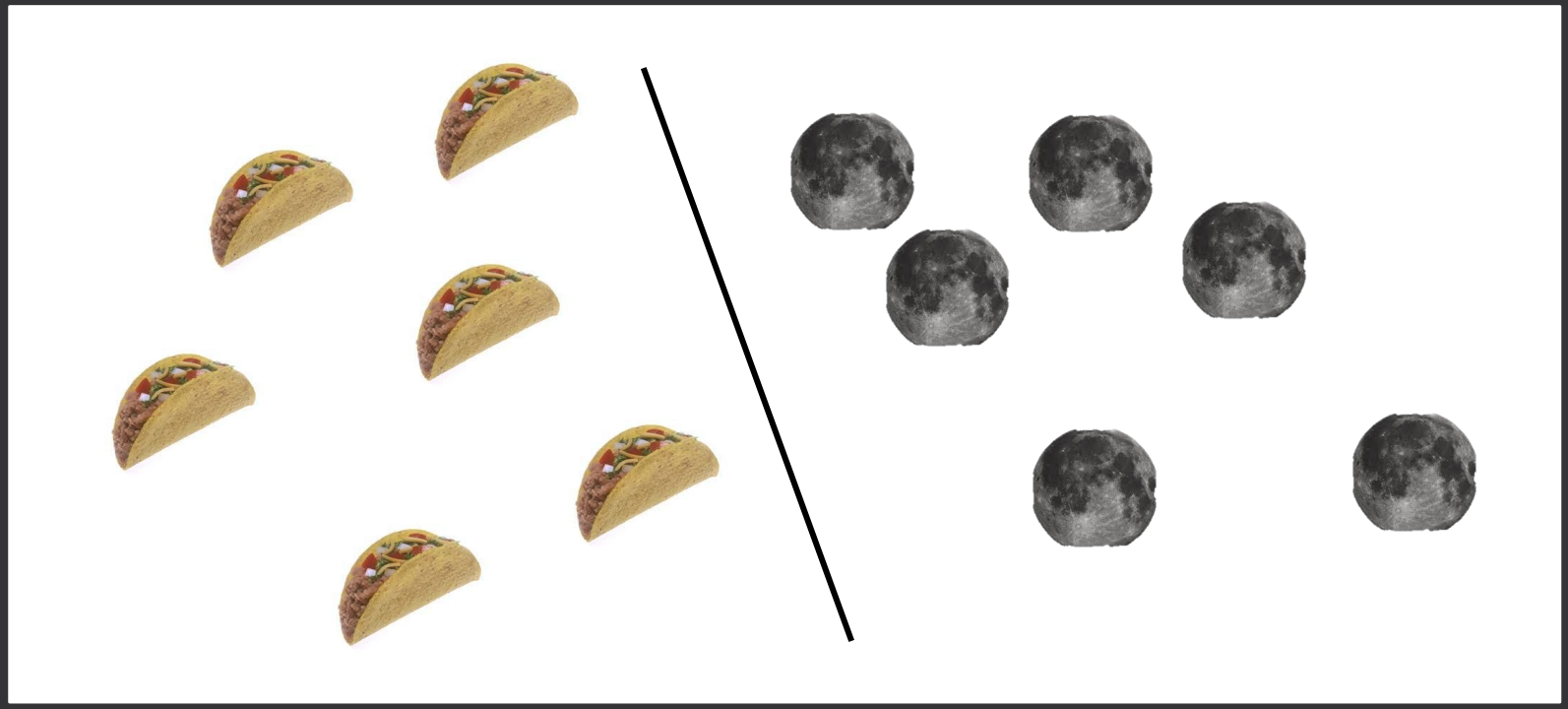
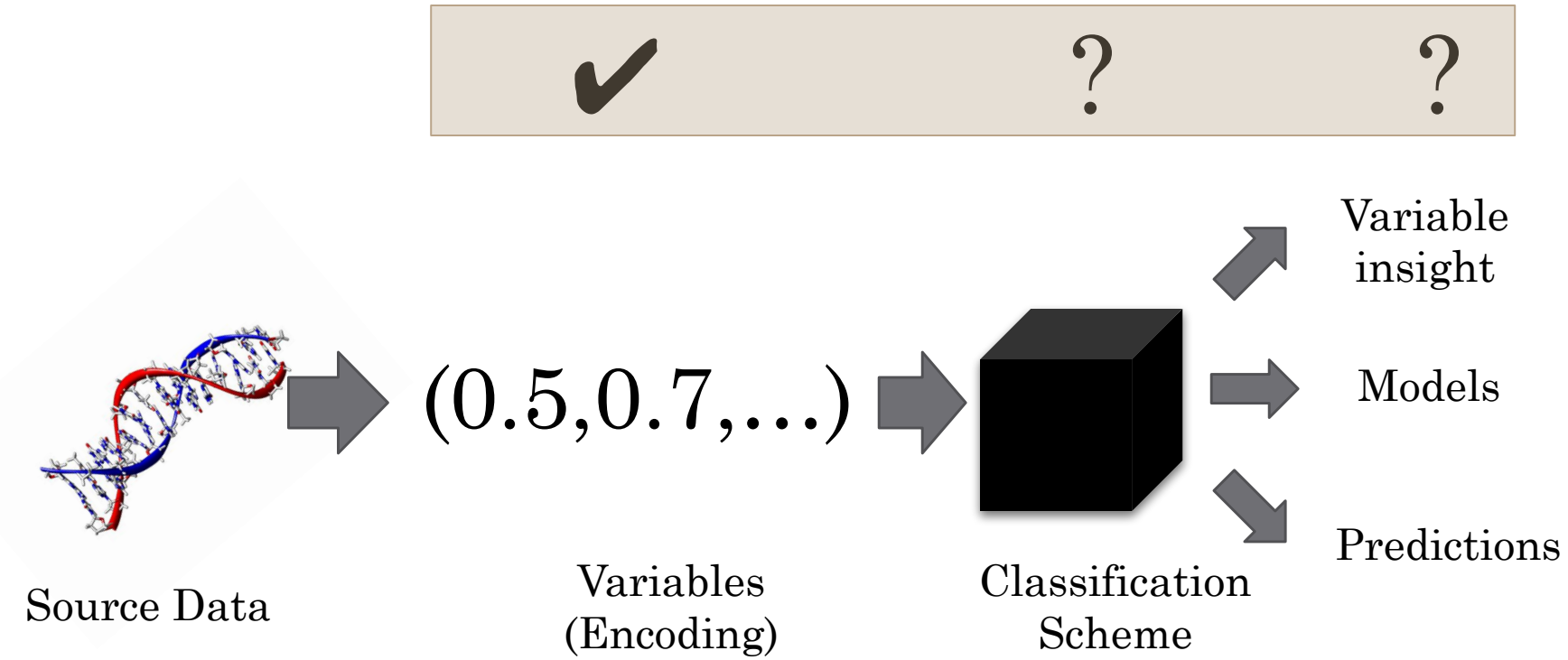


Training a classifier

CSCI 4181 / 6802 Module 1-TRAI

Overview

1. General properties of learning problems
2. Training, testing and quantifying accuracy
3. Choosing a classifier

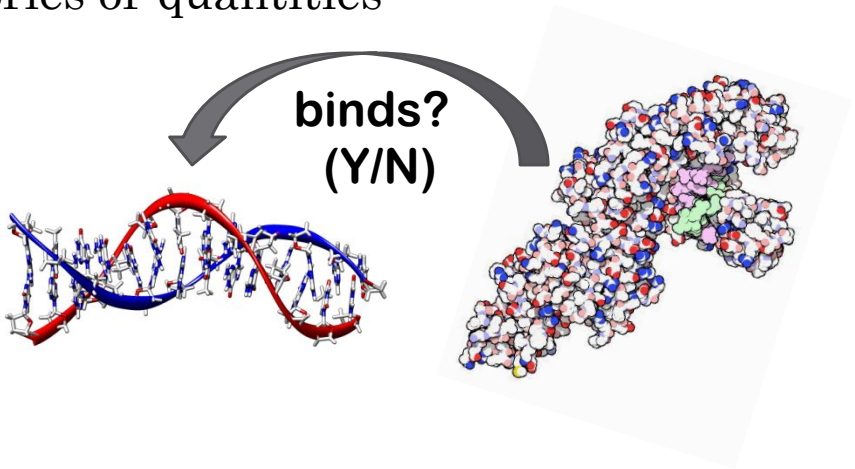

✔
?
?
Variable insight
Models
Predictions
(0.5,0.7,…)
Source Data
Variables (Encoding)
Classification Scheme

Learning Problems

Map input variables to categories or quantities

CLASSIFICATION

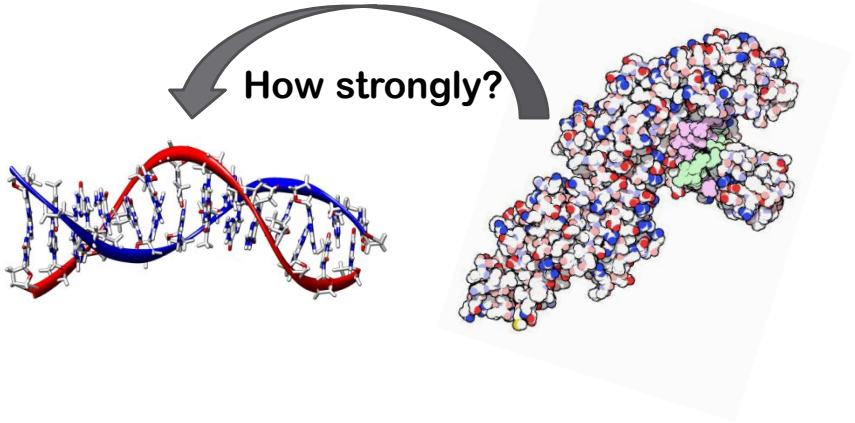
Predict qualitative traits (categories)



REGRESSION

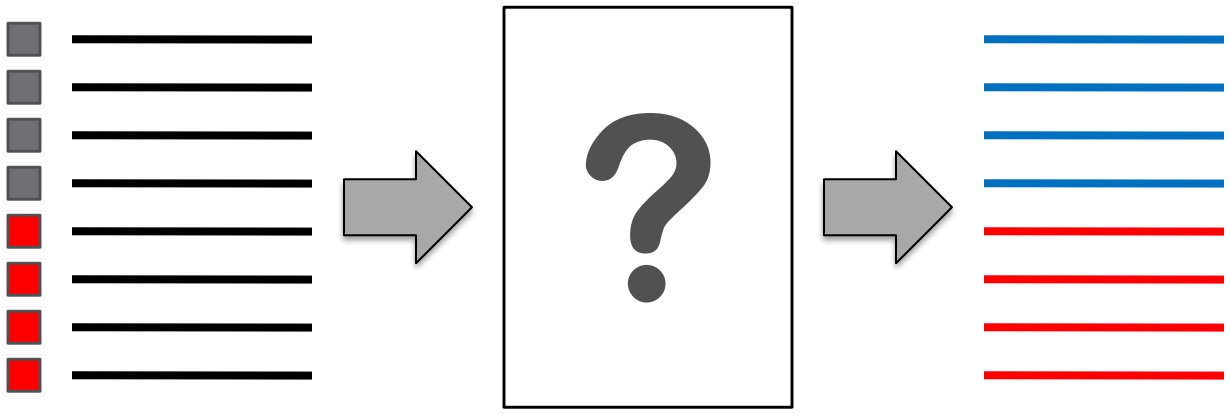
(and related methods)

Quantitative predictions



Training

Goal: to learn the rules (or fit functions) that distinguish classes



What are the properties of a good training set?

- Rnodam spamle from the population
- Sufficiently large
- All classes represented

Types of Learning

SUPERVISED

- Labeled classes
- Feedback: information about labeling is used to train classifier

UNSUPERVISED

- Classes may be labeled or unlabelled
- Classifier develops the classification scheme independently from class labels

SEMI-SUPERVISED

- Use both labeled and unlabeled data
- Unlabeled data can augment knowledge about probability distributions

REINFORCEMENT

- Identify optimal moves through a search space
- Good strategies are rewarded (consider short-term vs long-term tradeoffs)

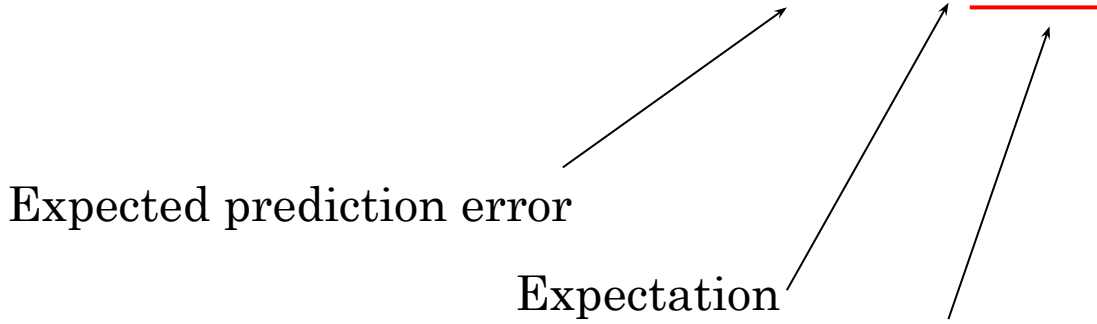
Goal of supervised learning

Minimize error

(via for instance a *loss function*)

on the training set

e.g., Squared error loss: $\mathrm{EPE}(f) = \mathrm{E}(\mathrm{Y} - f(\mathrm{X}))^2$



Hastie section 2.4

Some methods have closed-form solutions that are globally optimal on the cost function

- Many statistical methods e.g. discriminant function analysis, linear regression

Others must use heuristics (iterative training, greedy approaches)

- Neural networks
- Random forests
- Support vector machines

Generalization

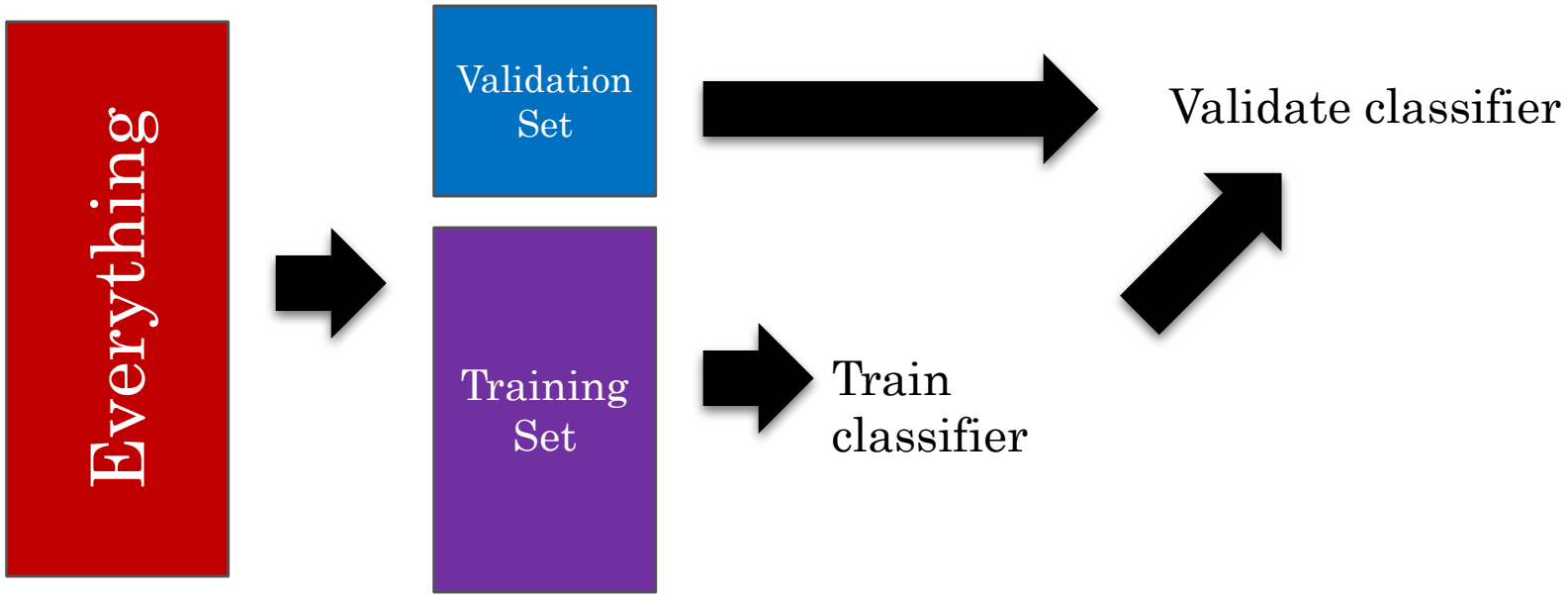
A classifier is of little use if it can only do well on data it has been trained on

How well does the classifier handle cases that were not present in the training set?

