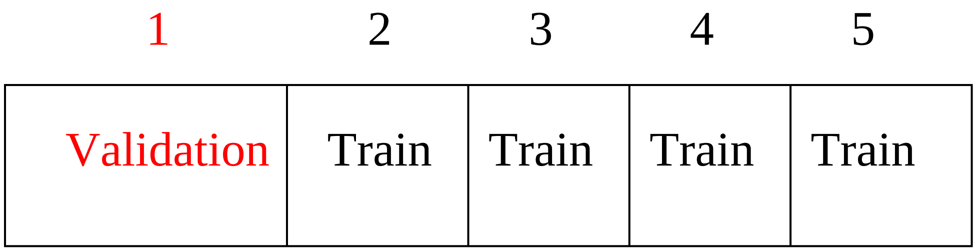
K-Fold Cross-Validation

Primary method for estimating a tuning parameter λ (such as subset size)

- Divide the data into K roughly equal parts



1	2	3	4	5
Validation	Train	Train	Train	Train

- for each $k = 1, 2, \ldots K$, fit the model with parameter λ to the other $K - 1$ parts, giving $\hat{\beta}^{-k}(\lambda)$ and compute its error in predicting the kth part:

  $E_k(\lambda) = \sum_{i \in kth\ part} (y_i - \mathbf{x}_i \hat{\beta}^{-k}(\lambda))^2.$

This gives the cross-validation error

$$CV(\lambda) = \frac{1}{K}\sum_{k=1}^{K} E_k(\lambda)$$

- do this for many values of λ and choose the value of λ that makes $CV(\lambda)$ smallest.

Typically we use $K = 5$ or 10.

Cross-validation- revisited

Consider a simple classifier for wide data:

- Starting with 5000 predictors and 50 samples, find the 100 predictors having the largest correlation with the class labels
- Conduct nearest-centroid classification using only these 100 genes

How do we estimate the test set performance of this classifier?

Apply cross-validation in step 2?

This is WRONG!

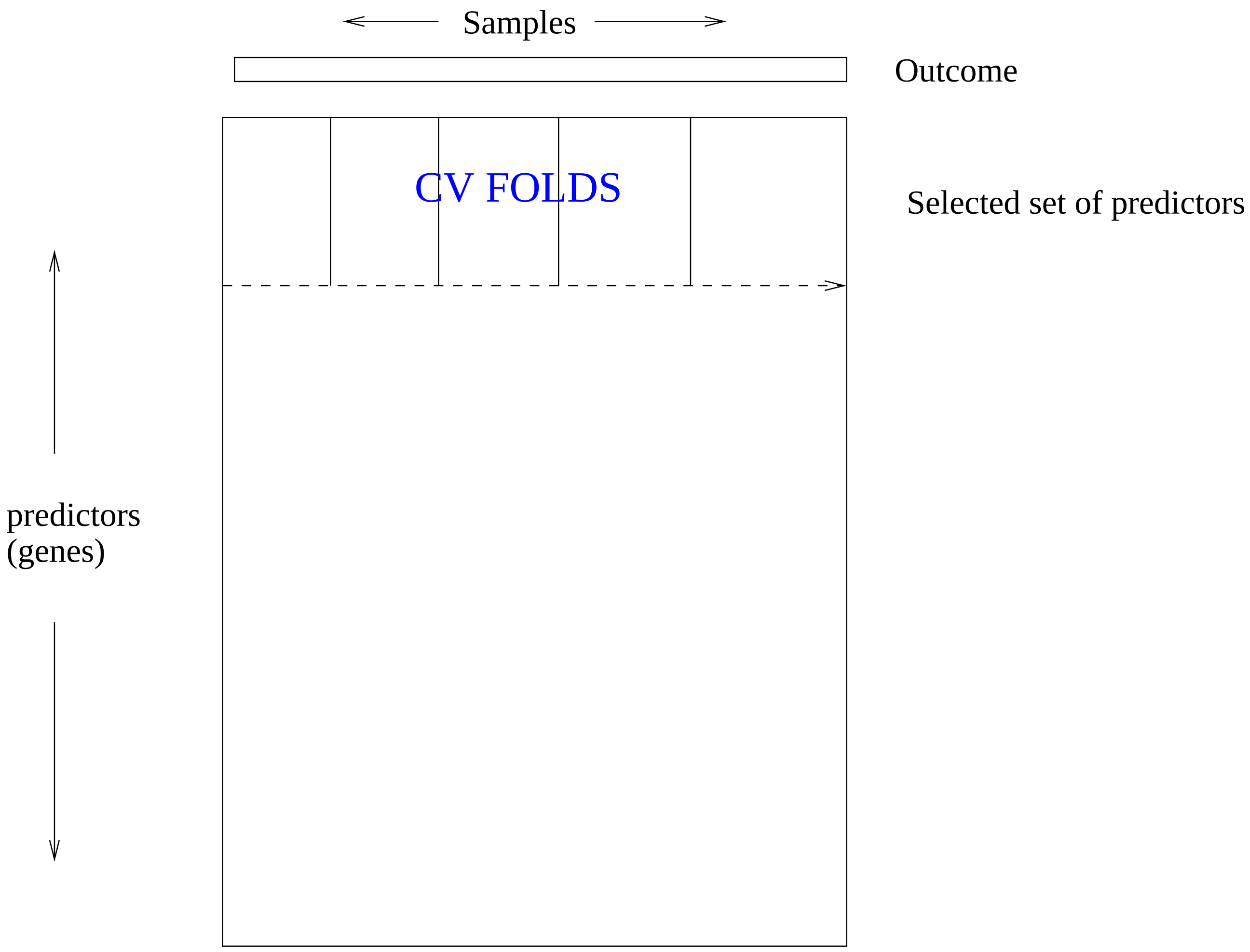
- It ignores the fact that the procedure has already "seen" the labels of the training data, and made use of them. This is a form of training and must be included in the validation process.
- It is easy to simulate realistic data with the class labels independent of the outcome, so that
  - true test error =50%, but
  - "Wrong" CV error estimate is zero!
- We have seen this error made in 4 high profile microarray papers in the last couple of years. See Ambroise & McLachlan (2002) .

