Pattern Classification Challenges in Bioinformatics

James R. Green

Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada

Active Research Areas

1. Bioinformatics
   - Prediction of protein structure, PTM, function, interaction
   - ChIP-chip/Mutation Spectrum Analysis
   - miRNA prediction in unannotated species
2. Hardware acceleration of scientific computing
   - GPGPU, heterogeneous multicore, manycore
   - Proteome-wide analysis, real-time mass spectrometry
   - Real-time patient monitoring using stream processing
3. Assistive devices for disabled and elderly
   - Promote independent living

CU “Wet Lab” Collaborations

- Ashkan Golshani/Alex Wong/Frank Dehne/Kyle Biggar: PIPE, PIPE-Sites, SNP-PIPE, InSIPS
- Jeff Smith: real-time mass spec.
- Bill Willmore/Kyle Biggar: PTM (hydrox., Kme) prediction
- Ken Storey/Kyle Biggar: miRNA prediction in unannotated species
- Ashkan Golshani: image processing for functional genomics, PTM (sumoylation) prediction
- Maria DeRosa (et al): Computational aptamer design
- Carole Yauk (et al): ChIP-chip analysis for THR
- Paul White/Francesco Marchetti: NGS for MSA
- Susan Aitken: Comparative genomics

Bioinformatics

- Biology is becoming an information science
  - You can go on the web and download the entire human genome in a text file.
  - High-throughput tests examine 1000’s of molecules simultaneously → BIG data!
- In Bioinformatics, we apply computational techniques to help biologists conduct biomedical research
- Machine Learning is a computational tool that can be applied to a set of solved examples to generalize to new data.
  - Automation (high-througput)
  - Cost savings (pre-screen before bio validation)
  - Suggest future biological experiments

Seminar Goals

- What is pattern classification?
  - Why do you need pattern classification?
  - Understand the structure of a pattern classification system
  - How to evaluate classification accuracy
- Case studies from current collaborations
  - PIPE
  - PTM prediction
  - miRNA prediction
- Other projects from our lab (time permitting)

Machine Perception

- Humans naturally recognize patterns

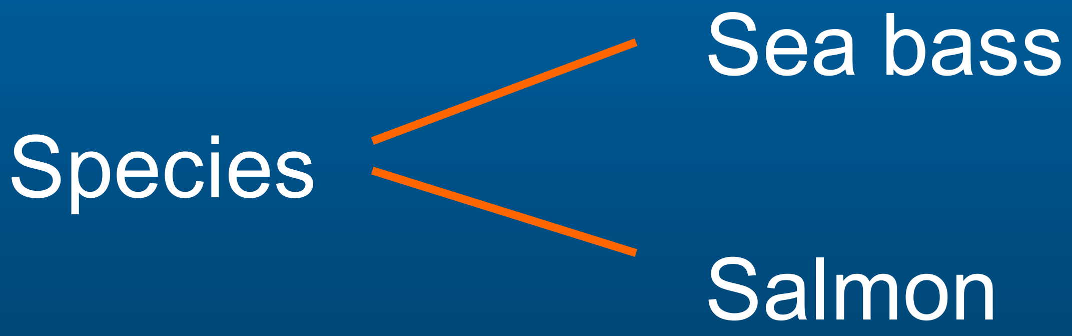
- These are all extremely difficult for a machine!

- Build a machine that can recognize patterns. e.g.:
  - Speech recognition
  - Fingerprint identification
  - Optical Character Recognition
  - DNA transcription factor binding sites
  - Gene identification
  - Protein structure, interaction, function prediction

This example and several illustrations in these slides are taken from Duda, Hart, and Stork, Pattern Classification, 2nd Edition, Wiley, 2001

An Example – fish sorter

- "Sorting incoming Fish on a conveyor according to species using optical sensing"



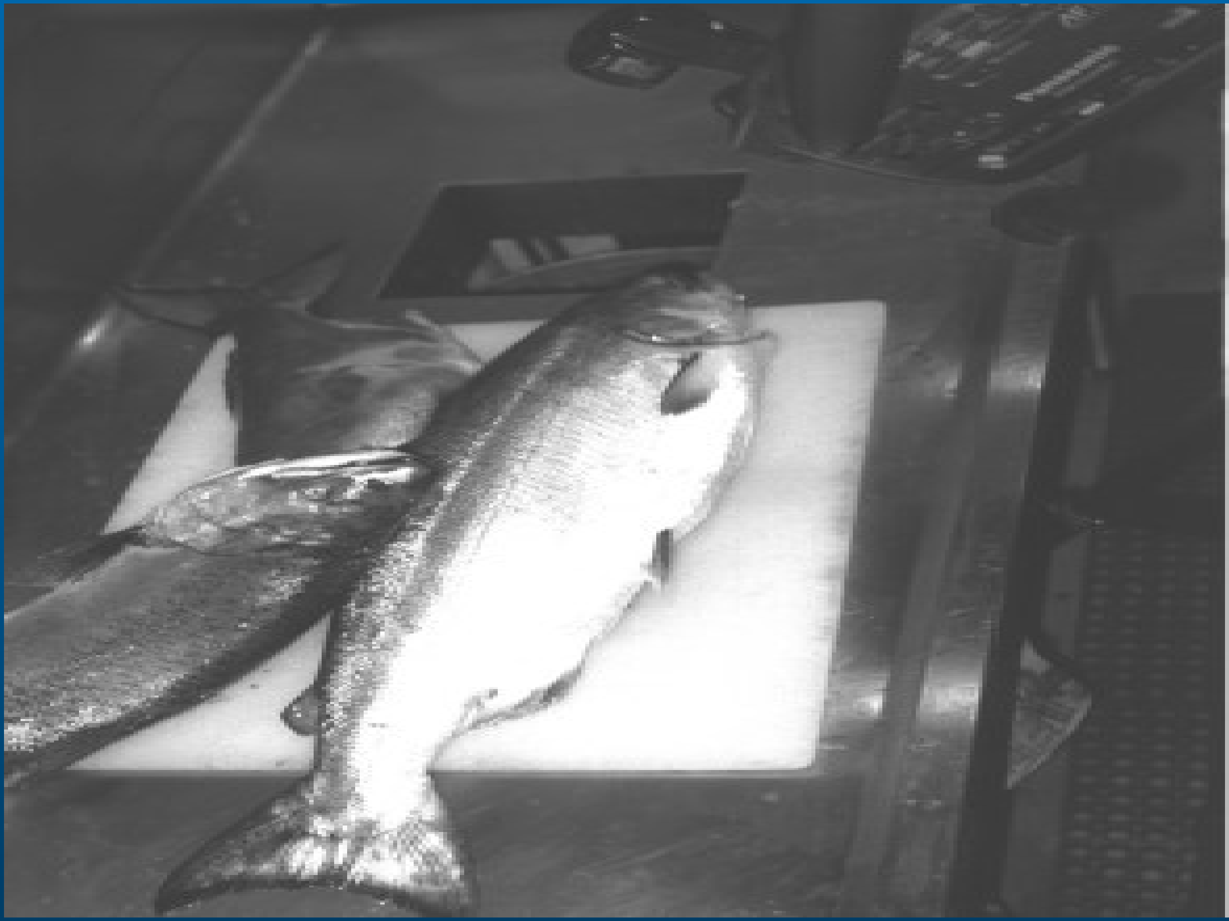
An Example – fish sorter

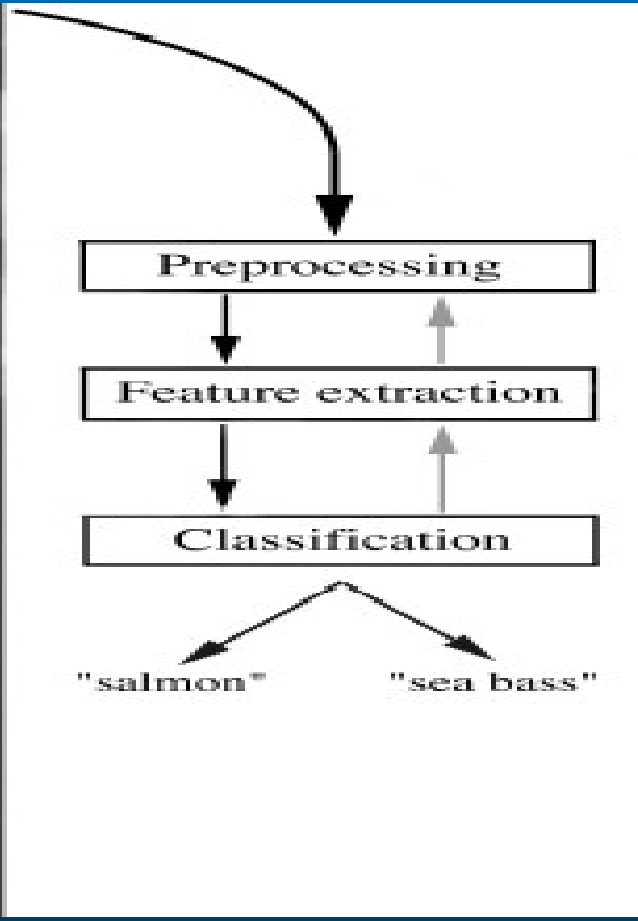
- Problem Analysis
  - Set up a camera and take some sample images to extract features
    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…
  - May be continuous, nominal/categorical, ordinal
  - We may use only a subset of these features in our classifier!

An Example – fish sorter

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
- Feature extraction
  - Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features
- The features are passed to a classifier

An Example – fish sorter



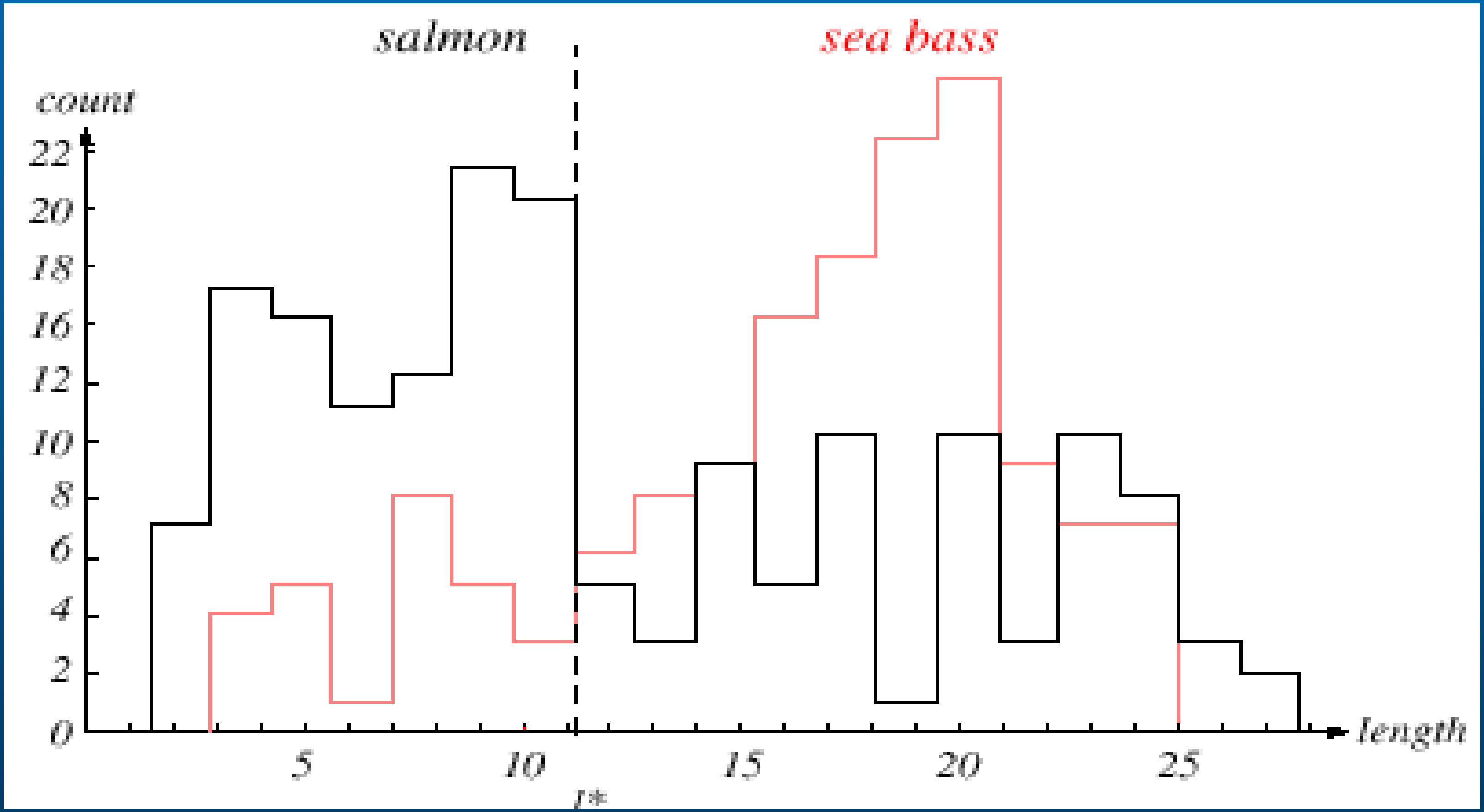


An Example – fish sorter

- Classification

  - Get some prior information:
    - Told that salmon are generally shorter than sea bass
  - Select the length of the fish as a possible feature for discrimination

