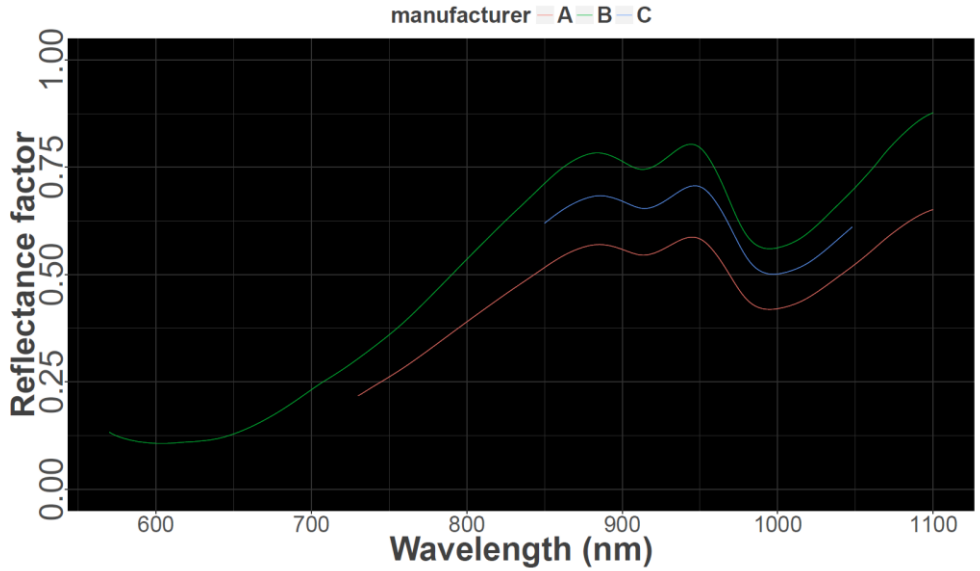
Step 3
- Select your calibration samples
- Multivariate calibration

Step 4
- (Optional) transfer data across different instruments

Spectra transfer

Product: Wheat
Property: Protein content
N samples: 248
Manufactureres: 3
- A: 4 instruments
- B: 4 instruments
- C: 1 instrument

Average spectra delivered by the manufacturers

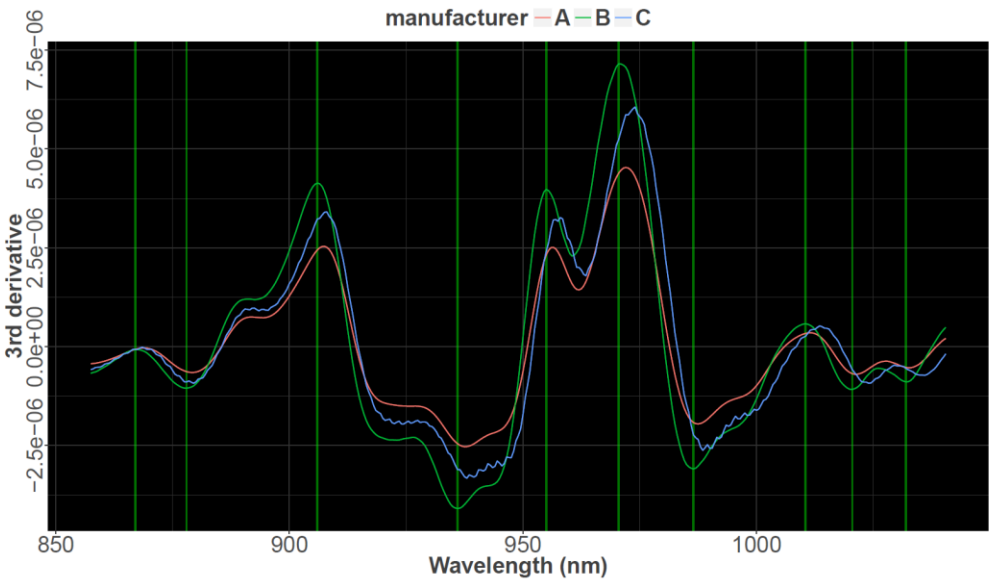


Different wavelength ranges
Different albedo

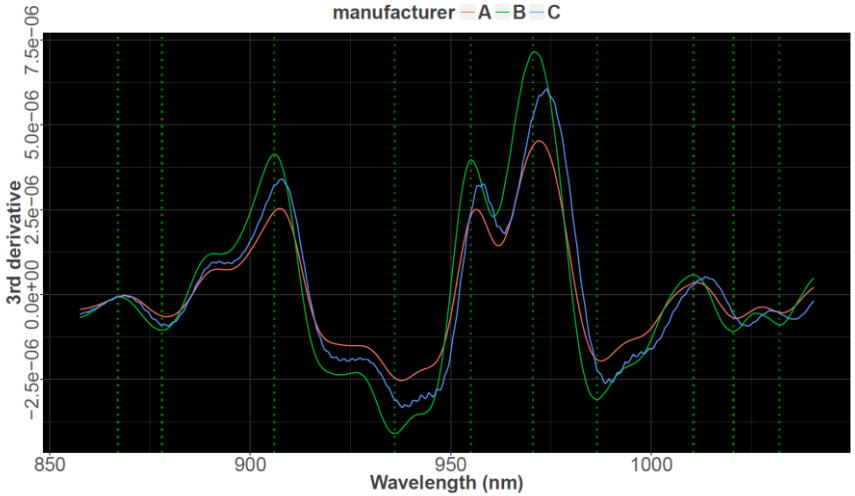
Spectra transfer

The problem:

Different sensors retrieve shifted signals



Differences between manufacturers

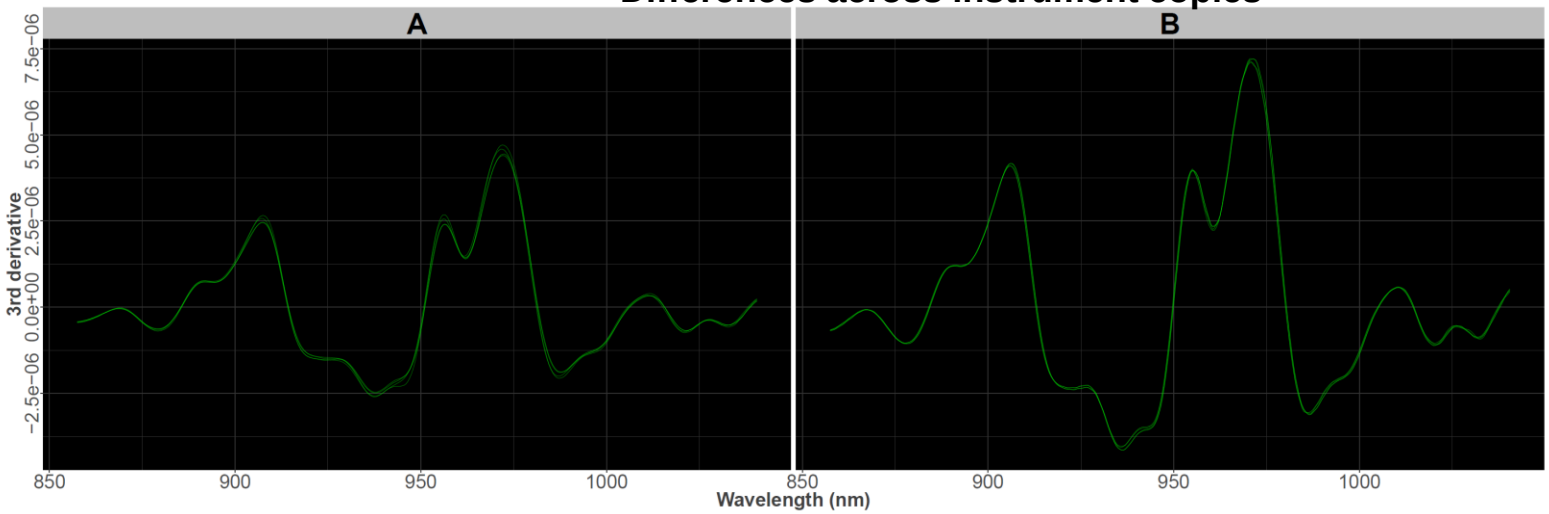


Reference/master instrument B1

Different peak positions (x axis):
up to 2.5 nm shifts

Different intensity

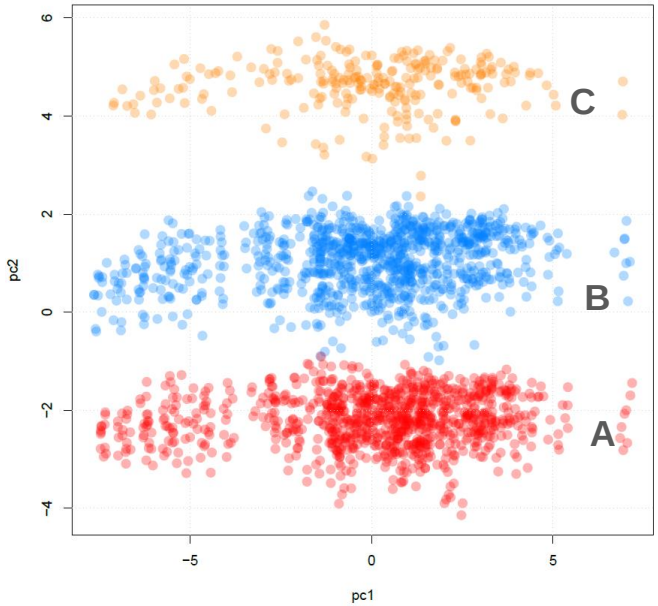
Differences across instrument copies



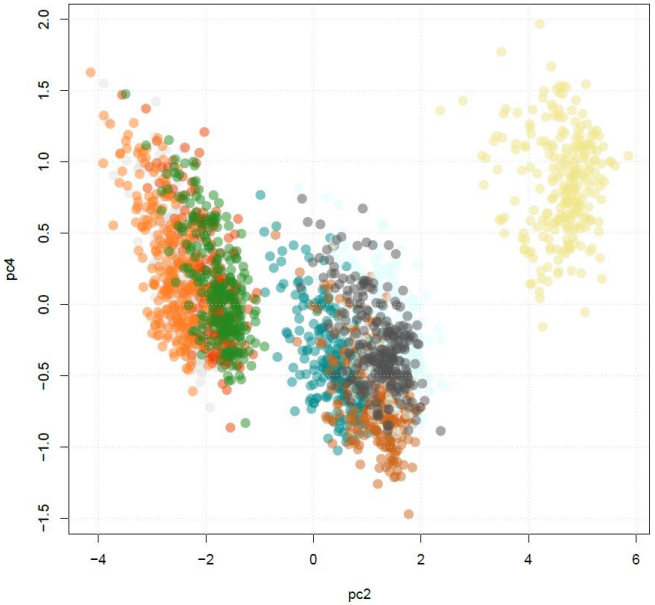
Principal component analysis

The spectra delivered by the different manufacturers are significantly different from the statistical point of view

