Cross-validation and performance measures

How to avoid prediction blunders

Filippo Biscarini

(Biostatistician, bioinformatician and quantitative geneticist) **CNR**, Milan (Italy)

What is overfitting?

You may fit a model to your data and then measure the predictive ability (e.g. “accuracy”) on the same data: **would this be correct?**

What is overfitting?

You may fit a deep learning model to your data and then measure the “accuracy”of predictions on the same data: **would this be correct?**

- short answer: **NO!**
- main reason: **overfitting**

What is overfitting?

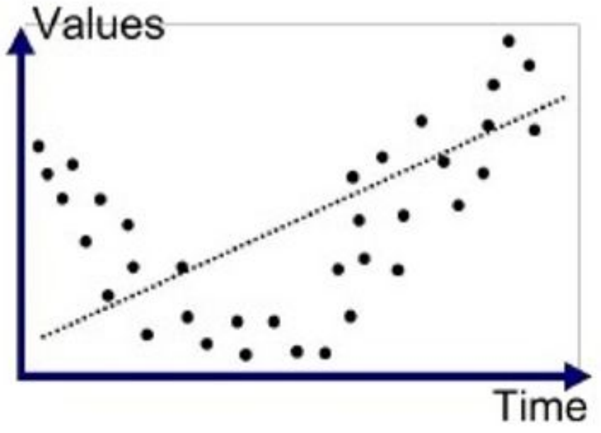
Overfitting:

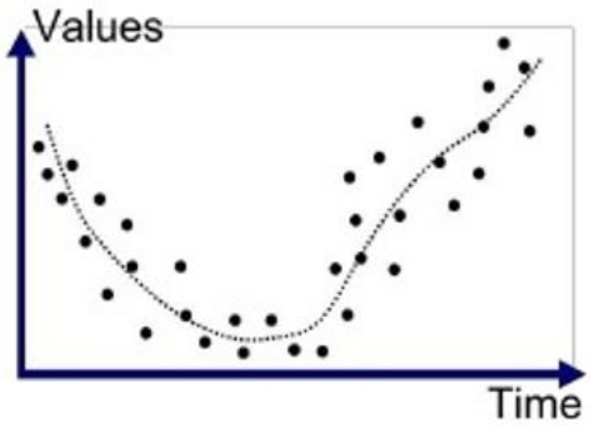
Fitting too well the data: R^2 too large (≃1)

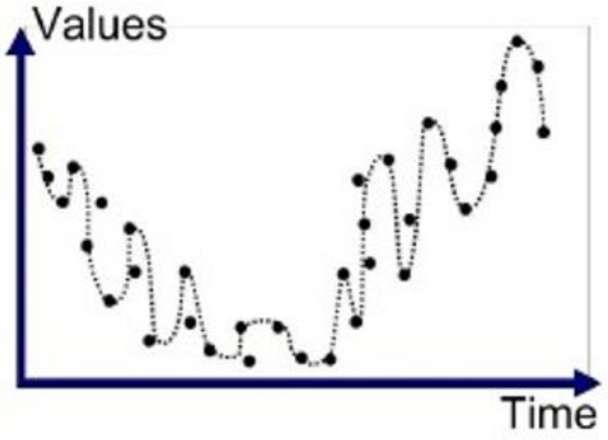
overfitting happens with:

- using the same data to fit the model and make predictions
- overparameterization of the model (e.g. too many effects)
- flexible methods (e.g. polynomial functions, splines, neural networks etc.)

What is overfitting?







Underfitted

Good Fit/Robust

Overfitted

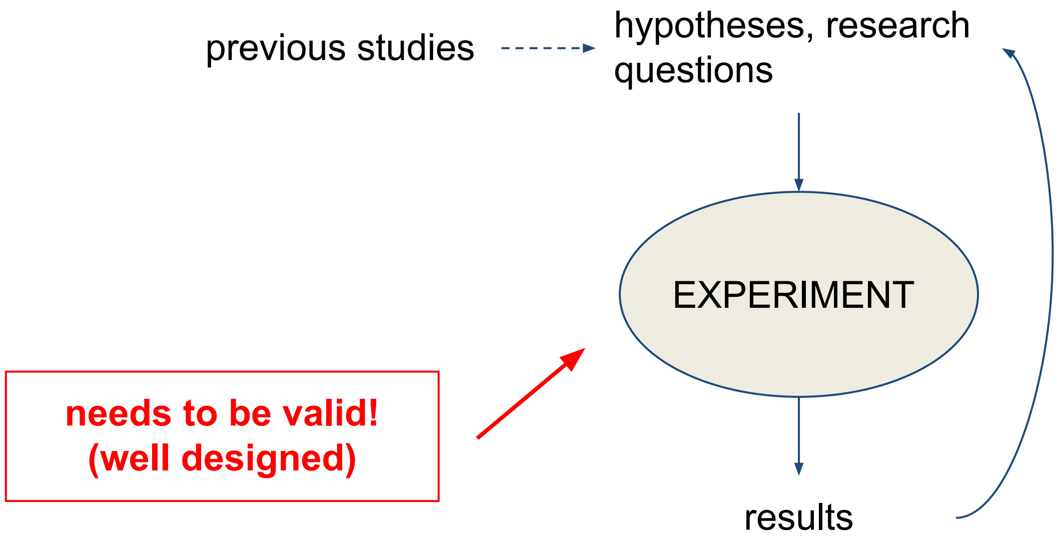
Linear regression

Polynomial (?)