General form

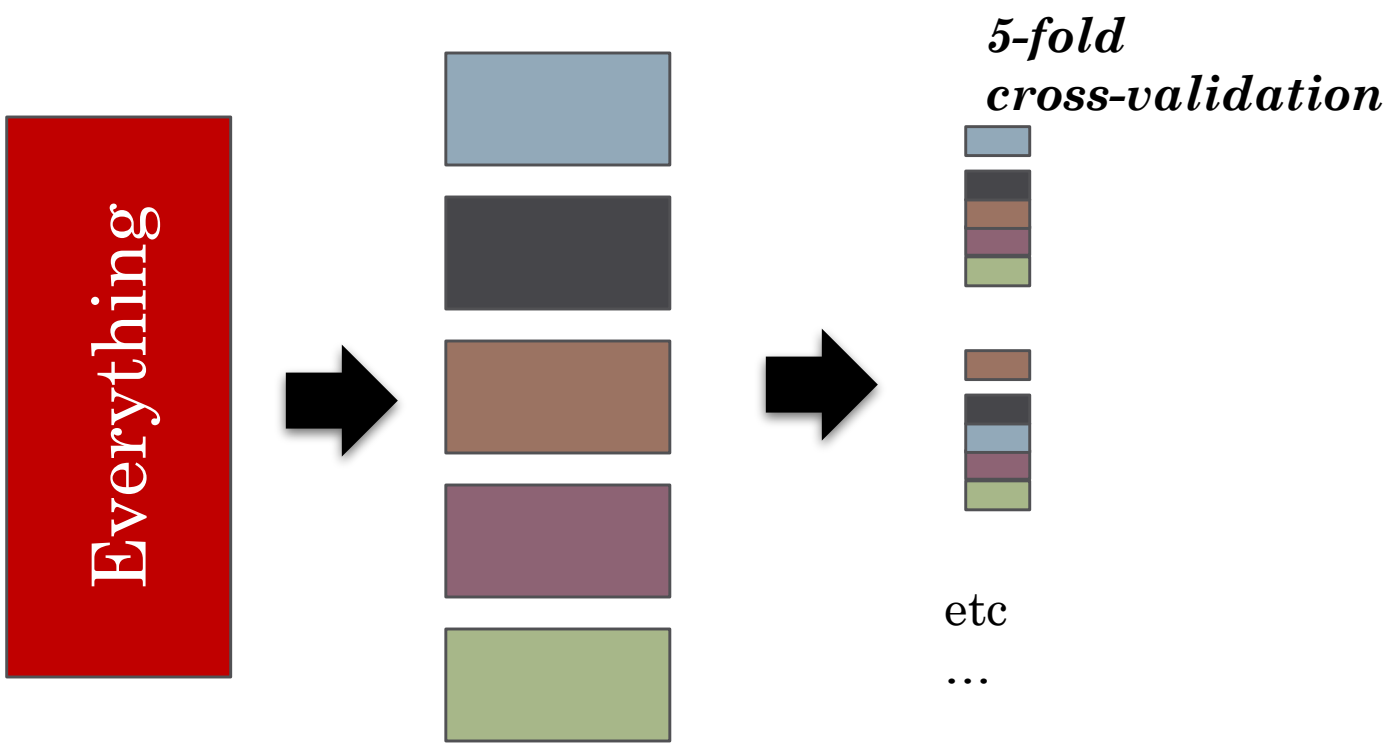
Data set splitting
(*holdout* method)

Use a fraction of available cases as the *training set*, reserve the remainder for a *validation* set



Cross-validation

Repeated training with different subsets



The *cross-validation score* is the average performance on all validation sets

Sample sets at random, but make sure every class is represented!

In the two-class case:

+ training set
- training set
+ validation set
- validation set

Classification Accuracy

CONFUSION MATRIX for a two-class (positive and negative set) problem

Predicted / True	+	-
+	True positive	False negative
-	False positive	True negative

may require THRESHOLDING of continuous predictions

Quantifying Accuracy

Sensitivity, recall, true positive rate	$= \frac{TP}{TP+FN} = \frac{TP}{n^+}$	Specificity, true negative rate	$= \frac{TN}{TN+FP} = \frac{TN}{n^-}$
Positive predictive value, precision	$= \frac{TP}{TP+FP}$	Negative predictive value	$= \frac{TN}{TN+FN}$
False positive rate, fallout	$= \frac{FP}{FP+TN} = \frac{FP}{n^-}$	False discovery rate	$= \frac{FP}{FP+TP}$

Don't forget that regularization may impact scoring!

Chicco and Jurman (2020) *BMC Bioinformatics*

Matthews Correlation Coefficient

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

F_1 Score

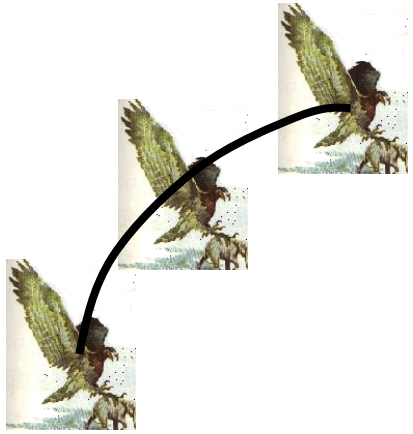
$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$

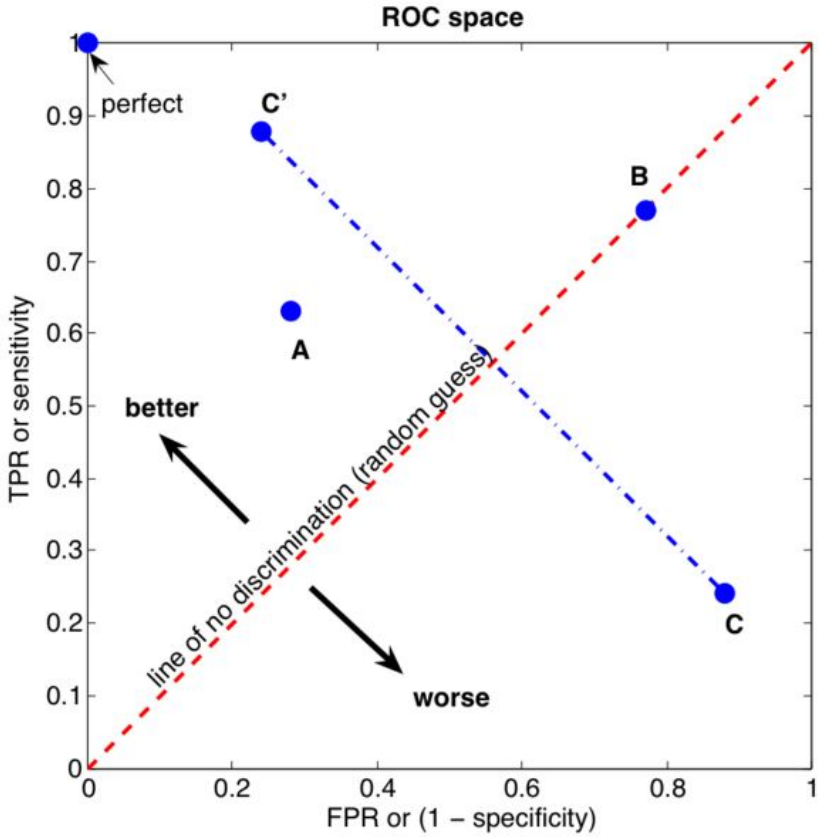
'Balanced' accuracy:

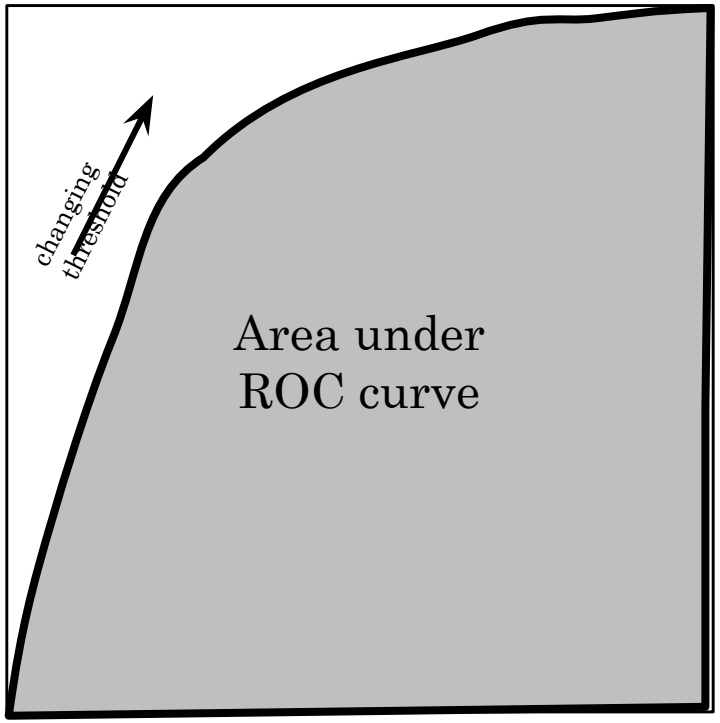
[**TP / (TP + FN)** + **TN / (TN + FP)**] / 2

Others: see Baldi et al. (2000) in *Bioinformatics*

ROC Curves







(Wikimedia Commons)

Regression problems

Mean absolute error

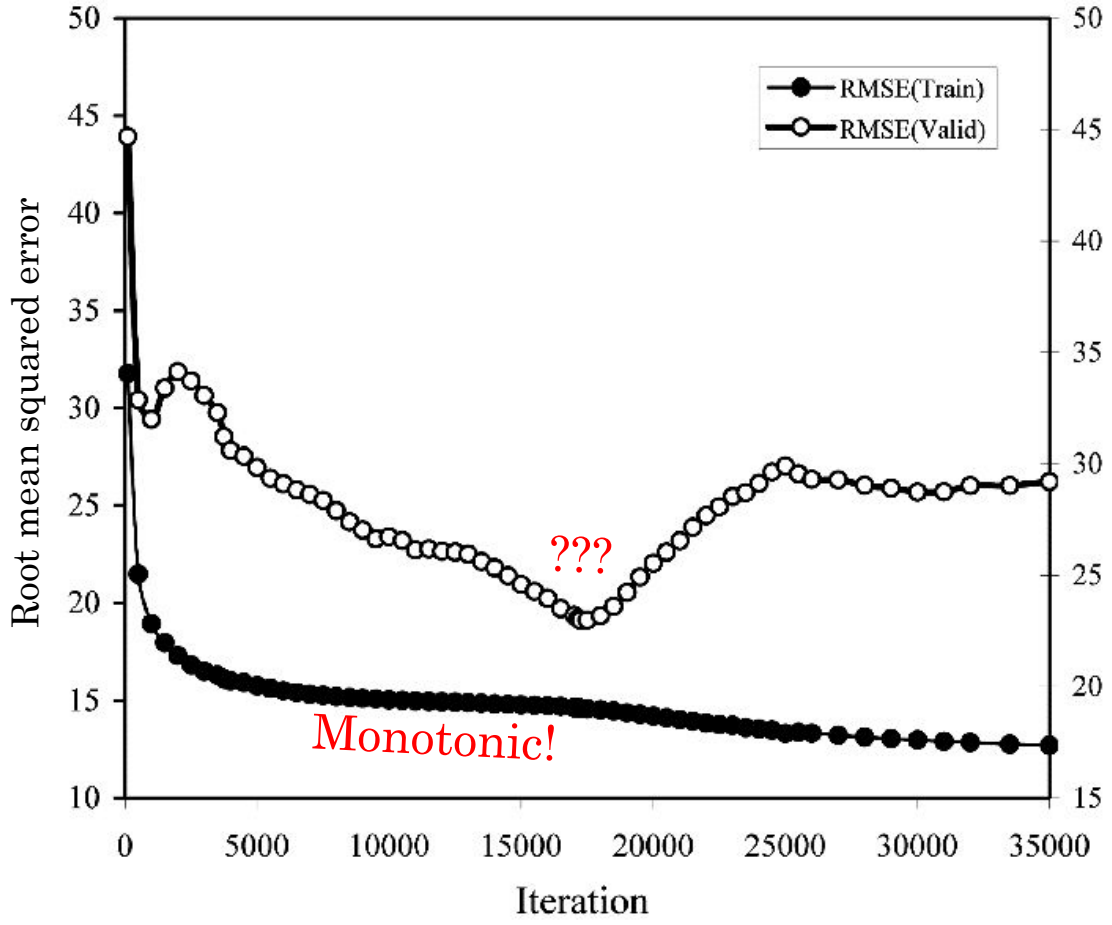
$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

Difference between predicted and true values

Root mean square deviation

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T} (\hat{y}_t - y_t)^2}{T}}.$$

Iterative training



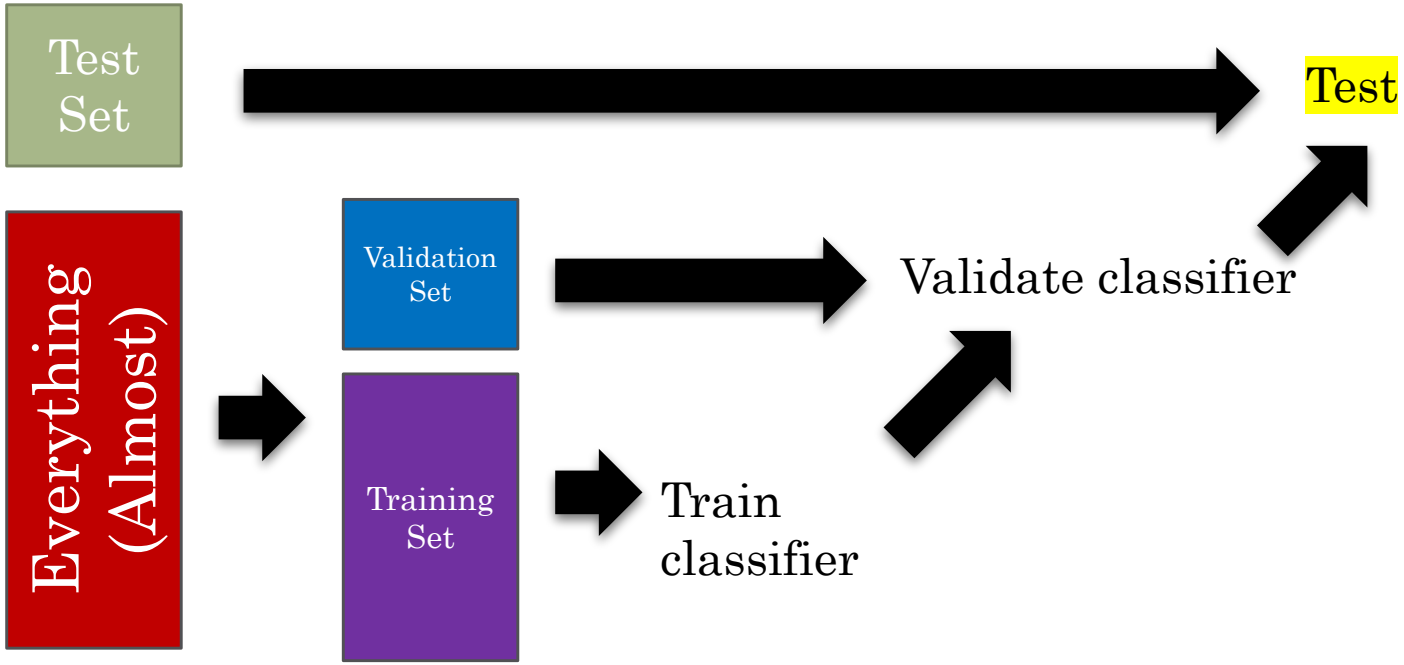
Training set accuracy improves, but at some point validation set accuracy may go boom

= OVERFITTING

Habibi-Yangjeh, Aziz, and Mahdi Esmailian. (2007) *Bull Kor Chem Soc*

Test set

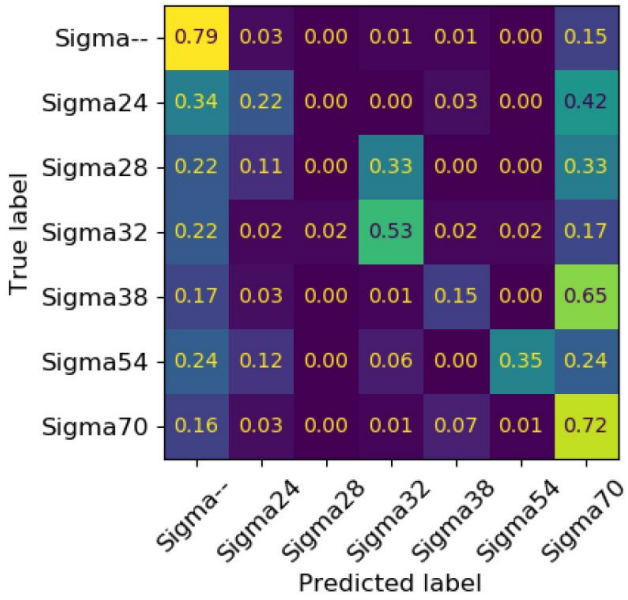
- With *k*-fold cross-validation, ALL data are used in training *k*-1 times
- So it is common practice to hold out part of the data entirely and assess only AFTER cross-validation / parameter selection has been completed



Expanding to multiple classes

- There's nothing special about "Positive" and "Negative" classes – invert the labels and corresponding scores would either change or map deterministically
- Do we weight all classes equally, or do we weight by abundance?

