# Outline for Part 2

Measuring prediction performance

Sample splitting

Resampling methods

Oslo Bioinformatics Workshop Week 2024

# Statistical principles in machine learning for small biomedical data

Manuela Zucknick

Oslo Centre for Biostatistics and Epidemiology, University of Oslo

manuela.zucknick@medisin.uio.no

December 10, 2024

Some of the figures in this presentation are taken from "Elements of Statistical Learning" (Springer, 2009) and "An Introduction to Statistical Learning, with applications in R" (Springer, 2021) with permission from the authors.

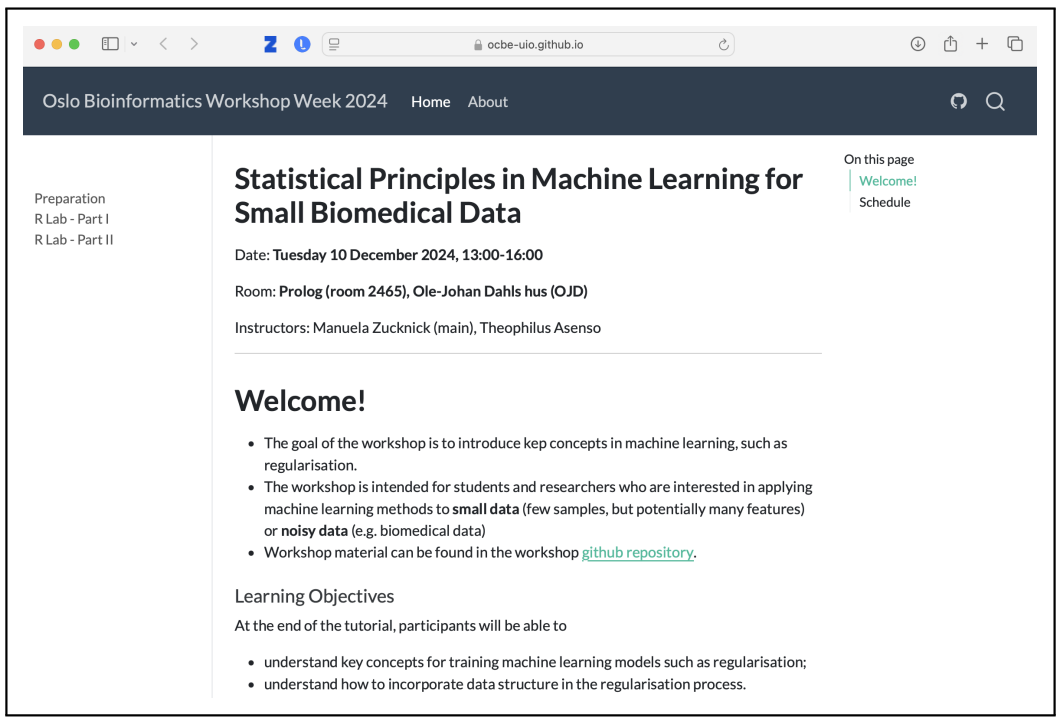
# Schedule for Today

## Schedule

| Time | Topic | Presenter |
|---|---|---|
| Now | Preparations | |
| 13:00 - 14:00 | (Supervised) machine learning with small data | Manuela Zucknick |
| | R lab 1 | Manuela Zucknick |
| 14:15 - 15:15 | Overfitting, regularisation and all that | Manuela Zucknick |
| | R lab 2 | Manuela Zucknick |
| 15:30 - 16:00 | Hierarchical models and structured penalties | Theophilus Asenso |

# Github, Workshop webpage and Posit Cloud project

- Github:
  `https://github.com/ocbe-uio/workshop-stat-highdim/`
- Workshop webpage:
  `https://ocbe-uio.github.io/workshop-stat-highdim/`
- Posit Cloud project: `https://posit.cloud/content/5131383/`



# Some topics for this morning

## Part 1

- What is supervised machine learning?
- What do we mean by small data?
- What can we do to improve ML with small data?
  - Restrict the model space → Regularisation
  - Borrow information → Include known structure in the model

## Part 2

- Overfitting
- Variance vs bias
- Model selection, assessment & validation
- Prediction performance
- Resampling: Cross-validation

# Further reading

**James G, Witten D, Hastie T and Tibshirani R (2021)**, An Introduction to Statistical Learning with Applications in R, Springer, 2nd edition. `https://www.statlearning.com`

**Hastie T, Tibshirani R, Friedman J (2009)**, The Elements of Statistical Learning, Springer, 2nd edition. `https://hastie.su.domains/ElemStatLearn/`

**Holmes S, Huber W (2019)**, Modern Statistics for Modern Biology, Cambridge University Press. `https://www.huber.embl.de/msmb/`

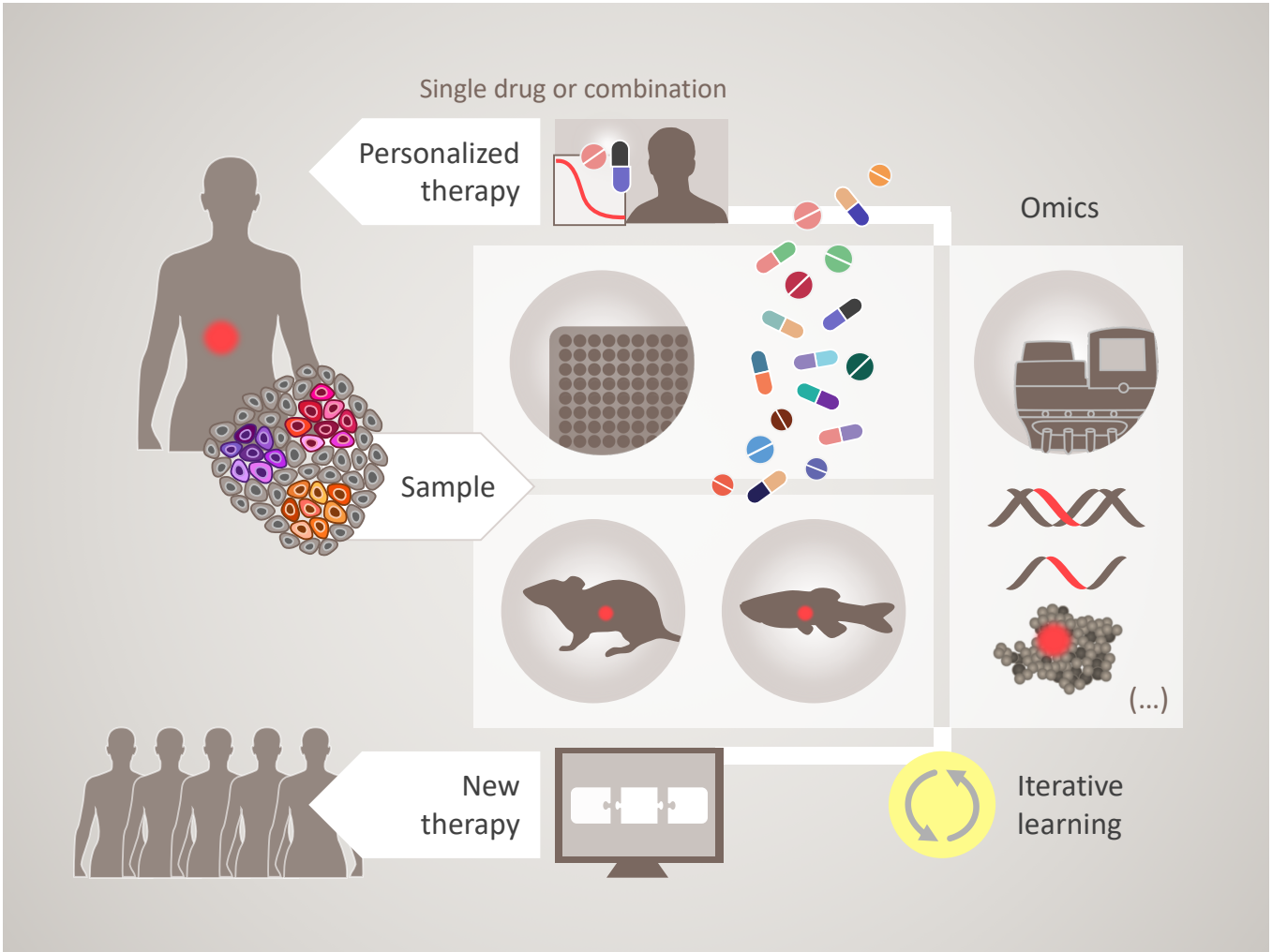
(some chapters on supervised/ unsupervised machine learning)

# Introductory example:

## Integrative omics for personalized cancer therapy

# Personalized cancer therapy

...aims to find the best therapy for each patient based on data about the patient and tumor (e.g. genomic data).



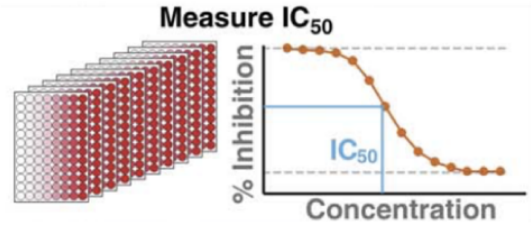

slide by Kjetil Taskén

# Predict sensitivity to multiple drugs **Y** from multi-omics **X**

$$\boxed{\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \epsilon}$$

- **Multivariate Y:**
  Drug dose response

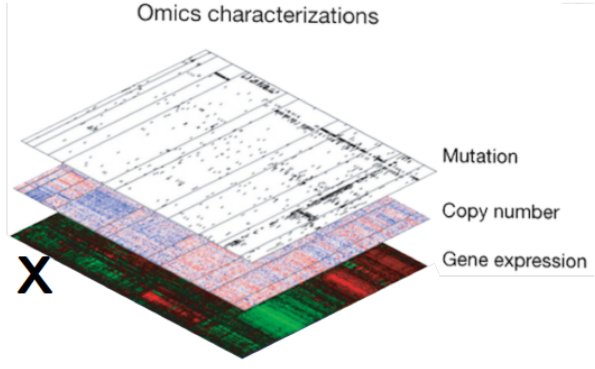
$$n \text{ cell lines} \overbrace{\begin{bmatrix} | & & | \\ \mathbf{y}_{\bullet 1} & \dots & \mathbf{y}_{\bullet m} \\ | & & | \end{bmatrix}}^{\text{drug sensitivity}} = \mathbf{Y}$$




Source: Yang, et al. 2017

- **Heterogeneous X:**
  Integrative omics

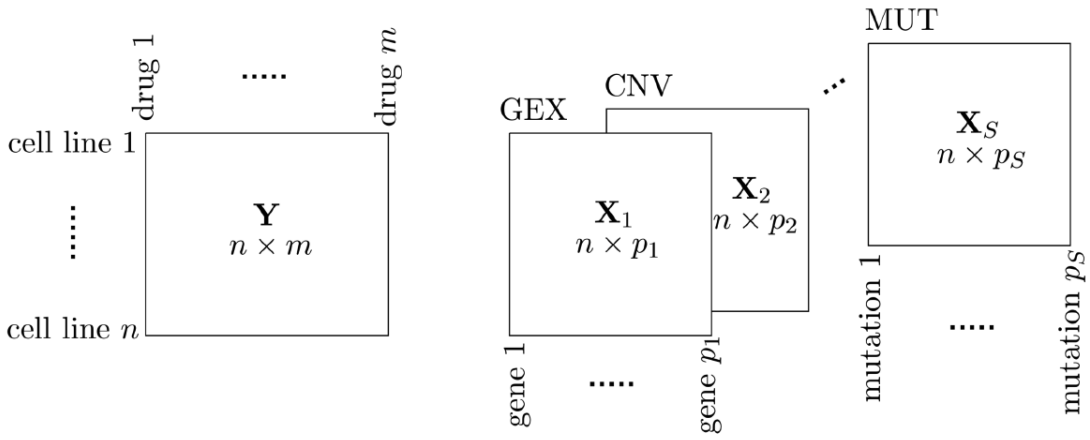
$$n \text{ cell lines} \overbrace{\begin{bmatrix} & \mathbf{X}_1 & | & \mathbf{X}_2 & | & \mathbf{X}_3 & \end{bmatrix}}^{\text{gene expression}\ \ \text{copy number}\ \ \text{mutation}} = \mathbf{X}$$



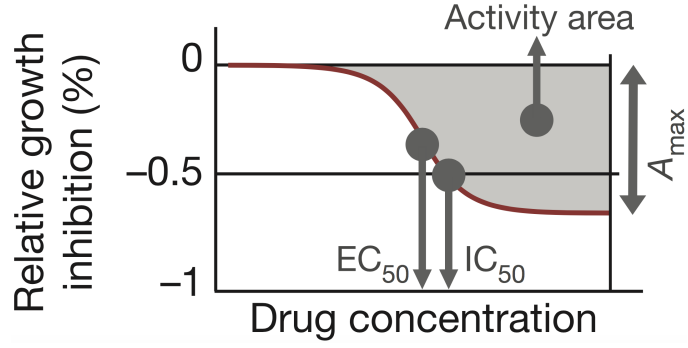

Source: TCGA, 2013

# Challenges and opportunities (1)

- Small sample size
- Several types of input data **X**:
  E.g., gene expression, copy number, mutation
- Multivariate response **Y**



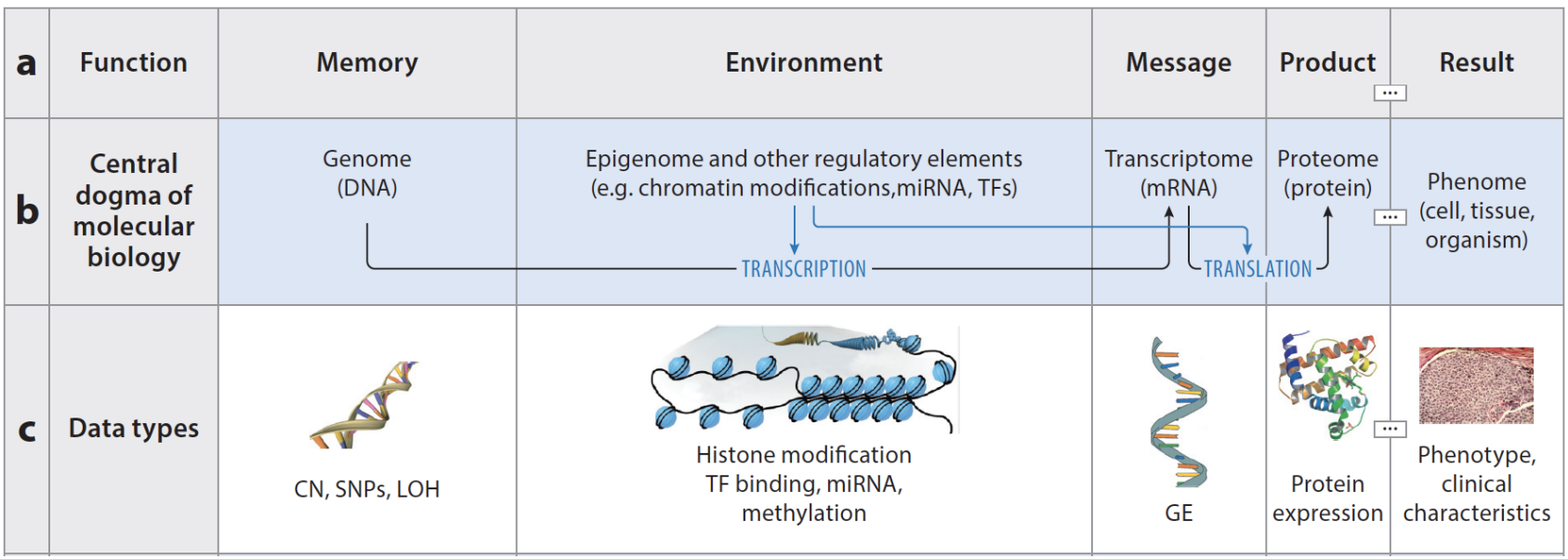

- Unclear how to define **Y**



# Challenges and opportunities (2)

The data are highly structured:

1. In **Y**: relationships between drugs, e.g. due to similar chemical drug composition, same target genes/pathways
2. In **X**: relationships between molecular data sources

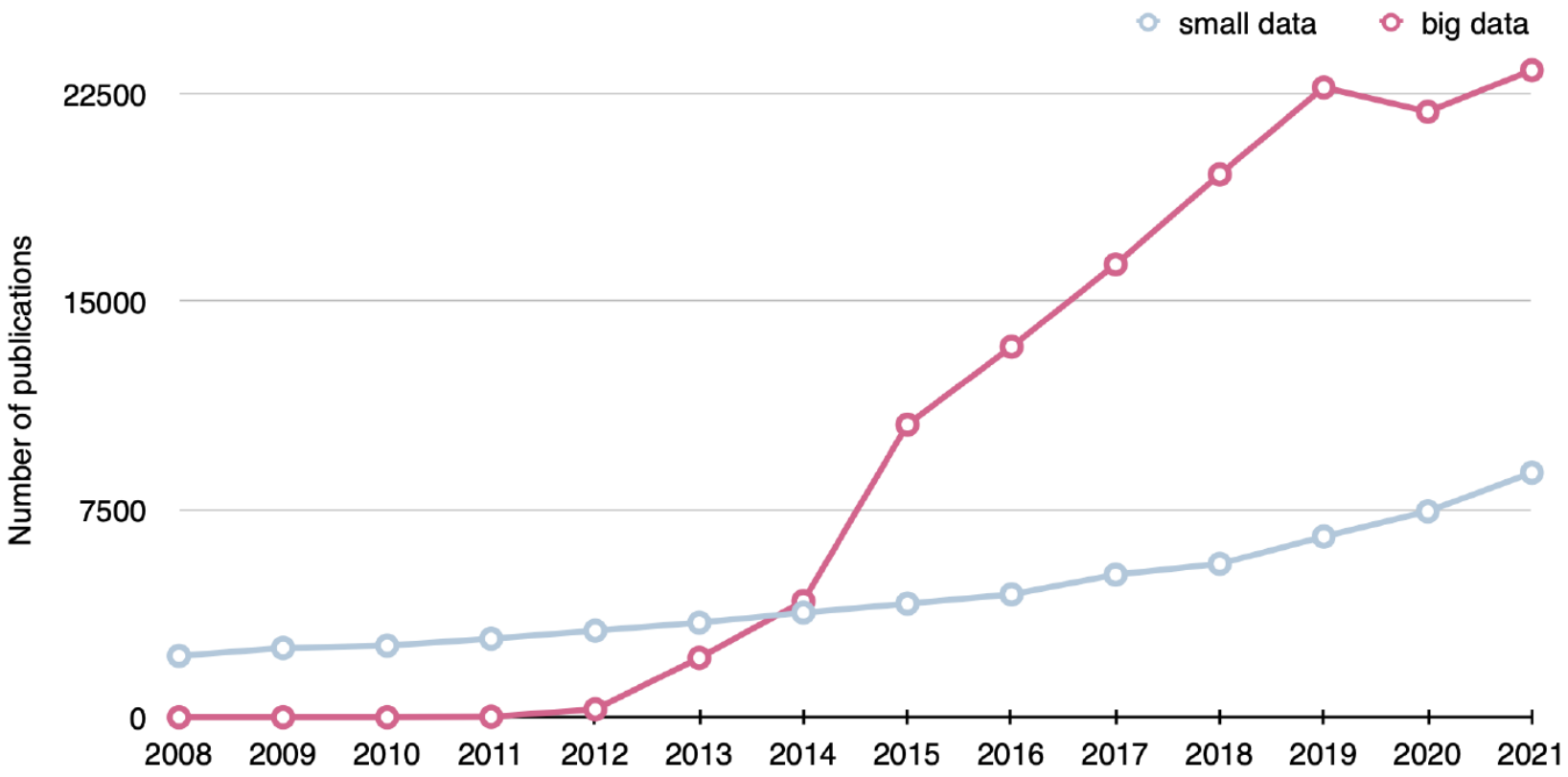



Ickstadt et al. (2018)

# (Supervised) Machine Learning with Small Data

Manuela Zucknick (with slides from Maren Hackenberg)

# Machine learning with small data



# Machine learning with small data

- What do we mean by "small data"?
- Implications for machine learning?

- Aspects when building (multi-omic) machine learning predictors of drug response (e.g. Sammut et al. Nature 2022):

  1. Biological knowledge +
  2. Feature selection +
  3. Prioritisation of accessible data types +
  4. Machine learning algorithms

$\rightarrow$ Develop ML methods that allow us to consider aspects 1 to 3.

# What is supervised machine learning?

**Supervised learning**

refers to the task of inferring a functional relationship between **input data matrix X** (e.g. gene expression array measurements) and **output data vector** $Y$ (= response/ outcome).

The input data are used for predicting the outcome.

$$Y = f_\beta(\mathbf{X}) + \epsilon,$$

where $\epsilon$ captures measurement errors and other discrepancies, e.g. by $\epsilon \sim N(0, \sigma^2 I_n)$.

In classical statistics, this task is usually performed by (generalised) linear regression models.

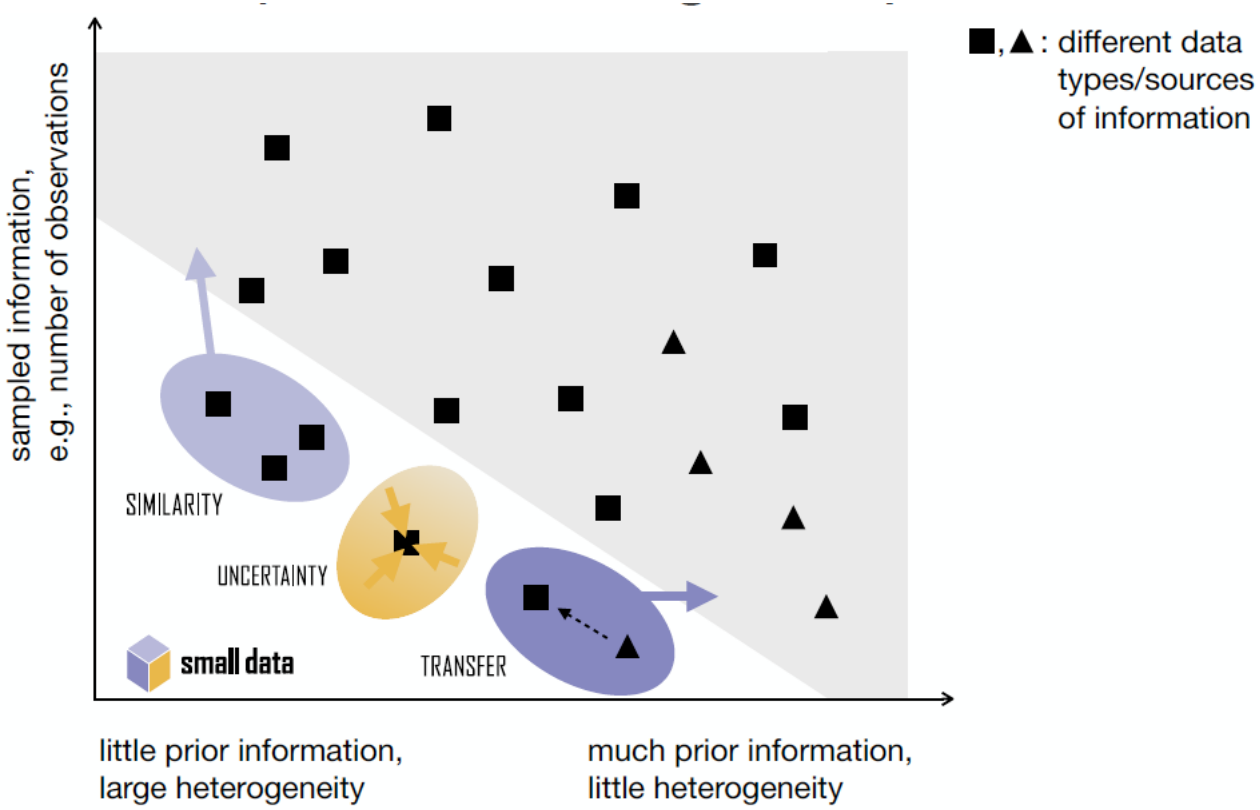
# What do we mean by small data?

- Large p, small n (p>n)
- Potentially, more variables in the model than we have samples

- Classical statistical methods (e.g. linear regression) do not work:
- More parameters (e.g. regression coefficients) to estimate than observations for estimating them
- Even if all parameters can be estimated: Danger of over-fitting

- Example: Predict treatment response using gene expression data (n ~ 100, p ~20000)

# What do we mean by small data?




What can we do?

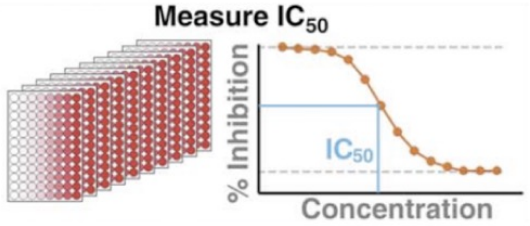
(1) Restrict the model space
(2) Borrow information across observations
(3) Increase sample size ☺

# Predict sensitivity to multiple drugs **Y** from multi-omics **X**

$$\mathbf{Y} = \mathbf{XB} + \epsilon$$

- **Multivariate Y:**
  Drug dose response

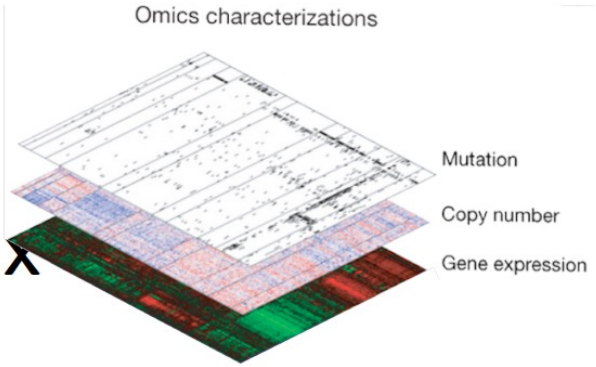
$$n \text{ cell lines} \overbrace{\begin{bmatrix} | & & | \\ \mathbf{y}_{\bullet 1} & \dots & \mathbf{y}_{\bullet m} \\ | & & | \end{bmatrix}}^{\text{drug sensitivity}} = \mathbf{Y}$$




Source: Yang, et al. 2017

- **Heterogeneous X:**
  Integrative omics

$$n \text{ cell lines} \overbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \end{bmatrix}}^{\text{gene expression}\ \ \text{copy number}\ \ \text{mutation}} = \mathbf{X}$$




Source: TCGA, 2013

# What can we do?
# (1) **Restrict the model space**

- (A) Careful feature engineering:
- Preselect variables by biological relevance
- Non-specific filtering, e.g. keep only variables with variance across observations larger than a threshold

- (B) Make use of known structure in the data (biological knowledge)

- (C) Use of regularisation techniques:
  - L1 and L2 penalisation
    - add a penalty term to the loss function to reduce the complexity of the model
    - Bayesian equivalents: restrictions on the prior (Bayesian variable selection)
  - Early stopping
    - train a model iteratively only until the validation error starts to decrease (boosting, neural networks)
  - Dropout regularisation
    - randomly dropping out neurons while training (neural networks) or
    - randomly dropping features when building a regression tree (random forest)

# Penalised regression

- Standard regression cannot deal with $p >> n$:
  - The maximum-likelihood estimate $\hat{\beta} = \arg\max_\beta \ell(\beta)$ does not exist ($\ell$ = log-likelihood).

- **Solution:**
  Penalise the likelihood function by subtracting a penalty term and maximise penalised log-likelihood instead:

$$\hat{\beta} = \arg\max_{\beta}(\ell(\beta) - \lambda||\beta||)$$

  - $\lambda$ is a penalty parameter,
  - $||\beta||$ represents the size of the regression coefficient vector,
  - The larger $\lambda$ is chosen, the more the algorithm is encouraged to find a solution where $||\beta||$ is small $\rightarrow$ shrinkage.

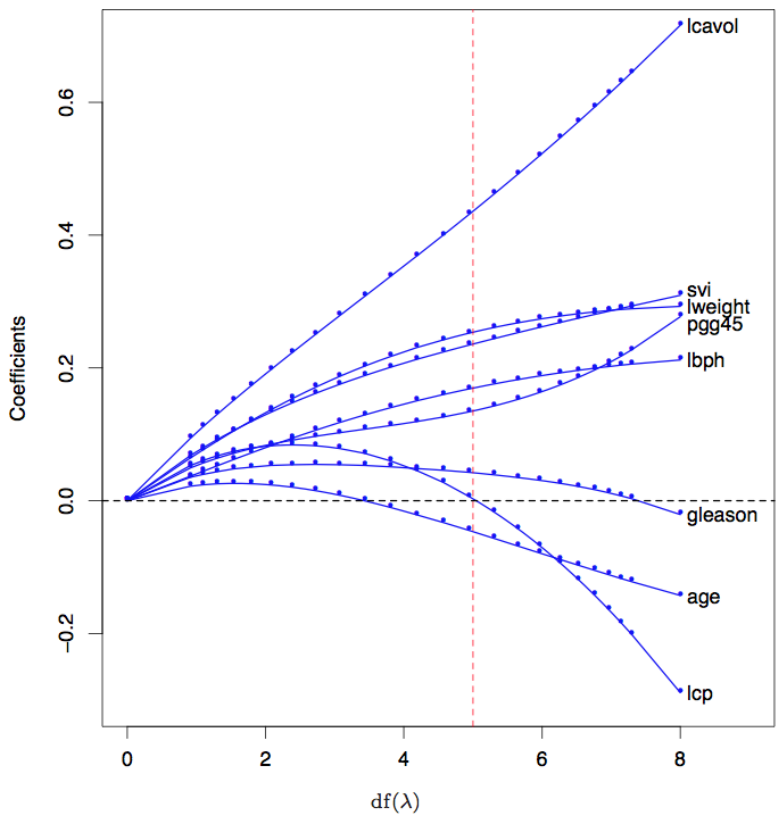
# Penalised regression

- Examples for penalty terms:
  - Ridge regression (Hoerl and Kennard 1970):
    $\lambda||\beta|| := \lambda \sum_{g=1}^{p} \beta_g^2$ $\rightarrow$ **$L_2$ penalty**
  - Lasso regression (Tibshirani 1996):
    $\lambda||\beta|| := \lambda \sum_{g=1}^{p} |\beta_g|$ $\rightarrow$ **$L_1$ penalty**
  - Elastic net (Zou and Hastie 2005):
    Combination of both ridge and lasso penalty:
    $\lambda_1 \sum_{g=1}^{p} |\beta_g| + \lambda_2 \sum_{g=1}^{p} \beta_g^2$
- Advantage of lasso and elastic net:
  Both will produce a sparse solution, where only a few genes have estimate $\hat{\beta}_g \neq 0$.