Histogram of fish length

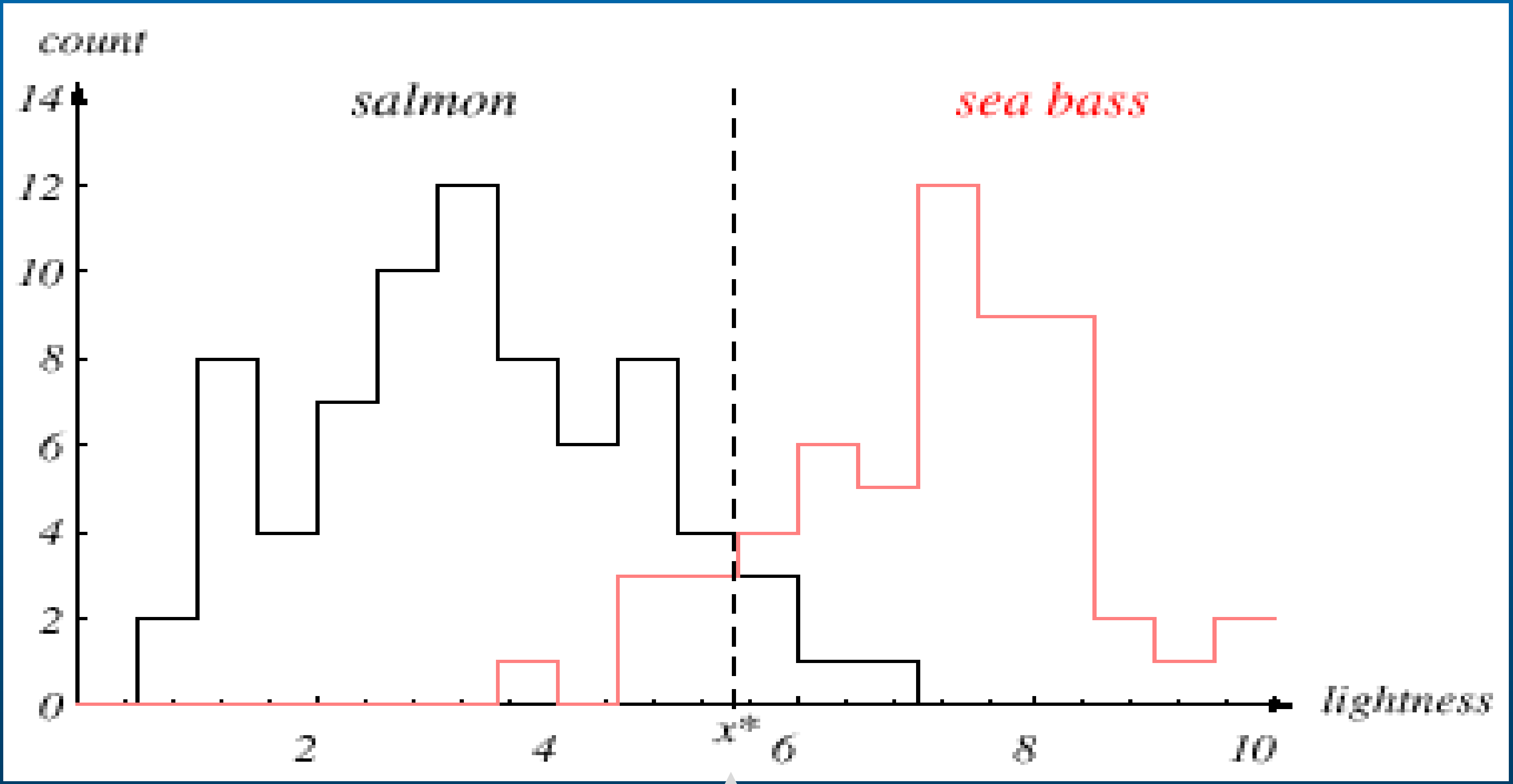


L*, Optimal decision boundary placement

An Example – fish sorter

- Although, on average, salmons are shorter than sea bass, length is a poor feature alone!
- Try selecting lightness as a possible feature.

Histogram of fish lightness



x*, Optimal decision boundary placement

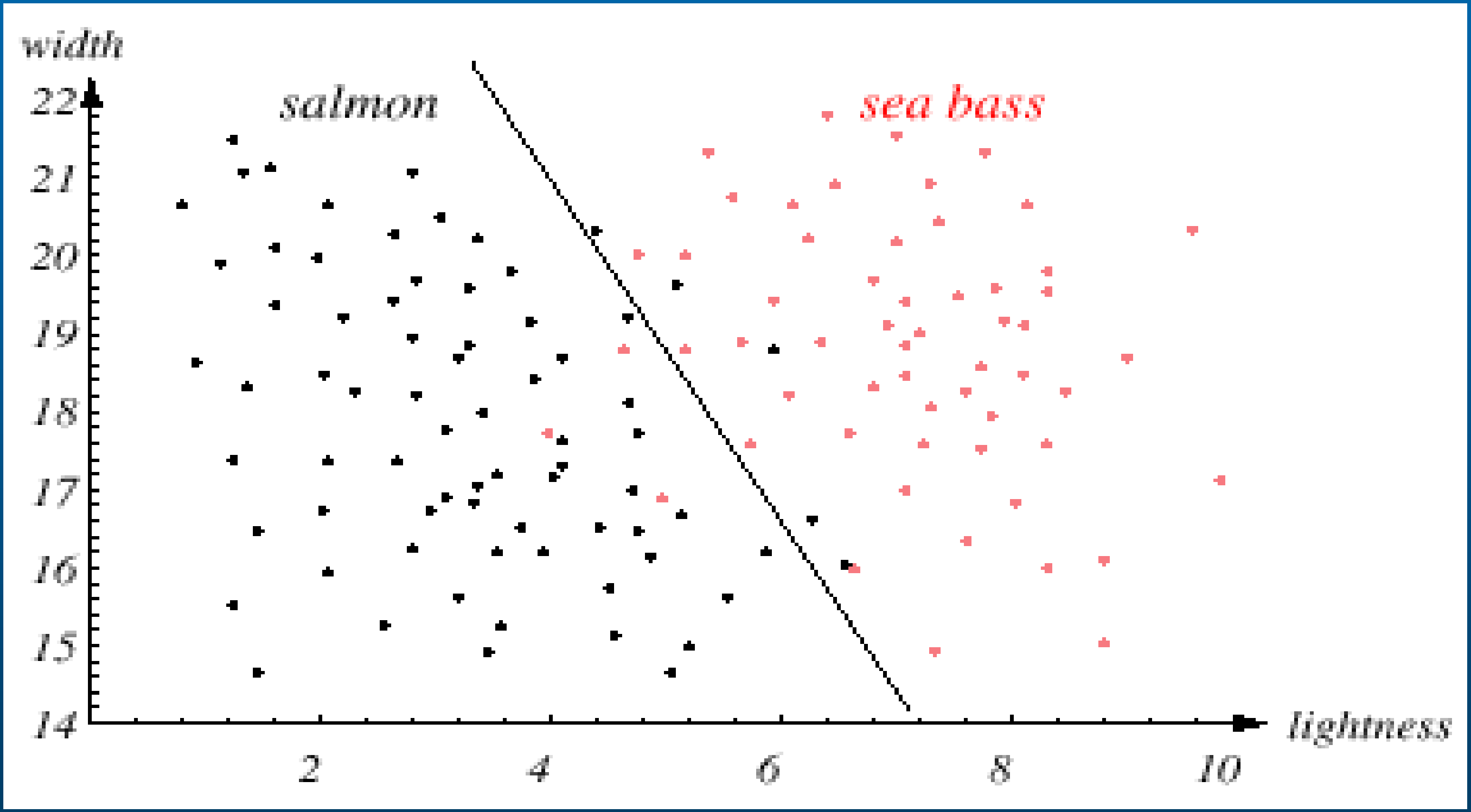
A new feature vector

- No single feature provides a good separation of the two fish types (classes)
- Try combining multiple features:
  - Adopt the lightness and add the width of the fish

$$\begin{tikzcd} \text{Fish} \arrow[r] & x^T = [& x_1 \arrow[dl] & , & x_2] \arrow[dr] & \\ & \textbf{Lightness} & & & & \textbf{Width} \end{tikzcd}$$