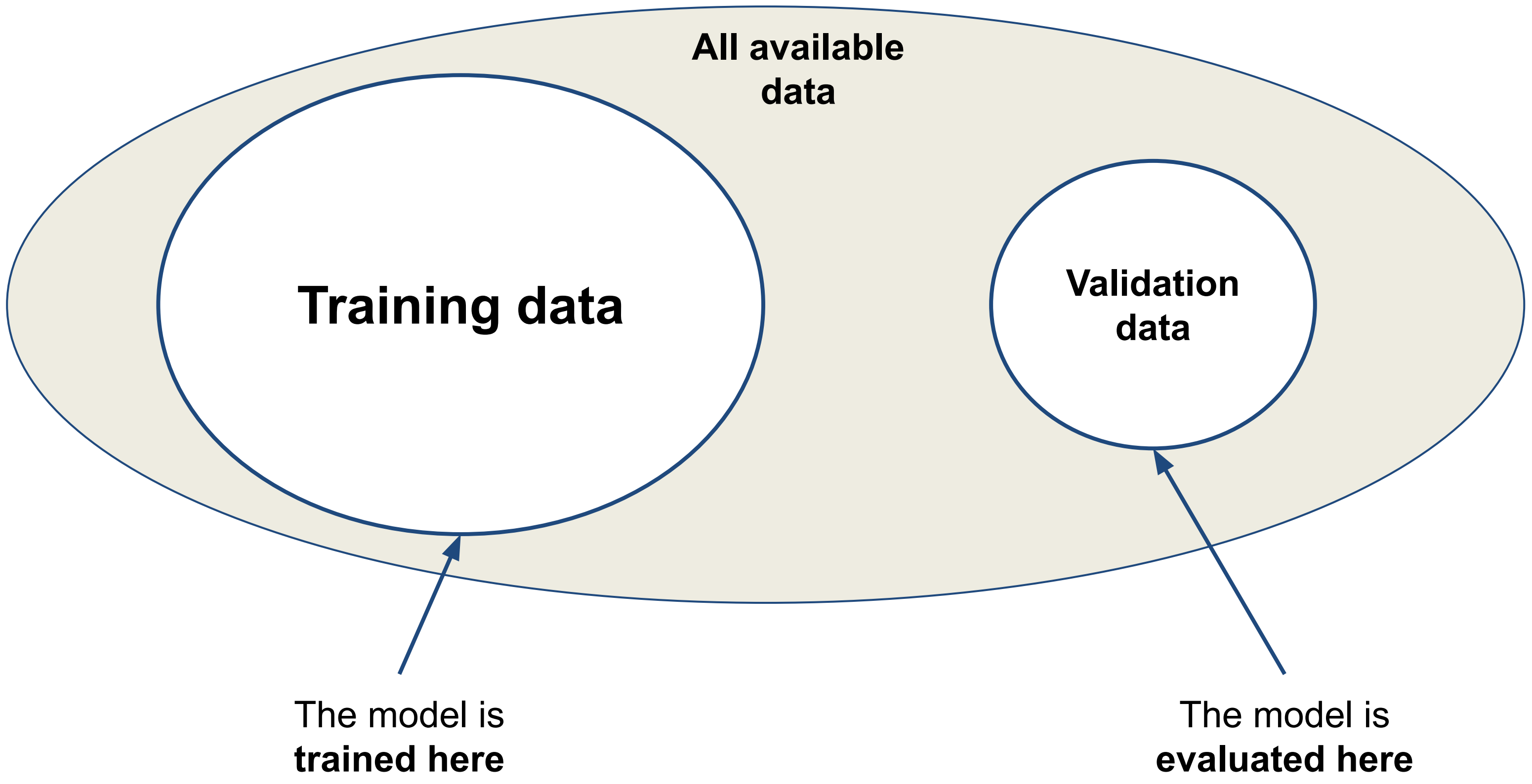
Highly non-linear model

Training and validation sets

“The test of all knowledge is experiment”


previous studies
hypotheses, research questions
EXPERIMENT
results
needs to be valid! (well designed)