Promoter predictions by class – different promoter types are active at different times



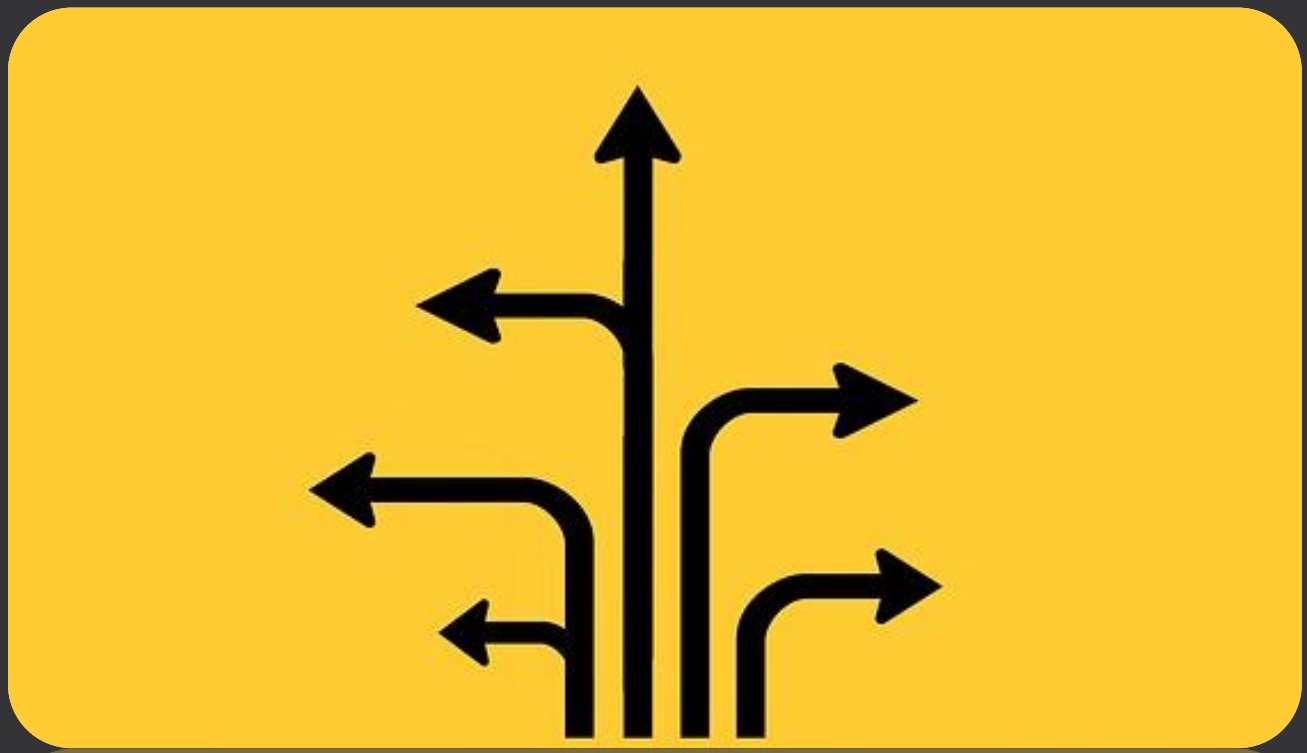
Rafante and Beiko (submitted)

Key questions

1. Which is more desirable, sensitivity or specificity?

2. How many folds of cross-validation is the right number of folds of cross-validation?

3. What is the value of our classifier if the accuracy on the test set is 60%?



Simple classifiers

CSCI 4181 / 6802

http://pixabay.com/en/road-sign-arrows-arrow-direction-64059/

Overview

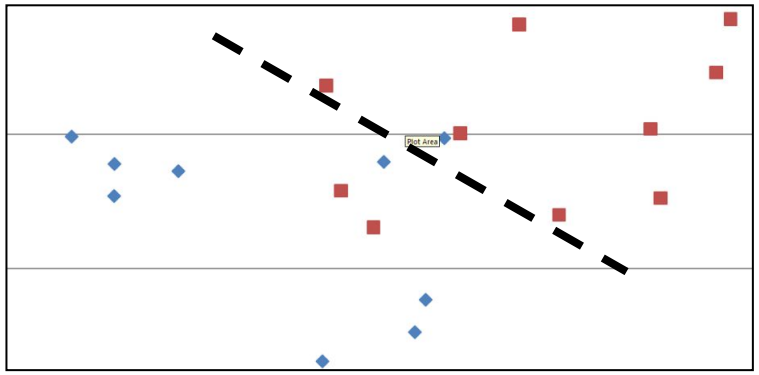
1. Decision criteria for choosing a classifier
2. Naïve Bayes
3. Random Forests
4. CART
5. K-NN

No one classifier is best for every classification problem

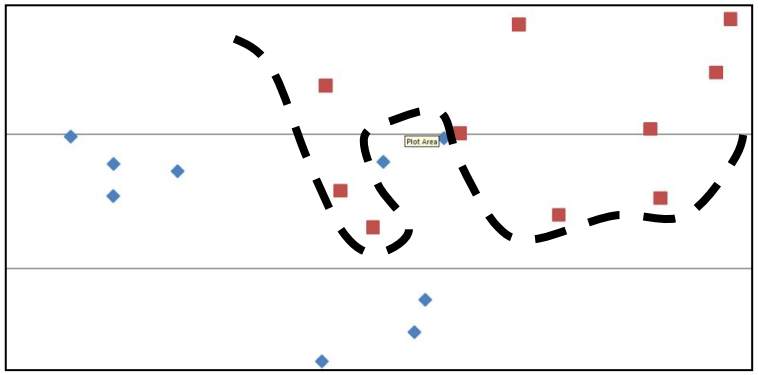
What criteria should we consider?

Bias-Variance Tradeoff

Do we want a classifier that is as simple as possible, or one that can make complex decisions?



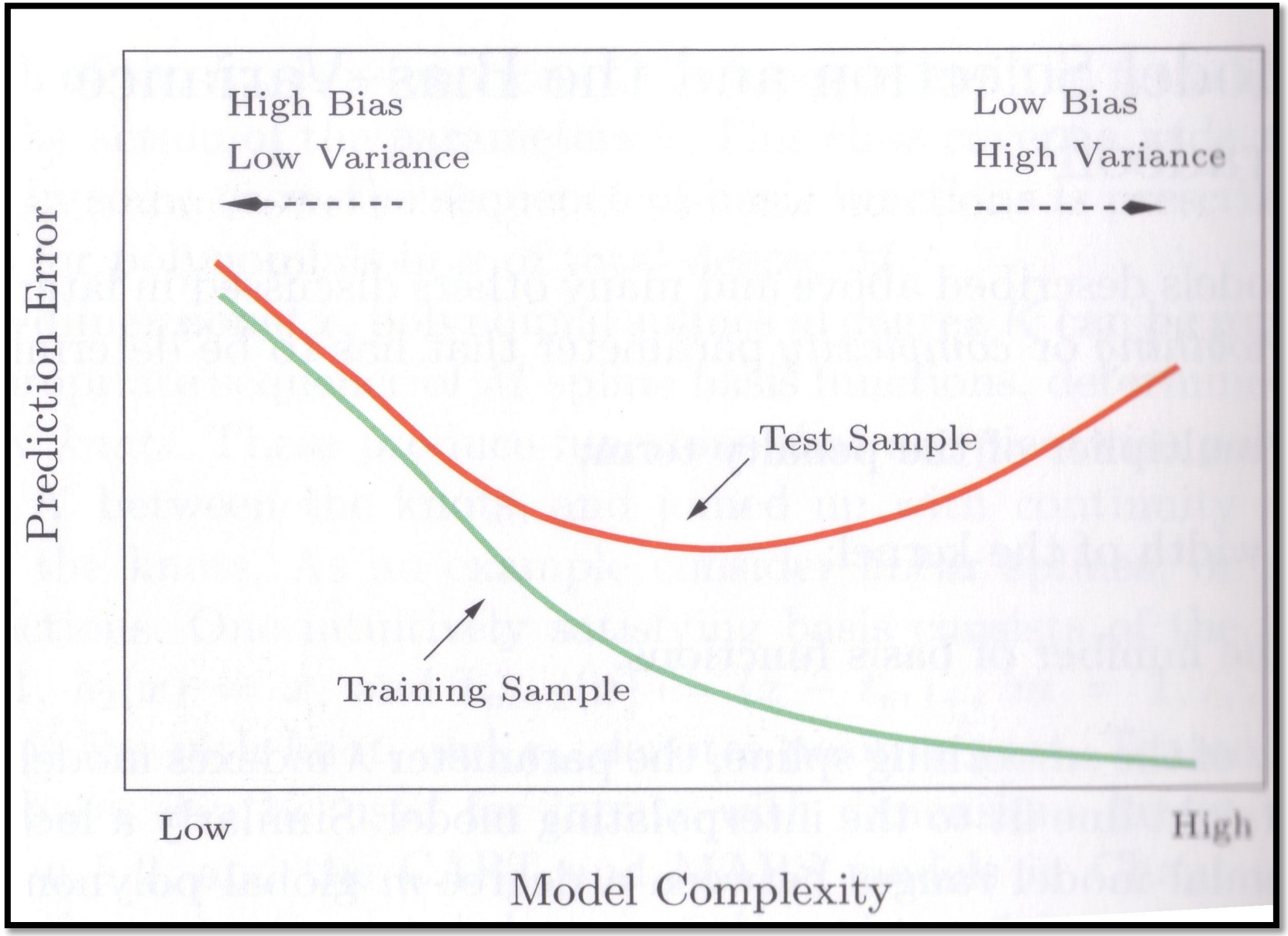
Bi
as



Variance
(overfitting)

the ability of the machine to learn any training set without error. A machine with too much capacity is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything she has seen before; a machine with too little capacity is like the botanist's lazy brother, who declares that if it's green, it's a tree. Neither can generalize well. The exploration and

Burges 1997, "A Tutorial on Support Vector Machines for Pattern Recognition".

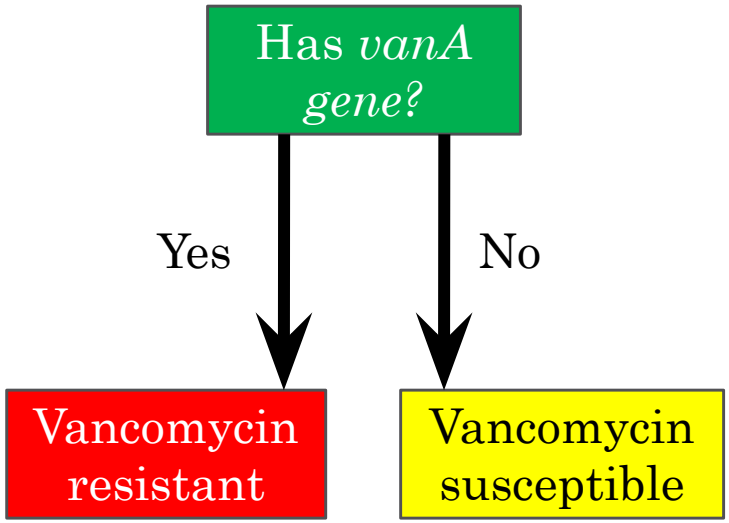

High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Training Sample
Low
High
Model Complexity

Hastie, p.38

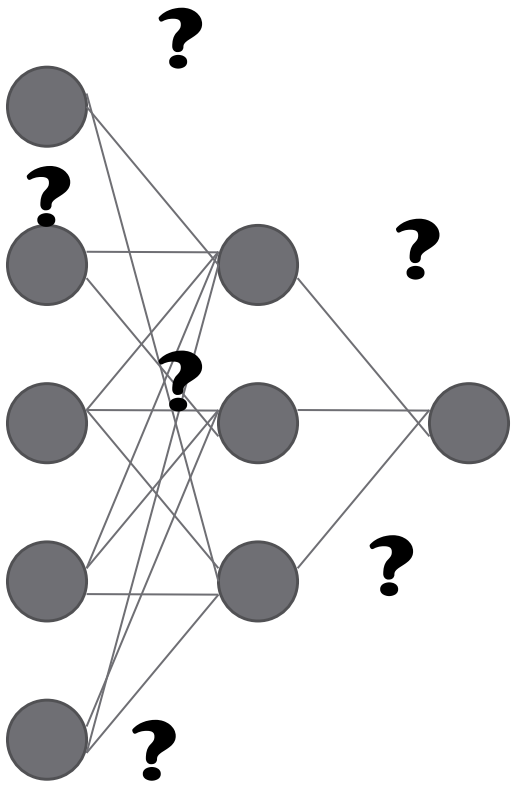
Interpretability

Some methods yield understandable (or almost understandable) rules, others do not

Decision tree



Artificial neural network



Tractability

If the training data are necessarily high-dimensional, then a simpler classifier may be necessary

(or we need to be more aggressive in our feature selection / extraction)

Naïve Bayes

http://www.polarbearsinternational.org/sites/default/files/legacy/D216674.jpg

The point

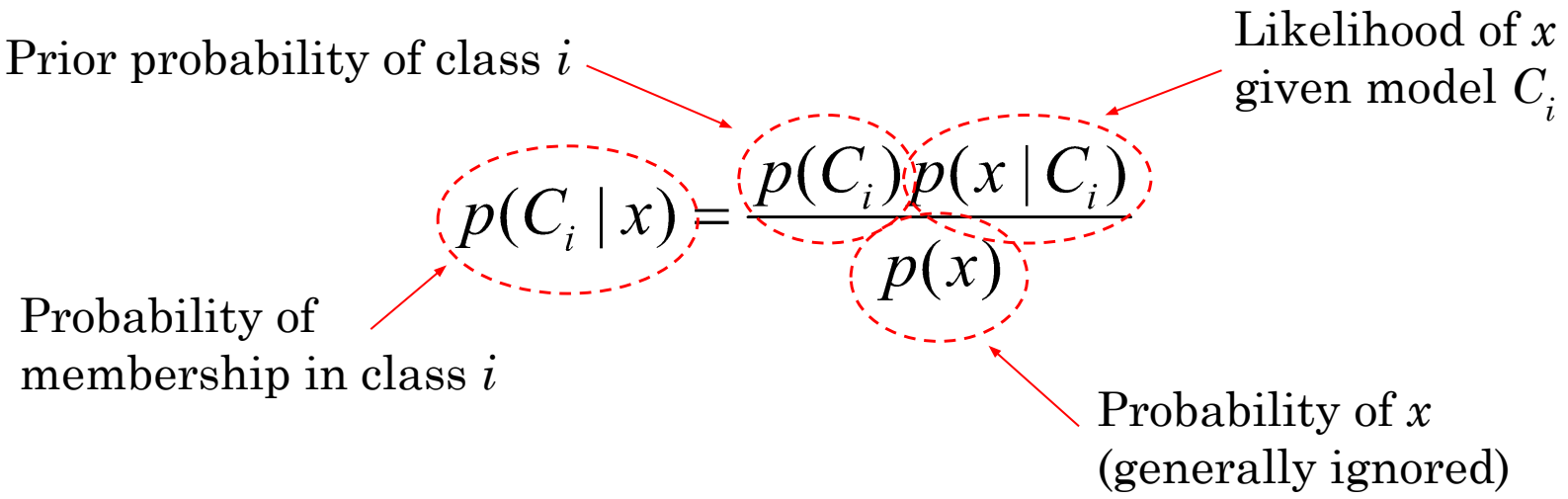
- Assign samples to different classes using a probabilistic approach
  - Think of it as competitive matching based on distributions of features
- Features are treated independently
  - A simplifying assumption that makes NB very fast
- Classes are assigned probabilities which can be modified by priors

Naïve Bayes

For a set of n classes C_1, C_2, C_3, …, C_n,

and a problem instance x

(usually represented with a feature vector),



Predicted class: C_i that maximizes $p(C_i \mid x)$

Priors

What are the expected probabilities of different classes?