The Wrong and Right Way

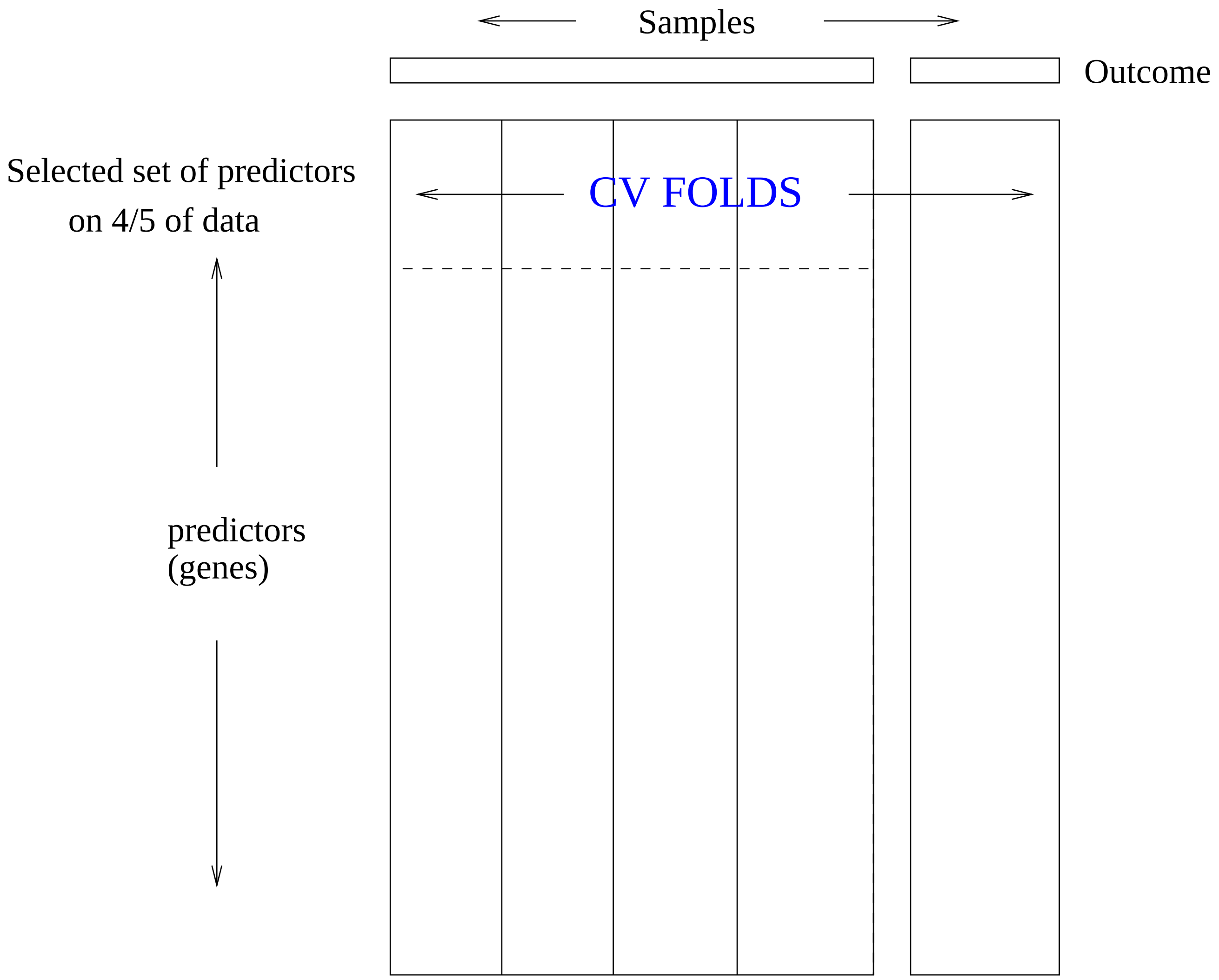
✗ *Wrong:* Apply cross-validation in step 2.

✓ *Right:* Apply cross-validation to steps 1 and 2.

A little cheating goes a long way



The Right way


Samples
Outcome
Selected set of predictors
on 4/5 of data
CV FOLDS
predictors
(genes)