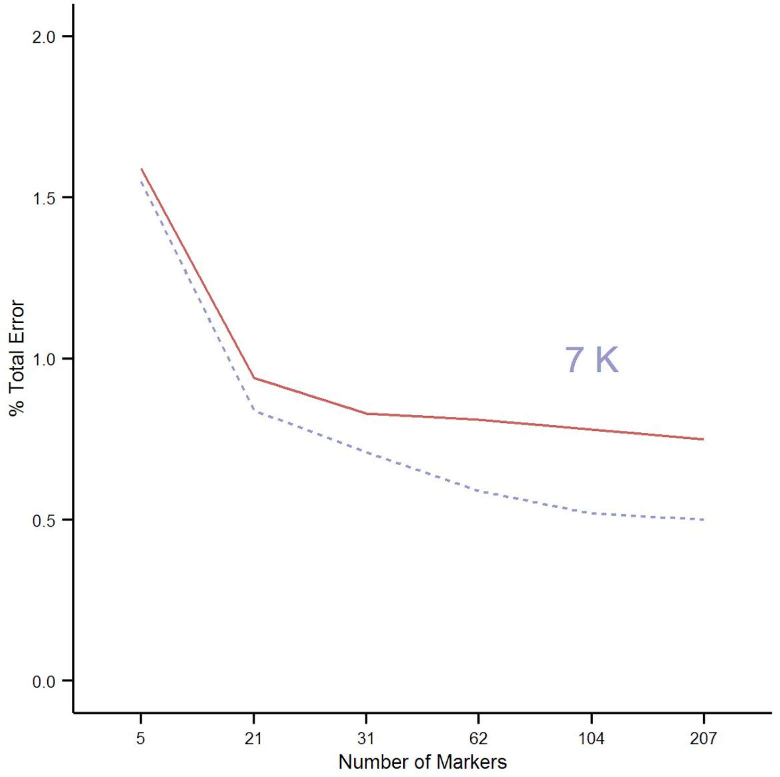
Training and validation sets

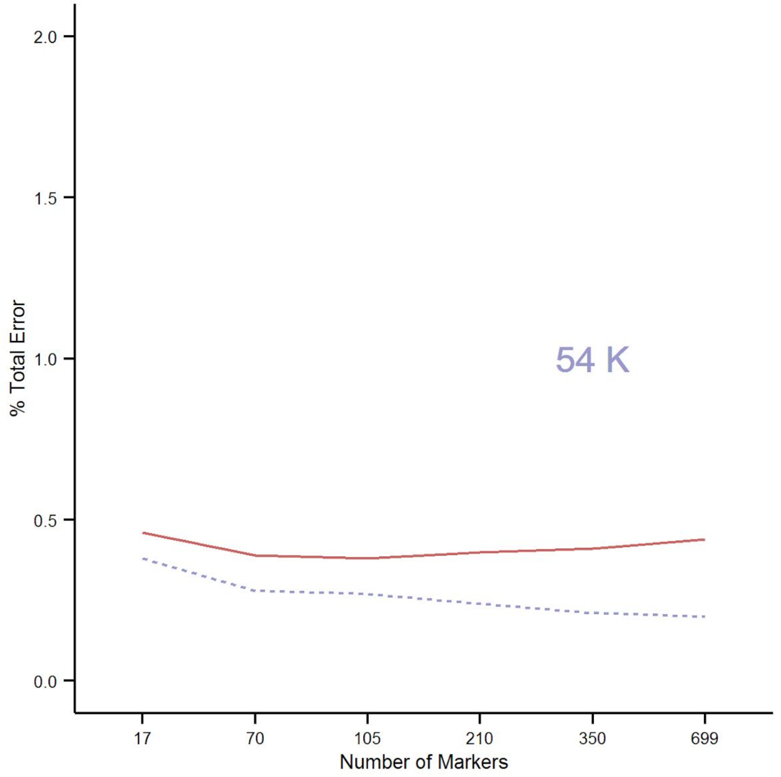

All available data
Training data
Validation data
The model is **trained here**
The model is **evaluated here**

Training and validation sets

- accuracy (model performance) on the training set is “optimistic” (biased upward ← *overfitting*)
- a better estimate of model performance can be obtained from independent data
- usually we are interested in the predictive performance on new data
- accuracy in the validation set is usually lower than in the training set

An example from genomics





And the test set?

- “Test” and “Validation” are often considered and used as synonyms
  - But they are not! (strictly speaking)
- A test set would be a third set
  - Completely new data
  - Used only once when you finished everything else
  - An estimate of performances in real world

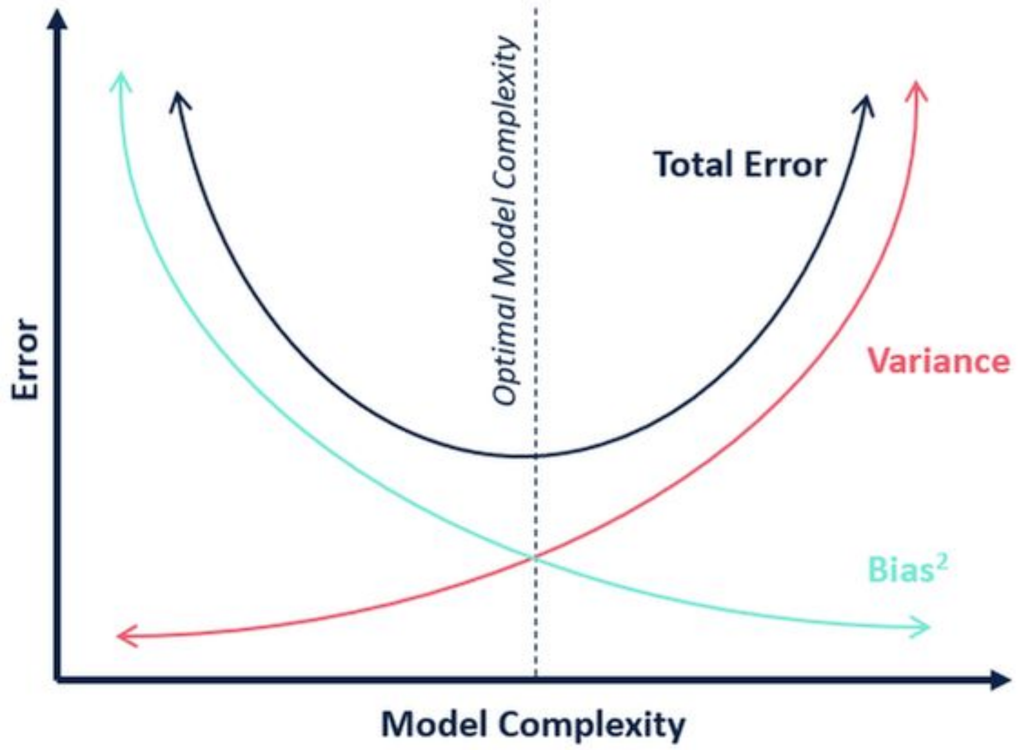
Prediction error

Prediction error

$$E\left(y-\hat{f}(x)\right)=Var\left(\hat{f}(x)\right)+\left[\text{Bias}\left(\hat{f}(x)\right)\right]^2+Var(\epsilon)$$