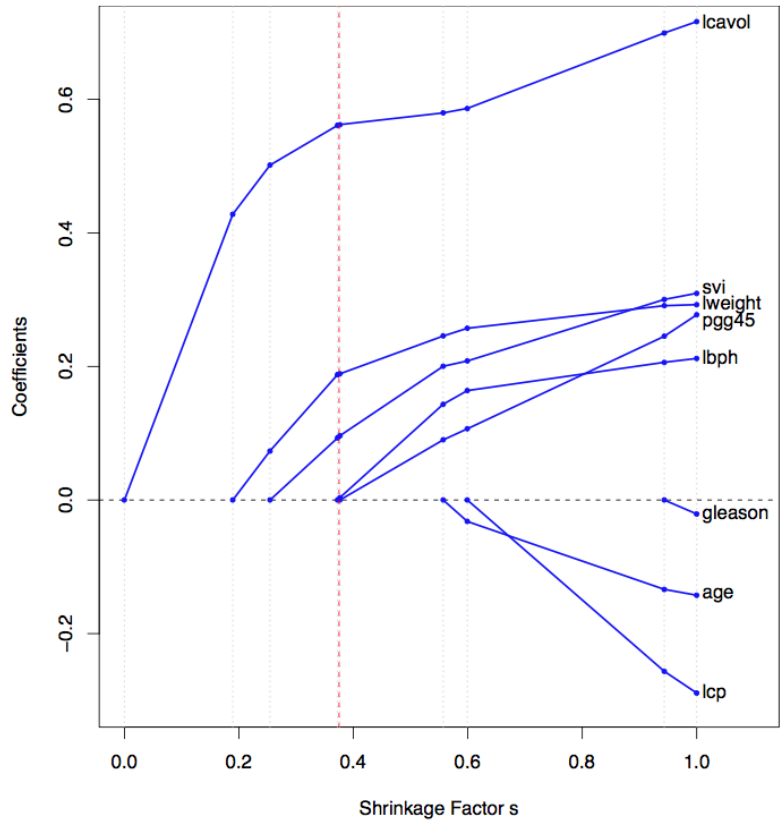
# Penalised regression

Examples for coefficient paths relative to penality $\lambda$:

## Ridge regresson




## Lasso regression




Hastie et al. (2009), Figures 3.8 and 3.10

# Penalised regression

- Ridge regression $L_2$: shrinks all coefficients to small, but non-zero values.
- Lasso regression $L_1$: shrinks some coefficients to exactly zero.
- Elastic net: mixture of the two: does shrink some coeffients to exactly zero. Keeps more variables if there is correlation.

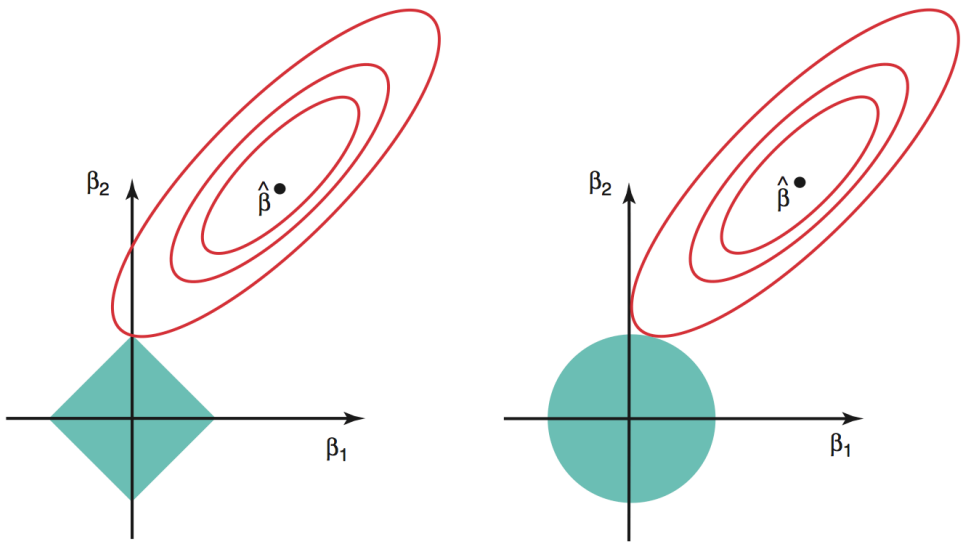



**FIGURE 6.7.** *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

# Different penalties for different types of data

Assume two data matrices **X** and **Z**:

$$Y = \mathbf{X}\beta + \mathbf{Z}\gamma + \epsilon$$

- Mandatory covariates: Do not penalise the parameters $\gamma$:

$$\ell_{\text{pen}}(\beta, \gamma) = \ell(\beta, \gamma) - \lambda||\beta||$$

e.g. with R packages `glmnet` or `penalized`

- Several types of molecular data sets:
Allow different penalties for $\beta$ and $\gamma$:

$$\ell_{\text{pen}}(\beta, \gamma) = \ell(\beta, \gamma) - \lambda_\beta||\beta|| - \lambda_\gamma||\gamma||$$

e.g. with R packages `GRridge` (Van de Wiel *et al.*, 2016)
`http://www.few.vu.nl/~mavdwiel/grridge.html`)

# Different penalties for different types of data

Assume two data matrices **X** and **Z**:

$$Y = \mathbf{X}\beta + \mathbf{Z}\gamma + \epsilon$$

- Several types of molecular data sets:
- Alternative: **Combine all data and use one penalty**, after scaling all features to unit variance to ensure that the data sources are treated equally.
- Example: Elastic Net models in Barretina et al. (2012)

# What can we do?
# (2) Borrow information

- Borrow information across observations in the data set
- If there is correlation, include this in your model
  - between variables (e.g. MRF prior for defining which variables to include together)
  - between samples (covariance matrix)

- Borrow information from external knowledge
  - E.g., use pathways to determine which genes should be included together

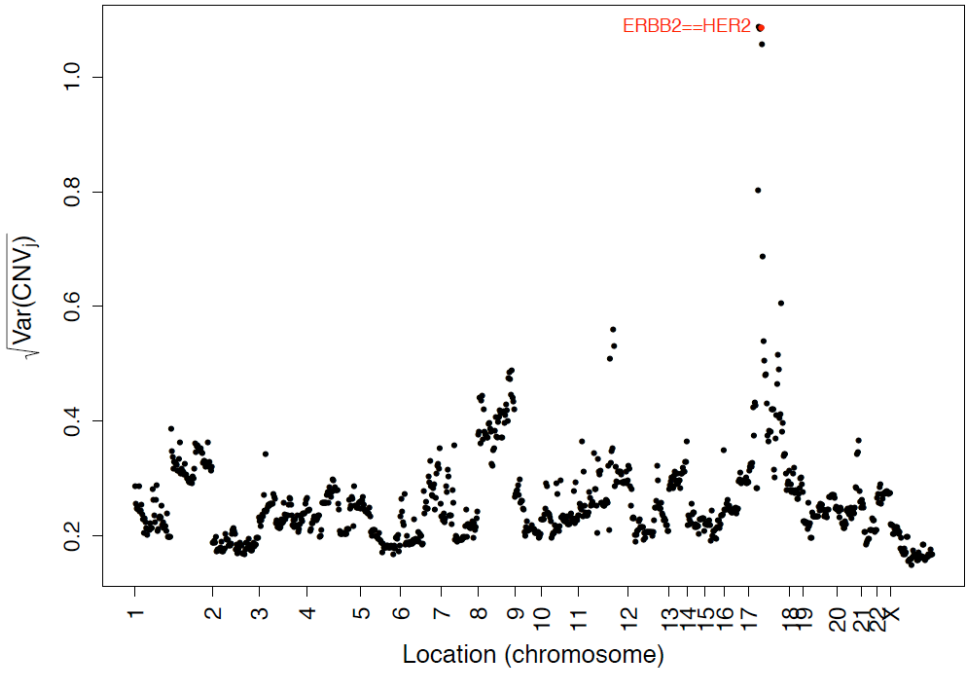
- Borrow information across data sets: transfer learning

# Make use of external (biological) knowledge

- (1) Use known relationships with one data source (CNV) to guide the variable selection in another (gene expression)
- (2) Combine the data-driven ML approach with knowledge-driven mechanistic modelling
- (3) Make use of correlations in the data
  - between input variables - to restrict the model space
  - between response variables - to borrow information

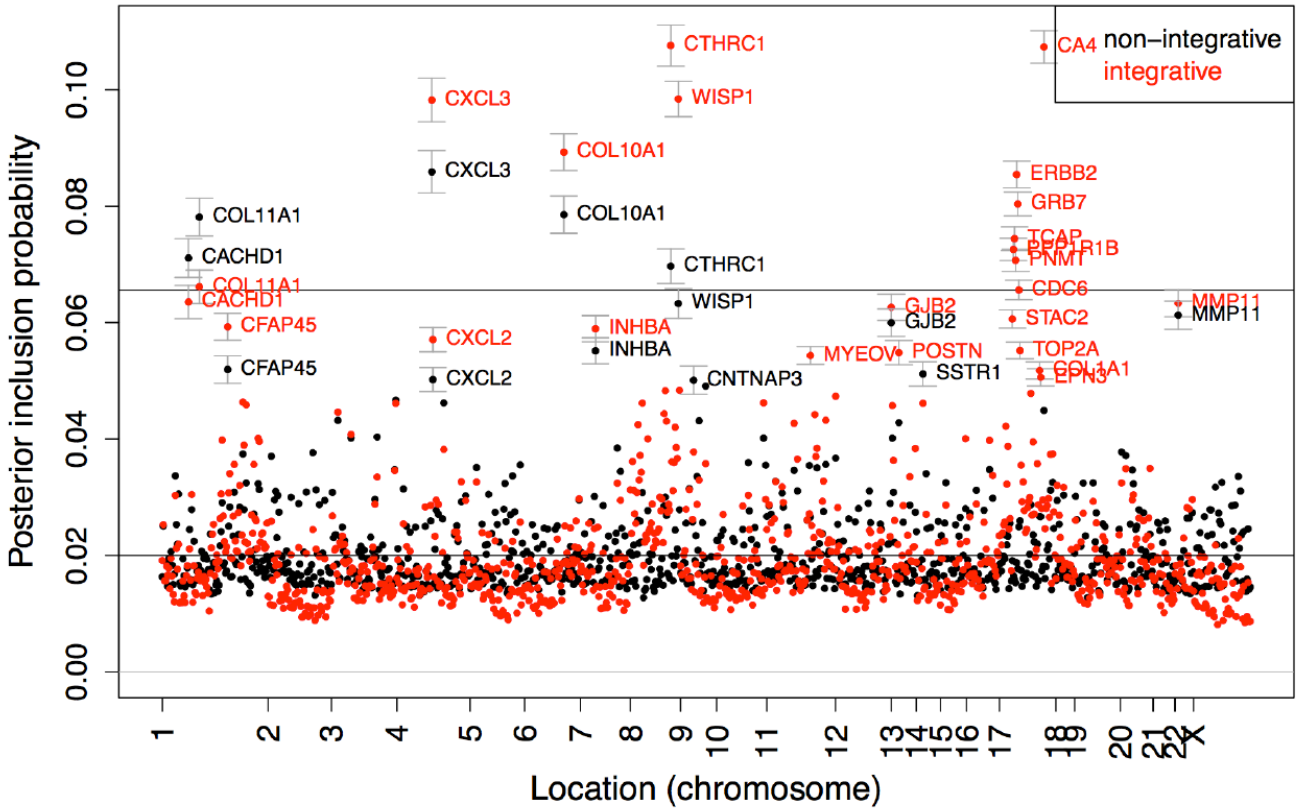
# (1) Use known relationships with one data source to guide the variable selection in another

Std. dev. of CNV data of HER2-pos. breast cancer and healthy tissue samples



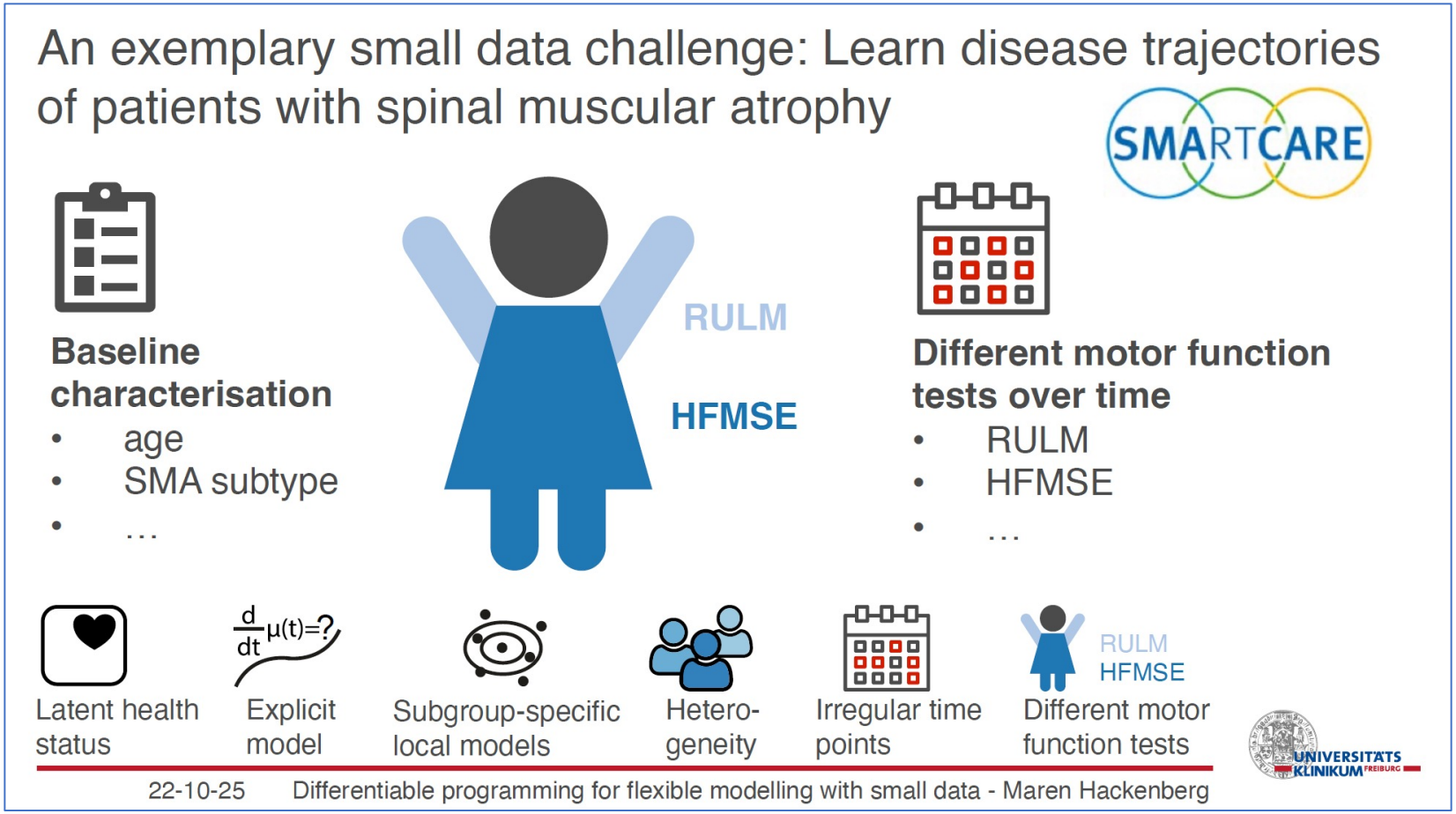

**Idea:** Use CNV information to weigh prior inclusion probabilities of gene expression variables in Bayesian variable selection