Differences across manufacturers

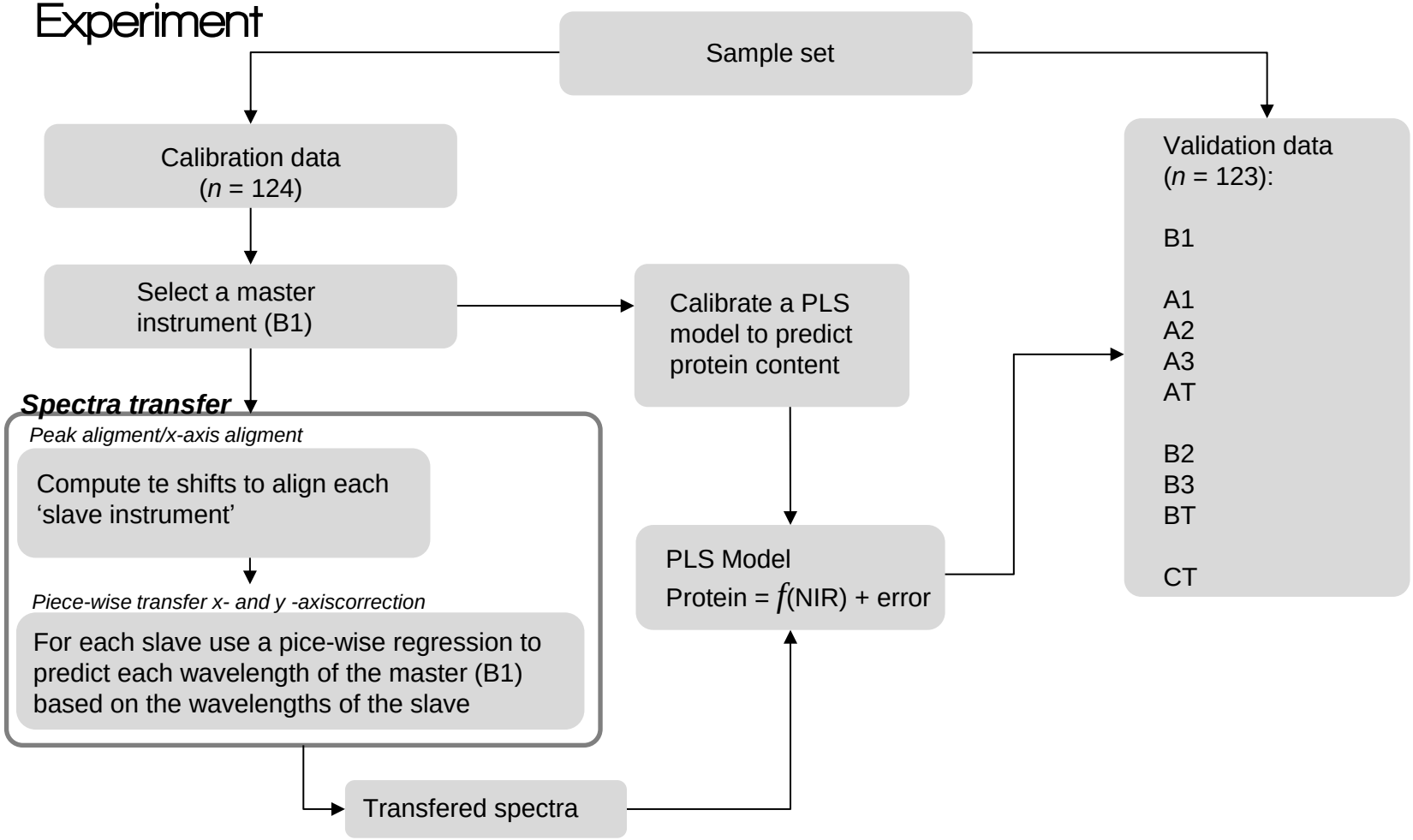


Differences across instrument copies



If the instruments would deliver similar NIR spectra, no grouping would be observed

Experiment


Sample set
Calibration data
(n = 124)
Select a master
instrument (B1)
Calibrate a PLS
model to predict
protein content
Validation data
(n = 123):
B1
A1
A2
A3
AT
B2
B3
BT
CT
Spectra transfer
Peak aligment/x-axis aligment
Compute te shifts to align each
'slave instrument'
Piece-wise transfer x- and y -axiscorrection
For each slave use a pice-wise regression to
predict each wavelength of the master (B1)
based on the wavelengths of the slave
PLS Model
Protein = f(NIR) + error
Transfered spectra

Experiment

Sample set

Calibration data
(*n* = 124)

Validation data
(*n* = 123):