- **variance** refers to the change of the predictor if estimated using different training data
- **bias** refers to the approximation of a real problem by a simpler model

Bias-variance trade-off



- models with low bias and high variance (e.g. KNN with k=1)
- models with high bias and low variance (e.g. horizontal line crossing the data)
- → find models/methods with both low variance and low bias

Source: https://ai-pool.com/a/s/bias-variance-tradeoff-in-machine-learning

Bias-variance trade-off

Methods

low variance and high bias

- Linear regression
- Logistic regression
- Penalised regression
- SVM (linear kernel)
- Naive Bayes
- etc.

high variance and low bias

- Random Forest
- Boosting
- Polynomial regression
- Regression splines
- GAM
- Deep learning
- KNN
- SVM (Rbf)
- Loess/Lowess (local regression)
- etc.

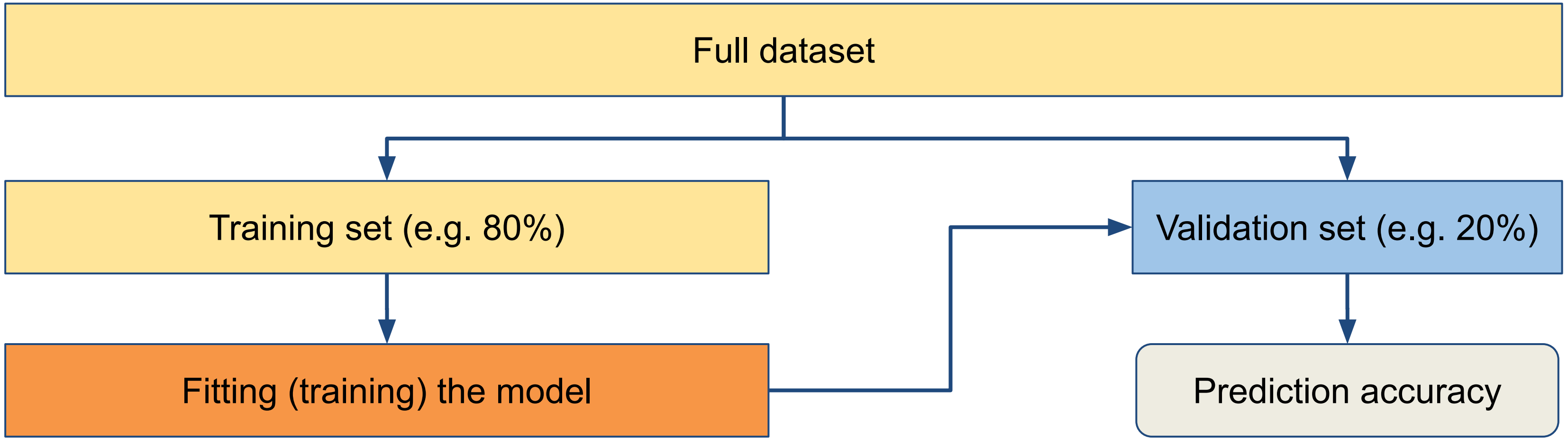
Bias-variance trade-off

Important for:

1. Correctly estimating the performance of a predictive machine
2. Correctly estimating model parameters
3. Selecting between models

Resampling methods

Sampling the training and the validation sets



- To correctly assess the performance of a predictive model we measure it on independent data → validation data
- However we can sample many different training and test sets!

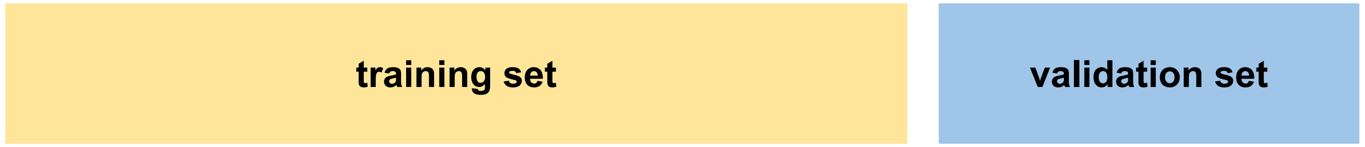
Resampling the data

- Resampling involves **repeatedly sampling** the training and validation datasets: each time, the model is **refitted** in the training set and **evaluated** in the validation set
- You can e.g. estimate the **variability** of a predictive model or the effect of modifying the model or method:
    - **Model assessment**
    - **Model selection**

Resampling the data

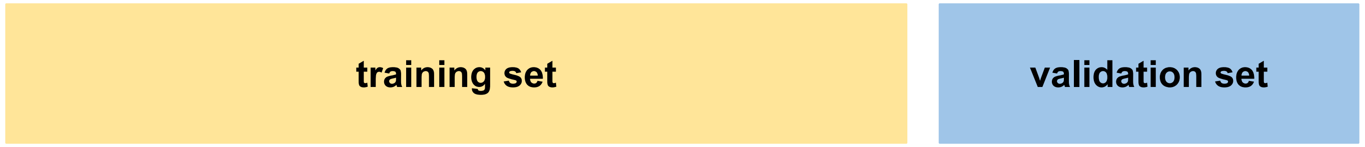
- Several resampling methods exist
- We will examine two such methods:
  1. **validation set approach**
  2. **cross-validation**

The validation set approach



- We split the data in **two random subsets**: training and validation (test) (20%/80%, 30%/70% etc.)
- This is what we already did!
- Repeat this *n times* and you get **robust estimates** of the model performance