# (1) Use known relationships with one data source to guide the variable selection in another



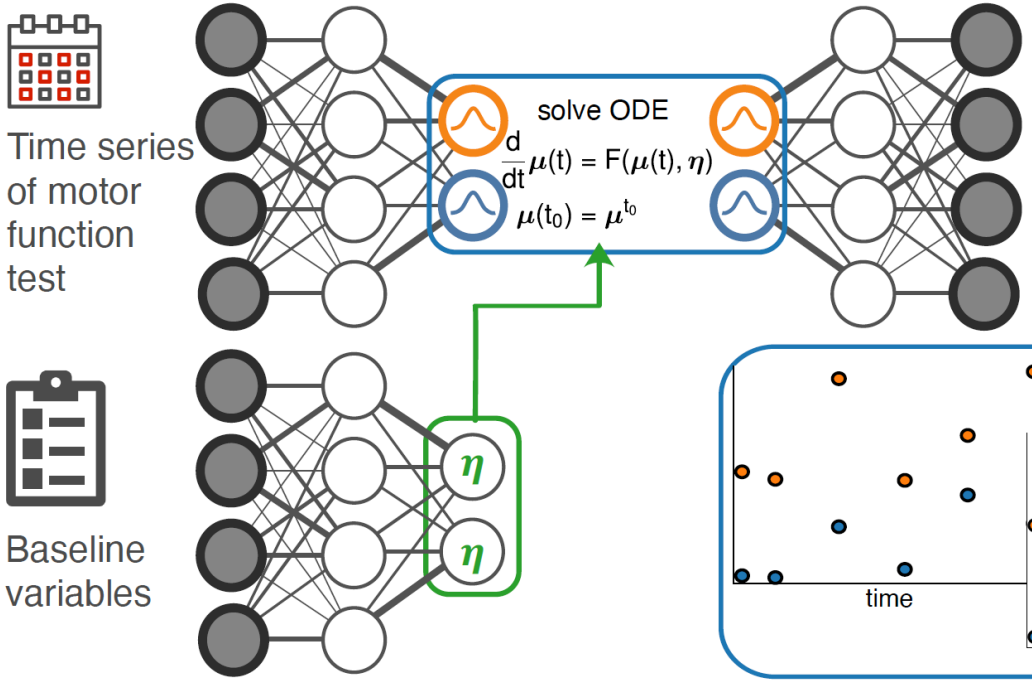

HER2 (= ERBB2) only selected in integrative analysis

# (2) Combine the data-driven ML approach with knowledge-driven mechanistic modelling



# (2) Combine the data-driven ML approach with knowledge-driven mechanistic modelling

## Describe individual SMA trajectories as ODEs in the latent space of a deep learning model




```
m = init_ODEVAE()
ps = getparams(m)
opt = ADAM(η)
trainingdata = zip(xs, xs_baseline, tvals)

for epoch in 1:epochs
    for (X, Y, t) in trainingdata
        grads = gradient(ps) do
            loss(X, Y, t, m, args=args)
        end
        update!(opt, ps, grads)
    end
end
```



UNIVERSITÄTS KLINIKUM FREIBURG

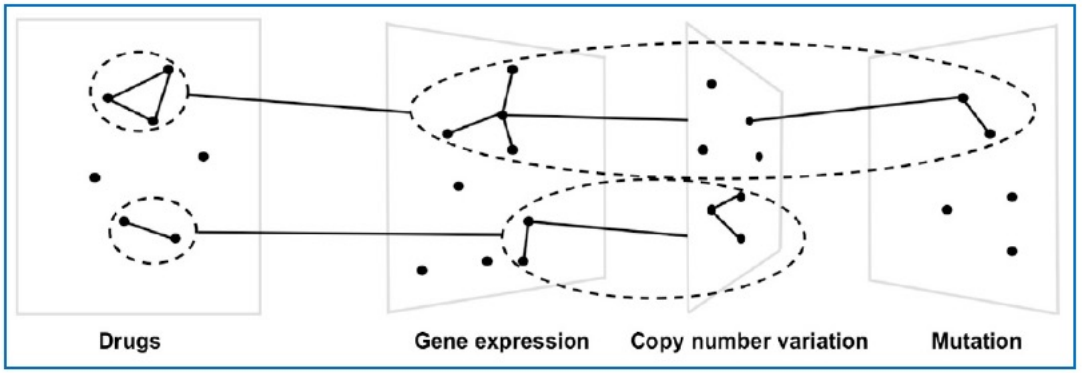
# (3) Make use of correlations in the data:
## between input variables - to restrict the model space

BayesSUR: An R Package for High-Dimensional Multivariate Bayesian Variable and Covariance Selection in Linear Regression

Zhi Zhao, Marco Banterle, Leonardo Bottolo, Sylvia Richardson, Alex Lewin, Manuela Zucknick

Vol. 100, Issue 11

Paper | R package (BayesSUR) | R replication code



# (3) Make use of correlations in the data: between input variables - to restrict the model space

- **Formulation of the model:**

$$\mathbf{Y} = \mathbf{XB} + \mathbf{U},$$
$$\text{vec}(\mathbf{U}) \sim \mathcal{N}(\mathbf{0},\ C \otimes \mathbb{I}_n)$$

$$\beta_{kj}|\gamma_{kj}, w \sim \gamma_{kj}\mathcal{N}(0,\ w) + (1-\gamma_{kj})\delta_0(\beta_{kj})$$

for each element $\beta_{kj}$ in $\mathbf{B}$.

- $\mathbf{Y}$ $n \times m$ matrix of outcomes with $m \times m$ covariance matrix $C$,
- $\mathbf{X}$ $n \times p$ matrix of predictors for all outcomes,
- $\mathbf{B}$ $p \times m$ matrix of regression coefficients,
- $\mathbf{\Gamma} = \{\gamma_{jk}\}$ $p \times m$ binary indicator matrix for variable selection.

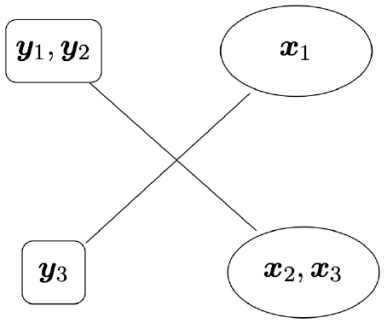
| | $\gamma_{jk} \sim$ Bernoulli | $\gamma_{jk} \sim$ Hotspot | $\boldsymbol{\gamma} \sim$ MRF |
|---|---|---|---|
| $C \sim$ indep | HRR-B | HRR-H | HRR-M |
| $C \sim \mathcal{IW}$ | dSUR-B | dSUR-H | dSUR-M |
| $C \sim \mathcal{HIW}_{\mathcal{G}}$ | SSUR-B | SSUR-H | SSUR-M |

# (3) Make use of correlations in the data:
## between input variables - to restrict the model space

MRF prior for pharmacogenomics

$$f(\gamma|d, e, G) \propto \exp\{d\mathbf{1}^\top\gamma + e \cdot \gamma^\top G\gamma\}$$

- $d$ controls the model sparsity,
- $e$ the strength of relations between responses and predictors,
- $G$ is an adjacency matrix of the structure prior knowledge.

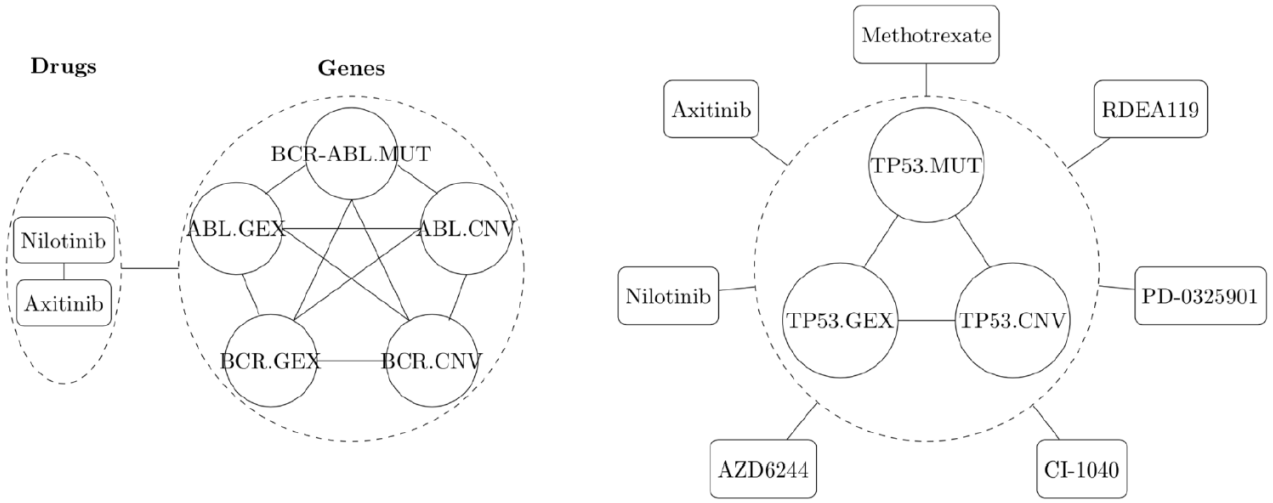



$$G = \begin{array}{c} \\ \gamma_{11} \\ \gamma_{21} \\ \gamma_{31} \\ \gamma_{12} \\ \gamma_{22} \\ \gamma_{32} \\ \gamma_{13} \\ \gamma_{23} \\ \gamma_{33} \end{array} \begin{array}{c} \begin{array}{ccccccccc} \gamma_{11} & \gamma_{21} & \gamma_{31} & \gamma_{12} & \gamma_{22} & \gamma_{32} & \gamma_{13} & \gamma_{23} & \gamma_{33} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

# (3) Make use of correlations in the data:
## between input variables - to restrict the model space

### Application to Genomics of Drug Sensitivity in Cancer data

- Same data as before, but now only use $m = 7$ cancer drugs



# (3) Make use of correlations in the data:
## between input variables - to restrict the model space

### Results ($\Gamma$): Which covariates are important?

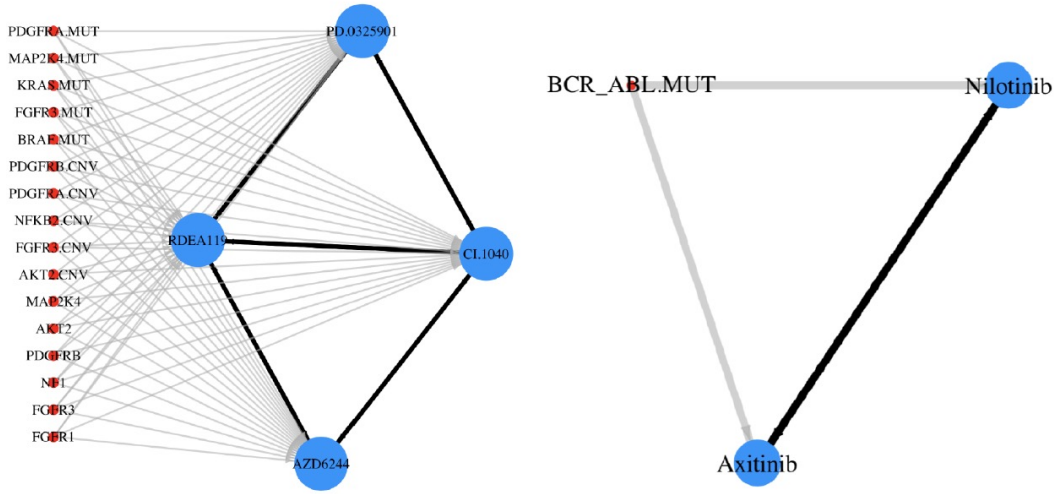



**Fig:** Important covariates related to the MEK inhibitors (left) or Bcr-Abl inhibitors (right) based on threshold for posterior marginal inclusion probabilities (mPIP $> 0.5$).