The real object ‘fish’ is now represented by a <u>feature vector</u>

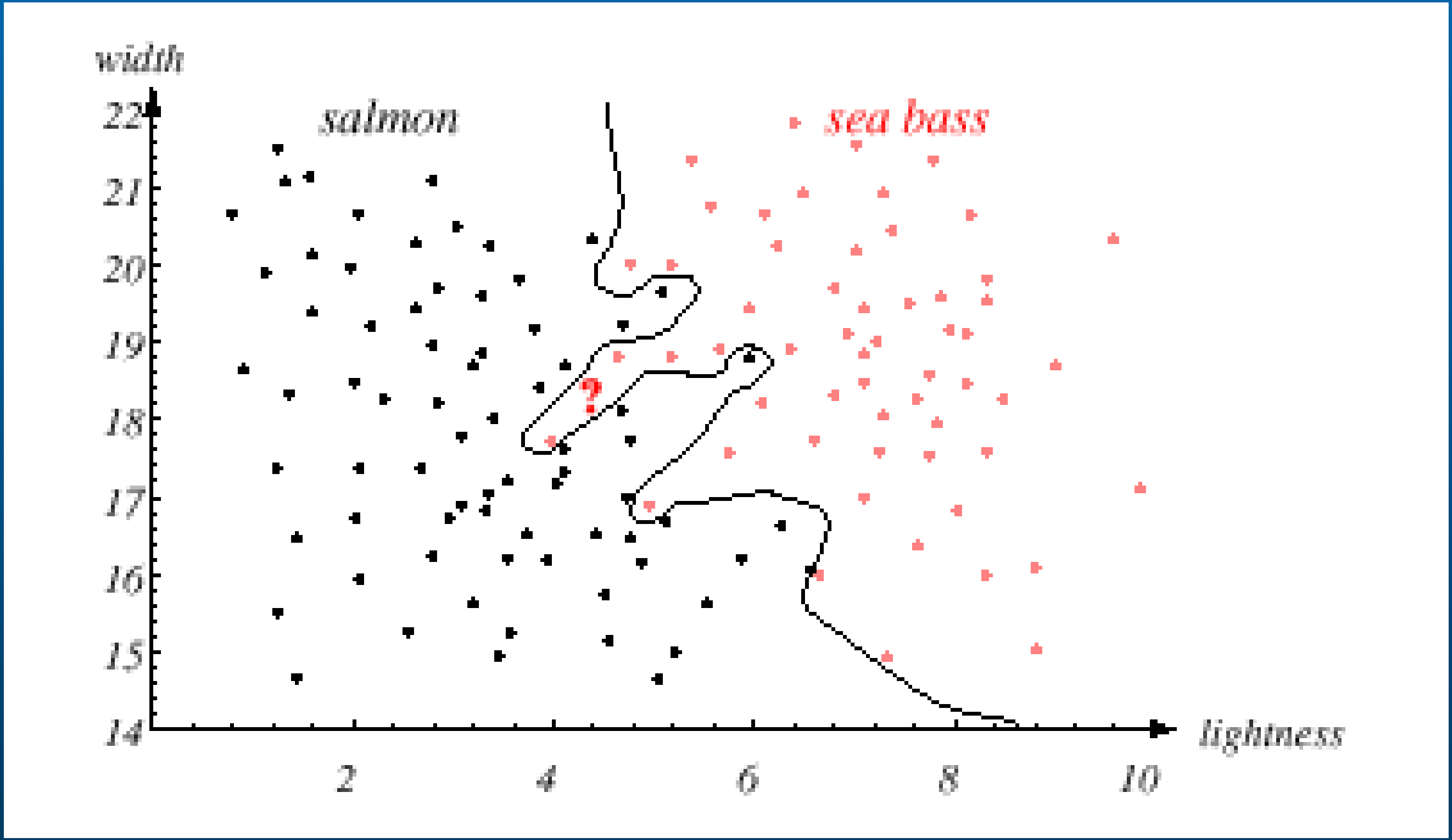
Scatter plot of fish width vs. lightness



Overfitting and generalization

- We might add other features that are not correlated with the ones we already have.
  - A precaution should be taken not to reduce the performance by adding “noisy features”
- We need to be careful of our “complexity”:

An ‘optimal’ decision boundary?



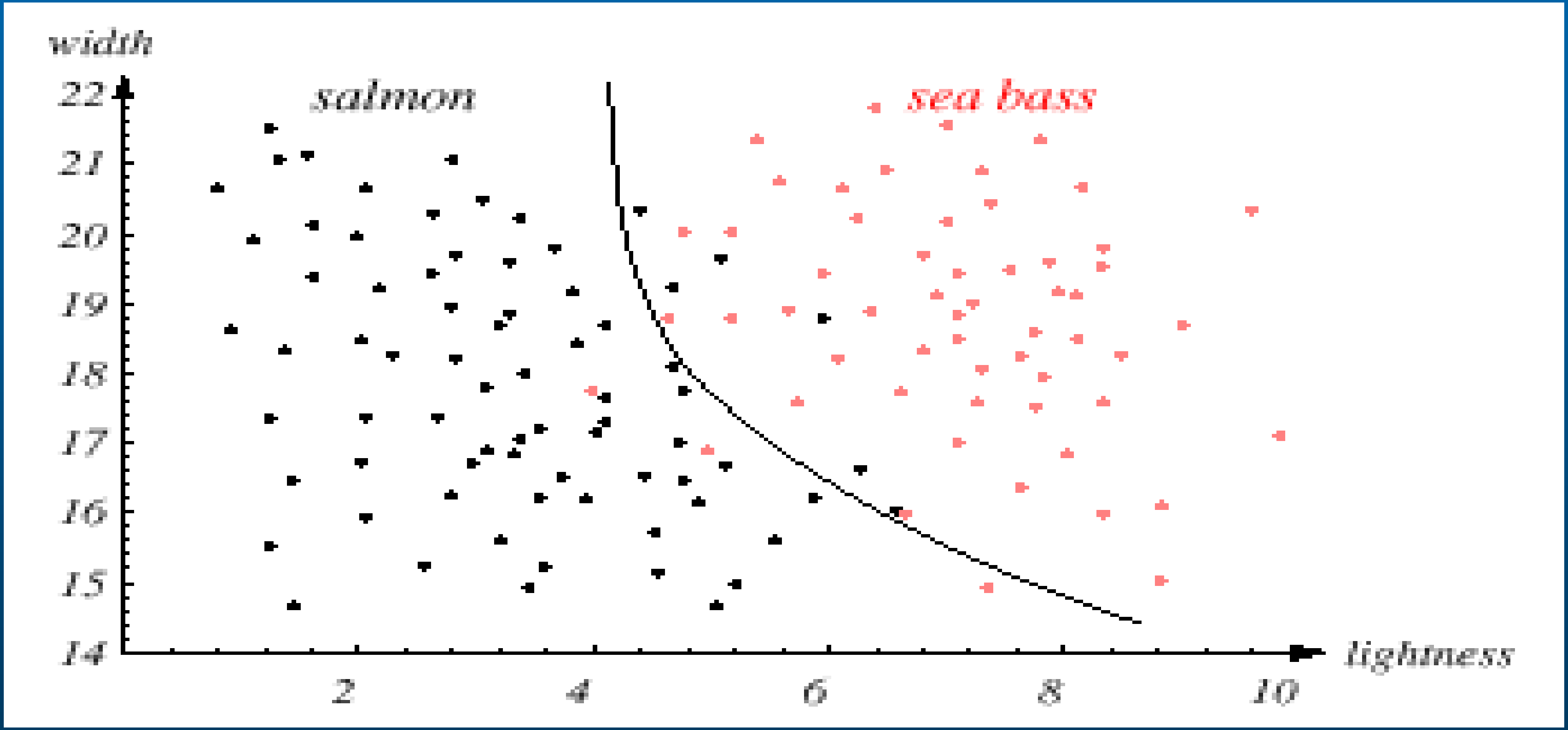
Overfitting and generalization

- The central aim of designing a classifier is to correctly classify novel input, not just training example inputs.

Issue of generalization!

- Performance on the training data is not always indicative of performance on future test data

An improved decision boundary?

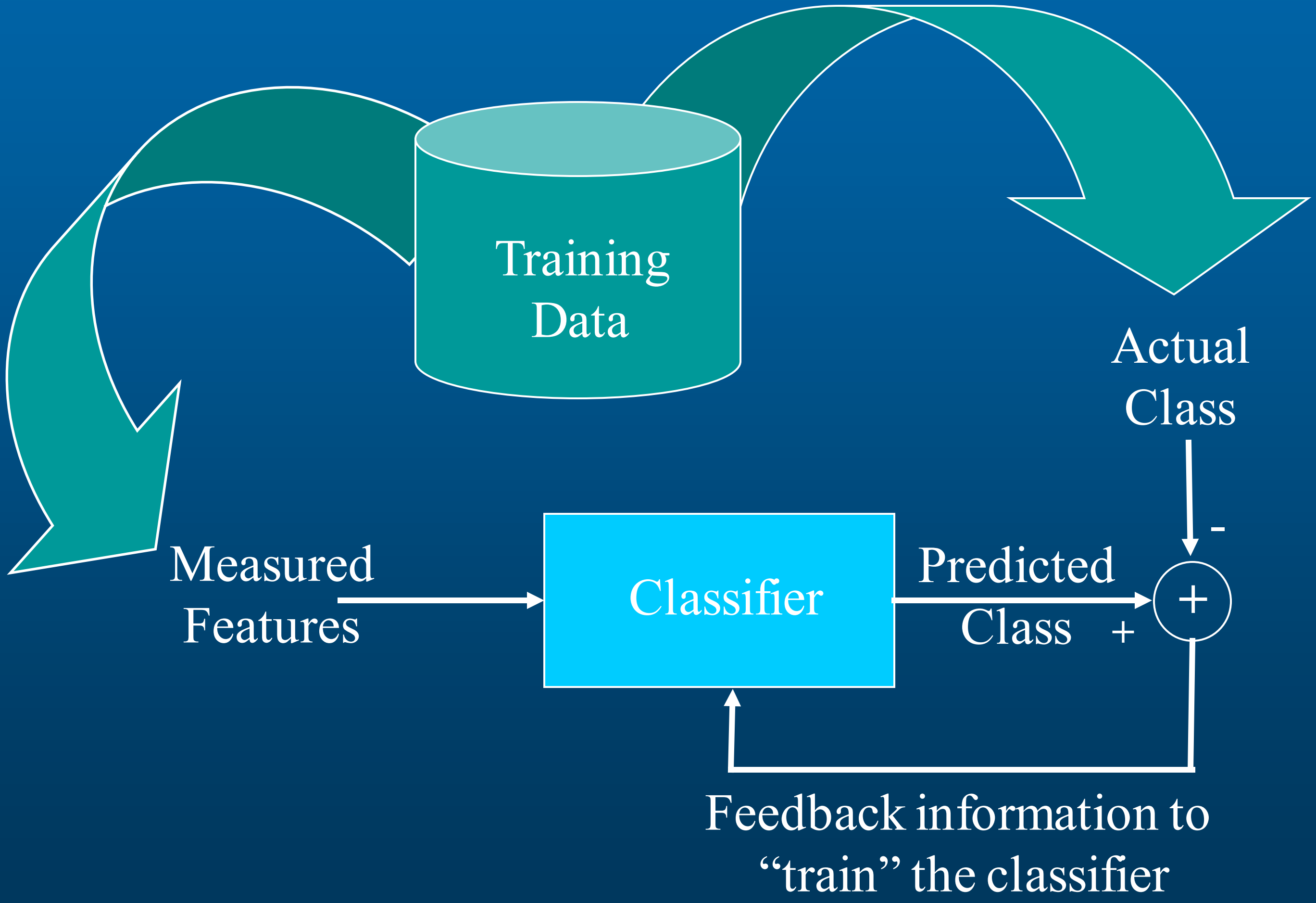


The big picture (supervised learning)

- Training
  - Collect some training samples where the class is known
  - Make some measurements to extract features
  - Train a classifier using measured features and known class
- Testing
  - Evaluate the accuracy of the classifier on test data that was not used to train the classifier.
- Operation
  - Ultimately, system will work for NEW data
  - i.e. examine features for a new sample, guess at class