Flat prior: $p(C_1) = p(C_2) = p(C_n) = 1 / n$

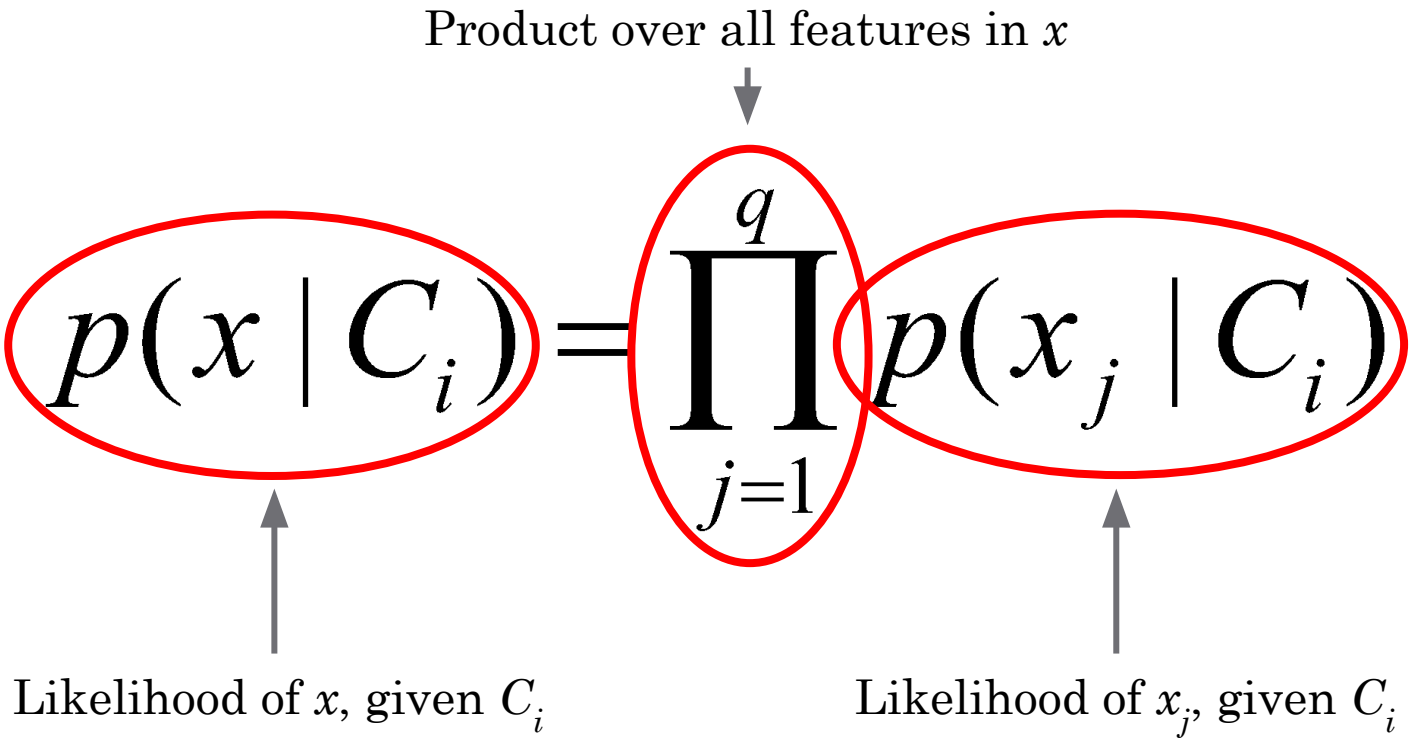
Informative prior: $p(C_i) \neq p(C_j)$ for some $i \neq j$

(based on what?)

Calculating likelihoods

"The probability of the data, given the model"

What is the likelihood of x, given class C_i?



Calculating likelihoods: the independence assumption

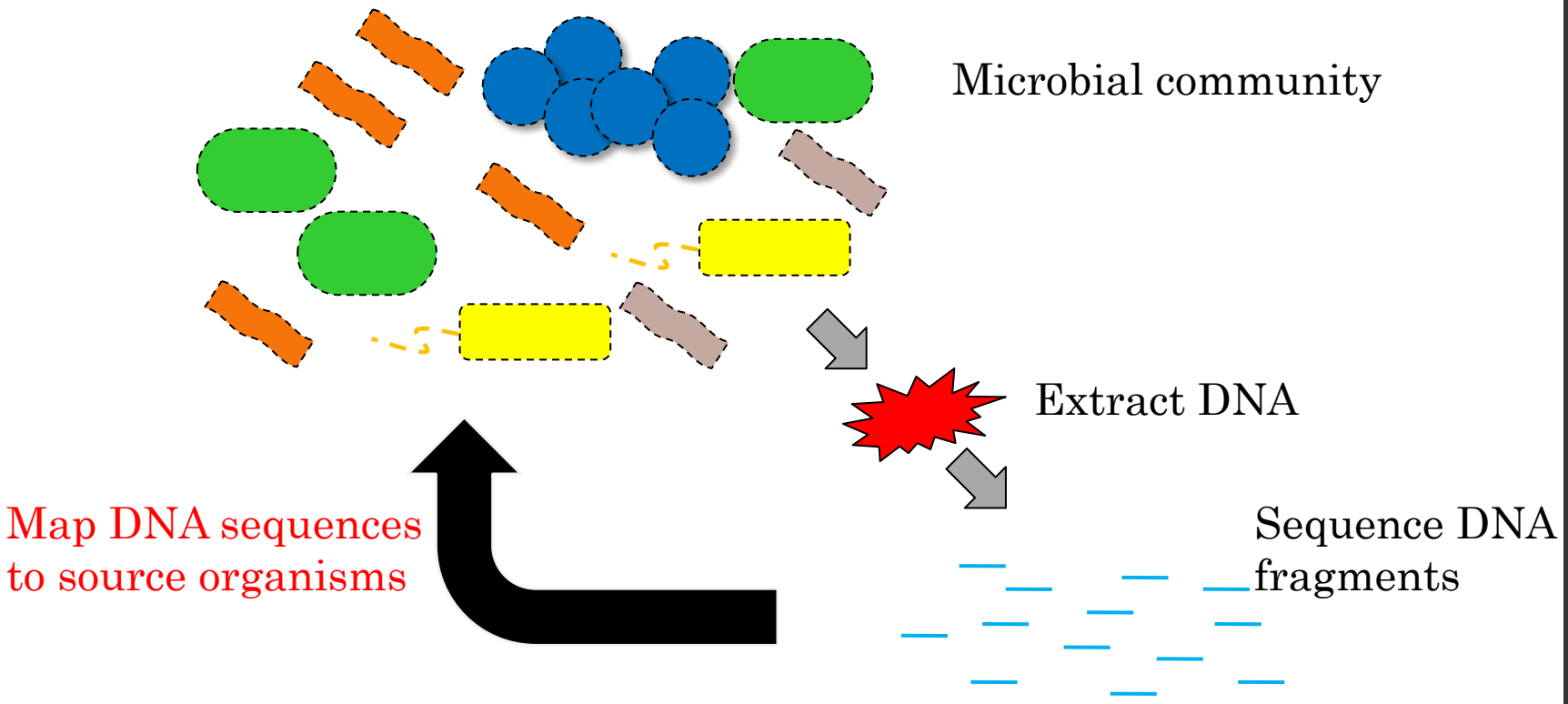
The calculation can be very complicated if x has many elements and we consider all possible dependencies among elements

(as most classifiers do)

The solution is to treat each element of x independently

NB example

Fragment Classification Package (FCP): Assigning DNA sequence fragments to originating organisms from a metagenomic sample



Parks DH, MacDonald NJ, and Beiko, RG (2011) *BMC Bioinformatics*

How do we classify these sequences?

k-mer decomposition: Build compositional models for each genome in a reference database

C_i	x
E. coli	{ AA = 0.05, AC = 0.03, AG = 0.08, AT = 0.04, CA = …
Y. pestis	{ AA = 0.05, AC = 0.04, AG = 0.10, AT = 0.02, CA = …
C. difficile	{ AA = 0.01, AC = 0.05, AG = 0.08, AT = 0.02, CA = …
E. faecium	{ AA = 0.03, AC = 0.03, AG = 0.09, AT = 0.01, CA = …

How do we classify these sequences?

k-mer decomposition: Build compositional models for each genome in a reference database

$$p(x \mid C_i) = \prod_{j=1}^{q} p(x_j \mid C_i)$$

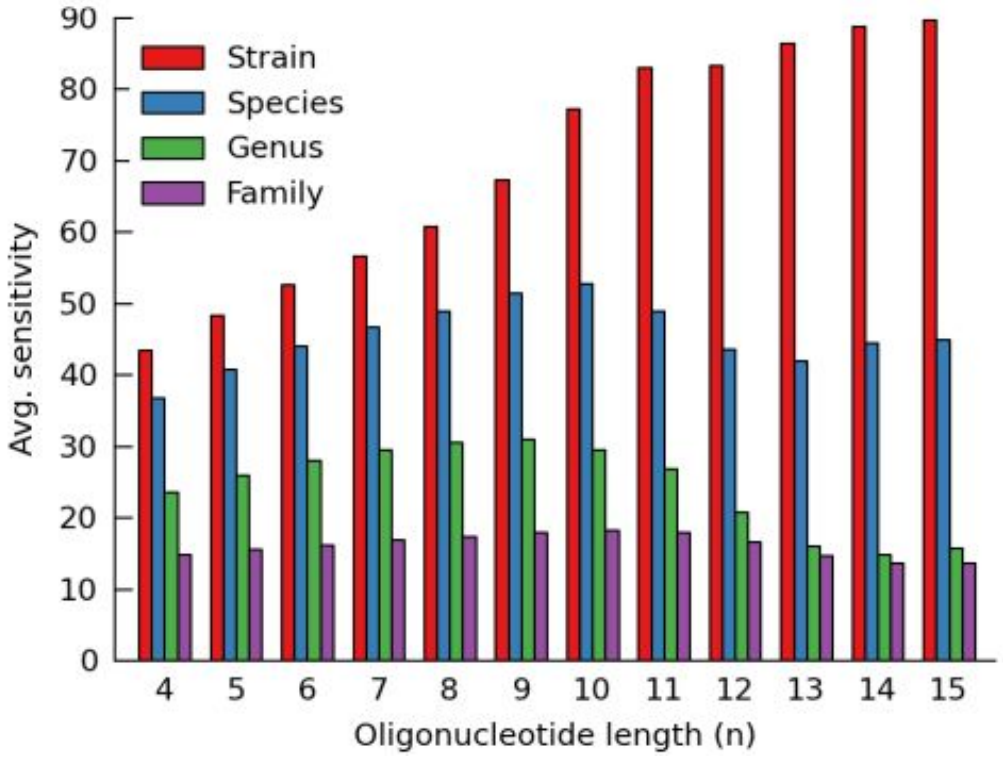
For all k-mers in fragment x…

Equal to the frequency of the k-mer j in model (= genome) C_i

The "winning" genome is the one that maximizes the likelihood (assuming a flat prior)

Trials

Simulated data



Optimal k = 9 from trials

Challenges in sequence classification

(1) *k*-mer frequencies are averages calculated over the entire genome: individual genes can and do vary

> Worst offenders: recent acquisitions from other genomes (e.g. plasmids!), viral genes, rapidly evolving genes

(2) We are restricted to the genomes in our training set; new genomes are not modeled

> "rank-flexible" classification: classify at a reasonable taxonomic level

(3) Sparse representations when *k* is large

> Spoiler: it doesn't seem to matter all that much (we checked!)

Parks DH, MacDonald NJ, and Beiko, RG (2011) *BMC Bioinformatics*

Example: classifying glacier metagenomes

Rank flexible

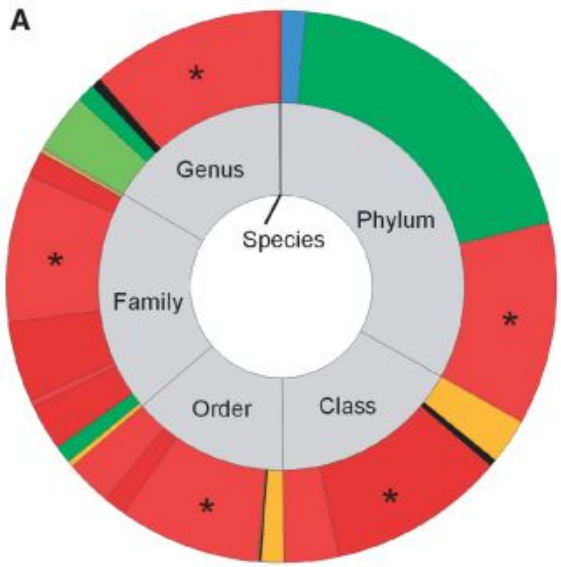


Figure 5. RITA classifications of the glacier metagenome of (26). (**A**) Rank-flexible classifications in Groups 1–3 to ranks between species and phylum. The inner ring identifies the rank at which different fragments were classified, while the outer ring shows the distribution across different labels at that rank, colored by the phylum to which the taxon belongs. Phylum colors: blue = Acidobacteria, green = Bacteroidetes, red = Proteobacteria, orange = Actinobacteria, black = other. Alternating shades of the same color are used to distinguish different taxa at the same rank from the same phylum. The taxonomic lineage of *Polaromonas* is identified with asterisks. (**B**) Rank-specific classifications at the phylum (outer ring) and genus (inner ring) levels, with color scheme as in panel A. Deepest red and green represent aggregated 'other' genera of Proteobacteria and Bacteroidetes.

MacDonald NJ, Parks DH, Beiko RG (2012) Rapid identification of high-confidence taxonomic assignments for metagenomic data. *Nucleic Acids Res.*

What about the independence assumption?

AGGGCCTAGCATT gets decomposed into

AGGG, GGGC, GGCC, GCCT, …

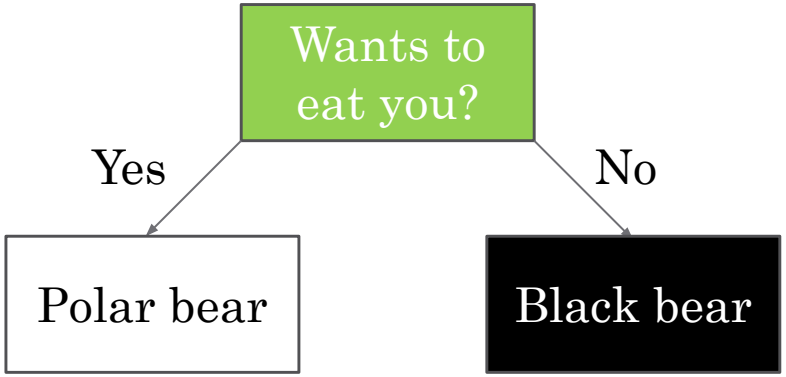
Which will be highly correlated!

Whatever.

Random Forests

Decision trees

- Classify two or more groups by creating *decision nodes* based on specific criteria



Training

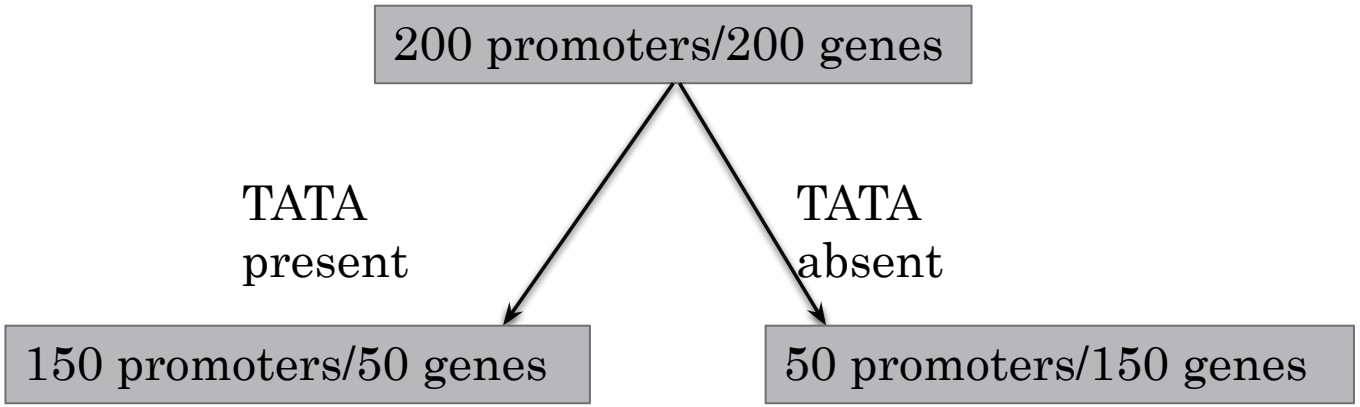
Decision Node (cases c *consisting of variables* x_{ic}*):*

If (stopping criterion not reached)

1. Find variable x_i and threshold t such that optimal separation is achieved between cases with different labels
2. *Decision Node* ($c_{xi} \leq t$)
3. *Decision Node* ($c_{xi} > t$)

Optimal separation

Minimize node 'impurity'