The validation set approach

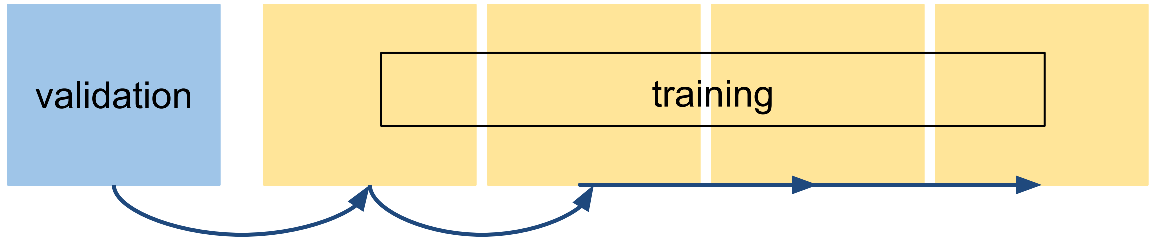


- We split the data in **two random subsets**: training and validation (test) (20%/80%, 30%/70% etc.)
- This is what we already did!
- Repeat this *n times* and you get **robust estimates** of the model performance

Drawbacks:

- **highly variable** (depending on the random partition of the data)
- only a subset of the data is used to train (fit) the model → **potentially underestimate model performance**

k-fold cross-validation



- *k* random partitions of equal size
- each partition in turn is used for validation, the rest for training
- ***k* estimates** of model performance → $$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

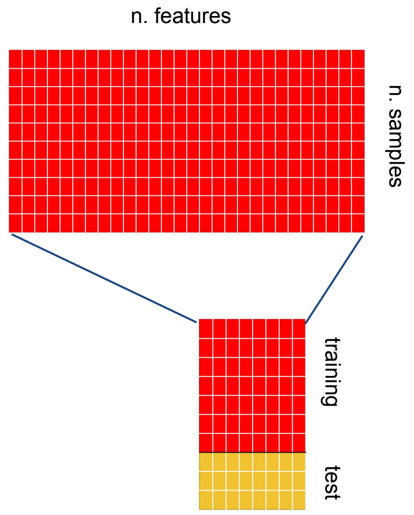
k-fold cross-validation

- Lower variability than the validation set approach
- cross-validation works well in **finding the minimum point** in the estimated test MSE curve → model selection
- In cross-validation each observation/record is used both to train the model and to test it → more data are used here than in the validation set approach → lower bias
- cross-validation is therefore expected to have **both lower variance** and **lower bias** than the validation set approach → more accurate estimate of model performance
- typical values for *k* are **k=5** and **k=10**

Cross-validation: right and wrong

Consider a **regression problem**: **100 samples**, **50,000 features** (variables, e.g. ‘omics data):

- Step 1: Find the 50 features with the **strongest correlation** with the response variable
- Step 2: Apply a **predictive model** (e.g. multiple linear regression) with only these 50 **selected features**

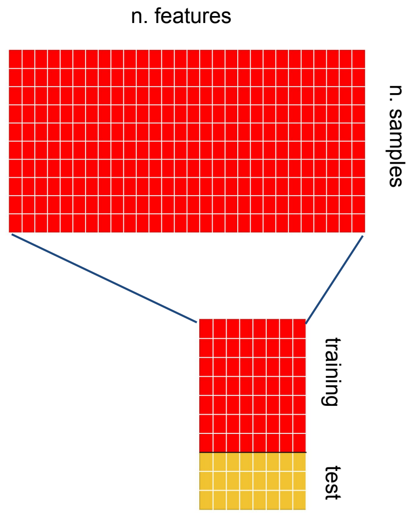


Estimate the **prediction error**: can we apply cross-validation in step 2?

Cross-validation: right and wrong

Consider a **regression problem**: **100 samples**, **50,000 features** (variables, e.g. ‘omics data):

- Step 1: Find the 50 features with the **strongest correlation** with the response variable
- Step 2: Apply a **predictive model** (e.g. multiple linear regression) with only these 50 **selected features**



Estimate the **prediction error**: can we apply cross-validation in step 2? → **NO!**

Cross-validation: right and wrong

Estimate the **prediction error**: can we apply cross-validation in step 2? → **NO!**