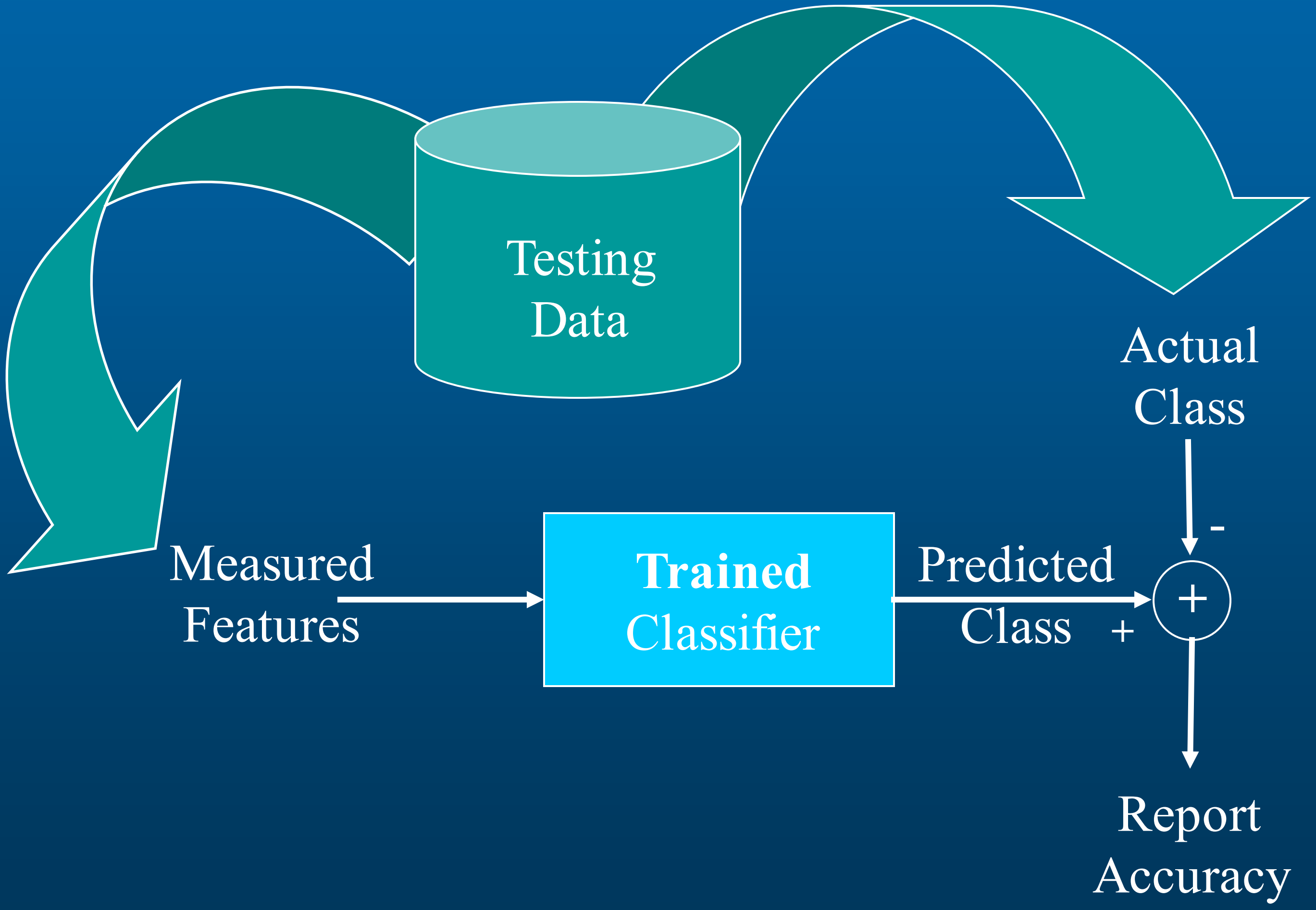
The big picture (supervised learning)

- Training:



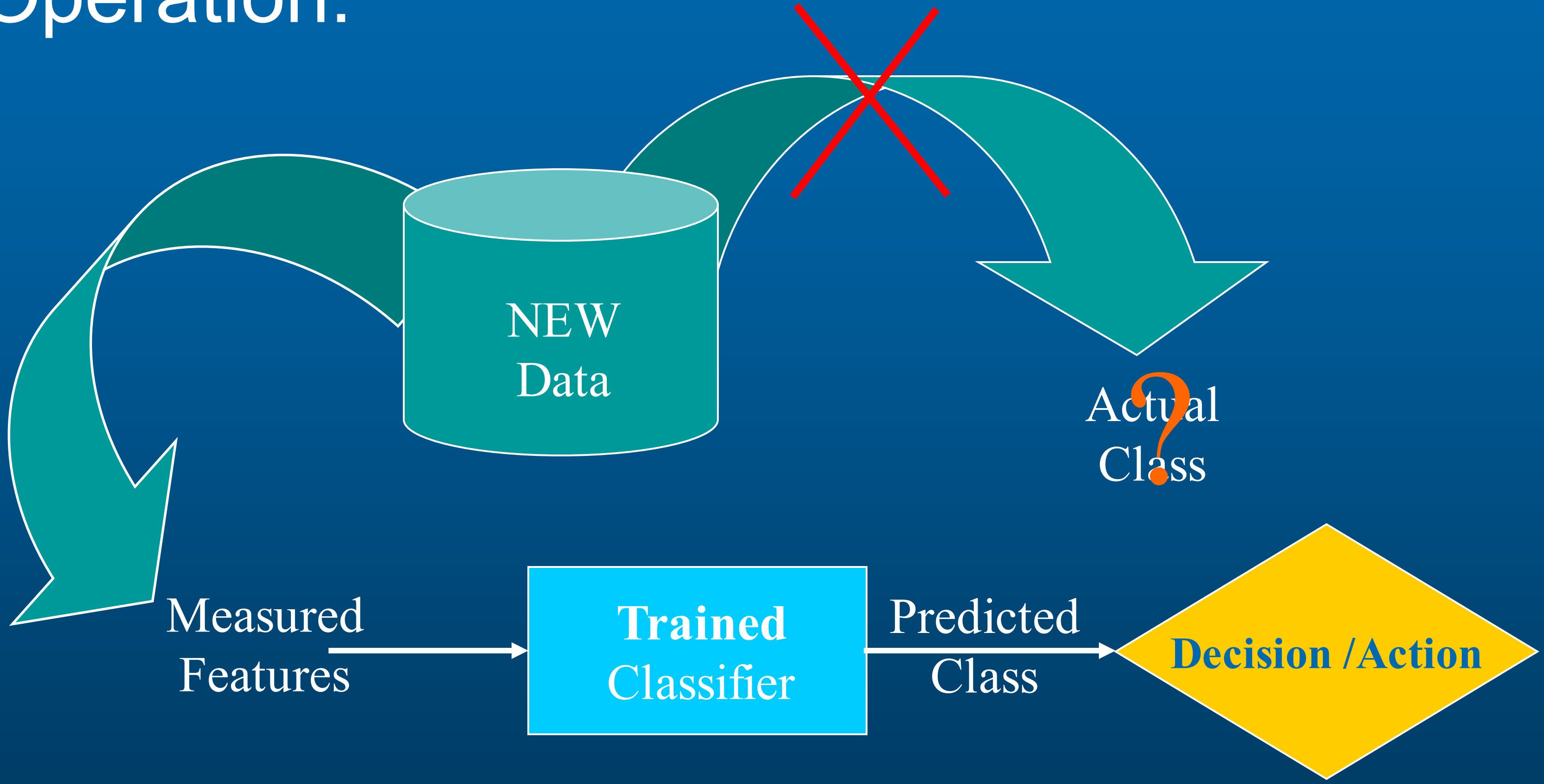
The big picture (supervised learning)

- Testing:



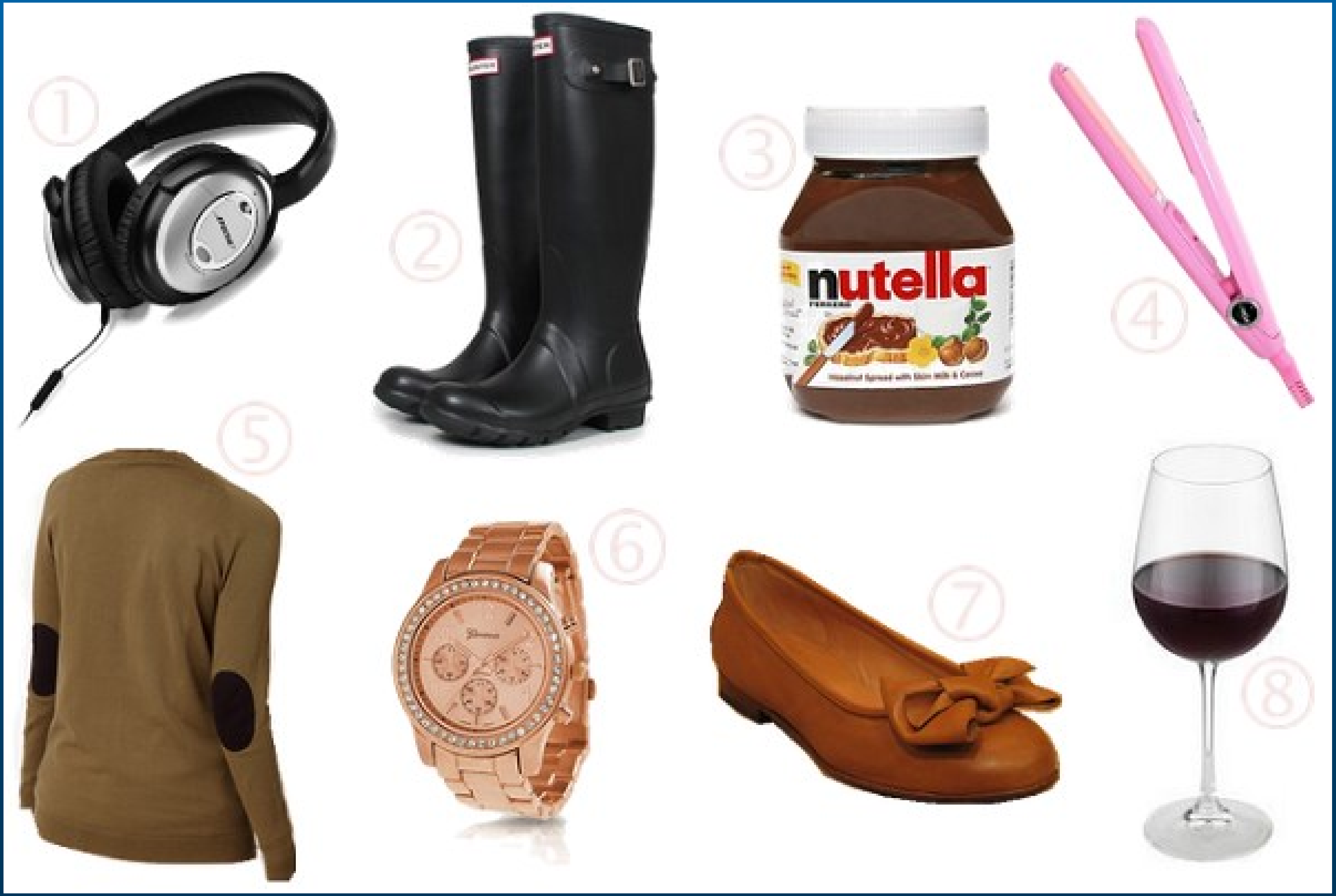
The big picture (supervised learning)

- Operation:



Unsupervised Learning

- Cluster these items:



http://www.lifeofcharmings.com/2012/11/obsessed-random-favs-from-our-trip-to.html

Selecting a learning algorithm

- Many forms of pattern classifier are available
  - Artificial neural networks, support vector machines, decision trees, decision forests, linear discriminant analysis, K-nearest neighbour, parallel cascade identification, rule-based systems, Bayesian networks, hidden Markov models, genetic algorithms, and many more!
- Be wary of claims such as '*SVMs are the BEST classifier*'
  - (*No Free Lunch Theorem*)
- In my experience:
  - If your problem is easy, any classifier will work
  - If your problem is hard, try a few classifiers
  - Find a good toolkit that implements the classifier structure
    - Many available for all the methods listed above (e.g. Weka)