Misclassification error: % of cases not belonging to majority class

Gini index: $\sum_{c} \left(\Pr(c)\right) \times \left(1 - \Pr(c)\right)$

Cross-entropy: $-\sum_{c} \left(\Pr(c)\right) \times \log \Pr(c)$

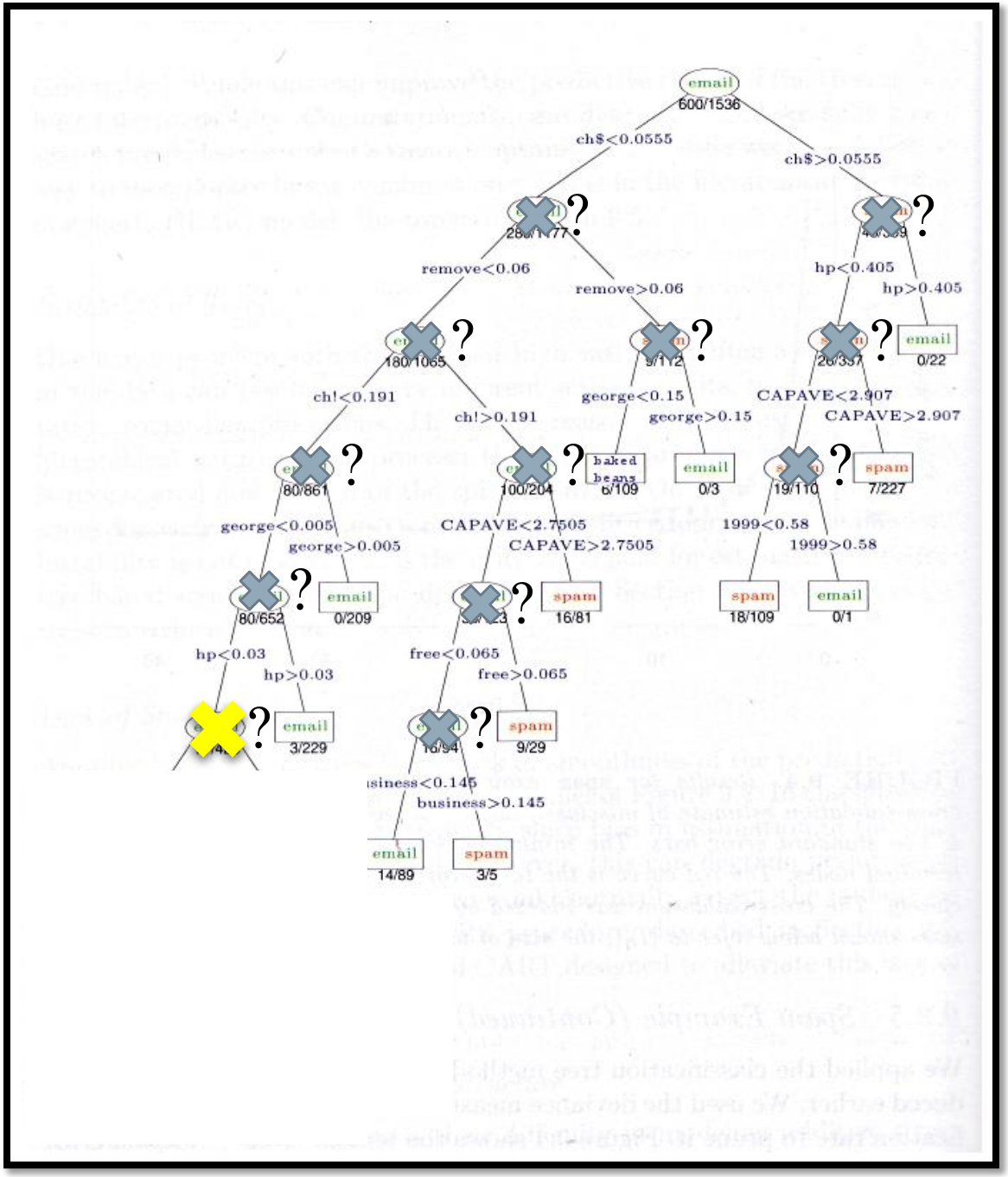
A greedy approach is most common

Is this guaranteed to find the best solution?

Of course not!!!

Stopping criteria

- There is no need to subdivide a pure class
- Other criteria (such as minimum number of classes at a node) might be used as well
- We can use *pruning* strategies to roll back a tree and achieve an optimal balance between size and separation (**bias** vs. **variance**)



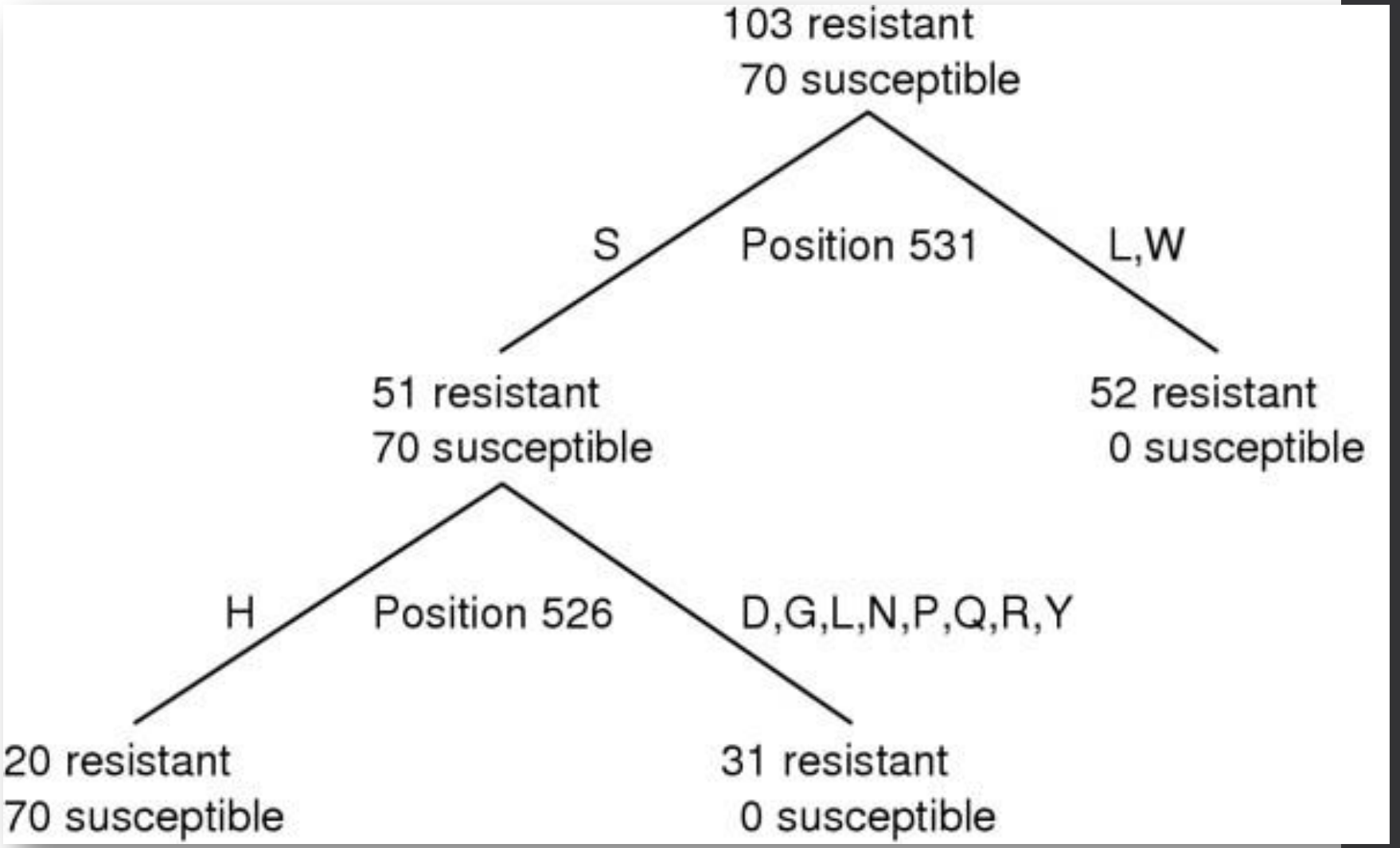
Weakest link pruning:

Find the subtree T_α that minimizes a cost function that balances accuracy and complexity

Rifampin sensitivity of bacterial RNA polymerase

- Rifampin interferes with transcription in *Mycobacterium tuberculosis*
- However, mutations can arise in certain parts of the *rpoB* gene that lead to rifampin resistance
- Can we classify *rpoB* variants based on the amino acids encoded by the gene?

Cummings *et al. BMC Bioinformatics* 2004 **5**:137



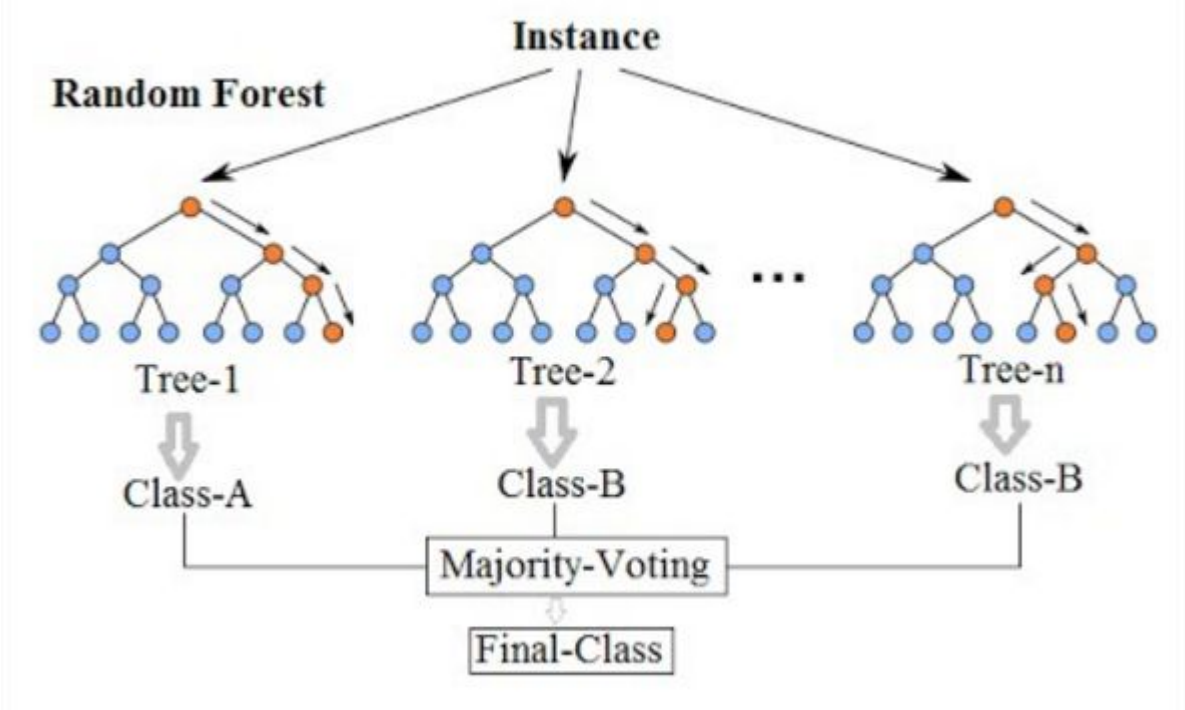
88.4% correctly classified (10-fold cross-validation)
Many nearby polymorphic sites (e.g. 511, 512, 515, 521 and 529) **rejected** as potentially good classifiers

That's great, but…

- Decision trees are **very** susceptible to overfitting during training
- Decisions are based on the training set and "nearly neutral" splitting decisions cannot be revisited
- So….?

Random forests:

why use only one tree when you can use many?



These trees must not all make the exact same predictions!

RFs part 1: trees, trees, trees!

- Each tree is trained on a randomly regenerated sample of the dataset, *with replacement*

Training Dataset: $\{T_1, T_2, T_3, T_4, \ldots, T_n\}$

Bootstrapped dataset 1: $\{T_1, T_2, T_3, T_2, \ldots, T_n\}$

Bootstrapped dataset 2: $\{T_3, T_1, T_3, T_4, \ldots, T_n\}$

Bootstrapped dataset 3: $\{T_4, T_4, T_1, T_1, \ldots, T_n\}$

Each time we *overrepresent* some cases, and *eliminate* others = boostrap aggregation (bagging)

RFs part 2: playing with features

- At each node in each tree, select only a random *subset* of features to draw from for decision making

- ???

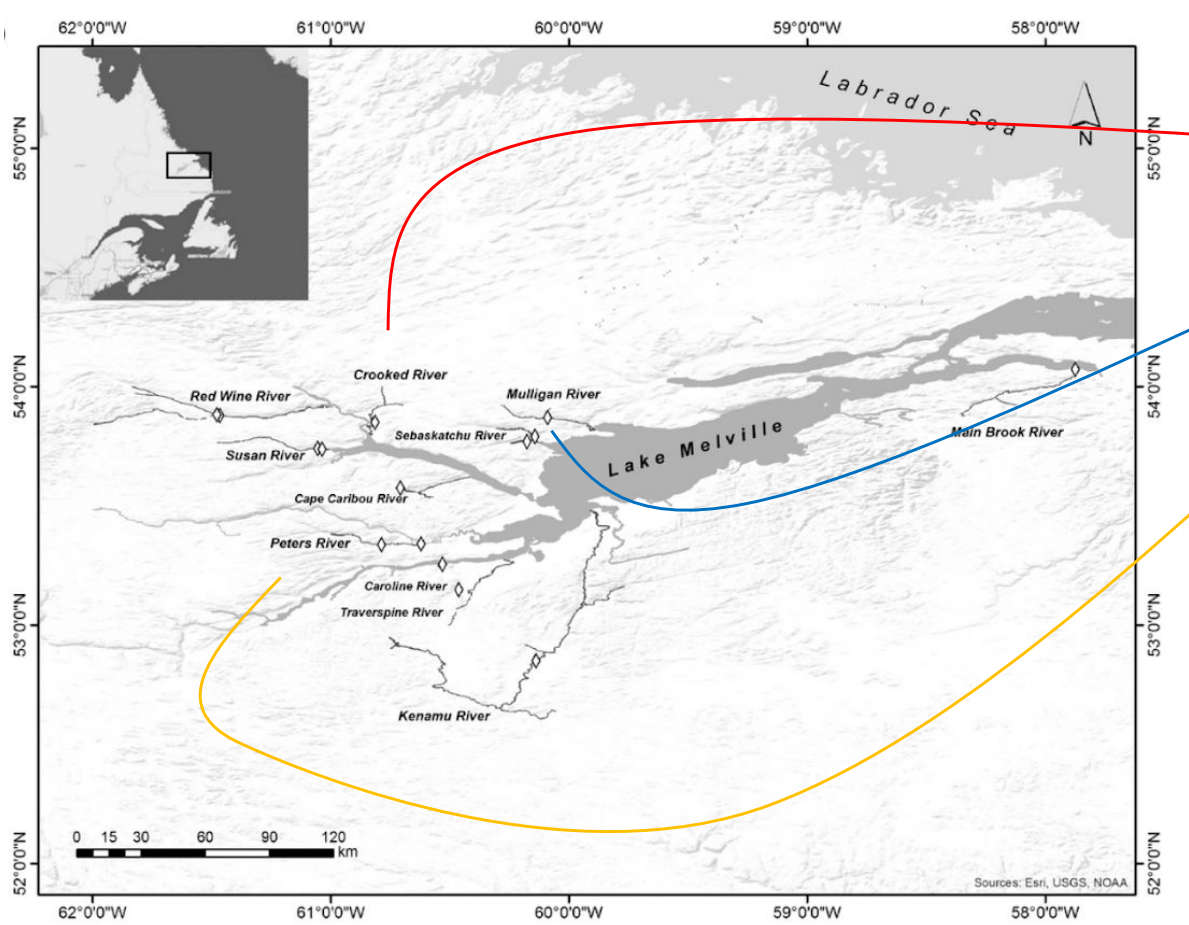
- The big idea: if you use the same features every time, **you'll get the same tree many times**. Random subset selection addresses this.

RFs: the point

- Individual trees can overfit, but taking many trees (averaging or voting) can smooth out the consequences of overfitting

- RFs can be very accurate, even if many of the individual trees aren't very good!

Example: fish!



Salmon breed here, here, here, etc.

Atlantic salmon are **anadromous**

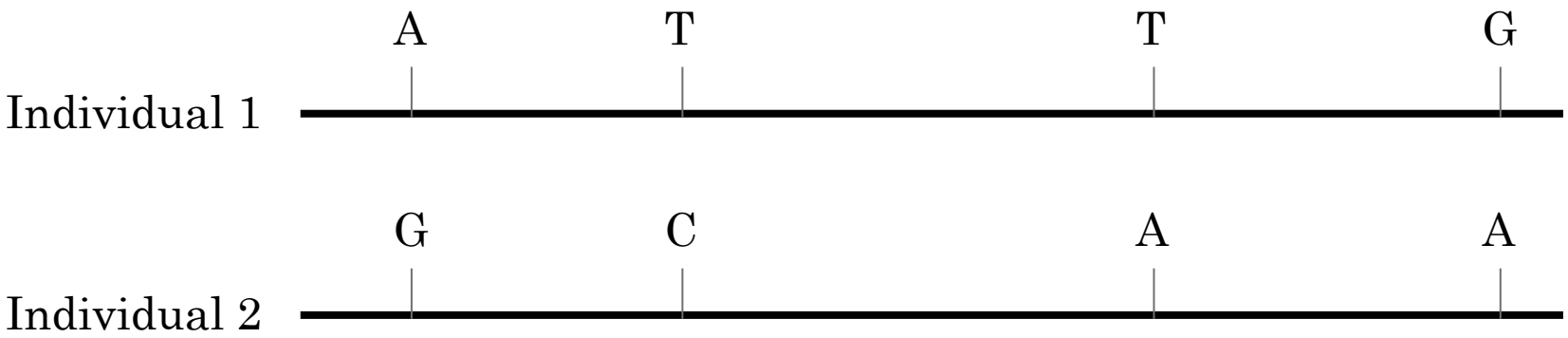
But are they the same population?