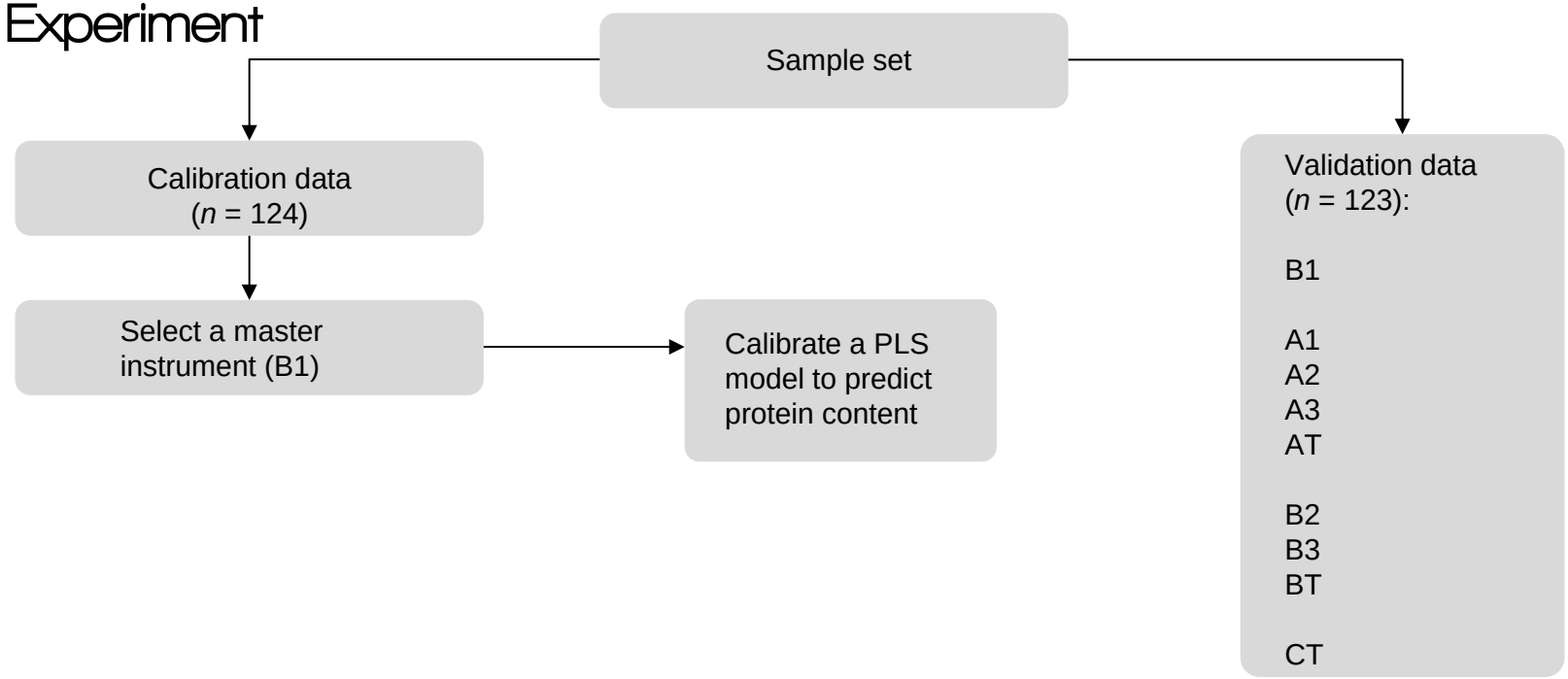
B1

A1
A2
A3
AT

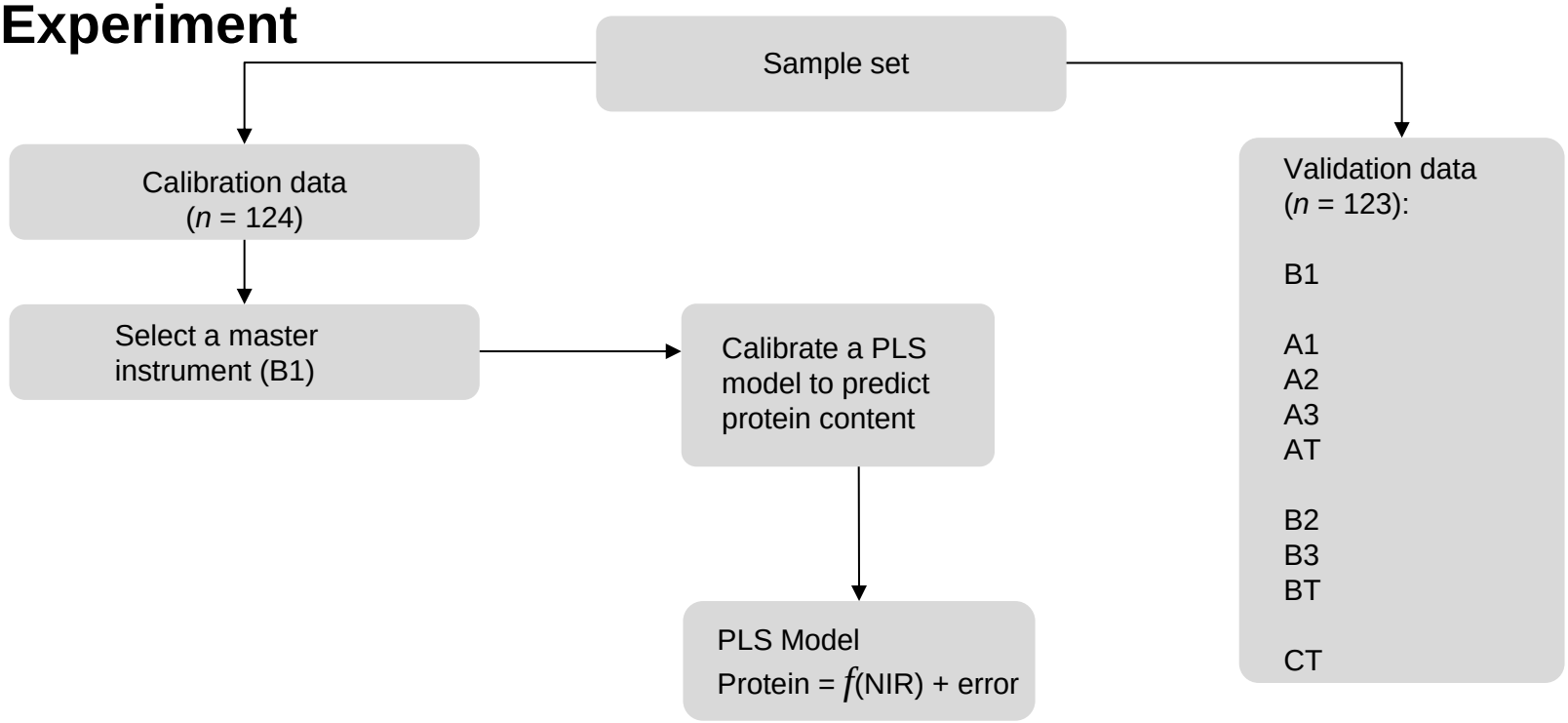
B2
B3
BT

CT

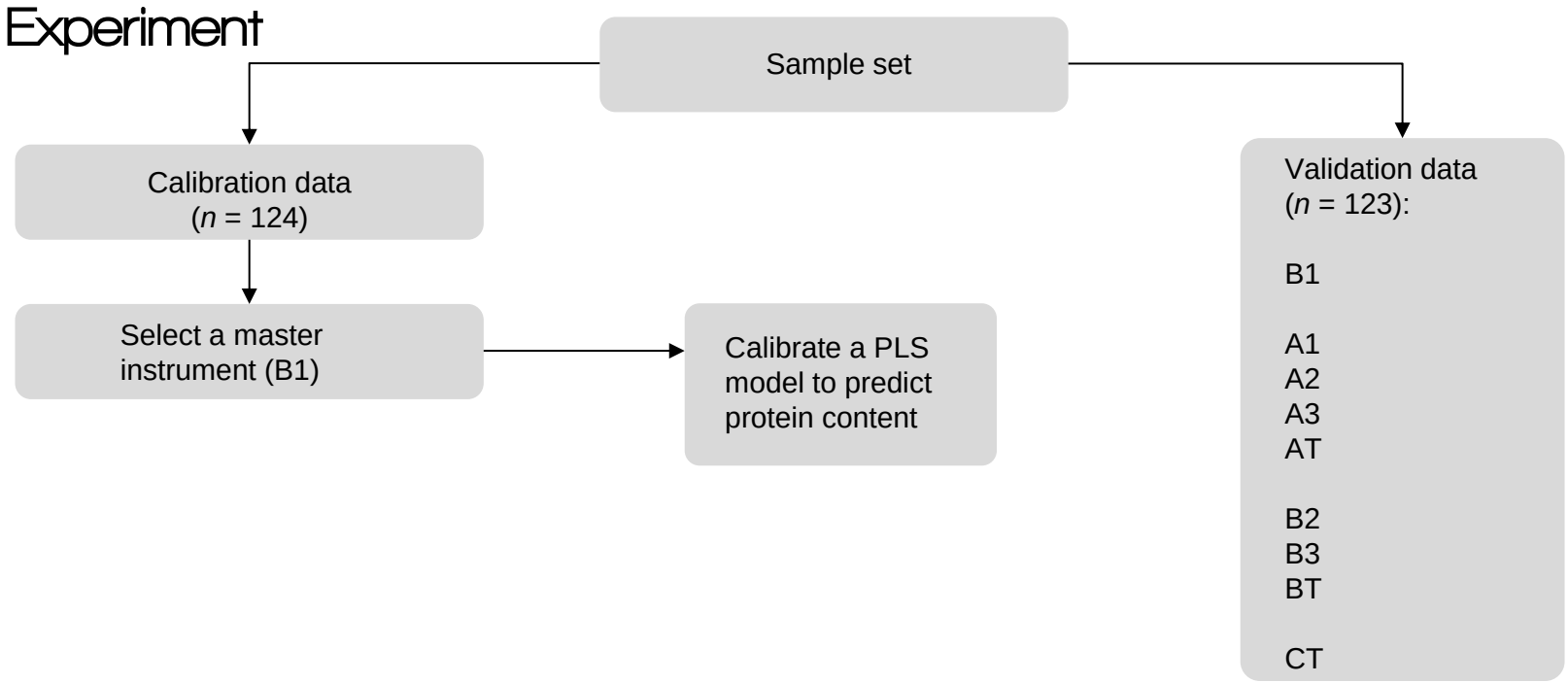
Experiment


Sample set
Calibration data
(*n* = 124)
Select a master instrument (B1)
Calibrate a PLS model to predict protein content
Validation data
(*n* = 123):
B1
A1
A2
A3
AT
B2
B3
BT
CT

Experiment


Sample set
Calibration data
(*n* = 124)
Select a master instrument (B1)
Calibrate a PLS model to predict protein content
PLS Model
Protein = *f*(NIR) + error
Validation data
(*n* = 123):
B1
A1
A2
A3
AT
B2
B3
BT
CT

Experiment


Sample set
Calibration data
(*n* = 124)
Select a master instrument (B1)
Calibrate a PLS model to predict protein content
Validation data
(*n* = 123):
B1
A1
A2
A3
AT
B2
B3
BT
CT

Experiment

Sample set

Calibration data
(n = 124)

Select a master instrument (B1)

Calibrate a PLS model to predict protein content

PLS Model
Protein = f(NIR) + error

Validation data
(n = 123):