Computational Complexity

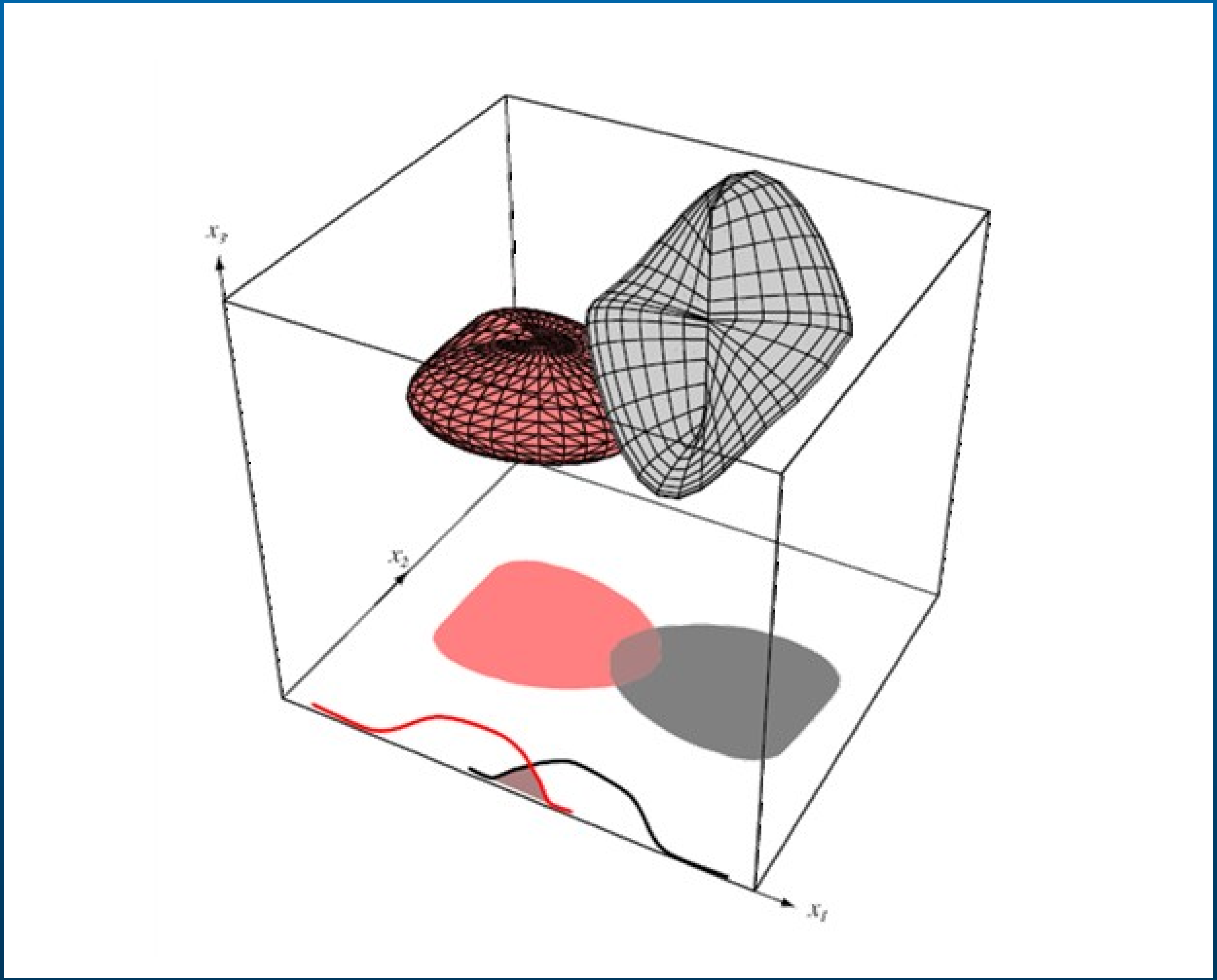
- What is the trade-off between computational ease and performance?
  - How does the algorithm scale as a function of the number of features, patterns or categories?
  - Starts to be important when you want to search an entire genome for a pattern…

Problems of dimensionality

- How does accuracy depend on the dimensionality of your features?

- The good news:
  - More features may increase accuracy
- The bad news:
  - The "*curse of dimensionality*"

Accuracy, dimension and training sample size



Duda, Hart, Stork, Pattern Classification, Wiley, 2001.

Feature selection

- "If all features have good predictive capabilities, any one of many classification methods should do well. Otherwise the situation is much less predictable"*
- Some methods will actually do worse with more features
  - May be overly sensitive to noisy features
  - May overweight redundant features
- Can use *feature selection* to mitigate these effects
  - Choose a subset of features based on merit

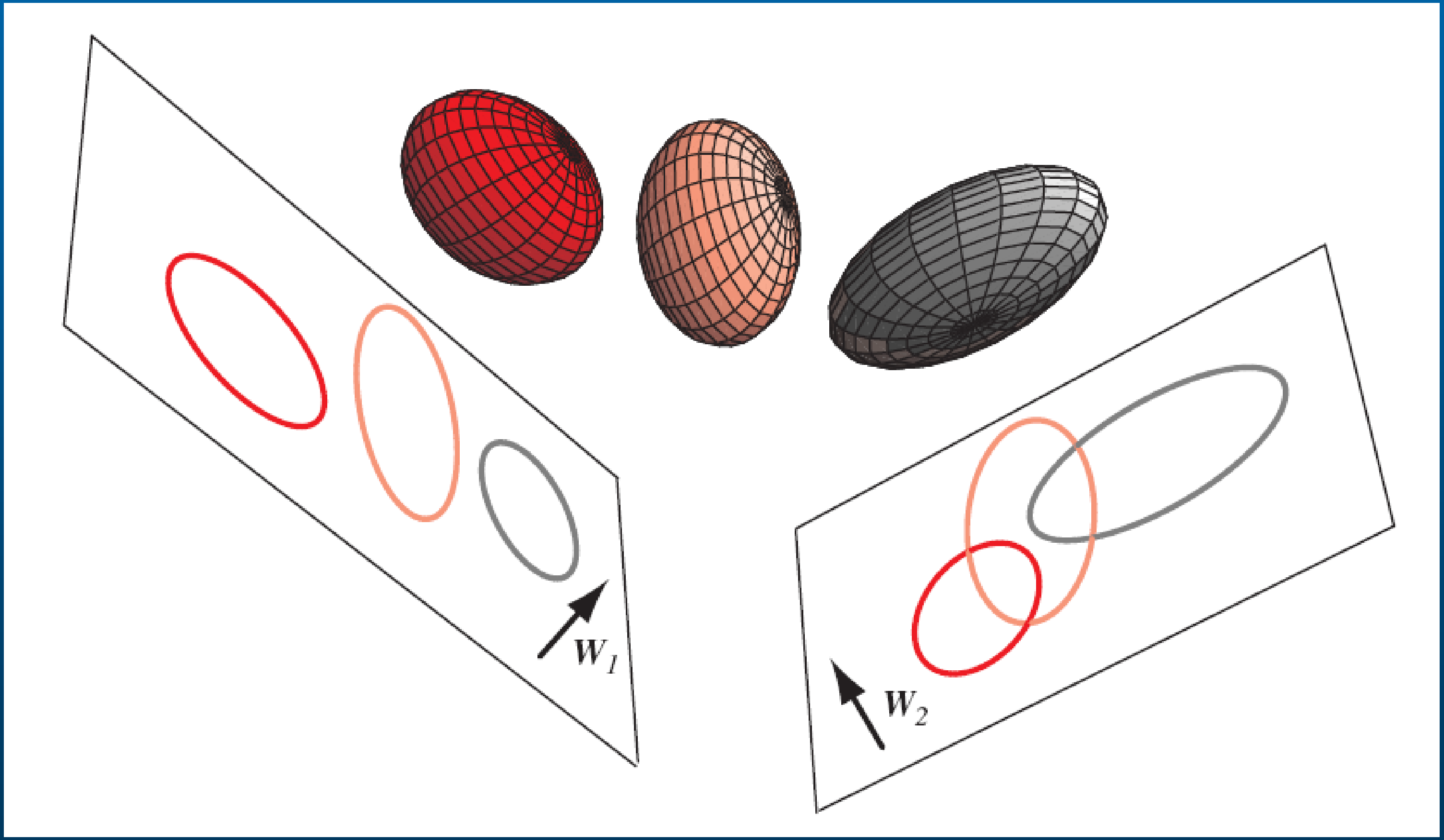
*Sholom Weiss and Casmir Kulikowski, Computer Systems That Learn, Morgan Kaufmann, 1991.

Reducing dimensionality

- Several options for reducing dimensionality
  - Manually select subset of features
    - Can pre-screen individual features for ability to discriminate between classes
    - Cluster similar/redundant features based on covariance
  - Automated dimension reduction
    - Use a linear combination of features
      - Principal Component Analysis
      - Fisher's Linear Discriminant
      - Multiple Discriminant Analysis

Reducing dimensionality

- Multiple discriminant analysis example



Duda, Hart, Stork, Pattern Classification, Wiley, 2001.

Data set partitioning

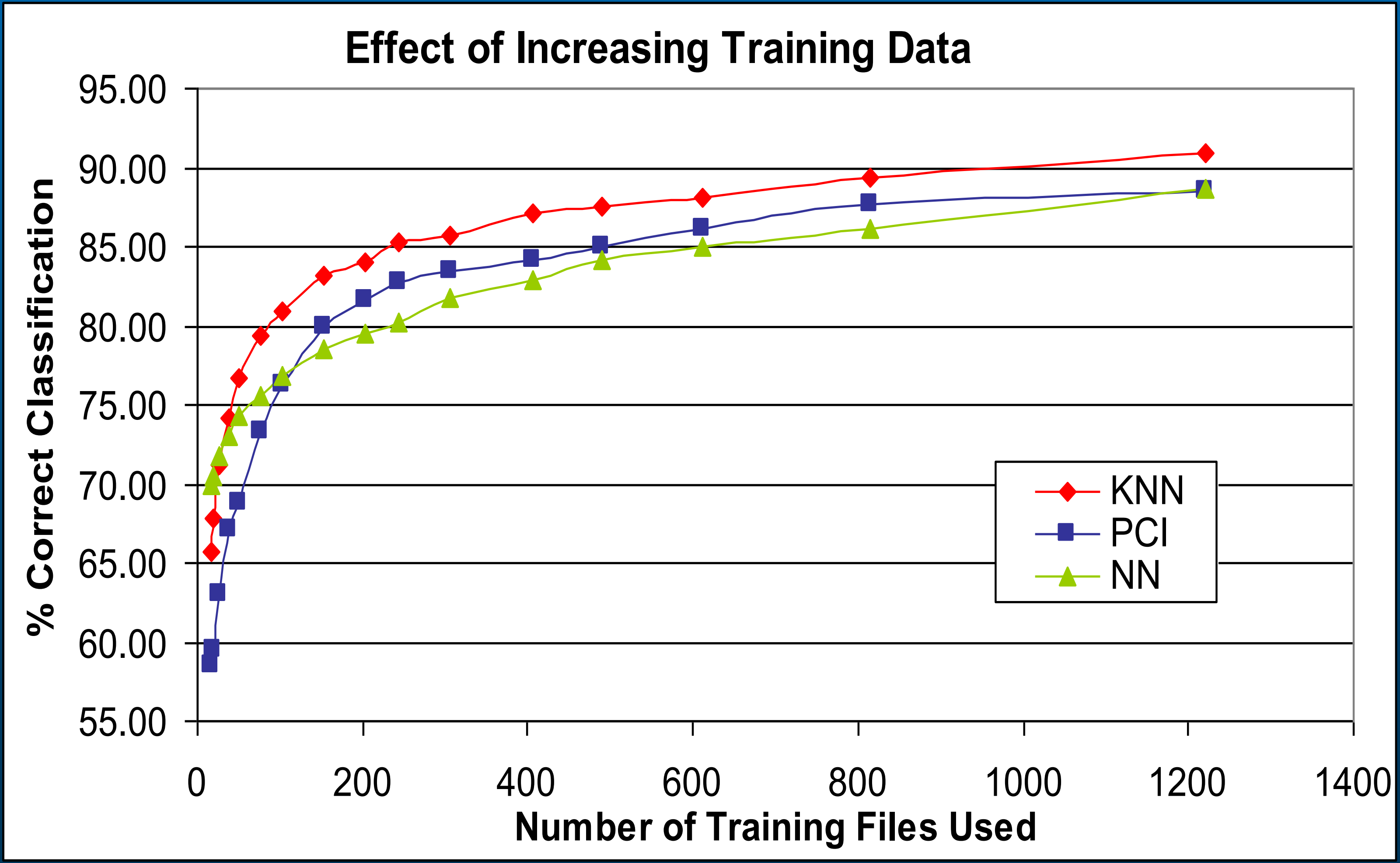
- Goal of pattern classification is to learn from training data in order to perform accurately over <u>new future data</u> (generalization)