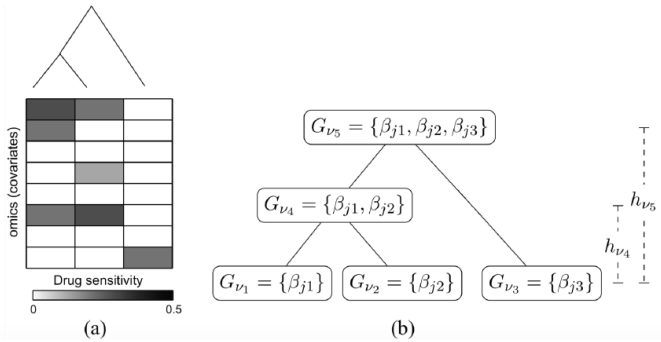
# (3) Make use of correlations in the data: between response variables - to borrow information

## (Multi-response) Tree-guided group lasso (Kim & Xing 2012)

- Include dependencies between columns of **Y** in a group lasso
- Extension to IPF-tree lasso

$$\textbf{Tree lasso: } \text{pen}(\mathbf{B}) = \lambda \sum_{j=1}^{p} \sum_{\nu \in \{V_{\text{int}}, V_{\text{leaf}}\}} \omega_\nu \|\beta_j^{G_\nu}\|_{\ell_2}$$

$$\textbf{IPF-tree lasso: } \text{pen}(\mathbf{B}) = \sum_{s} \lambda_s \left( \sum_{j_s} \sum_{\nu \in \{V_{\text{int}}, V_{\text{leaf}}\}} \omega_\nu \|\beta_{j_s}^{G_\nu}\|_{\ell_2} \right)$$



# (3) Make use of correlations in the data:
## between response variables - to borrow information

### Drug screens for precision cancer medicine:
### How to predict the drugs' effect with data on drugs and tumour?

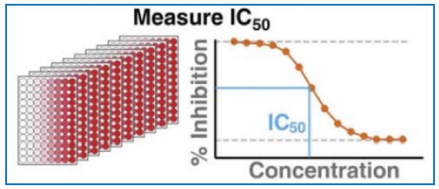
ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

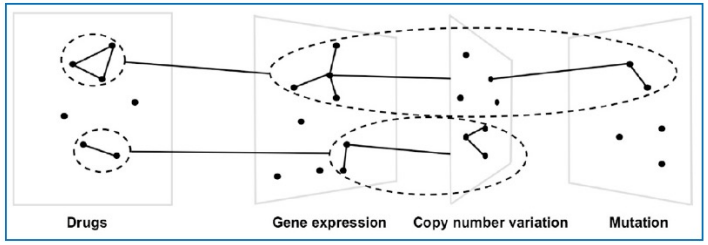
Journal of the Royal Statistical Society
Applied Statistics
Series C

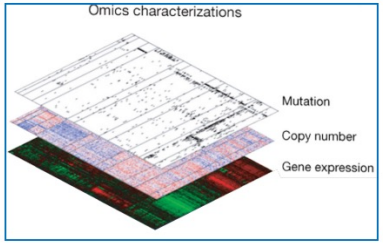
Original Article | Open Access

**Structured penalized regression for drug sensitivity prediction**

Zhi Zhao, Manuela Zucknick



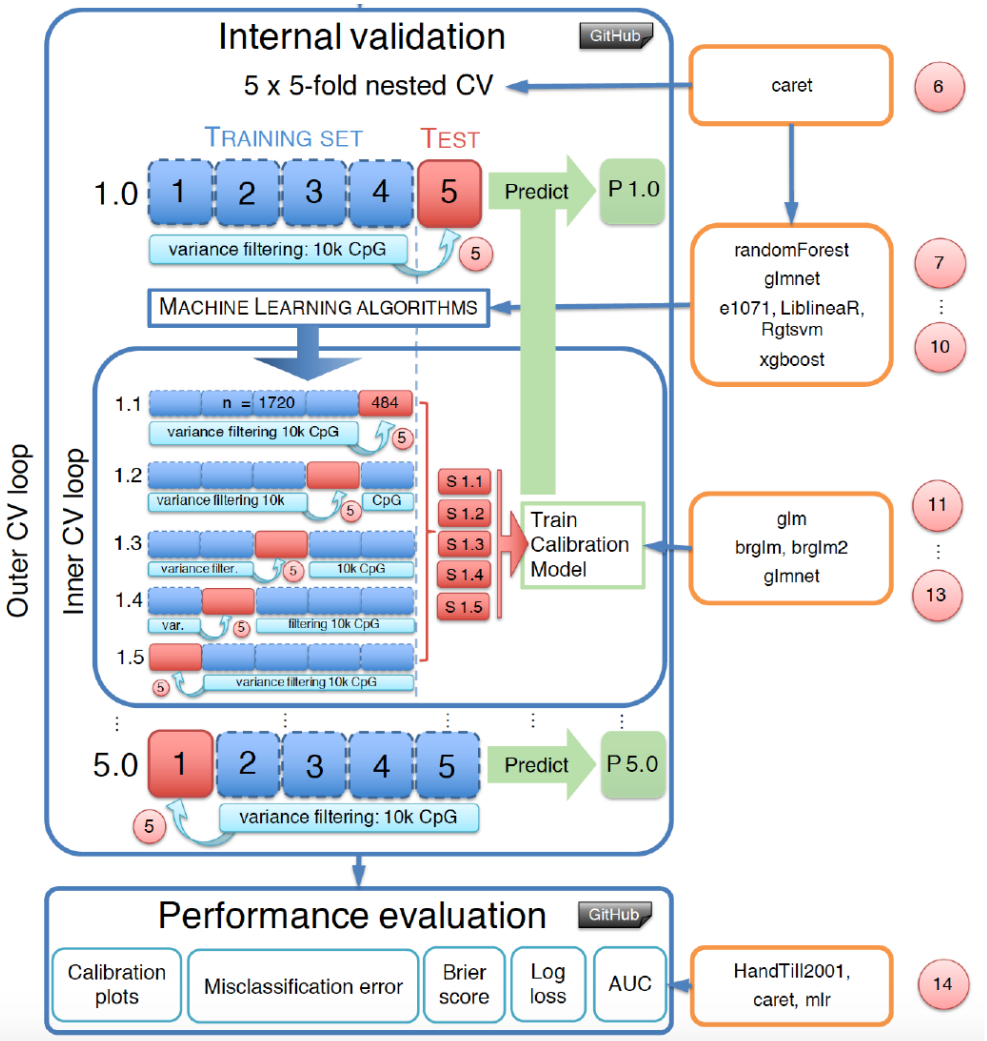



# Model validation is crucial with small data

- Careful and correctly set up the model validation framework is even more important with small data
- To avoid over-fitting when selecting tuning parameters or selecting models
- To avoid being too optimistic when estimating prediction error

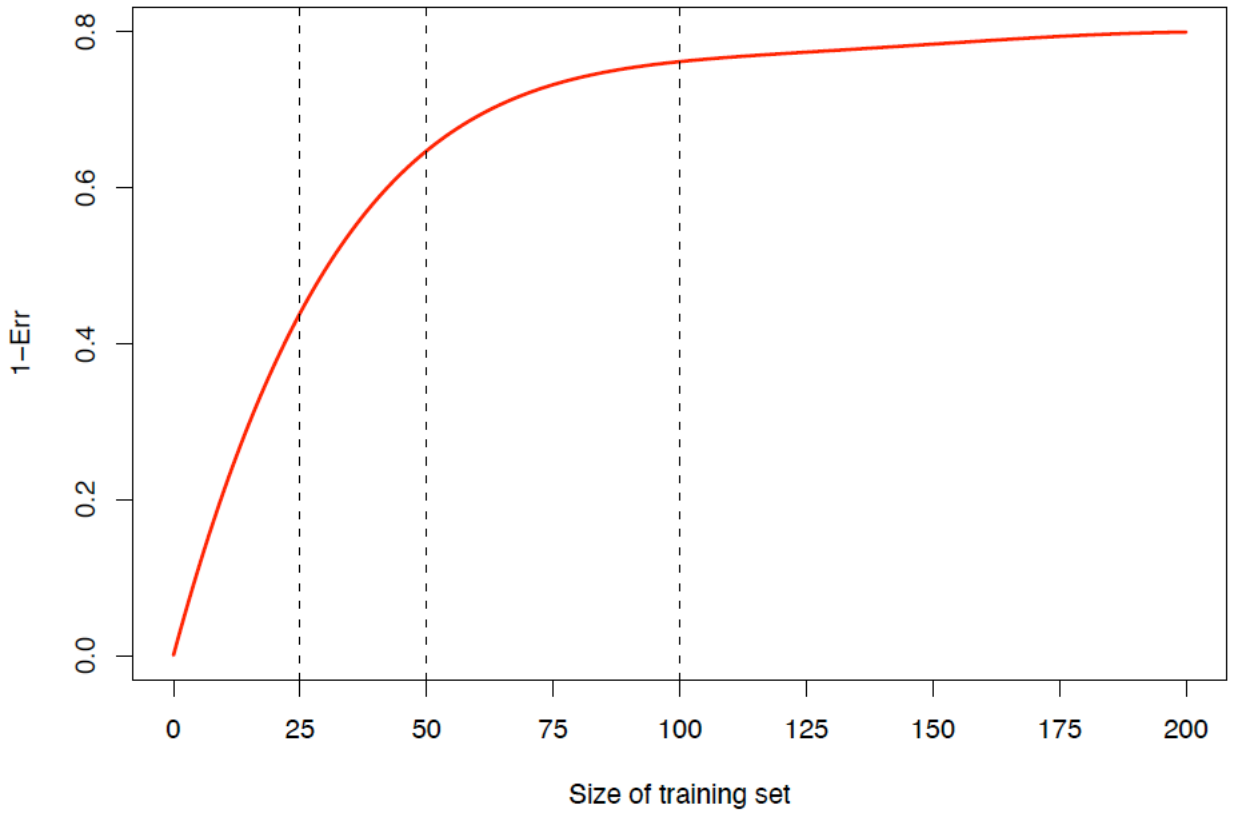
- Learning curve: How many samples are needed in the training set to approach optimal model training?
- Nested cross-validation
- .632+ bootstrapping vs .632+ subsampling

# Nested cross-validation




from: Maros et al. (2020)

# Learning curve:
# How many samples are needed in the training data?



# Which model is best for prediction?

## Example: Regularization/Variable selection by Lasso

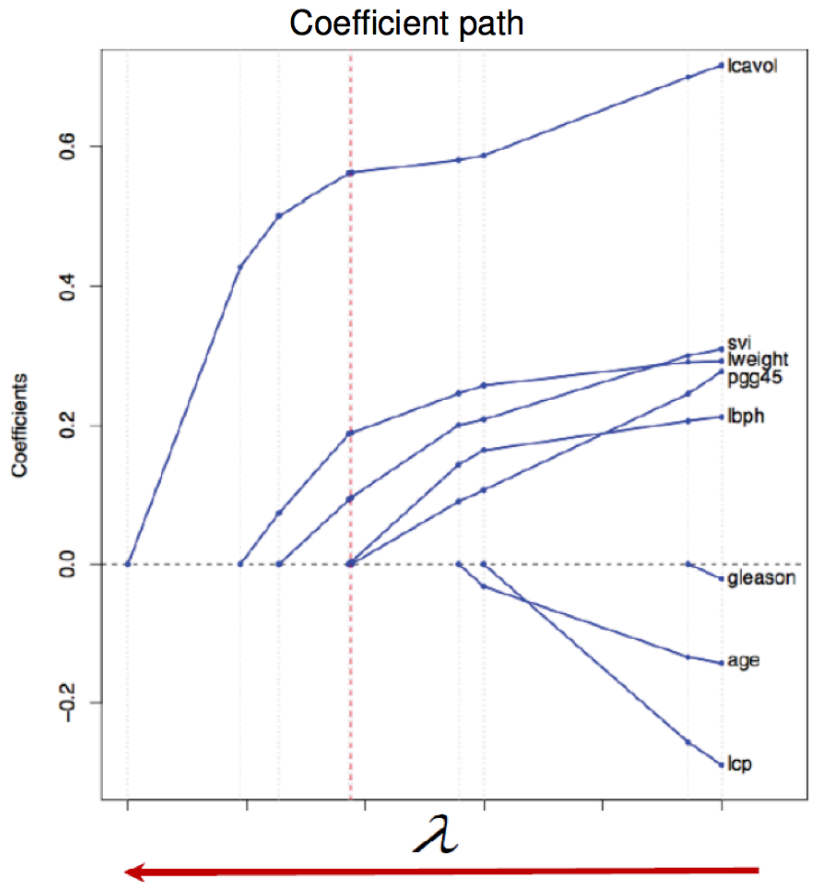
**Idea:**
Penalize (shrink towards zero) regression coefficients by adding penalty term to LS criterion.

Thereby, "non-relevant" coefficients are estimated as exactly 0 and can be excluded.

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Penalty controlled by regularization parameter $\lambda$:

- small $\lambda \Rightarrow$ many variables in model
- large $\lambda \Rightarrow$ few variables in model




$\Rightarrow$ How to select $\lambda$ to minimize prediction error?

# Measuring prediction performance

To evaluate model performance on a given data set, measure how well its predictions actually match the observed data.

How close is the predicted value to the true value for that observation?

- **Linear Regression:** Mean squared error:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **2-class Classification:** Brier score:

$$BS = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{p}(y_i = 1|x_i))^2$$

# Performance measures

Some models are used only for parameter estimation and testing

But:

- If used for prediction/classification, need to consider accuracy of predictions
- Two major aspects of prediction accuracy that need to be assessed:

  (1) Reliability or calibration of a model:
  - ability of the model to make unbiased estimates of the outcome
  - observed responses agree with predicted responses

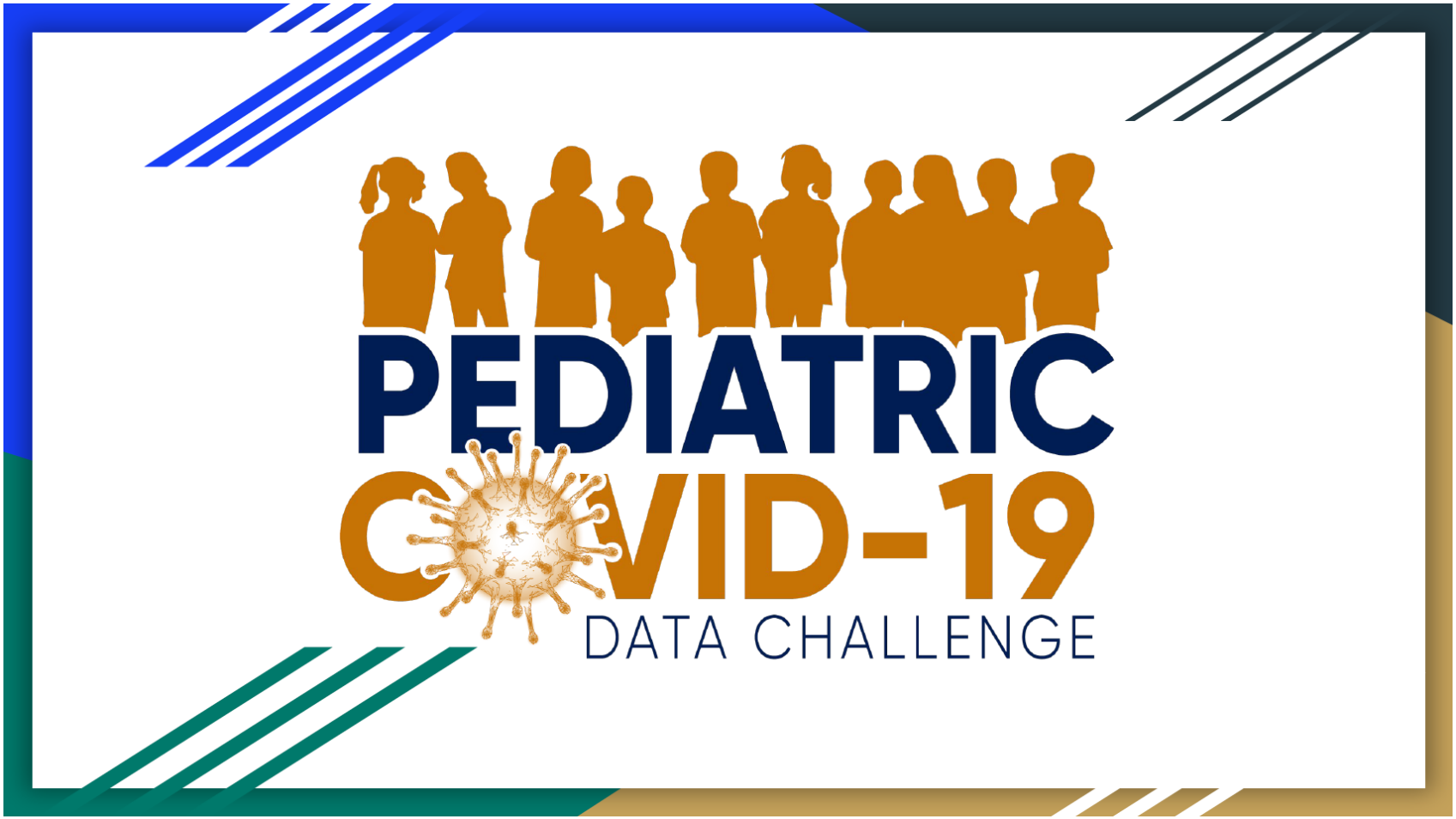
  (2) Discrimination ability:
  - the model is able, through the use of predicted responses, to separate subjects