B1

A1
A2
A3
AT

B2
B3
BT

CT

Spectra transfer

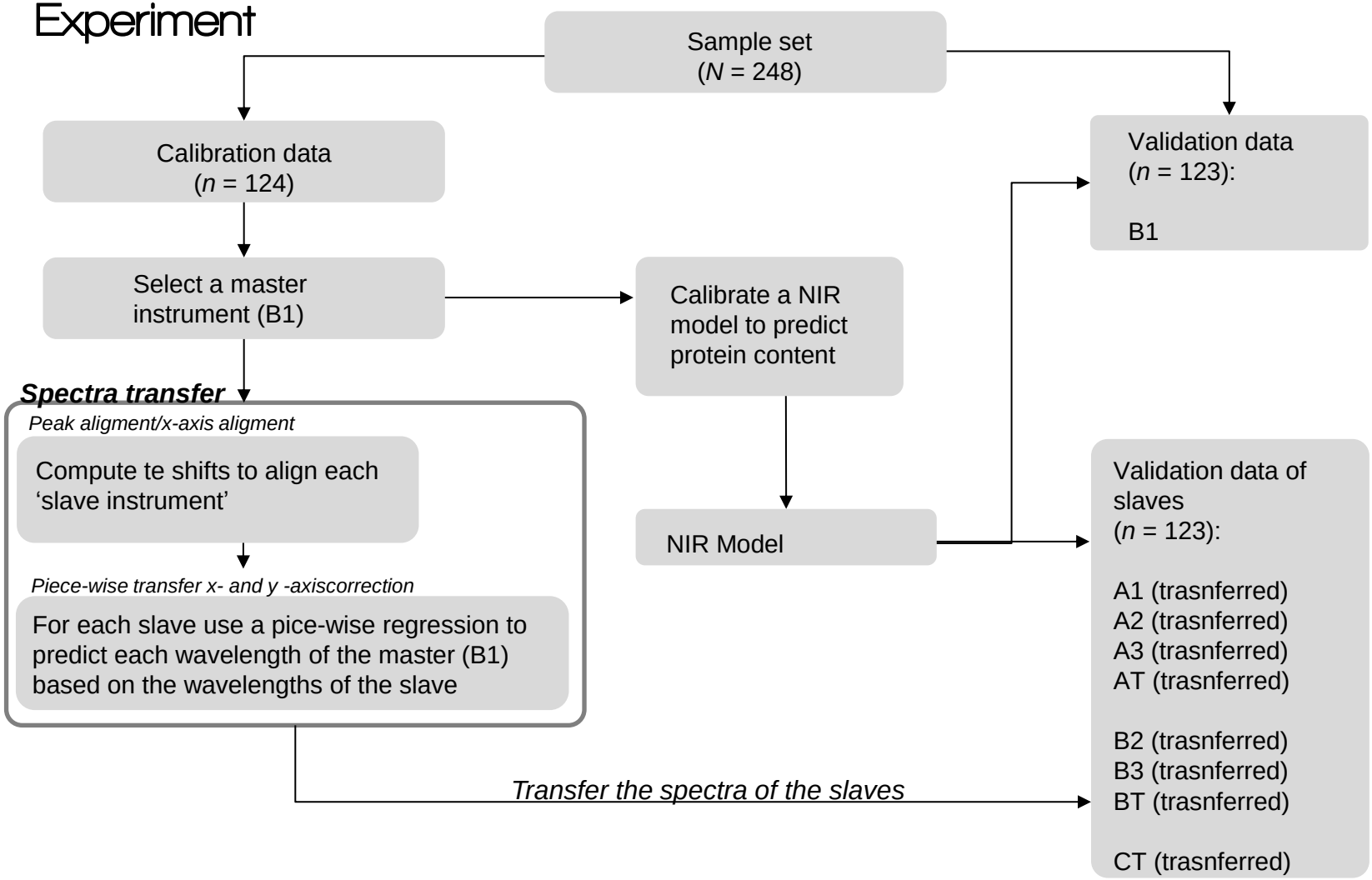
Peak aligment/x-axis aligment

Compute te shifts to align each 'slave instrument'

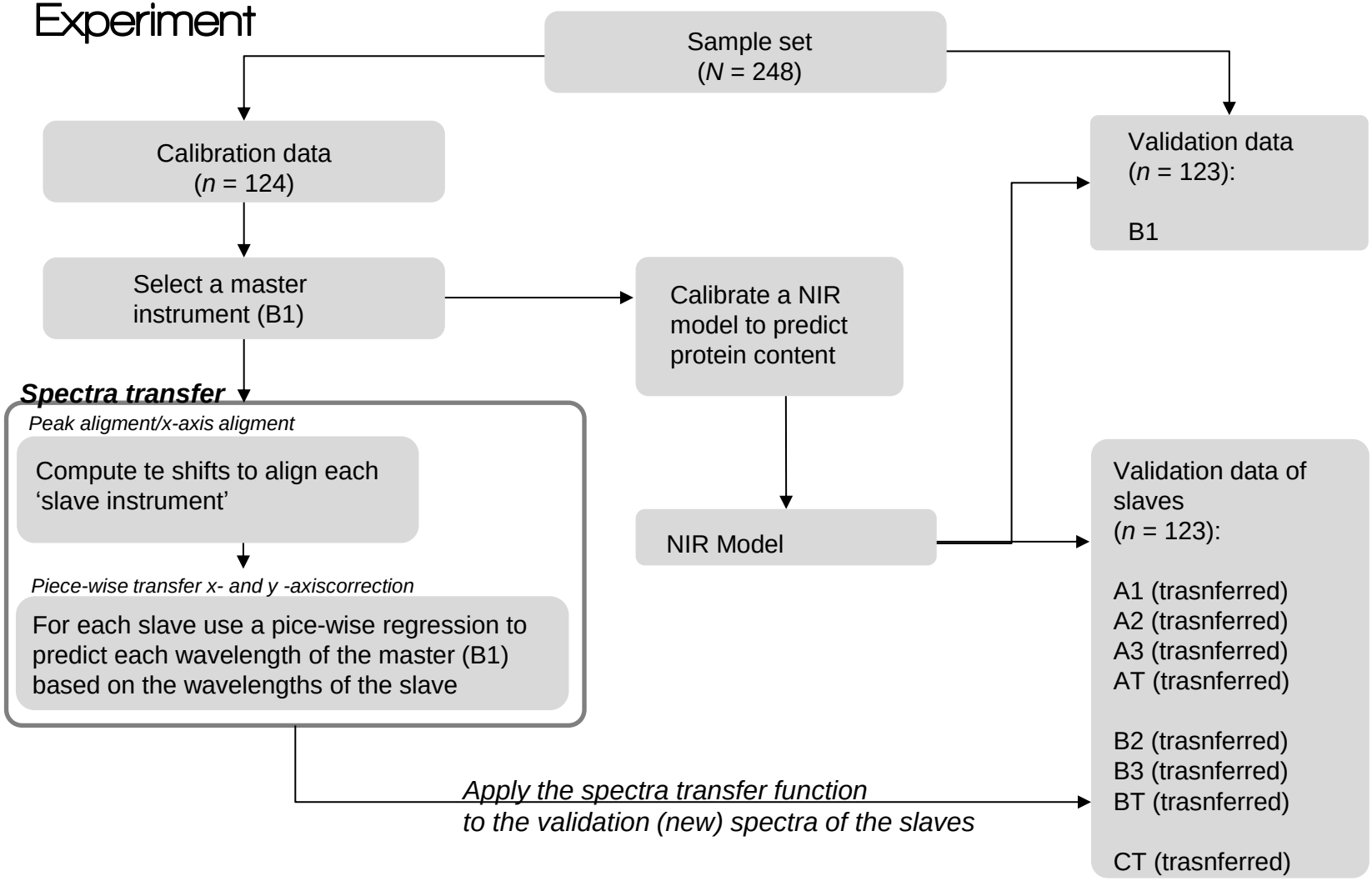
Piece-wise transfer x- and y -axiscorrection

For each slave use a pice-wise regression to predict each wavelength of the master (B1) based on the wavelengths of the slave