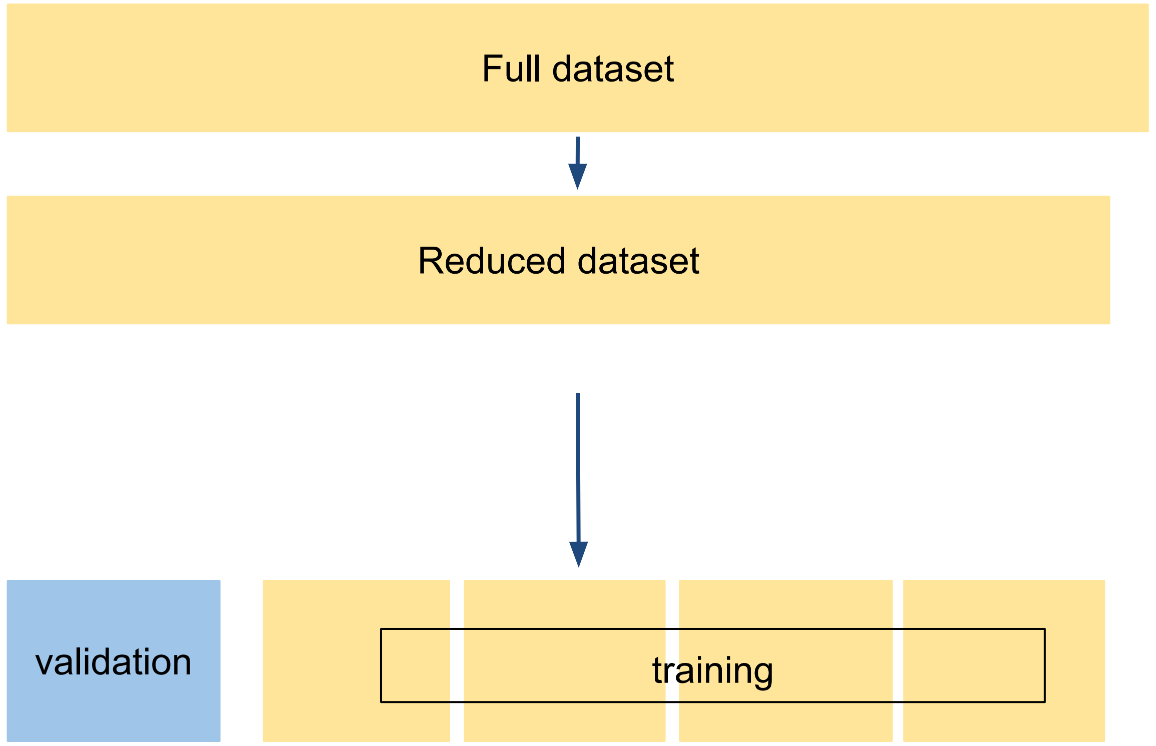
- in Step 1, the **model has already used the response** of the training data
- Features have been **“cherry picked”** based on the data: this is already **training**, and the correlation with the response may be a result of the specific configuration of this dataset (a “quirk” in the data)

Cross-validation: right and wrong

Estimate the **prediction error**: can we apply cross-validation in step 2? → **NO!**

- **Wrong!** → select variables on the whole dataset, then apply cross-validation
- **Right!** → first split the data in training and test sets, then select variables (part of training)