Sylvester et al. (Feb 2018) *Evol Applications*

One fish, two fish, Kenamu fish

Every fish has its own distinctive genotype, with mutations that distinguish it from other members of the species

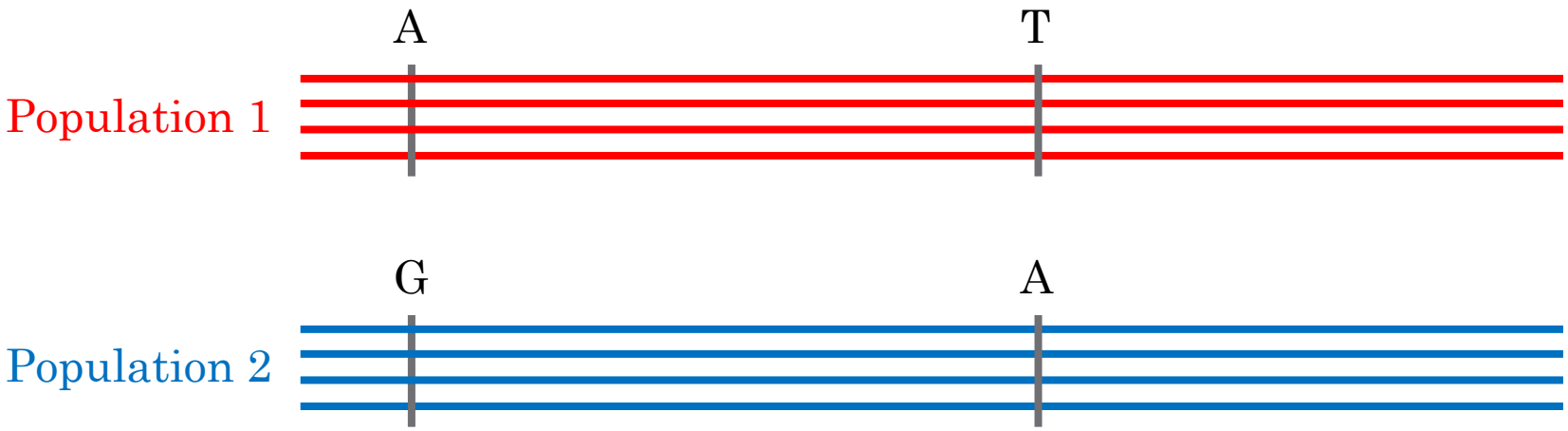
Single Nucleotide Polymorphisms (SNPs)



Sylvester et al. (Feb 2018) *Evol Applications*

Can we distinguish populations?

Sets of features that distinguish individuals by their river of origin suggest that they constitute separate populations and should be treated as such



Of course it isn't that simple, which is why we need RFs!
Starting point: 220,000 salmon SNPs
Real starting point: 8000 salmon SNPs

Sylvester et al. (Feb 2018) *Evol Applications*

Step 1: feature selection!

Mean decrease in accuracy: train RFs without a given SNP, and assess the impact on classification accuracy

Which features consistently gave substantial MDAs across many trials?

Click to add text

Click to add text

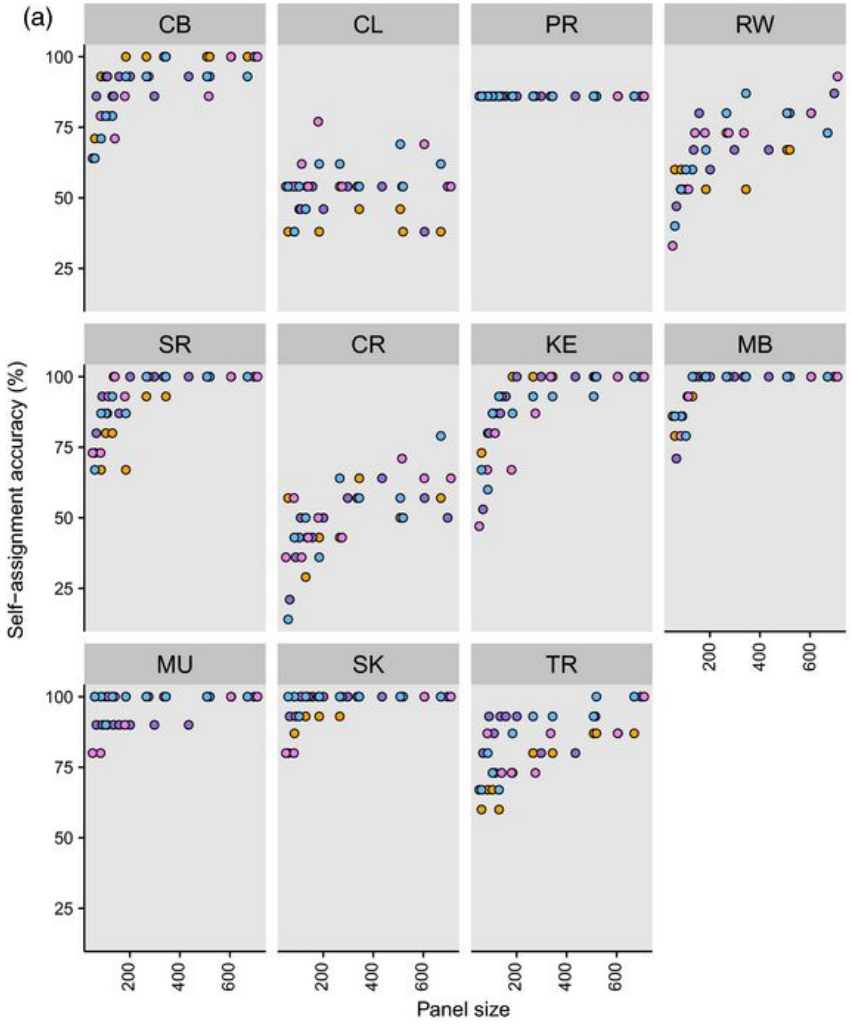
Interestingly, relatively few features came up again and again (due to redundancy among features?)

Use this set of features for subsequent classification

Best test set error with 2000 trees, and twice the square root of the number of features considered at each node

Sylvester et al. (Feb 2018) *Evol Applications*

Step 2: actually classifying!



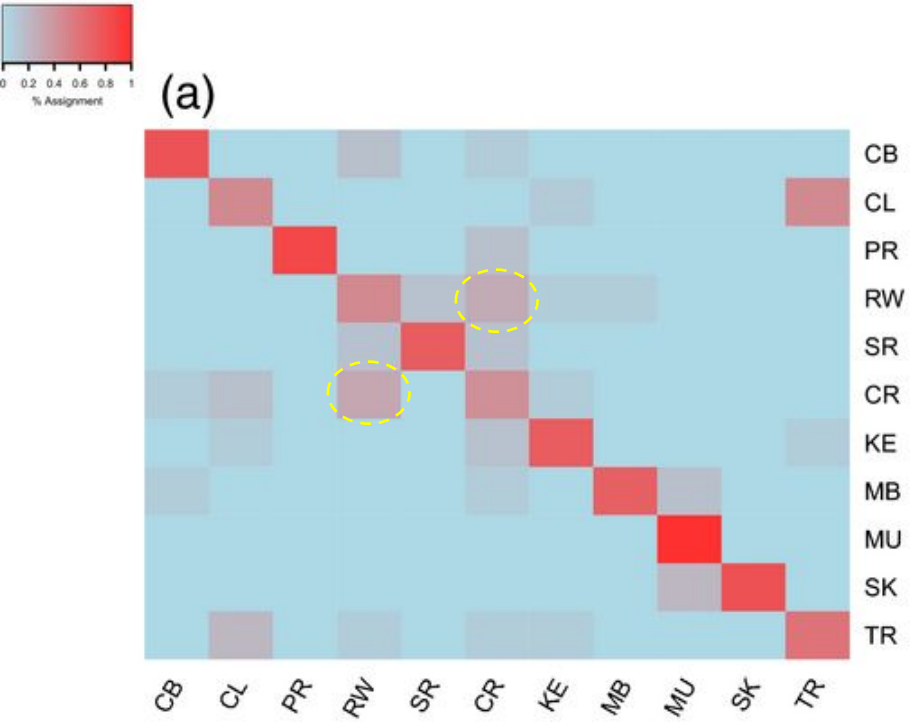
River by river, four types of classification (different coloured dots)

Note:

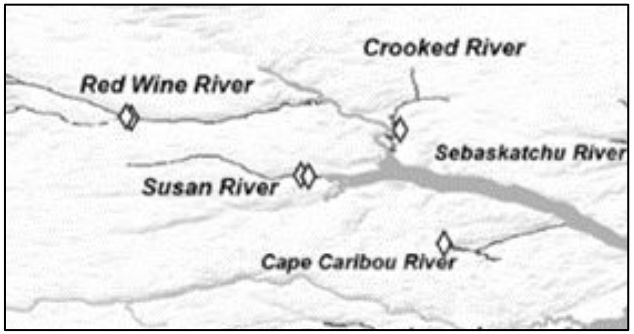
- Many rivers have near-perfect accuracy
- Some rivers (RW, CR) are awful

Sylvester et al. (Feb 2018) *Evol Applications*

Misclassifications are not random



CR and RW are frequently misclassified as one another



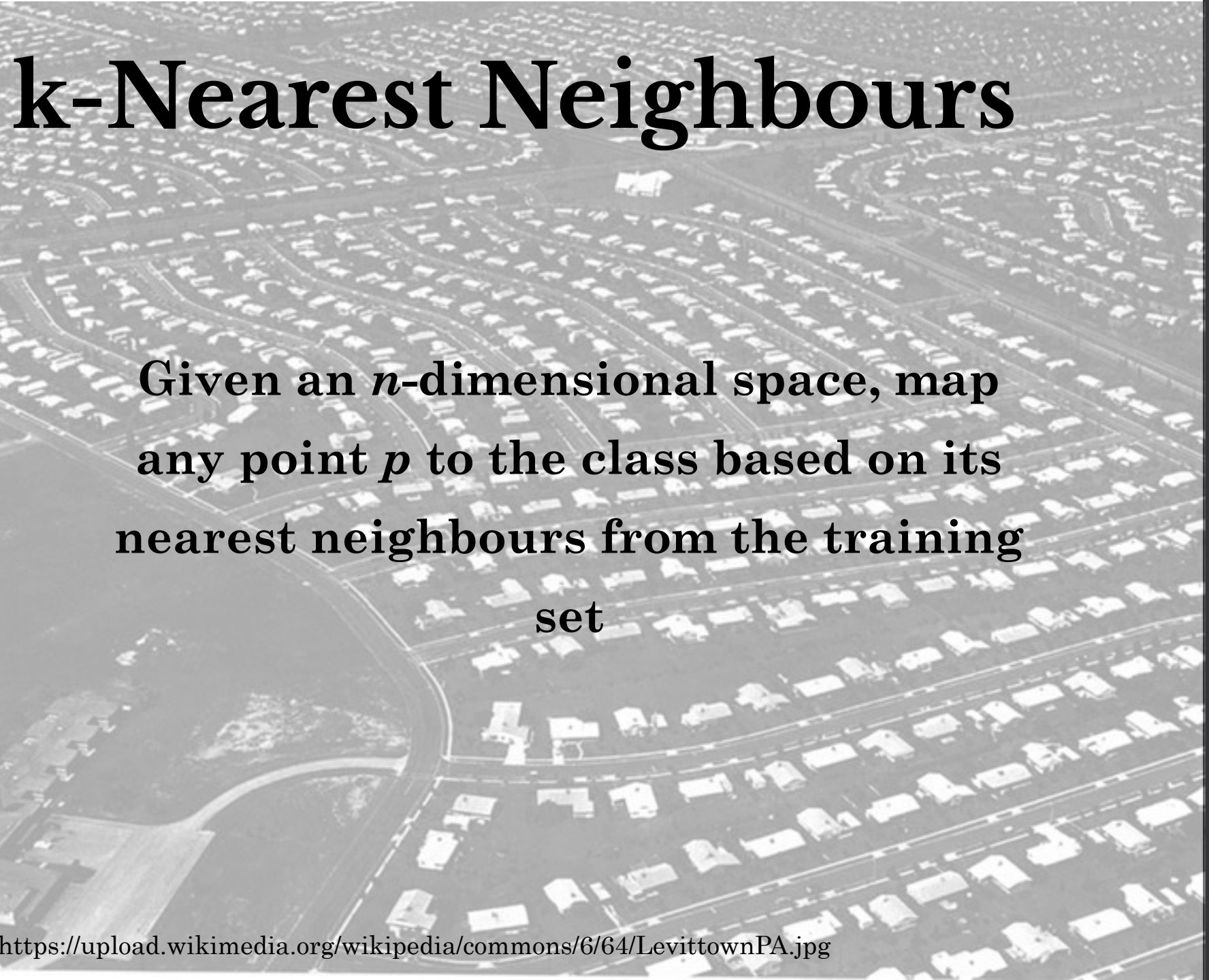
Sylvester et al. (Feb 2018) *Evol Applications*

Conclusions

Yes, rivers do constitute separate populations!
(although our ability to distinguish individuals is not perfect).

Sylvester et al. (Feb 2018) *Evol Applications*

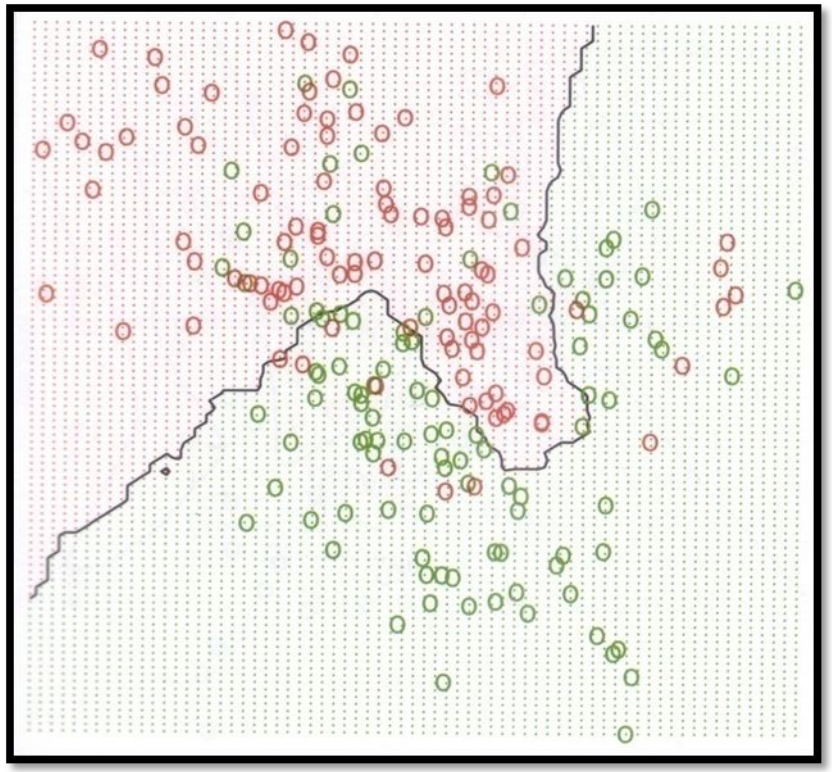
k-Nearest Neighbours



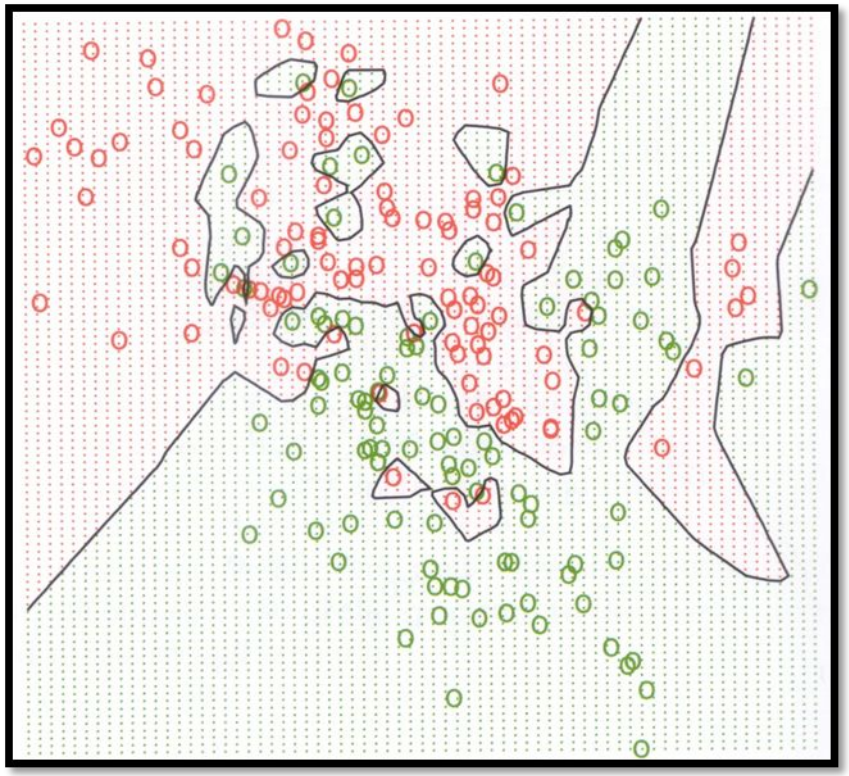
Given an *n*-dimensional space, map any point *p* to the class based on its nearest neighbours from the training set

https://upload.wikimedia.org/wikipedia/commons/6/64/LevittownPA.jpg

Procedure



15-NN



1-NN
(Voronoi
tesselation)