Experiment

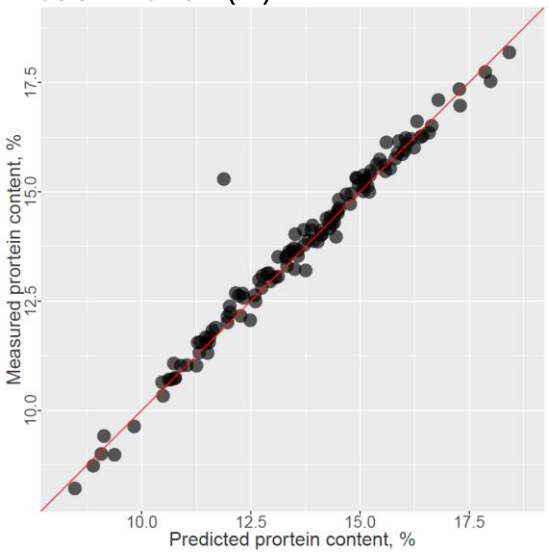

Sample set
(*N* = 248)
Calibration data
(*n* = 124)
Validation data
(*n* = 123):
B1
Select a master instrument (B1)
Calibrate a NIR model to predict protein content
Spectra transfer
Peak aligment/x-axis aligment
Compute te shifts to align each 'slave instrument'
Piece-wise transfer x- and y -axiscorrection
For each slave use a pice-wise regression to predict each wavelength of the master (B1) based on the wavelengths of the slave
NIR Model
Validation data of slaves
(*n* = 123):
A1 (trasnferred)
A2 (trasnferred)
A3 (trasnferred)
AT (trasnferred)
B2 (trasnferred)
B3 (trasnferred)
BT (trasnferred)
CT (trasnferred)
Transfer the spectra of the slaves

Experiment


Sample set
(*N* = 248)
Calibration data
(*n* = 124)
Validation data
(*n* = 123):
B1
Select a master instrument (B1)
Calibrate a NIR model to predict protein content
Spectra transfer
Peak aligment/x-axis aligment
Compute te shifts to align each 'slave instrument'
Piece-wise transfer x- and y -axiscorrection
For each slave use a pice-wise regression to predict each wavelength of the master (B1) based on the wavelengths of the slave
NIR Model
Validation data of slaves
(*n* = 123):
A1 (trasnferred)
A2 (trasnferred)
A3 (trasnferred)
AT (trasnferred)
B2 (trasnferred)
B3 (trasnferred)
BT (trasnferred)
CT (trasnferred)
Apply the spectra transfer function to the validation (new) spectra of the slaves

Results

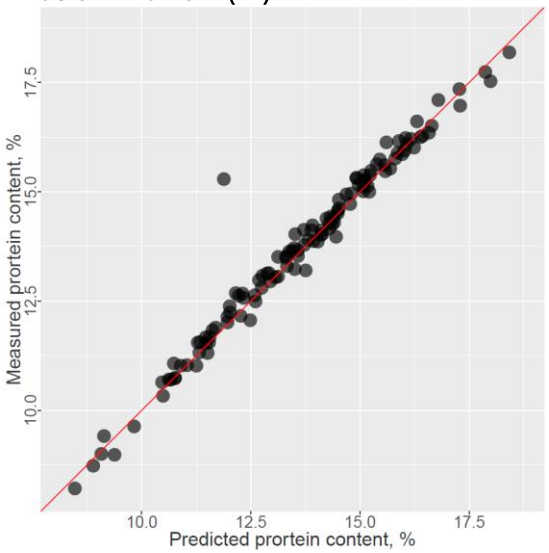
Validation results for the master intrument (B1)



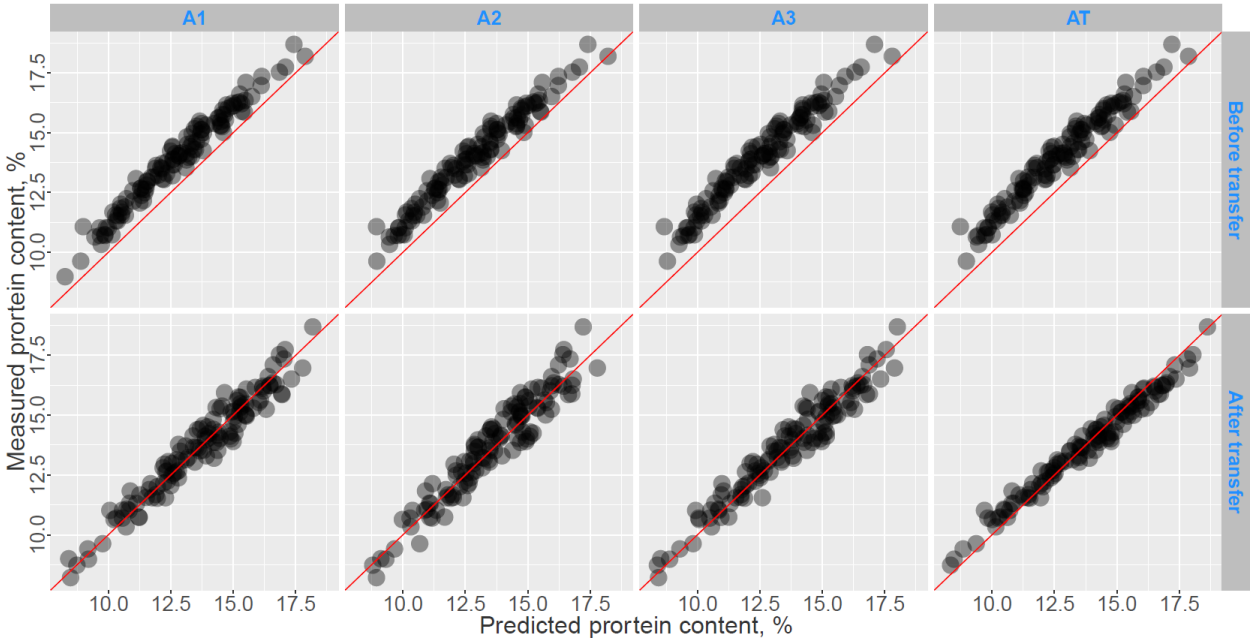
	After transfer		Before transfer	
	R^2	RMSE	R^2	RMSE
A1	0.97	1.20	0.94	0.53
A2	0.97	1.20	0.91	0.63
A3	0.96	1.51	0.94	0.57
AT	0.96	1.33	0.98	0.48
B1	**0.97**	**0.38**	**0.97**	**0.38**
B2	0.93	0.57	0.93	0.57
B3	0.97	0.38	0.97	0.40
BT	0.99	0.31	0.99	0.27
CT	0.86	2.14	0.93	0.55