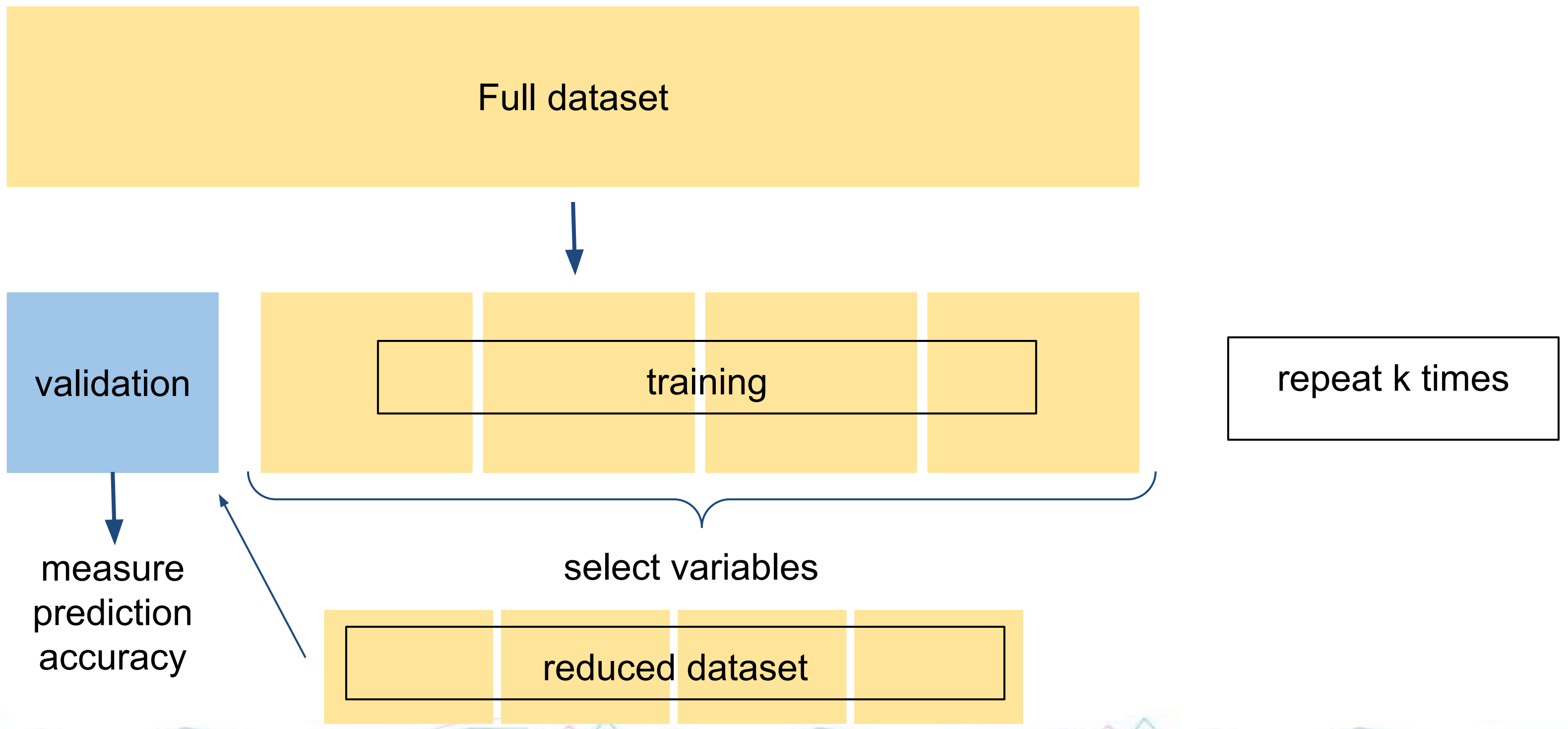
Cross-validation: wrong way



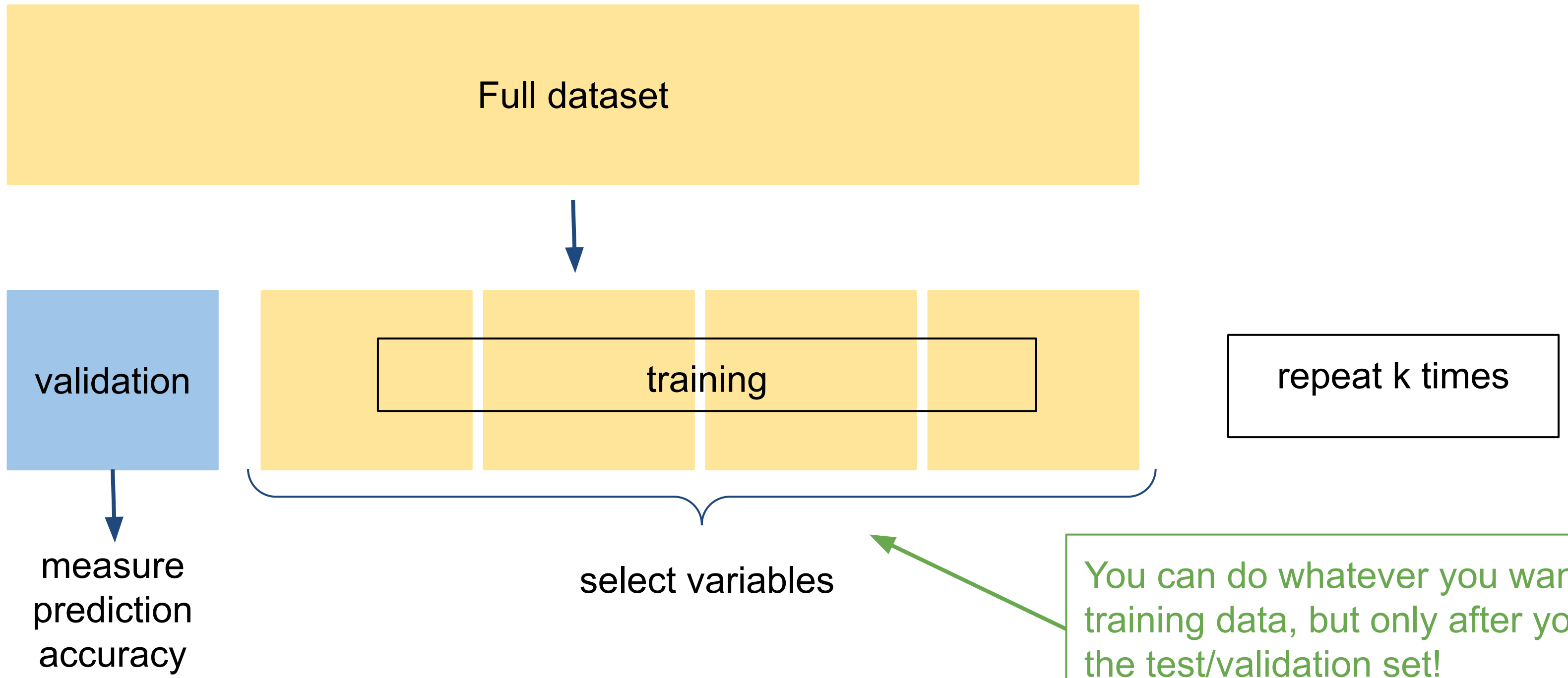
select variables

measure prediction accuracy

Cross-validation: right way



Cross-validation: right way



You can do whatever you want in the training data, but only after you hold out the test/validation set!

Cross-validation with structured data

- So far we have discussed cross-validation based on **random data resampling and data splitting**
- However, data may present **dependence structures** (e.g. geography, time, phylogeny, treatment/group)
- If such structure is not taken into account when designing the cross-validation scheme, there may be **dependencies (connections) between the training and validation data sets**
- When validation data are sampled nearby training data (in the dependence structure: e.g. close in space, time, treatment, genetic relatedness etc.), the **assumption of independence of the validation set is compromised**
- non-independence of validation data makes the models appear better than they really are

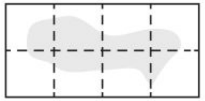
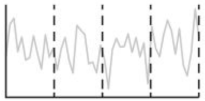
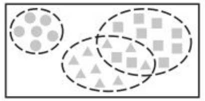
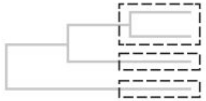
Cross-validation with structured data

- So far we have discussed cross-validation based on **random data resampling and data splitting**
- However, data may present **dependence structures** (e.g. geography, time, phylogeny, treatment/group)
- If such structure is not taken into account when designing the cross-validation scheme, there may be **dependencies (connections) between the training and validation data sets**
- When validation data are sampled nearby training data (in the dependence structure: e.g. close in space, time, treatment, genetic relatedness etc.), the **assumption of independence of the validation set is compromised**
- non-independence of validation data makes the models appear better than they really are

Sometimes you may want to have connections between training and validation data: e.g. genomic selection of young animals/plants related to ancestors whose semen/seeds we want to test