Hastie, p. 15-16

Training

No training *per se*, since modeling the decision boundary could be quite complex

Instead, find the labels of the k closest (e.g., Euclidean distance) training vectors

$$D_{Euclidean} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Analysis of microarray leukemia data using an efficient MapReduce-based K-nearest-neighbor classifier

Mukesh Kumar *, Nitish Kumar Rath, Santanu Kumar Rath

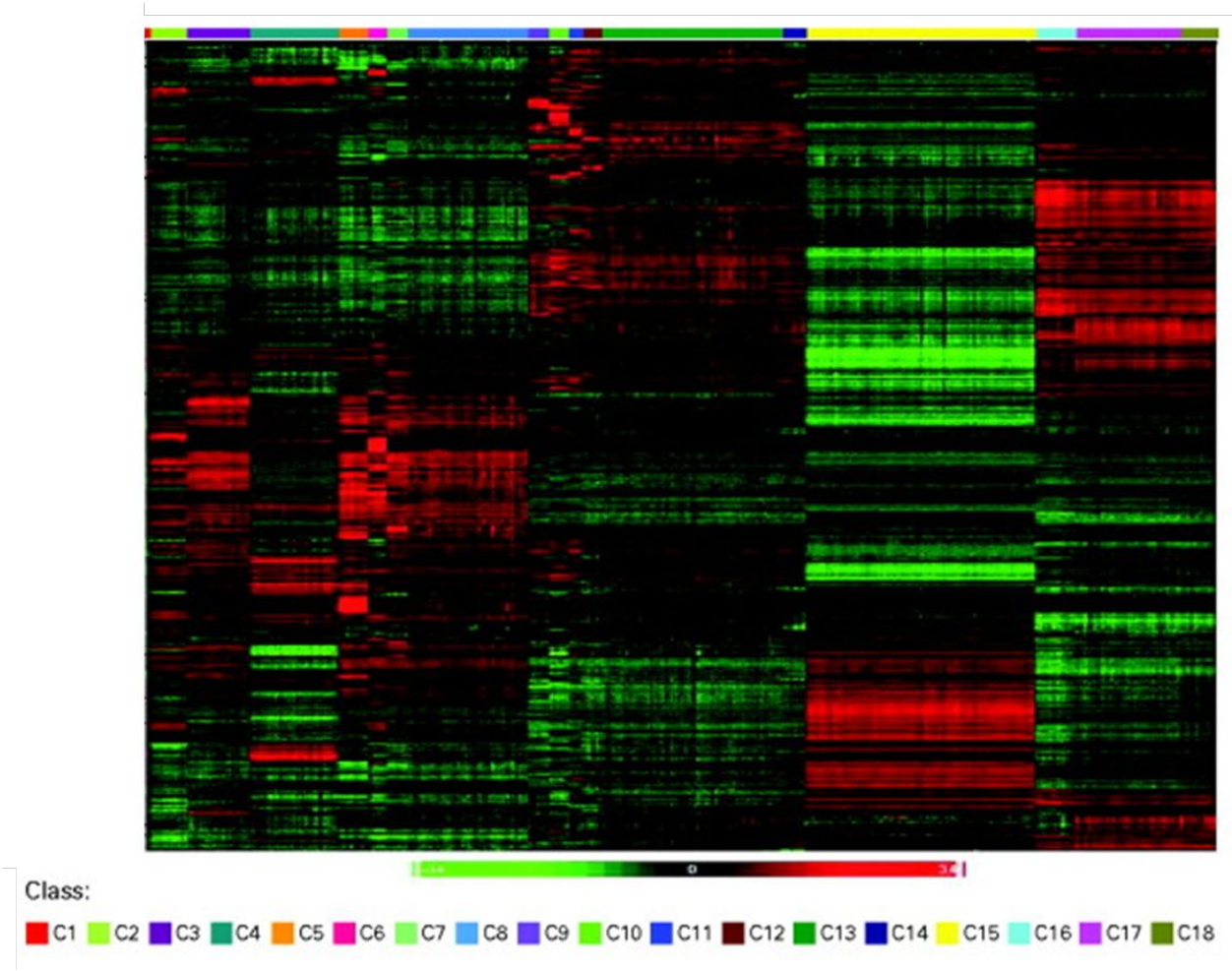
Department of Computer Science and Engineering, NIT Rourkela, Orissa 769008, India

Objective: classify different groups of leukemia subjects based on shared patterns of gene expression

Note: this paper spends a *lot* of time on MapReduce aspects of feature selection and k-NN classification – we're not going to discuss that part

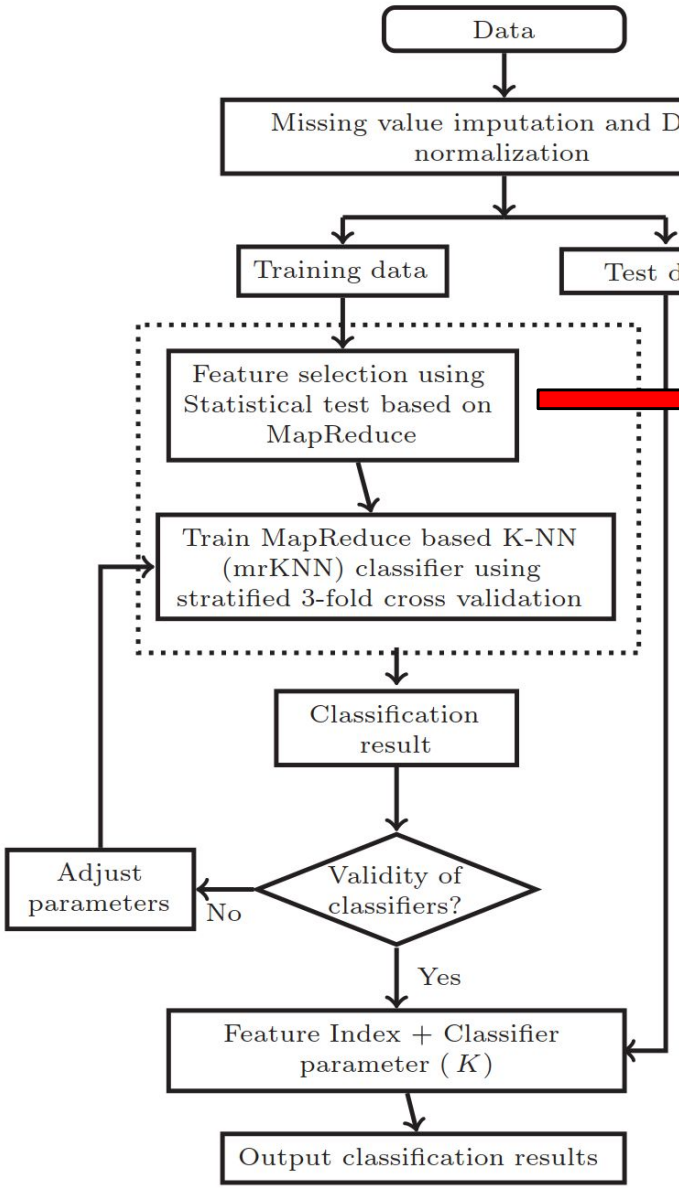
Kumar et al. (2016)

Expression Microarrays



Haferlach et al. (2010) *J Clin Oncol*

Workflow



Filling in the blanks

Simple statistical tests (e.g., ANOVA)

Choose *K* using cross-validation in the *training* set

Now try the test data

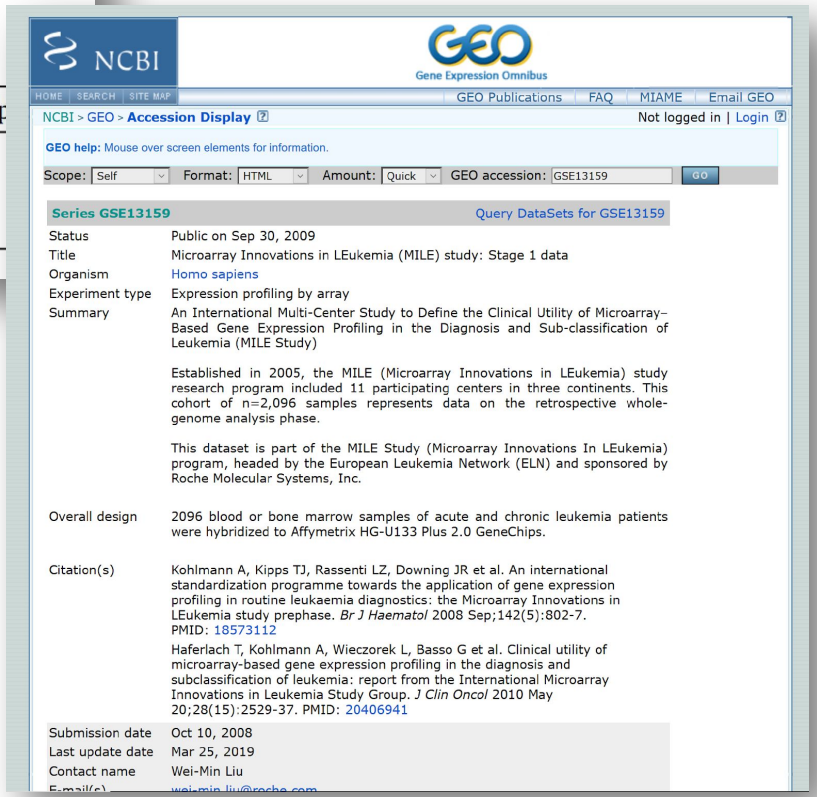
Datasets

Table 2
Microarray dataset used.

Dataset	Number of samples	Number of features	Number of classes	Size
GSE13159 [28]	2096	54,675	18	1.93 GB
GSE13204 [29]	3248	1480	18	1.96 GB
GSE15061 [31]	870	54,675	3	650 MB

Table 3
Details of partitioning into training and testing datasets.

Dataset	#Samples	#Features	#Training samples	#Testing samp
GSE15061	870	54,675	580	290
GSE13159	2096	54,675	1397	699
GSE13204	3248	1480	2165	1083



An impressive number of classes

GSE15061 classes	Class label	#Samples
Disease state: AML	1	135
Disease state: MDS	2	109
Disease state: none-of-the-targets	3	46

GSE13159 classes	Class label	#Samples
ALL with hyperdiploid karyotype	1	14
ALL with t(12;21)	2	19
ALL with t(1;19)	3	12
AML complex aberrant karyotype	4	16
AML with inv(16)/t(16;16)	5	9
AML with normal karyotype + other abnormalities	6	117
AML with t(11q23)/MLL	7	13
AML with t(15;17)	8	12
AML with t(8;21)	9	14
CLL	10	149
CML	11	25
MDS	12	69
Non-leukemia and healthy bone marrow	13	25
Pro-B-ALL with t(11q23)/MLL	14	23
T-ALL	15	58
c-ALL/Pre-B-ALL with t(9;22)	16	41
c-ALL/Pre-B-ALL without t(9;22)	17	79
mature B-ALL with t(8;14)	18	4

Feature selection

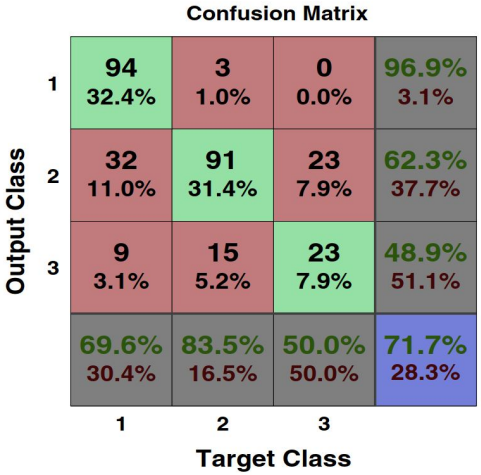
- We probably don't want to use over 54,000 features.
- The authors used basic statistical approaches such as ANOVA to determine which features best differentiated different classes.
- p-value filter for inclusion
- Feature selection results were…weird

Dataset	ANOVA	Kruskal–Wallis	Friedman
GSE15061 (54,675)	6786	9741	54,318
GSE13159 (54,675)	37,016	36,897	17,593
GSE13204 (1480)	1423	1427	1225

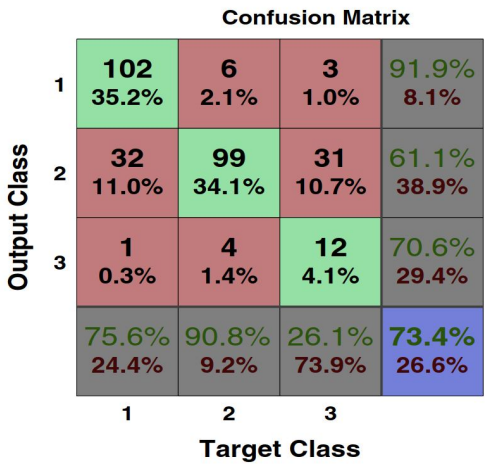
K-NN operation

- For each test data point, identify the *K* closest points from the training set (i.e., most similar profiles according to Euclidean distance)
- What are the labels of these *K* points?
- Choose the modal class, i.e., the class that is most frequently represented in the neighbour set
- Accuracy = % of all samples that were correctly classified

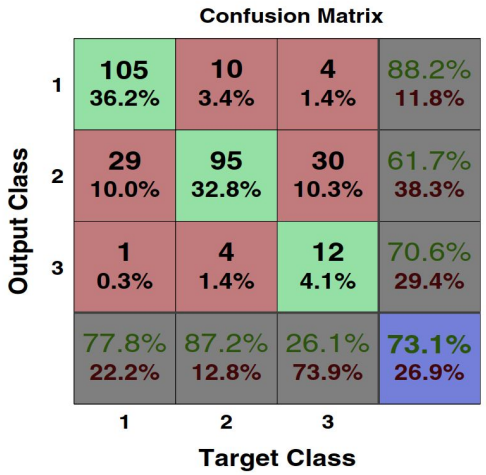
Results



(a) ANOVA
(f=6,786,K=21)



(b) Kruskal-Wallis
(f=9,741,K=17)



(c) Friedman
(f=54,318,K=21)