Results

Validation results for the master intrument (B1)

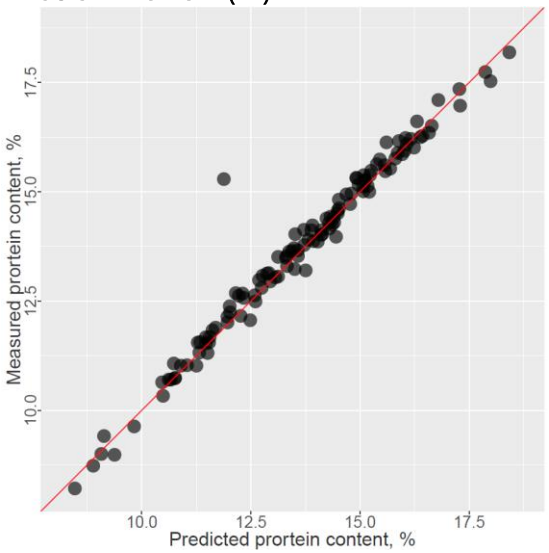


Results for intruments A

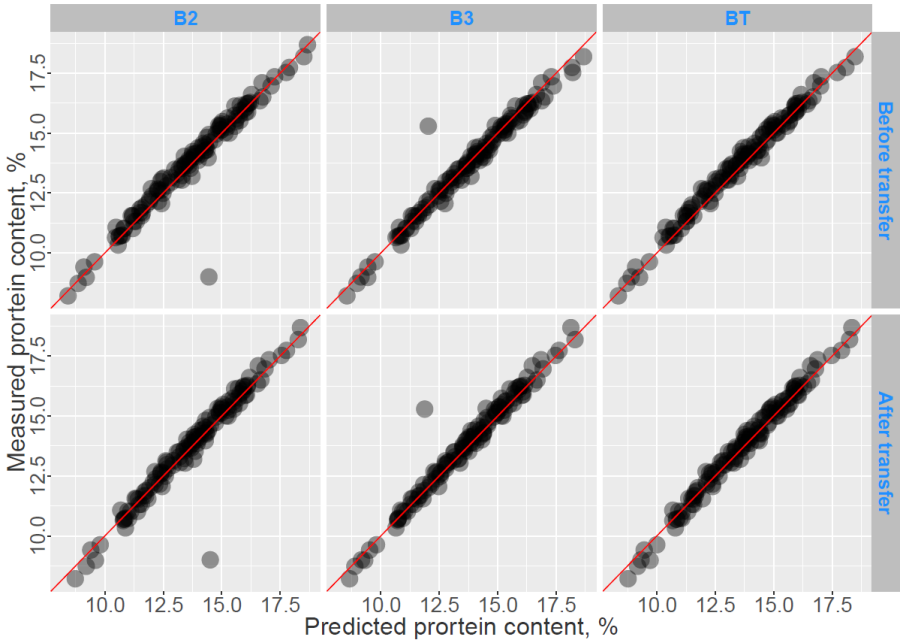


Results

Validation results for the master intrument (B1)

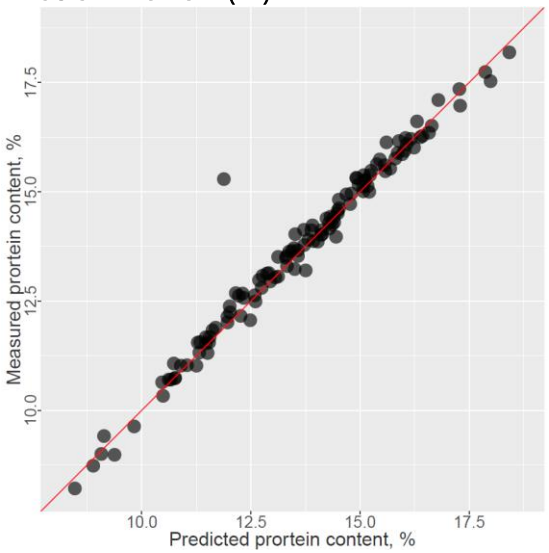
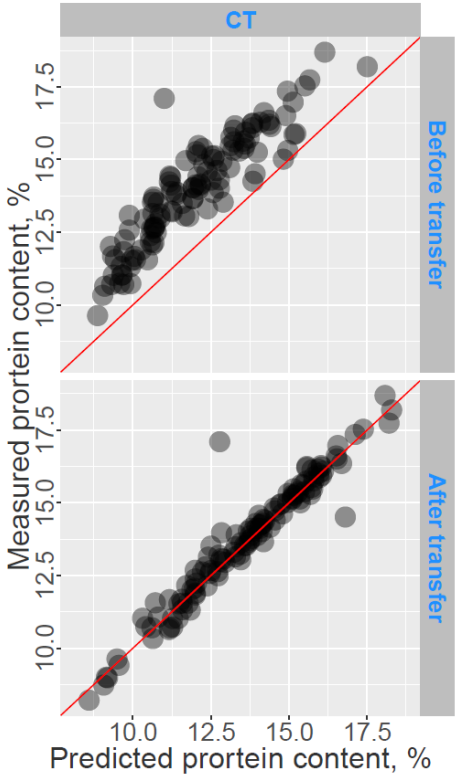


Results for intruments B



Results

Validation results for the master intrument (B1)



Results for intrument C