Cross-validation with structured data

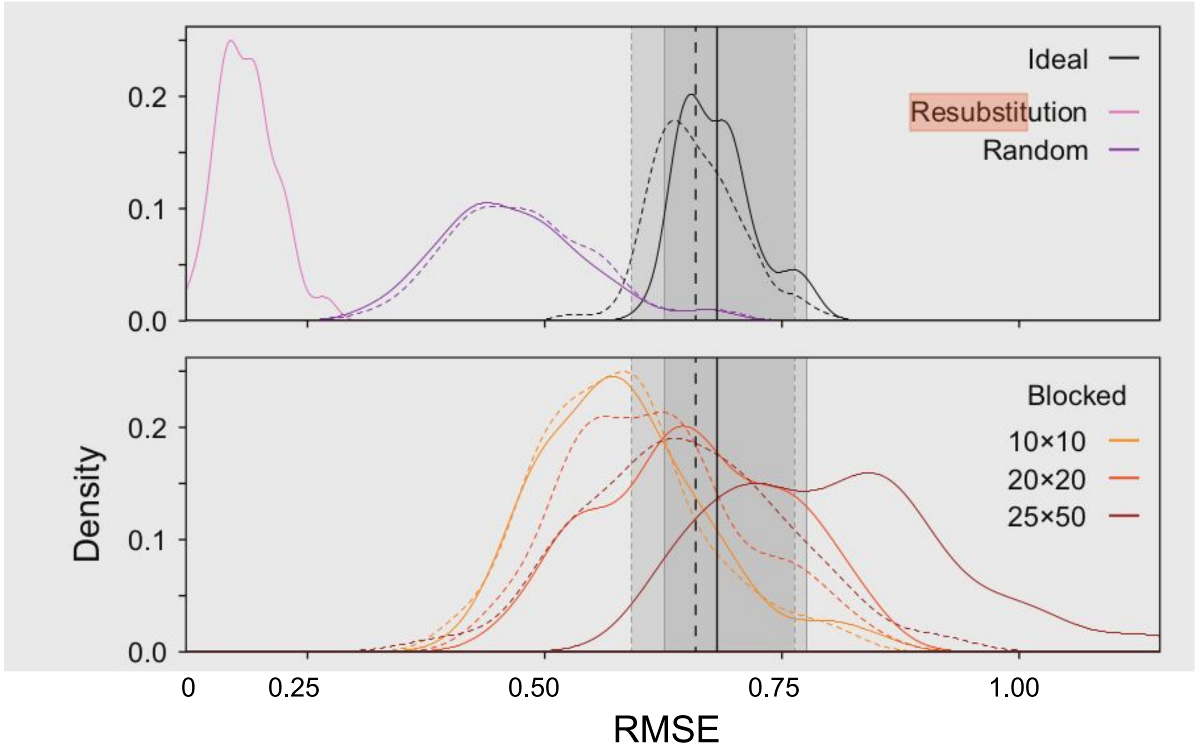
dependencies in the data? → **block cross-validation**

Dependence structure	Parametric solution	Blocking	Blocking illustration
Spatial	Spatial models (e.g.CAR, INLA, GWR)	Spatial	
Temporal	Time-series models (e.g.ARIMA)	Temporal	
Grouping	Mixed effect models (e.g. GLMM)	Group	
Hierarchical / Phylogenetic	Phylogenetic models (e.g. PGLS)	Hierarchical	

force testing more distant records

From Roberts et al. 2016

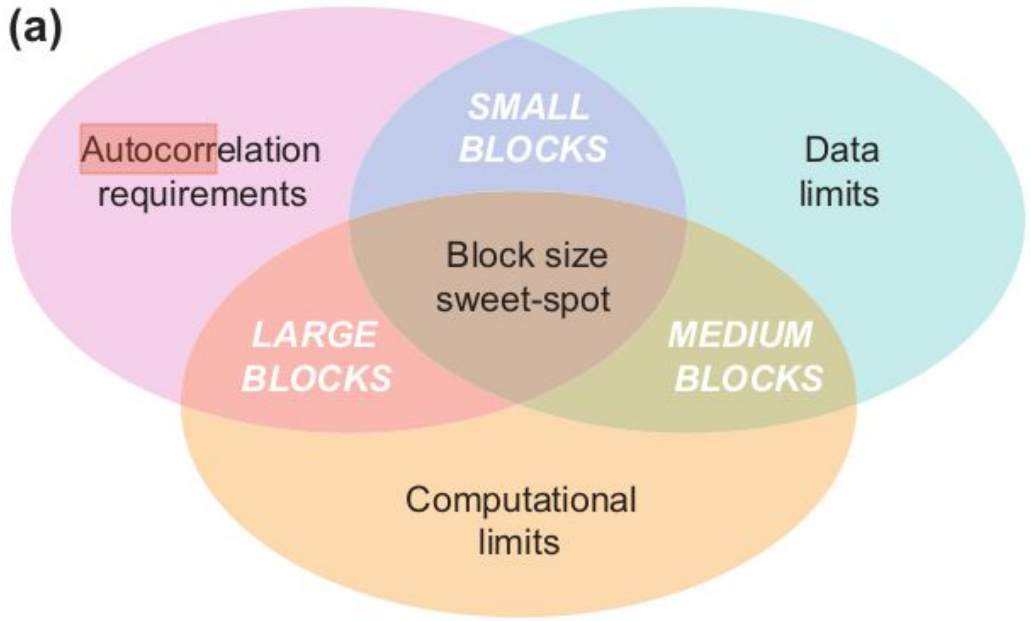
Cross-validation with structured data



simulated data

From Roberts et al. 2016

Cross-validation with structured data



- min block size → ~ extent of autocorrelation
- more data → more blocks / larger blocks
- more computational resources → more blocks / more replicates

From Roberts et al. 2016