  - Goal 1: Create an accurate classifier
    → need lots of training data

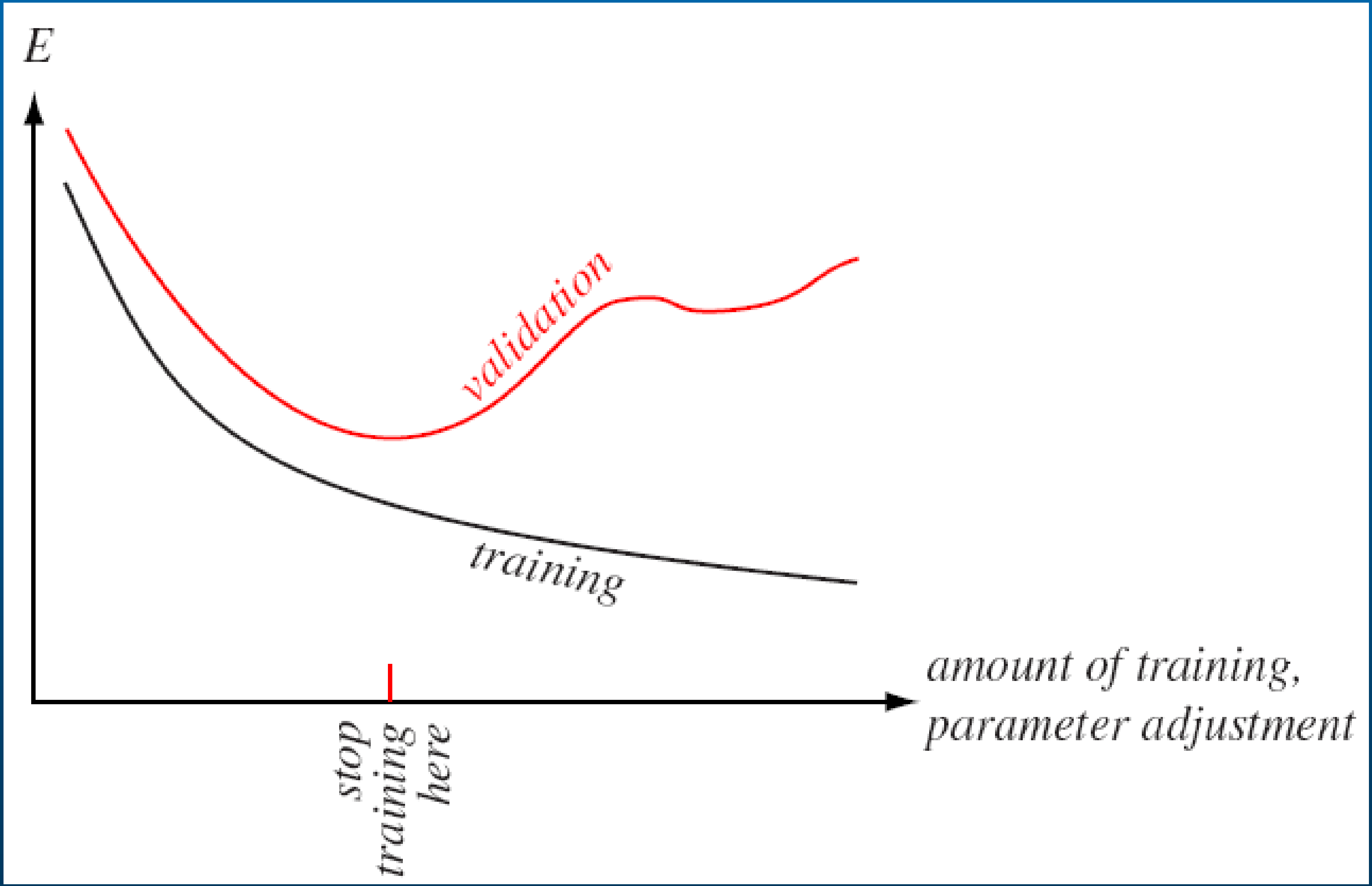
  - Goal 2: Estimate accuracy on future data
    → need lots of independent test data

Goal 1: Effect of Training Set Size

- Three techniques used for ATP protein binding site prediction

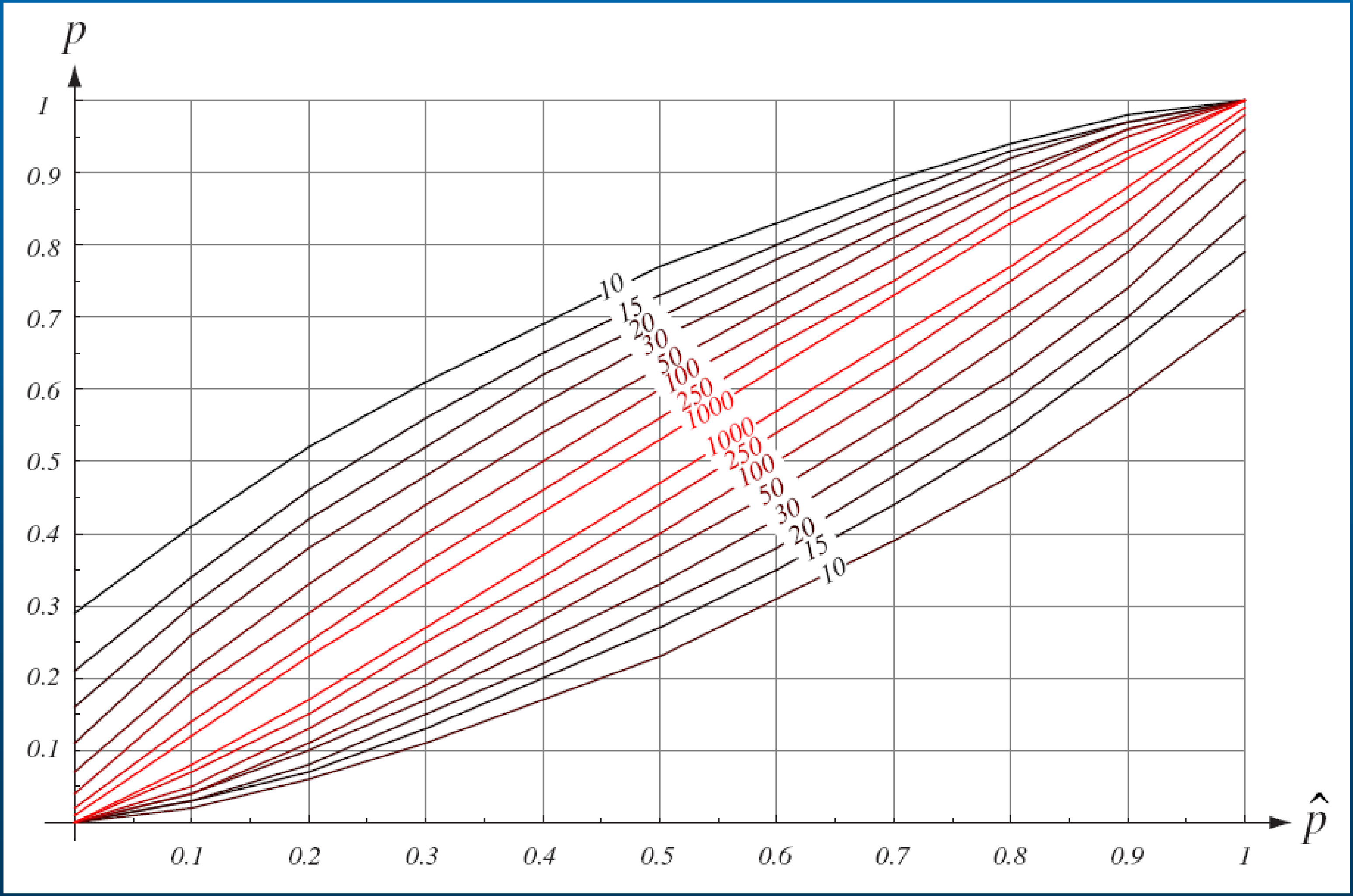


Goal 2: Need for Independent Test Data



Duda, Hart, Stork, Pattern Classification, Wiley, 2001.

Goal 2: Need for MANY Test Data



Duda, Hart, Stork, Pattern Classification, Wiley, 2001.

Data set partitioning

- BUT, most problems have <u>limited samples</u>
  - Must decide how many to use for training, validation, and testing.
  - Want sufficient <u>training</u> data to learn from
  - Want sufficient <u>test</u> data to accurately predict performance over future data
- Several strategies to maximize use of data
  - Hold-out
  - N-fold cross-validation
  - Leave-one-out / jackknife
  - Bootstrap

Testing & reporting results

1. How do we accurately measure and report the accuracy of a pattern classifier?
2. How do we objectively compare two classifiers over a given problem?
3. How can we predict how well a classifier will generalize, given it performance over our training data / testing data?

Measures of classification accuracy

- Confusion table/matrix
  - Accuracy
  - Sensitivity / recall / true positive rate
  - Specificity
  - False Positive Rate
  - False Negative Rate
  - Positive Predictive Value / precision
  - Negative Predictive Value
  - False Discovery Rate Sensitivity
  - Matthews' correlation coefficient
  - F-measure
  - G-mean
  - Application-specific measures
- Receiver Operator Characteristic Curves
  - Area under curve

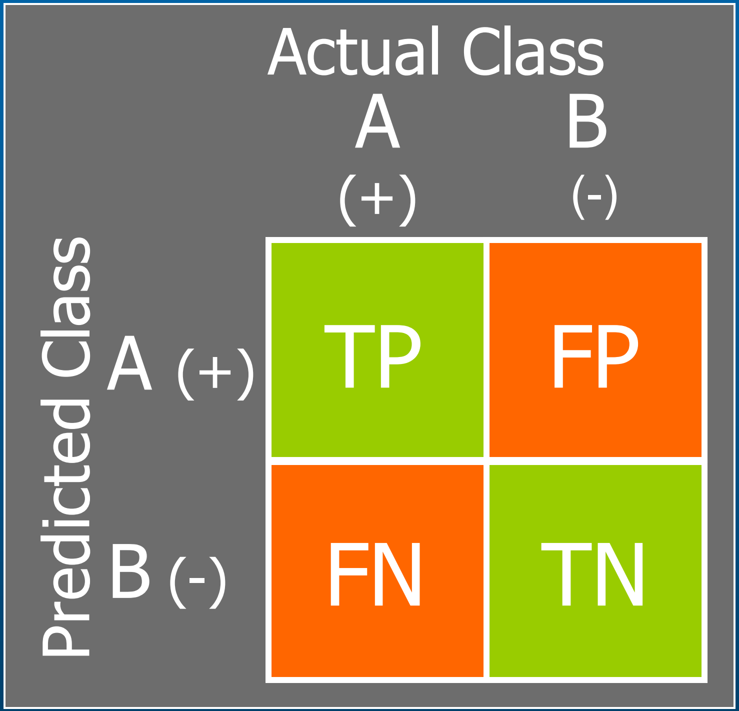
Confusion Table

- Correct predictions shown in green, errors in red.
  - Type I errors (or α error, or false positive)
  - Type II errors (β error, or a false negative)

		Actual Class	
		A (+)	B (-)
Predicted Class	A (+)	TP	FP
	B (-)	FN	TN

Confusion Table

- Accuracy = (TP+TN) / (TP+TN+FN+FP)
- Sensitivity = Sn = TP / (TP+FN)
  - aka 'recall', 'true positive rate'
- Specificity = Sp = TN / (TN+FP)
- False Positive Rate = 1-Sp
  - = FP/(TN+FP)
- False Negative Rate = 1-Sn
  - = FN/(TP+FN)
- Positive Predictive Value = TP / (TP+FP)
  - aka 'precision'
- Negative Predictive Value = TN / (TN+FN)
- False Discovery Rate = FP / (TP+FP)
- F-measure = harmonic mean of Sn & PPV
- G-mean = geometric mean of Sn&Sp



		Actual Class A (+)	Actual Class B (-)
Predicted Class	A (+)	TP	FP
Predicted Class	B (-)	FN	TN

- None of these measures in isolation can tell us how 'accurate' the classifier is.