# Performance measures for classification tasks

## Steyerberg et al, 2010 (Table 1)

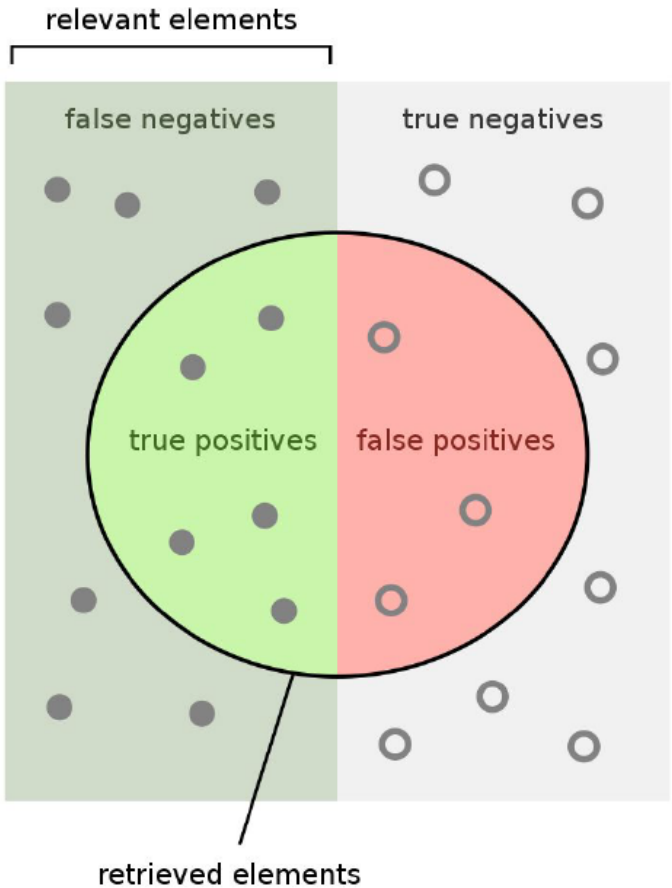
| Aspect | Measure | Visualization | Characteristics |
|---|---|---|---|
| Overall performance | $R^2$ Brier → Brier score | Validation graph | Better with lower distance between Y and $\hat{Y}$. Captures calibration and discrimination aspects. |
| Discrimination | C statistic → AUC | ROC curve | Rank order statistic; Interpretation for a pair of patients with and without the outcome |
| | Discrimination slope | Box plot | Difference in mean of predictions between outcomes; Easy visualization |
| Calibration | Calibration-in-the-large | Calibration or validation graph | Compare mean(y) versus mean($\hat{y}$); essential aspect for external validation |
| | Calibration slope | | Regression slope of linear predictor; essential aspect for internal and external validation related to 'shrinkage' of regression coefficients |
| | Hosmer-Lemeshow test | | Compares observed to predicted by decile of predicted probability |

# Example: Data challenge model performance evaluation




https://drive.hhs.gov/pediatric_challenge.html

# Example: Data challenge model performance evaluation



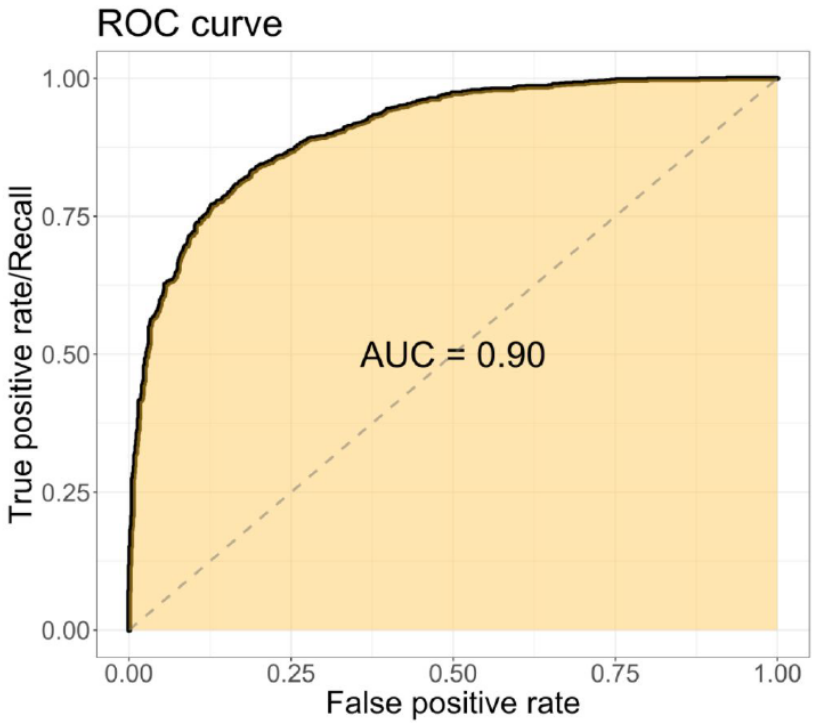

How many retrieved items are relevant?

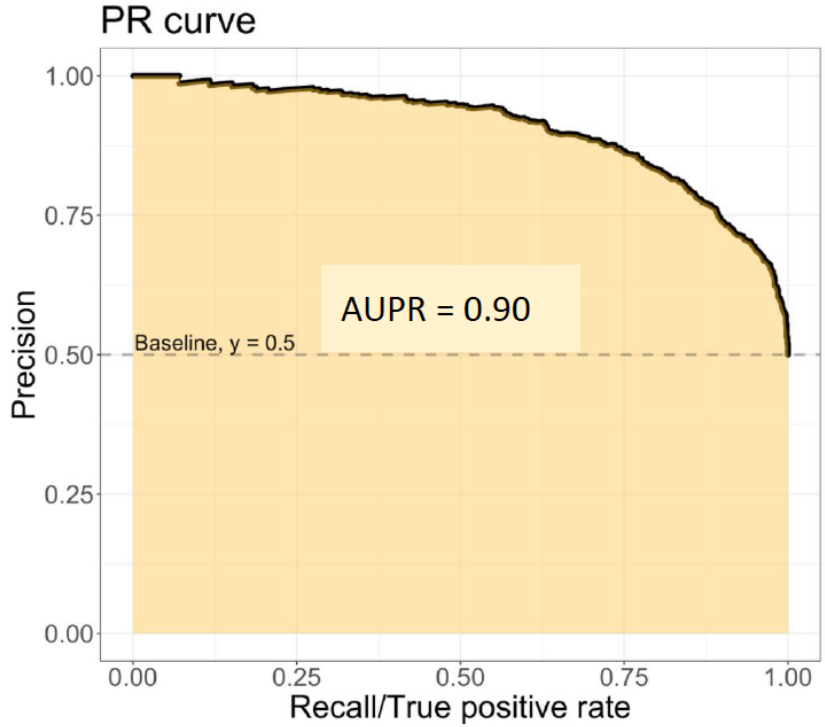
$$\text{Precision} = \frac{\text{true positives}}{\text{retrieved elements}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

# Example: Data challenge model performance evaluation




$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

# Example: Data challenge model performance evaluation

## Quantitative score (85 %):

$$\frac{1}{3}\left(\left(\max_{\text{threshold } t} F_2(t)\right)^2 + \text{AUPR}^2 + \left(\text{Mean}(\text{AUROC}) - \text{Var}(\text{AUROC})\right)^2\right)$$

## Qualitative score (15 %):

- Timeliness
- Interpretability
- Context Utility
- Technical Reproducibility
- Prediction Reproducibility

How to estimate the performance measure

in an unbiased manner?

# How to estimate performance in an unbiased manner?

**Need:** Model assessment/validation to ascertain whether predicted values from the model are likely to accurately predict responses on future subjects or subjects not used to develop the model

**Two modes of validation**

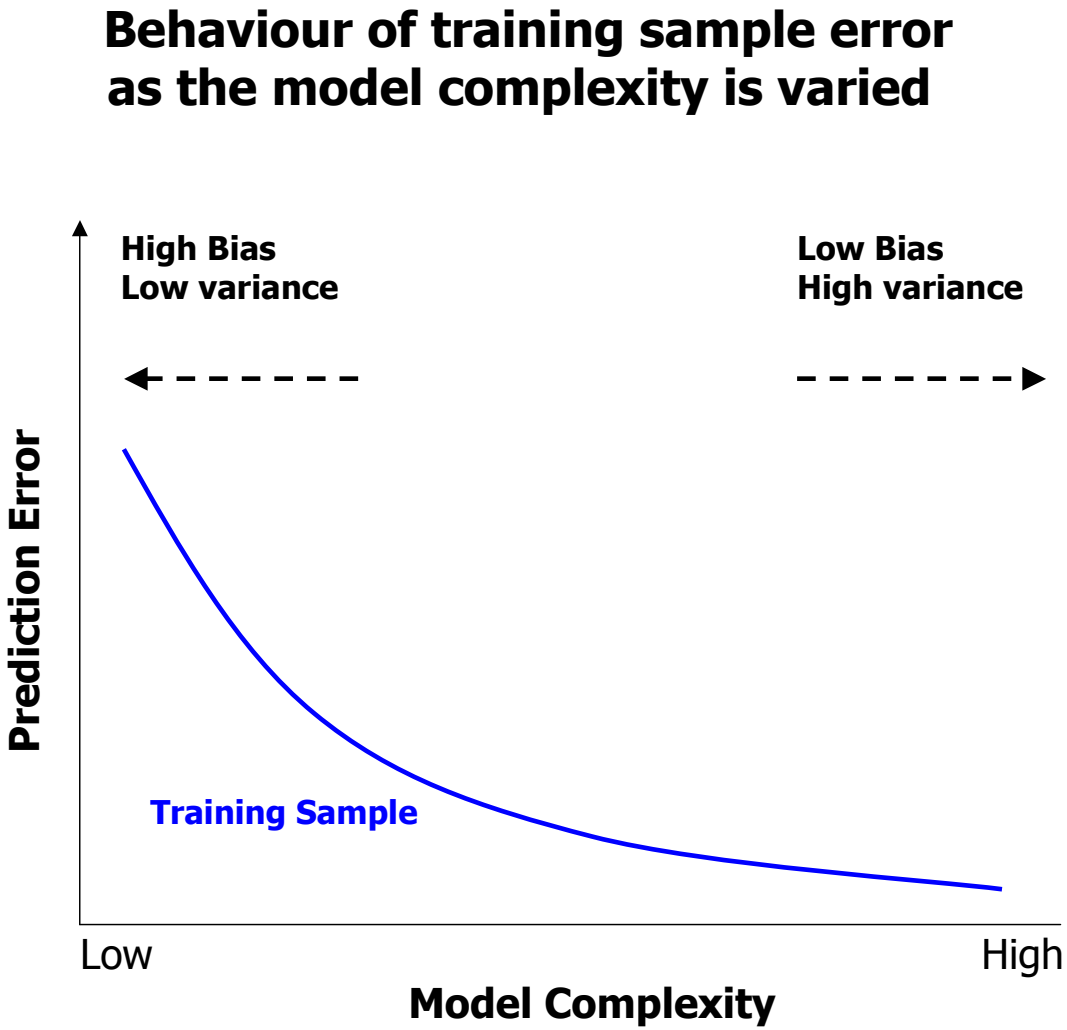
- External:
  Use different sets of subjects for building the model (including tuning) and testing
- Internal:
  (i) Apparent (or training) error: evaluate fit on same data used to create fit
  (ii) Data splitting and its extensions
  (iii) Resampling methods

- **Two fundamental problems with estimation on the training data:**
  - The final model will over-fit the training data. Problem is more pronounced with models with a large number of variables.
  - The error estimate will be overly optimistic (too low).
- A much better idea is to **split the data** into disjoint subsets or use **resampling methods**

- **Training error**: Classification error in the training data set
- **Generalisation error**: Expected error for the classification of new samples $\rightarrow$ This is what we want to estimate!

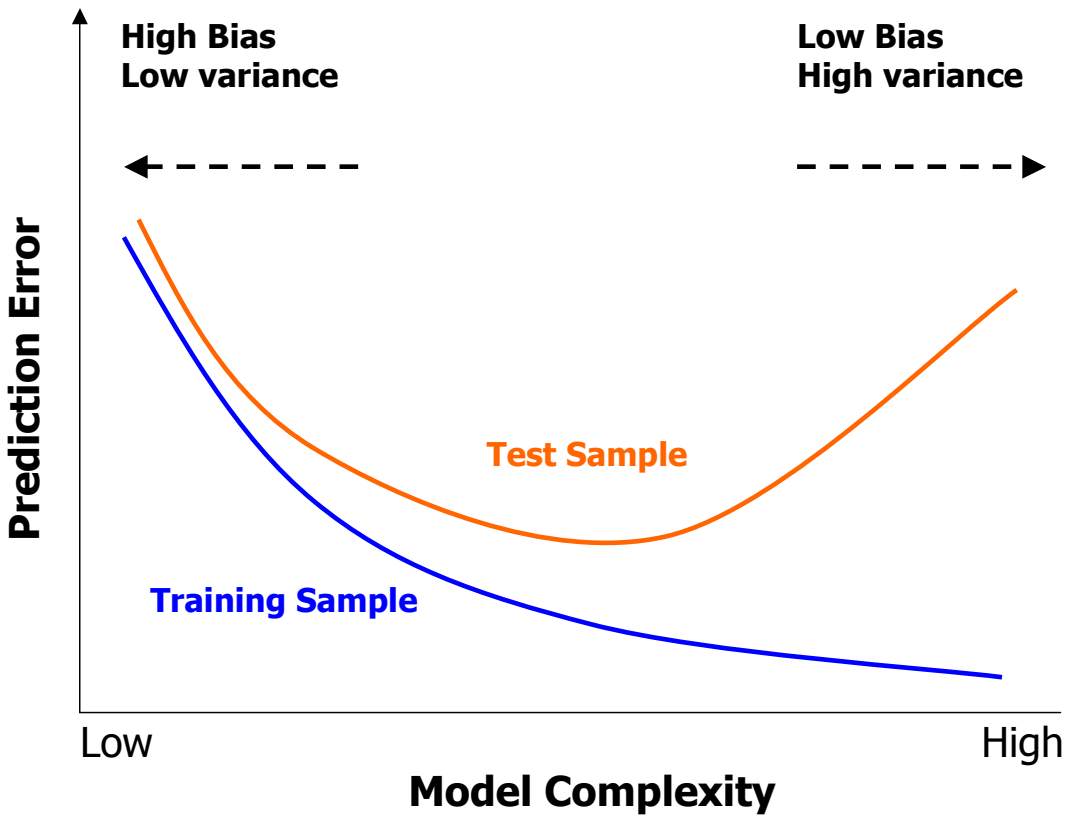
The training error is a bad estimator for the generalisation error!