Accuracy is similar across all three feature sets
K is reasonably consistent

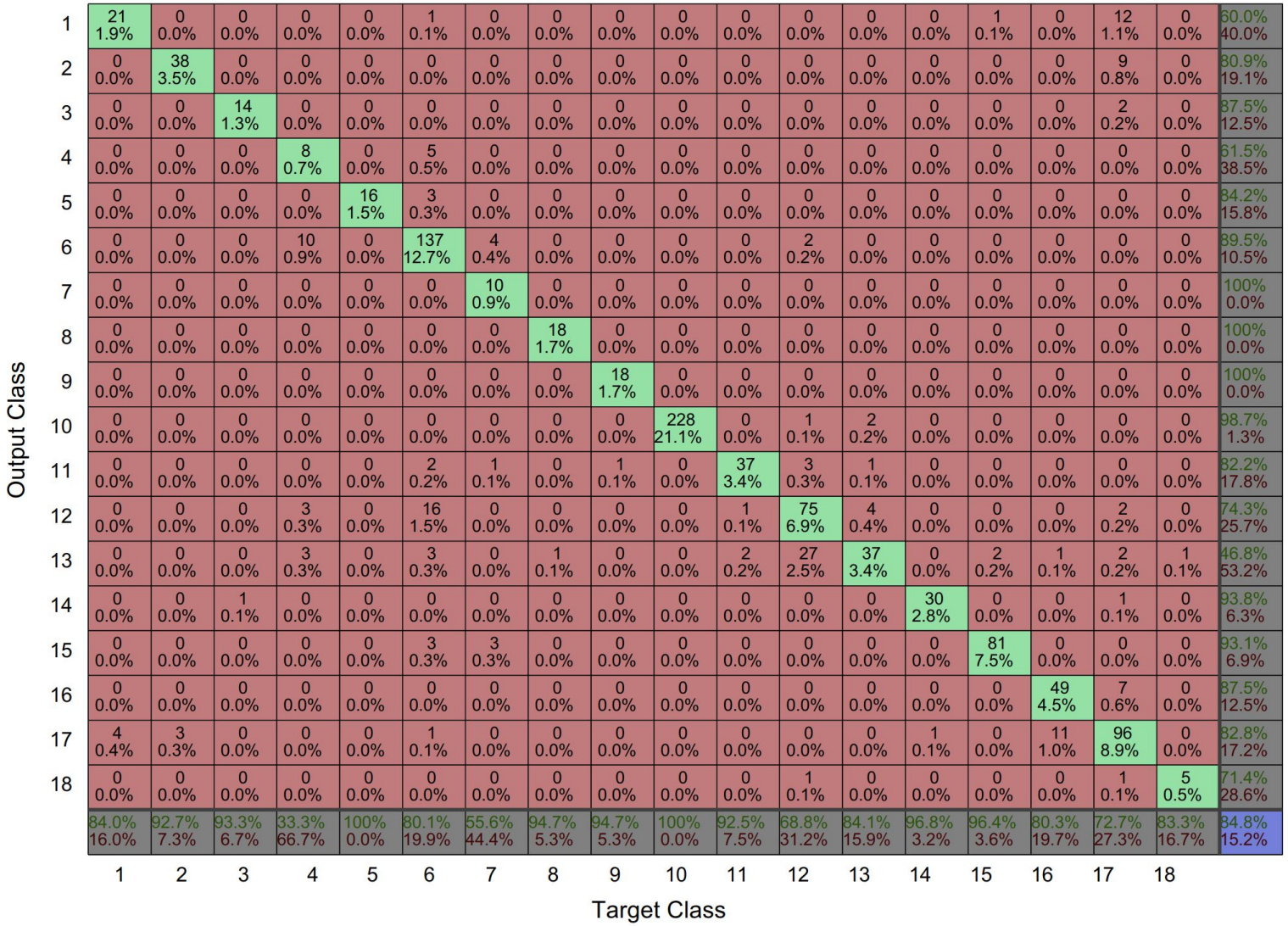
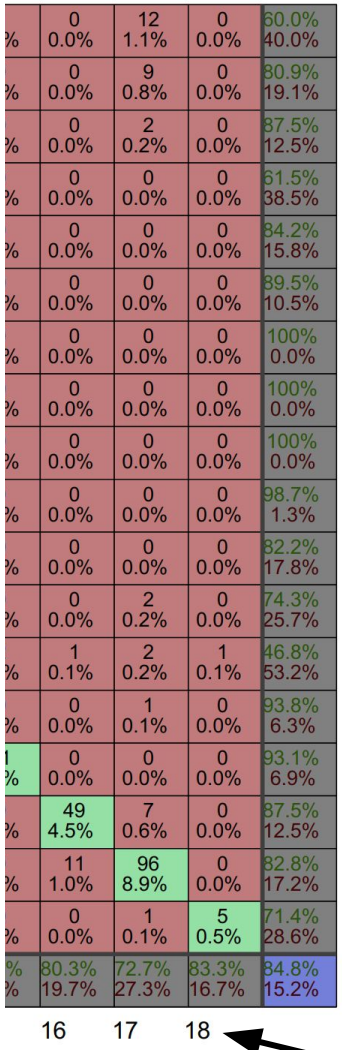


Fig. 11. Confusion matrix for mrKNN classifier with Friedman using GSE13204 dataset ($f = 1225,\ K = 3$).

K is a lot smaller – does this make sense?



Five examples. FIVE!!!

Accuracy is nothing to write home about

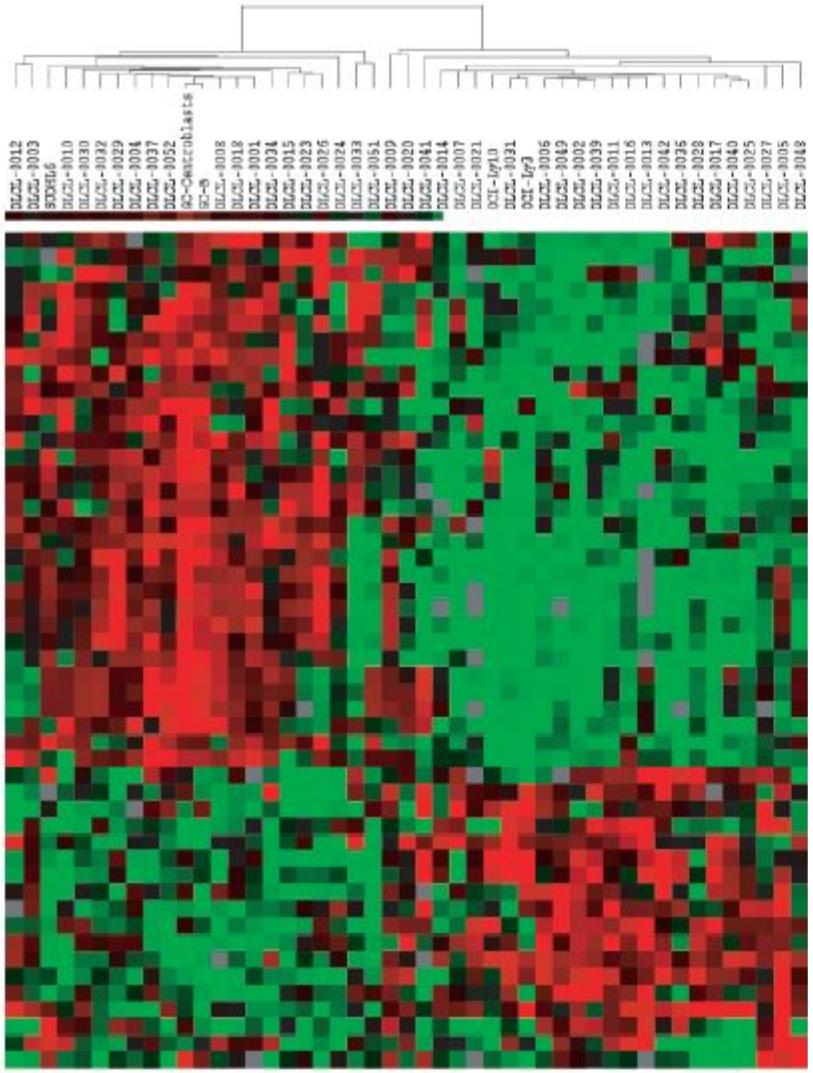
Modifications to basic k-NN

Try different distance definitions

Treat different dimensions differently (e.g., normalize, kernel methods)

Earlier k-NN example

Example: Gene selection from microarrays



s Samples (Diffuse large B cell lymphoma patients of two types, with different prognoses)

g Genes

Expression levels:
Low
High

Li et al (2001)

Approach

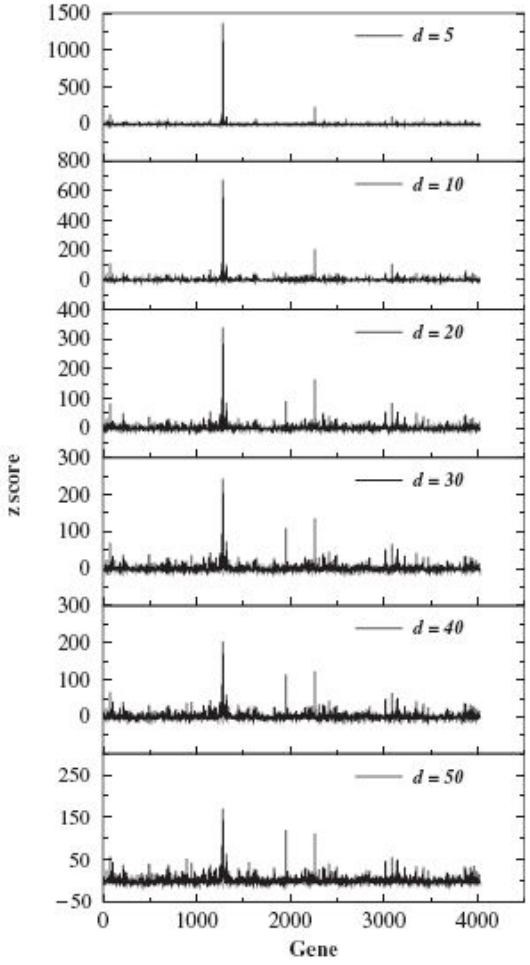
- Map all *s* training vectors (gene expression profiles) into *d*-dimensional space by subsampling genes

- A test vector is considered classified if its 3-NNs all represent the same lymphoma type, otherwise *unclassified*

Feature Selection

- When $g = 4026$ and $p = 34$, we have a few too many variables
- The authors used a *genetic algorithm* to select subsets of d genes for analysis
- Tried $d = \{ 5, 10, 20, 30, 40, 50 \}$

How often are different genes selected for classification by the genetic algorithm?



$$Z = [S_i - E(S_i)]/\sigma$$

Standard deviation

# of times selected (out of 10,000 chromsomes)

Expected # of times selected = d/g * 10,000

Table 1. The 50 most frequently selected genes for distinguishing germinal center B-like DLBCL and activated B-like DLBCL[a]

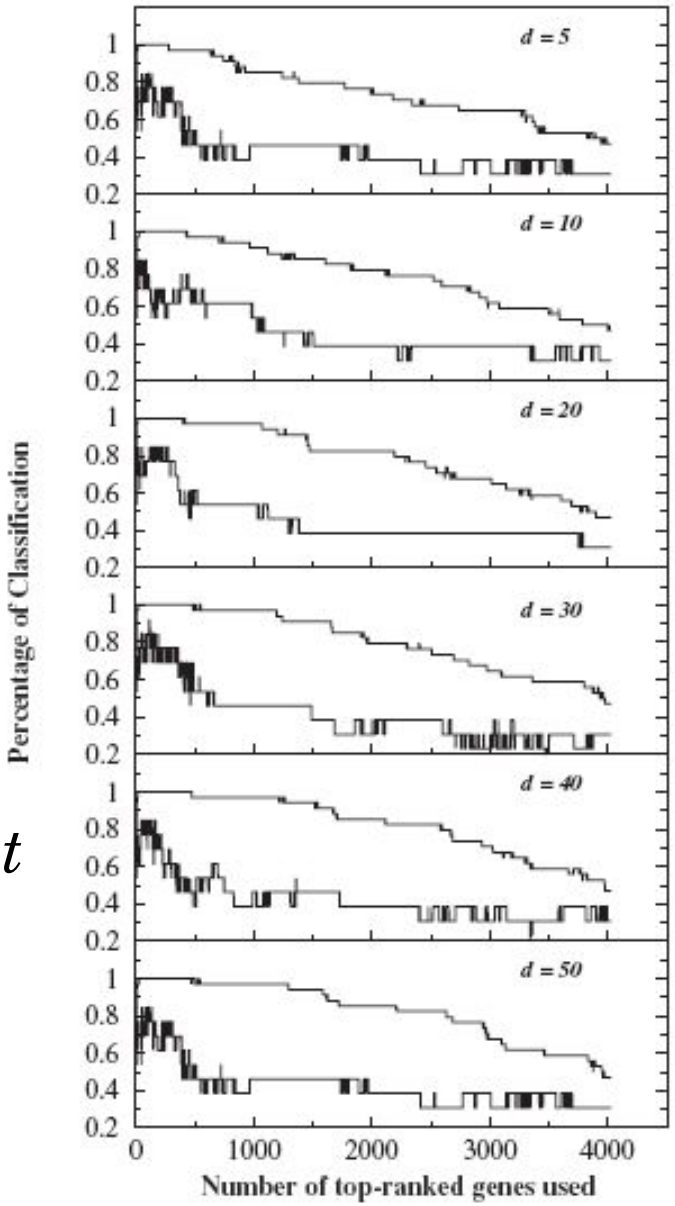
Student's *t*-statistics[b]	*z*-score[c]	Gene name
−9.044	203.231	Unknown UG Hs.120716 ESTs; clone = 1334260
−7.342	124.082	Unknown UG Hs.136345 ESTs; clone = 746300
7.675	122.267	MCL1 = myeloid cell differentiation protein; clone = 711870
4.147	114.604	Smad4 = DPC4; clone = 774619
−7.180	98.169	Unknown clone = 825199
−7.468	94.136	Unknown UG Hs.169565 ESTs; clone = 825217
−7.700	87.179	Unknown UG Hs.224323 ESTs; clone = 1338448
−7.955	86.978	CD10 = CALLA = neprilysin = enkepalinase; clone = 200814
−6.478	81.029	Unknown UG Hs.105261 EST; clone = 824088
−6.994	77.399	CD10 = CALLA = neprilysin = enkepalinase; clone = 1286850
6.442	66.611	Unknown UG Hs.169081 EST variant gene 6 TEL oncogene; clone = 1355435
−6.635	63.788	Unknown UG Hs.140559 EST; clone = 1339835
−6.662	60.158	Unknown clone = 1353015
−6.516	58.545	CD10 = CALLA = neprilysin = enkepalinase; clone = 701606
4.603	50.983	Erk3 = extracellular signal-regulated kinase 3; clone = 50506
−5.263	48.361	PARP = poly ADP-ribose polymerase; clone = 712849
−5.107	47.554	Unknown UG Hs.163222 ESTs; clone = 1338044
−6.820	47.454	Unknown UG Hs.208410 EST; clone = 1353036
−6.475	46.143	Unknown UG Hs.137038 EST; clone = 1338981
4.766	45.840	IRAK = interleukin-1 receptor-associated kinase; clone = 345588

What is the best classification accuracy that can be achieved?

What is the optimal value of d?

How many genes do we need?

Top line: accuracy on training set
Bottom line: accuracy on test set



Example results

- "Simulated" data (chop out 200 nt pieces from 534 sequenced genomes with known taxonomy)
- Leave-one-out strategy: if we delete all reference models from the same group at rank [x], can we classify to the correct [x+1] rank?

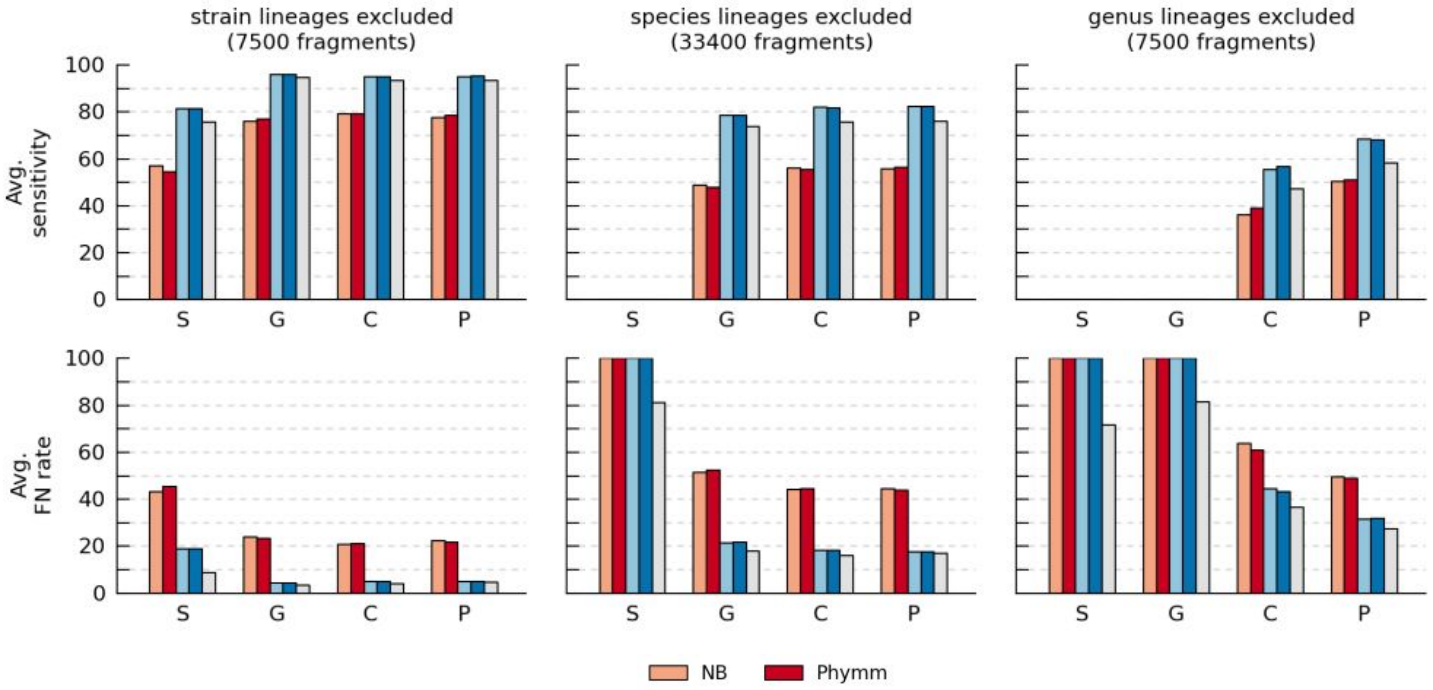


Figure 2 Average classification performance of rank-specific classifiers on 200 bp query fragments. Performance was evaluated using a leave-one-out framework with lineages excluded at the strain, species, or genus rank. Fragments were considered correctly classified if they were assigned to the species (S), genus (G), class (C), or phylum (P) of their source genome.

Everything looks like a nail.

START SIMPLE!!!

https://www.bandsintown.com/a/1119