Case study: PIPE II

- The challenge:
  - Yeast has 6200 proteins in its proteome.
  - Every possible pair of yeast proteins could potentially interact.
  - Based on biological evidence, it is believed that approx 50K interactions exist in yeast.
  - Would like to computationally predict from sequence alone whether a given pair will interact.
  - It is very expensive to verify a prediction experimentally.
- The solution:
  - We have developed a classifier which tests a given pair of protein sequences and predict whether they will interact *in vivo*.
  - We have reduced the computational complexity to the point where we can run it on all 18million pairs.
  - Through parameter tuning, we can achieve either:
    - 1) High specificity of 99% with medium sensitivity (%50)
    - 2) Very high specificity of 99.9% at the cost of a low sensitivity (25%)
- The $1M questions:
  - **Which parameter set is preferred?**
  - **How many of the predicted interactions are likely to be true interactions?**

Case study: PIPE II

Case 1

		Actual Class	
		A (+)	B (-)
Predicted Class	A (+)	25K	180K
	B (-)	25K	17.8M

Sn=50%
Sp=99%
Prec=25K/205K=12%

Case 2

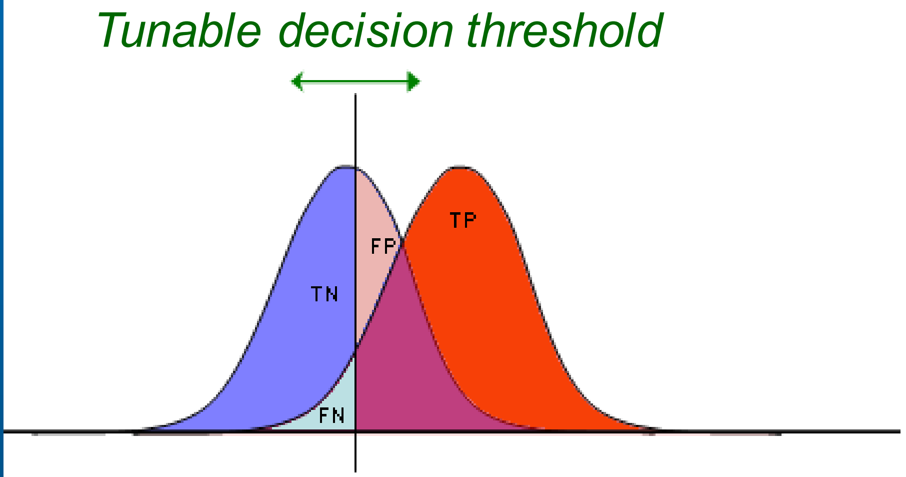
		Actual Class	
		A (+)	B (-)
Predicted Class	A (+)	12.5K	18K
	B (-)	37.5K	18M

Sn=25%
Sp=99.9%
Prec=12.5K/30.5K=42%

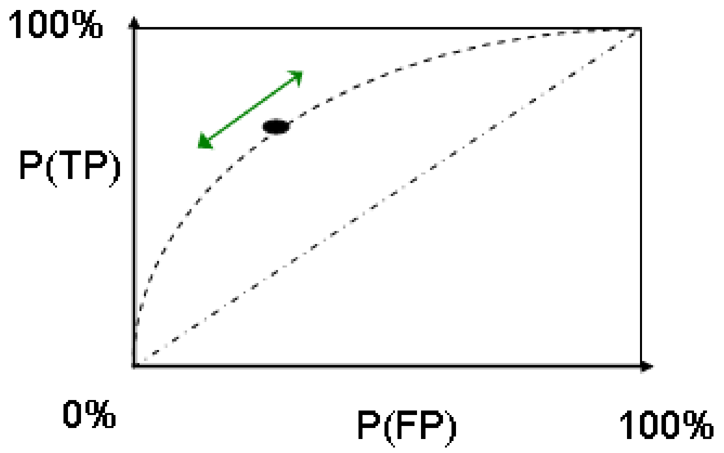
Class Imbalance

- Many events of interest are rare
  - ~500K interactions among ~250M human protein pairs (1:500)
  - 40 protein hydroxylation targets with 61 positive N/D and 1,980 negative (1:32)
  - 4M non-redundant RNA hairpins; only ~2600 known miRNA in MiRBase (<1:1500)
- Problem:
  - Classifiers tend to always predict overrepresented class & ignore rare class
- Solution:
  - Use appropriate performance metrics!
  - Random undersampling/oversampling
    - Can also create new data by adding noise to existing data
  - Adjusting cost/loss function
    - Make errors on rare class more costly

ROC Curves



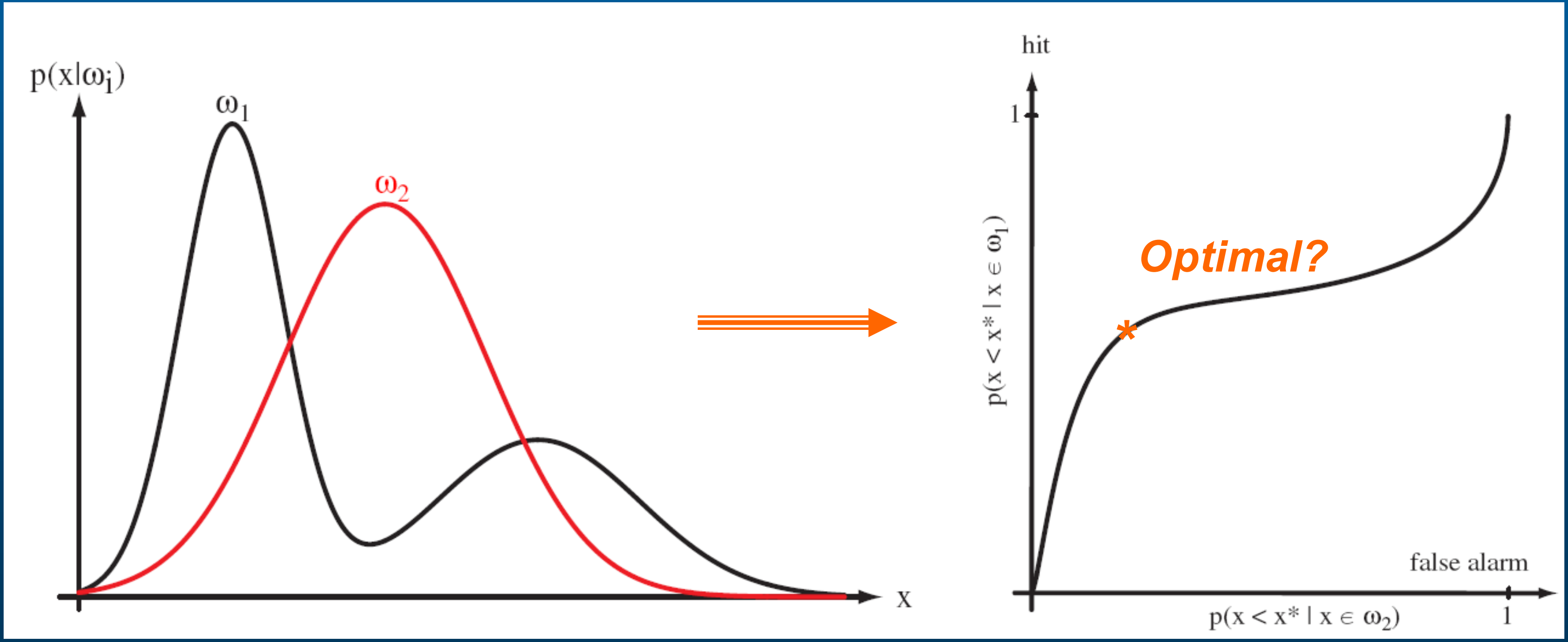
TP	FP
FN	TN
1	1



See http://www.anaesthetist.com/mnm/stats/roc/Findex.htm for a great ROC demo

ROC Curve

- Curve is not necessarily symmetric
- Can be informative in setting threshold to balance benefit of TP against cost of FP