# Over-fitting is a major problem



# Over-fitting is a major problem

**Behaviour of test and training sample error as the model complexity is varied**



# The Bias-Variance Trade-Off

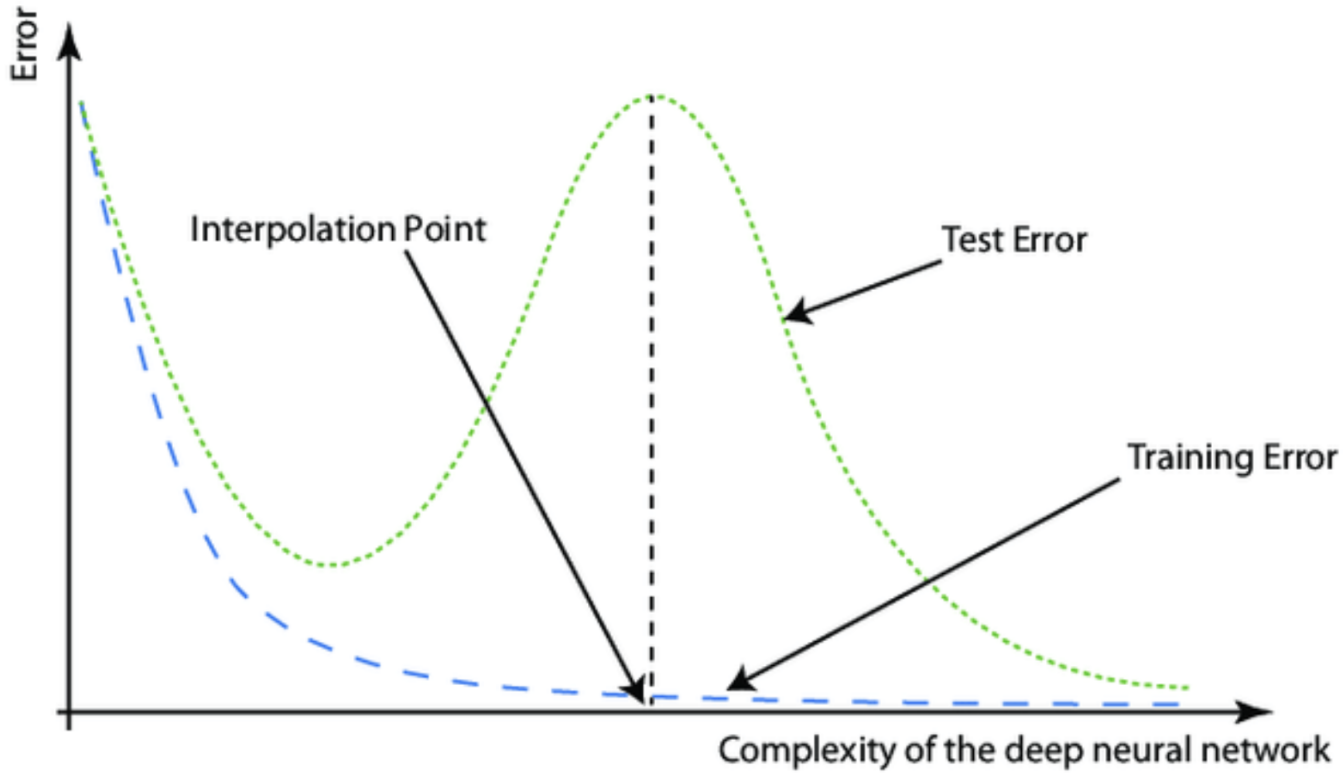
- A simple model might have more model bias, but
- A complex model has more model variance.

For $Y = f(X) + \epsilon$ with $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma_\epsilon^2$, the expected prediction error of $\hat{f}(X)$ at point $x_0$ with squared error loss is:

$$\begin{aligned}
\mathrm{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\
&= \sigma_\varepsilon^2 + [\mathrm{E}\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - \mathrm{E}\hat{f}(x_0)]^2 \\
&= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\
&= \mathrm{Irreducible\ Error} + \mathrm{Bias}^2 + \mathrm{Variance}. \qquad (7.9)
\end{aligned}$$

from Hastie et al. (2009), chapter 7.3

# Things are different for very large (deep learning) models




- Underparameterised region
- Overparameterised region
- Double descent region: beyond overfitting to training data.

Belkin et al. (2019). doi:10.1073/pnas.1903070116
Lafon & Thomas (2024). doi:10.48550/arXiv.2403.10459

# Model building, selection and assessment

1. How to decide which method is the "best", i.e. has the smallest generalisation error, in a specific situation?
2. And how large is that smallest generalisation error anyway?

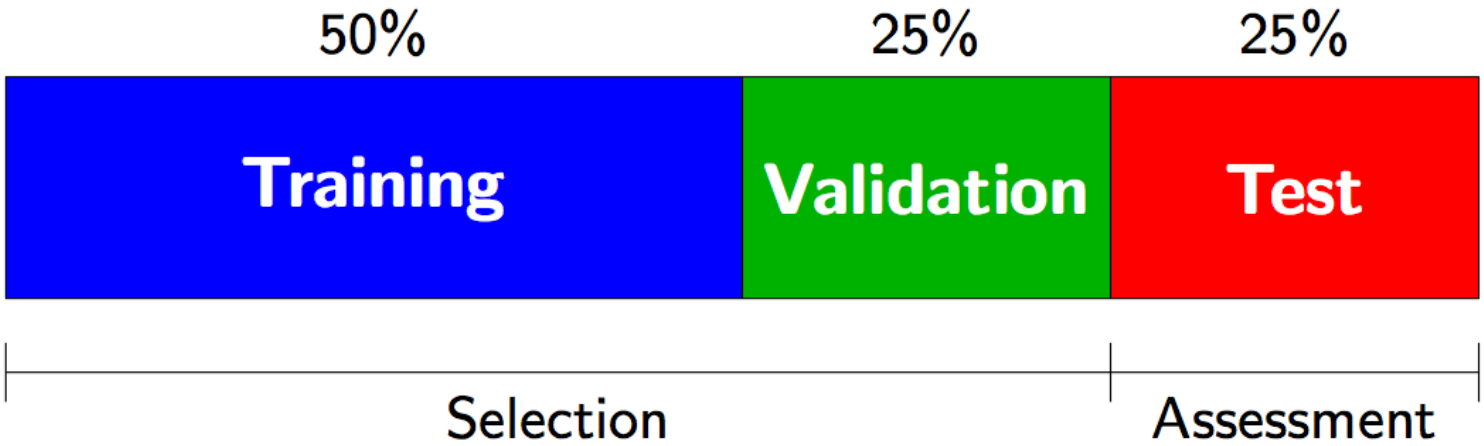
- **Model building and selection**: For a variety of different methods
  1. Fit ("train") the models,
     i.e. perform parameter tuning/ variable selection
  2. Estimate the prediction errors.
  3. Choose the "best" method for a specific situation.
- **Model assessment**
  - For the final selected model estimate the generalisation error on *new data*.

# Sample splitting

→ Split data in several independent subsets before model building.

# Sample splitting

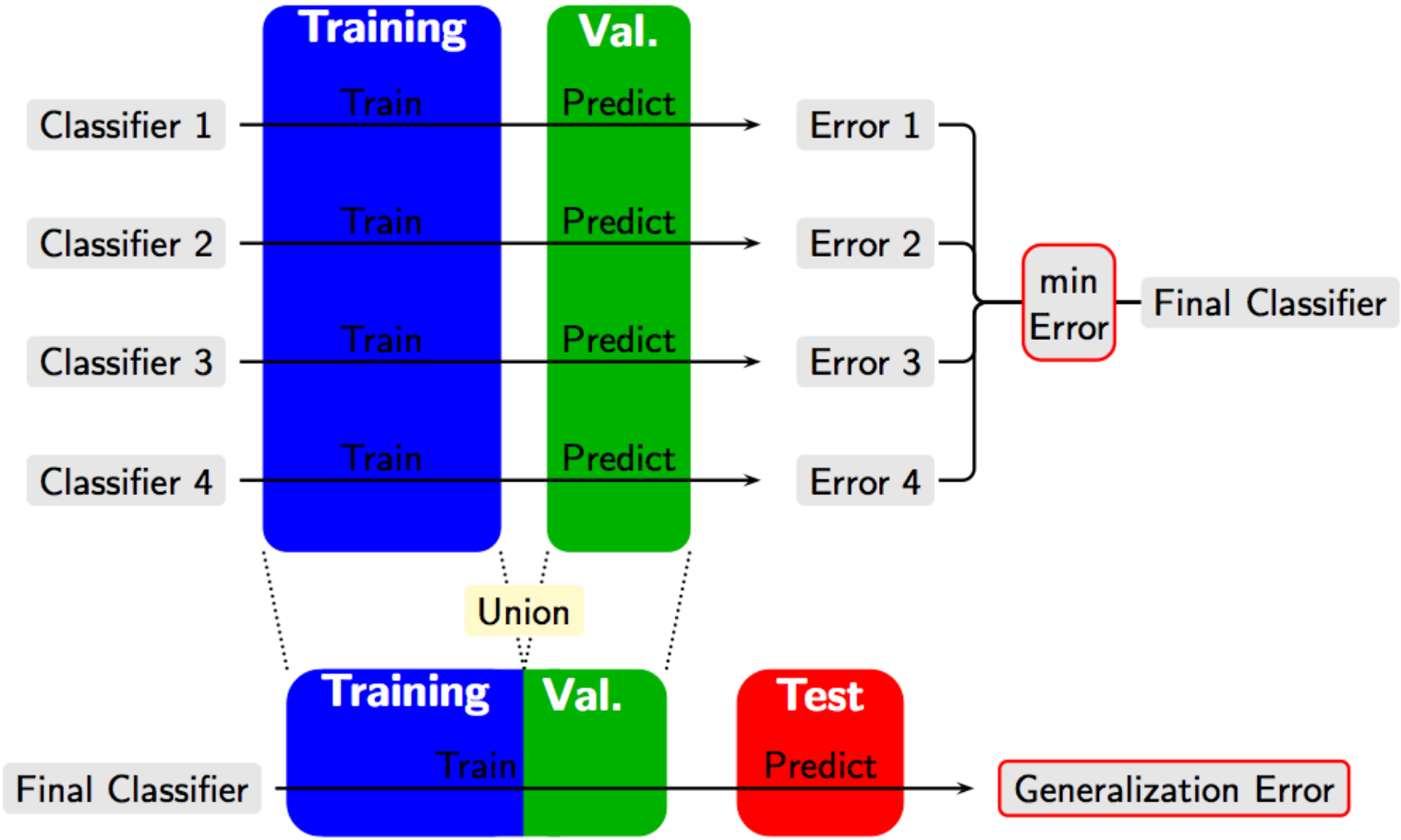
In a data-rich situation, we can split the available data.




- **Training set**: Fit ("train") the various prediction models
- **Validation set**:
  - Estimate the prediction errors of the models
  - Final model: Choose model with smallest prediction error
- **Test set**: Estimate the generalisation error by applying the final model to a new test data set

# Sample splitting

## Model building and selection →




→ Model assessment

# Drawbacks of sample splitting

One-time sample splitting has two **basic drawbacks**:

- We may not be able to afford the "luxury" of setting aside a portion of the data set for testing, as it might result in a large loss of power.

- The assessment can vary greatly when taking different splits:

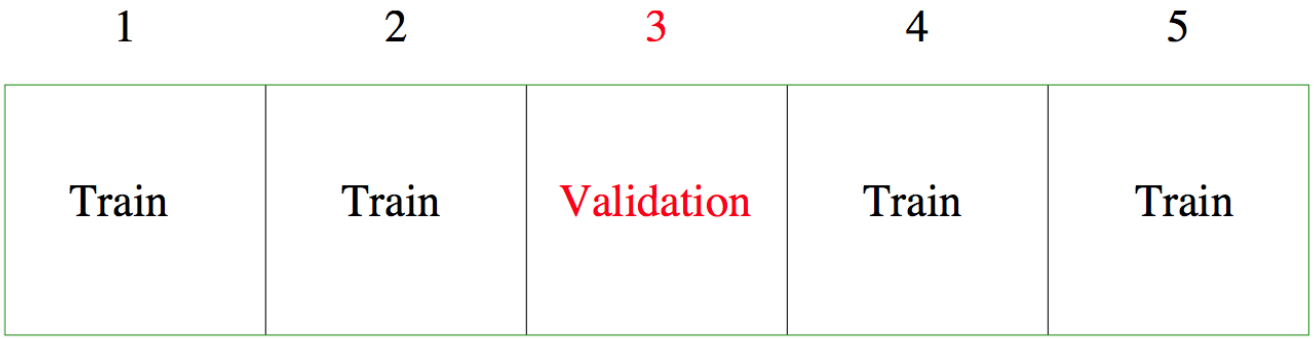
  Since it is a single train-and-test experiment, the estimate of the error rate will be misleading if we happen to get an "unfortunate" split.

# Resampling methods

- → Cross-validation
- → Bootstrapping

# Cross-validation

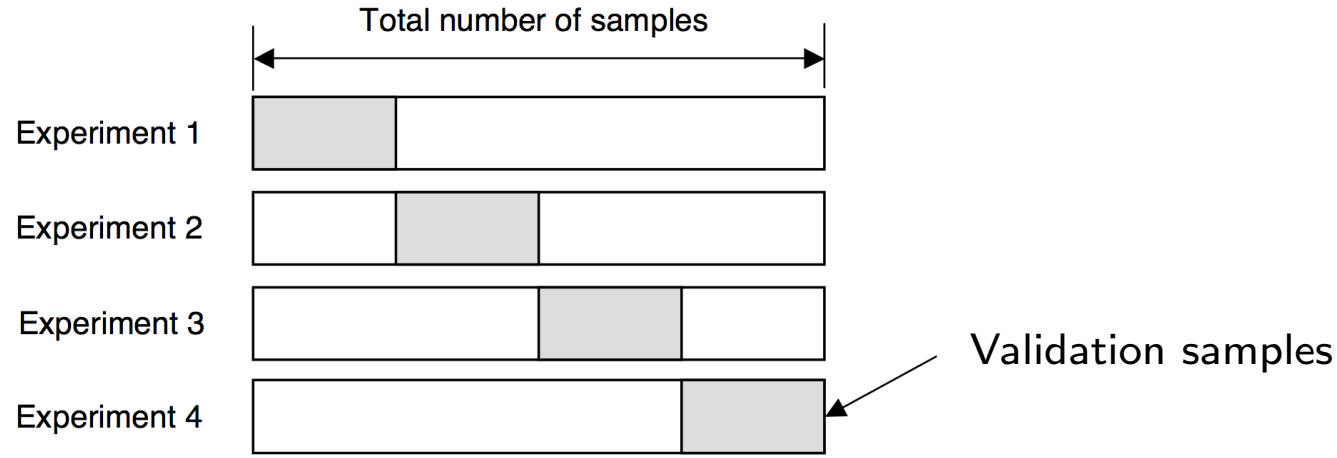
- Alternative to data splitting in not so data-rich situations (i.e. most of the time...)
- Partition the data set into $K$ roughly equal-sized subsets
- Each subset will be the test data set once, with the remaining samples making up the training data



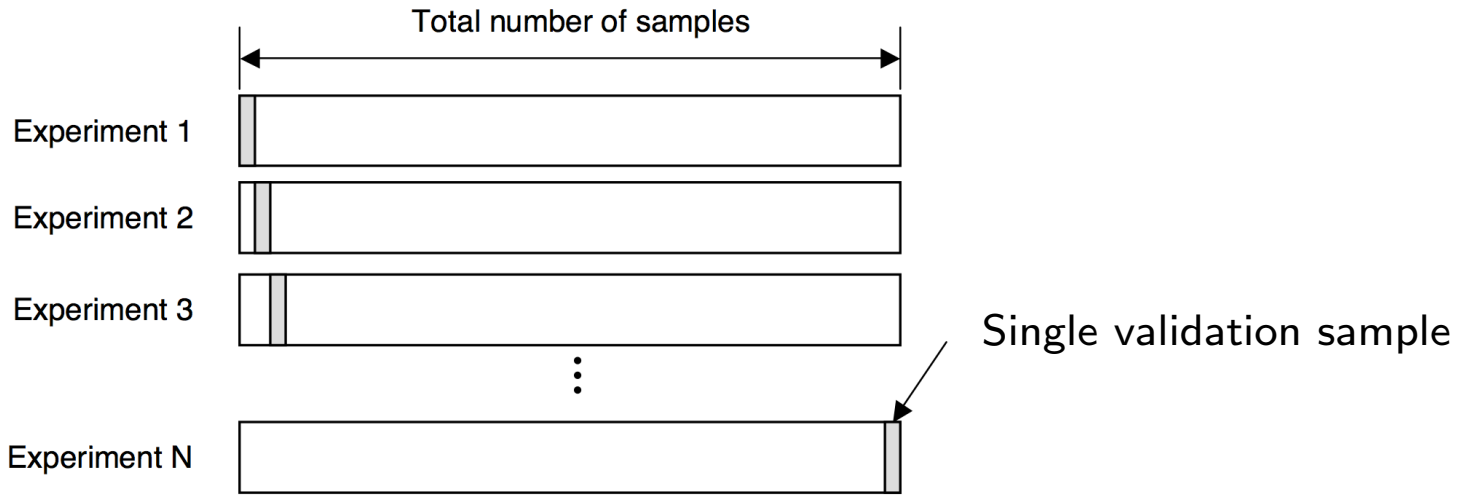

- Cross-validation error: The results are pooled from all test sets to estimate the performance of the model (each case is used exactly once).

# Cross-validation

- *K*-fold cross-validation

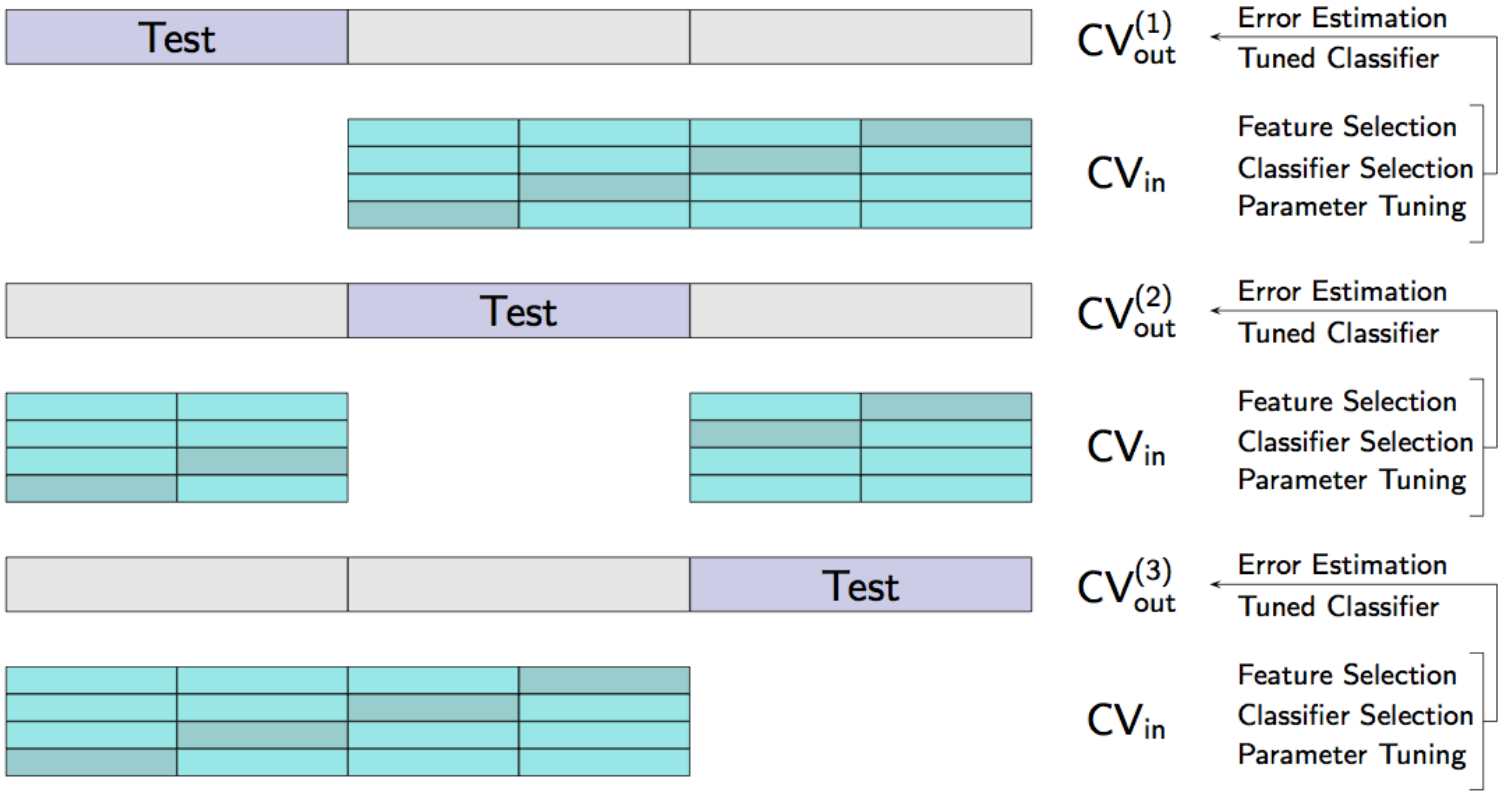


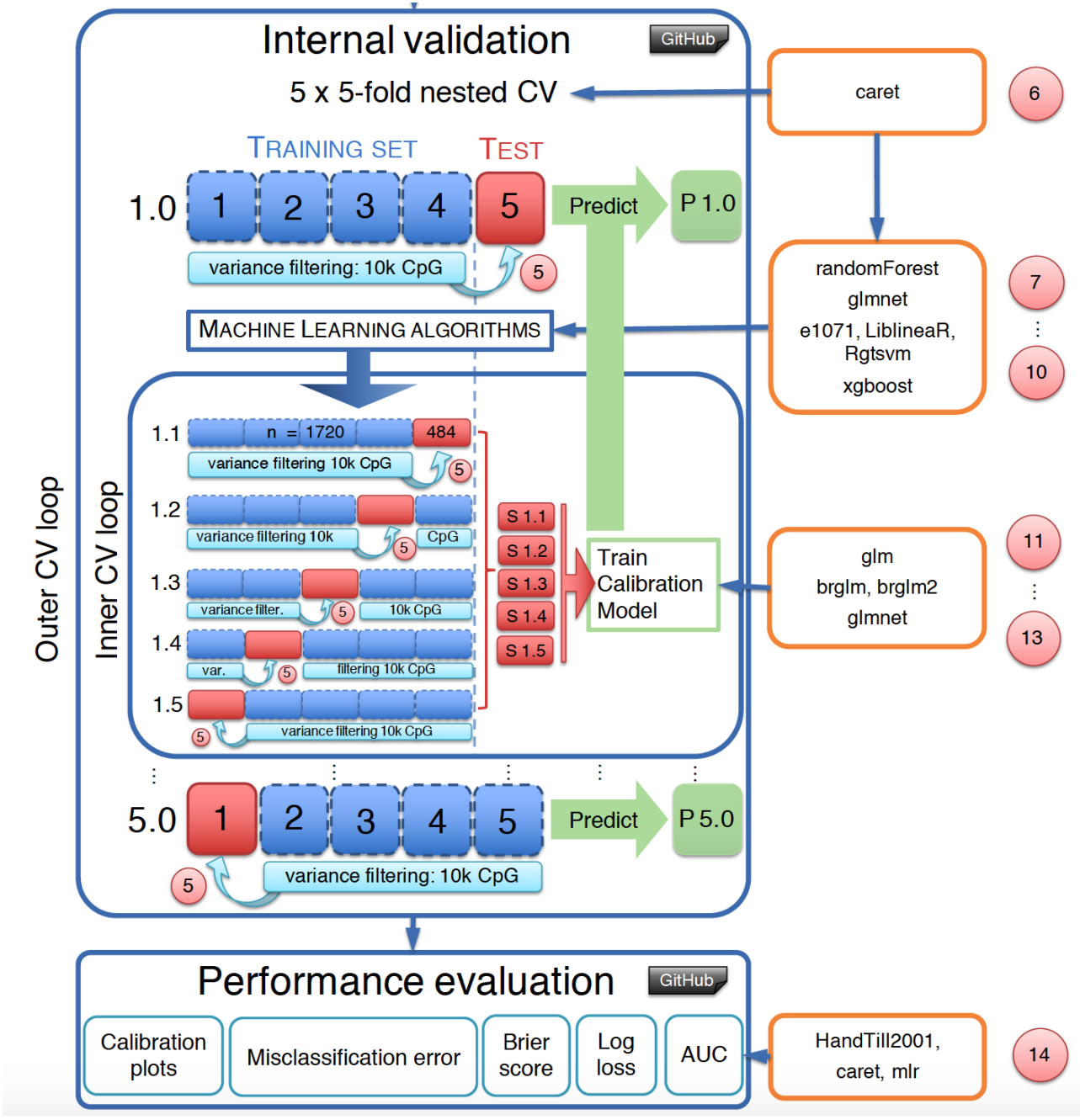

- Leave-one-out cross-validation



# Nested cross-validation

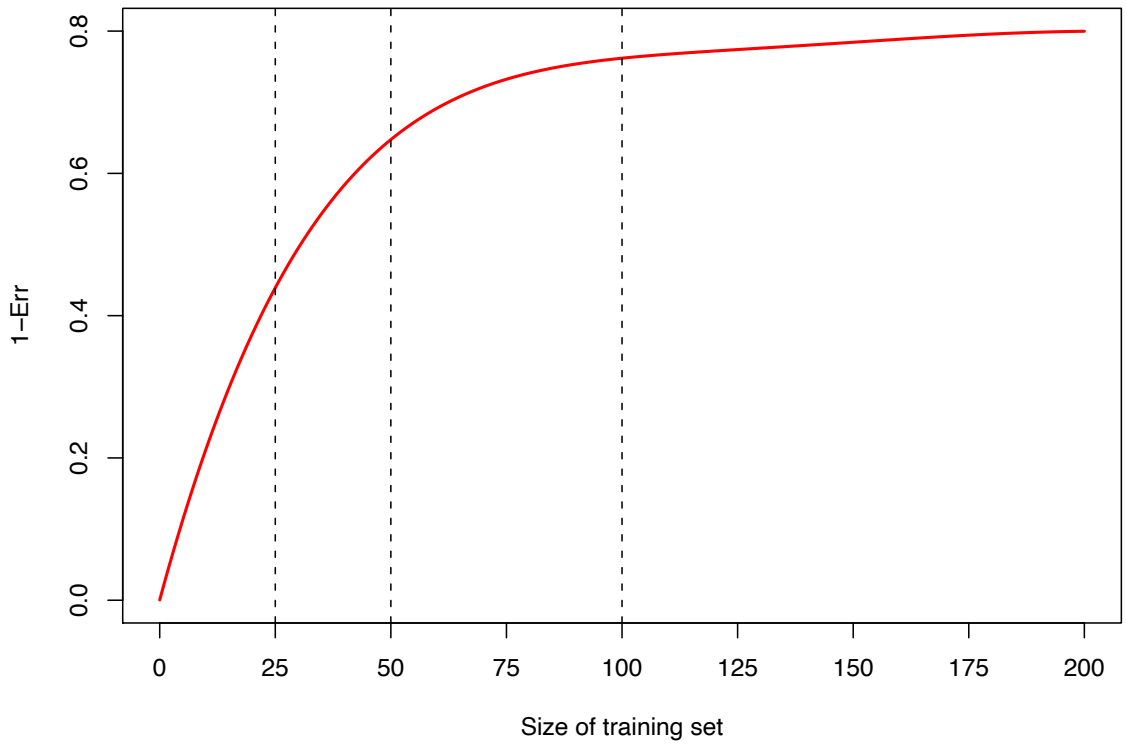
- Inner CV loop: Model building and selection
  - Feature selection, model selection, parameter tuning
  - Choose the model with the smallest CV error within inner loop
- Outer CV loop: Model assessment
  - Estimate the generalisation error for the final model






from: Maros et al. (2020)

# K-fold cross-validation: Training set size bias




**Hypothetical learning curve:**
The performance of the predictor improves as the training set size increases to about 100 observations.

Increasing this number further brings only a small benefit.

# Drawbacks of cross-validation

- **Leave-one-out CV:** may have large variance
- **K-fold CV:** may have large bias, depending on the choice of the number of observations to be held out from each fit. The bias is possibly severe for training set sizes $< 50$, say. If the learning curve has a considerable slope at the given training set size, 5 or 10-fold CV will strongly overestimate the true prediction error.

- Possible solution: estimate prediction error by **bootstrapping**