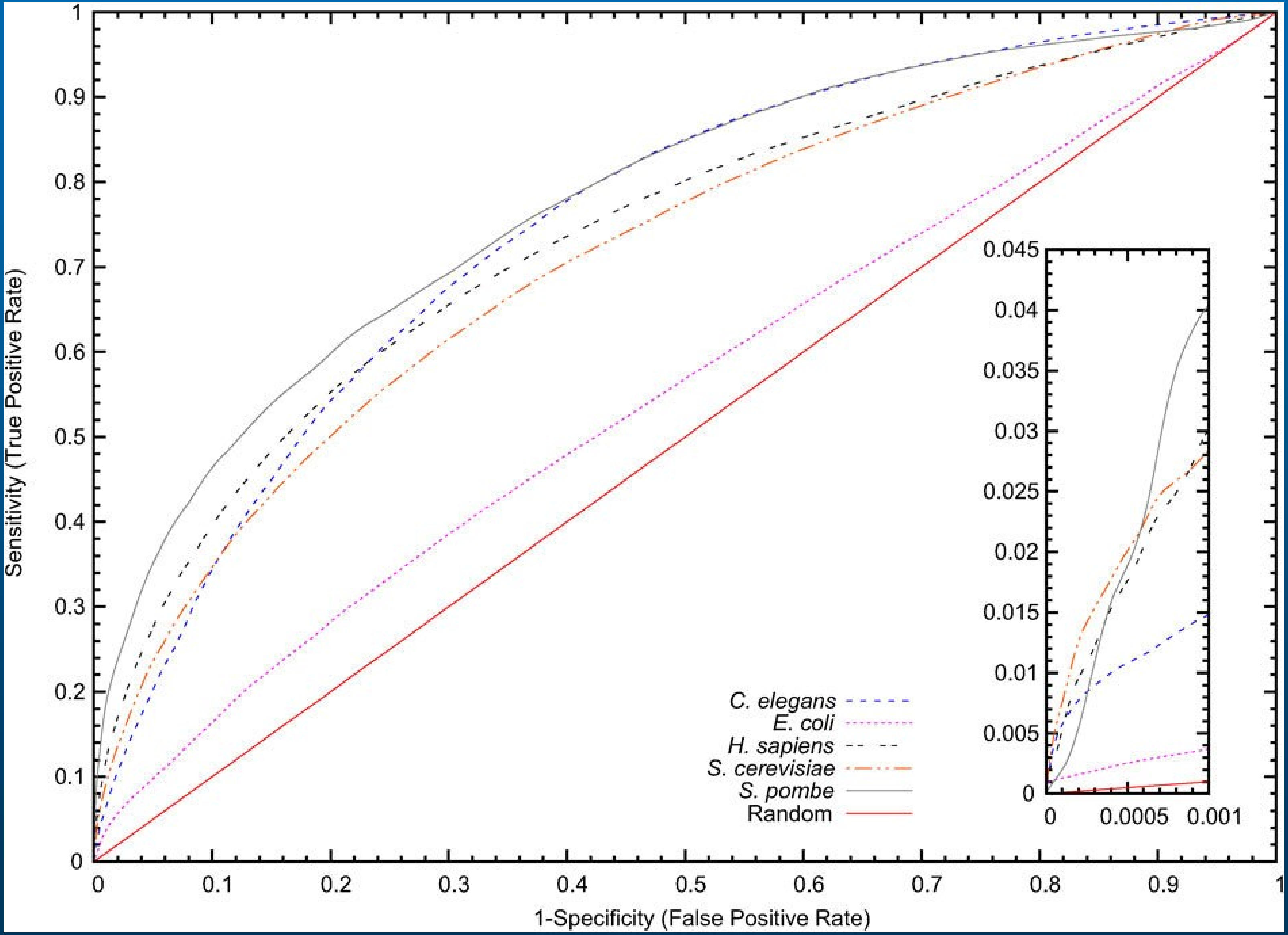


Area under the ROC Curve

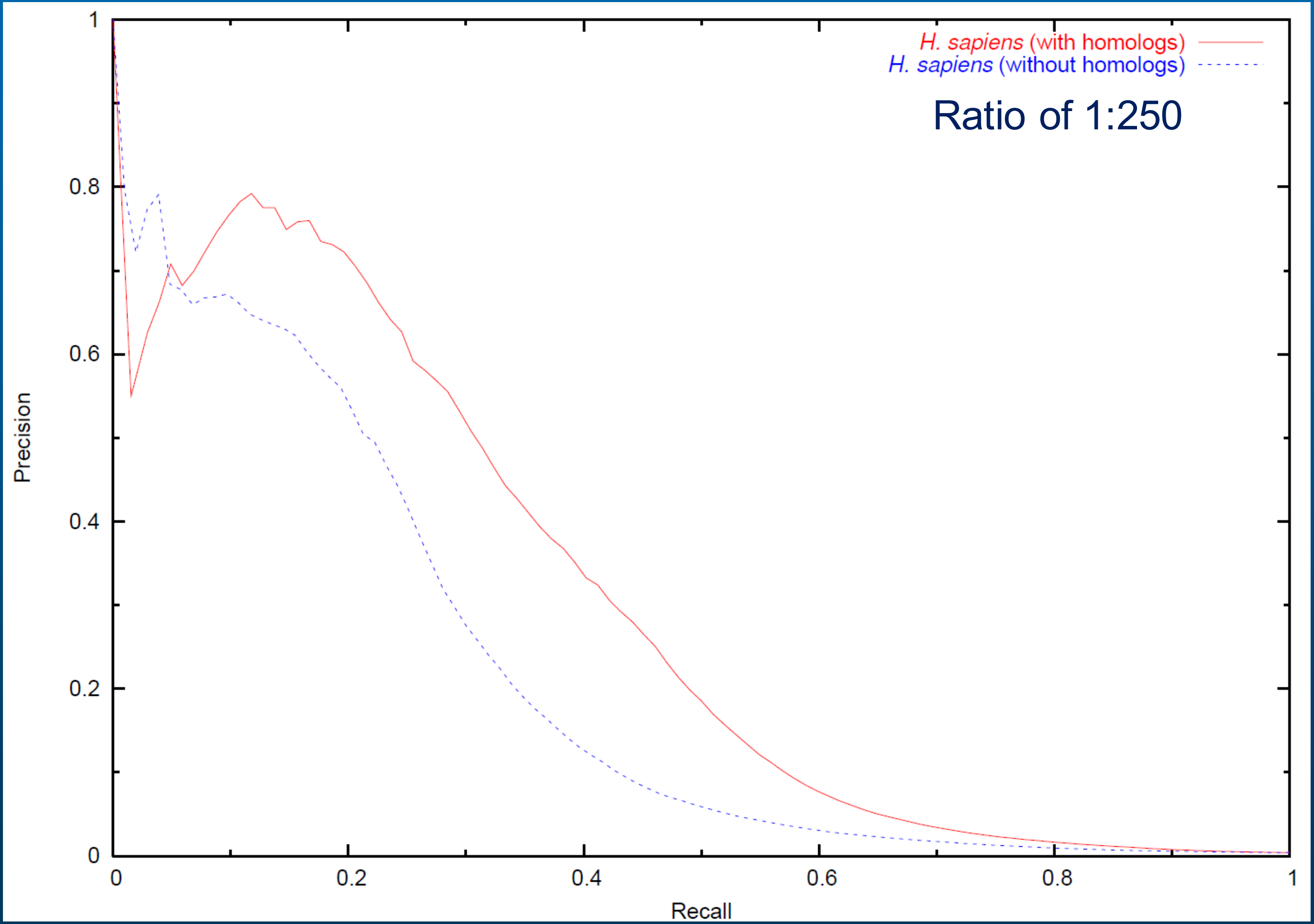
- Area under an ROC curve (AUC) summarizes performance of a classifier
  - Independent of particular cost function which might influence threshold placement
  - Ranges from 1 (perfect) to 0 (worst)
    - Random = 0.5

  - BUT, AUC is just one facet of classifier performance. May not be the most important one
    - E.g. PIPE must perform at one extreme end of the curve…

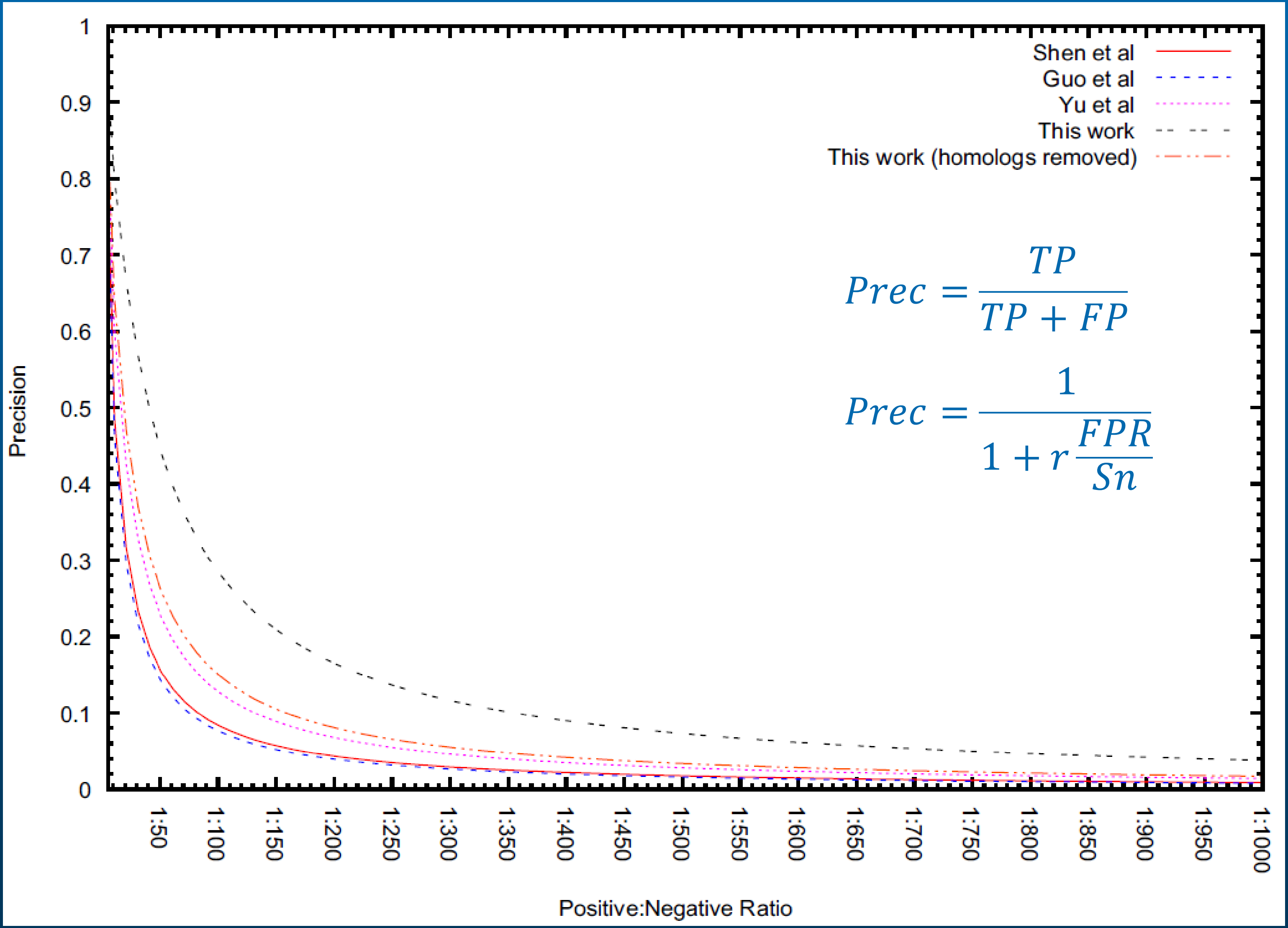
PIPE ROC Curve



Precision-Recall Curves



Precision vs. Prevalence

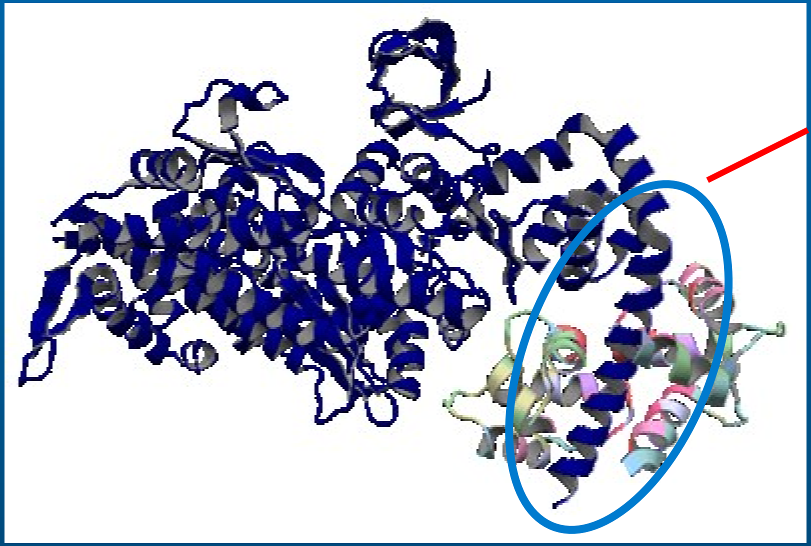


$$Prec = \frac{TP}{TP + FP}$$

$$Prec = \frac{1}{1 + r\frac{FPR}{Sn}}$$

Protein-Protein Interaction

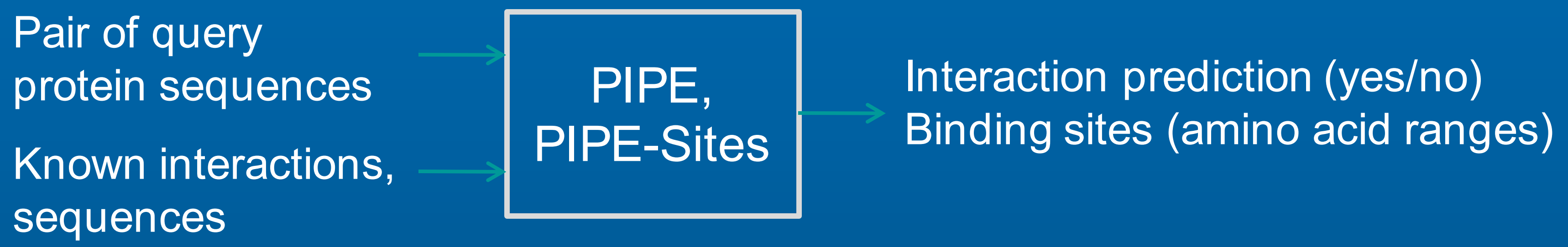
Myosin-VI ...HWLICS**<u>RWKKVQWCSLSVIKLKNKI</u>**KYRAE



MADQLTEEQI**<u>AEFKEAFSLF</u>**DKDGDG...EE**<u>EIREAFRVF</u>**DKD...

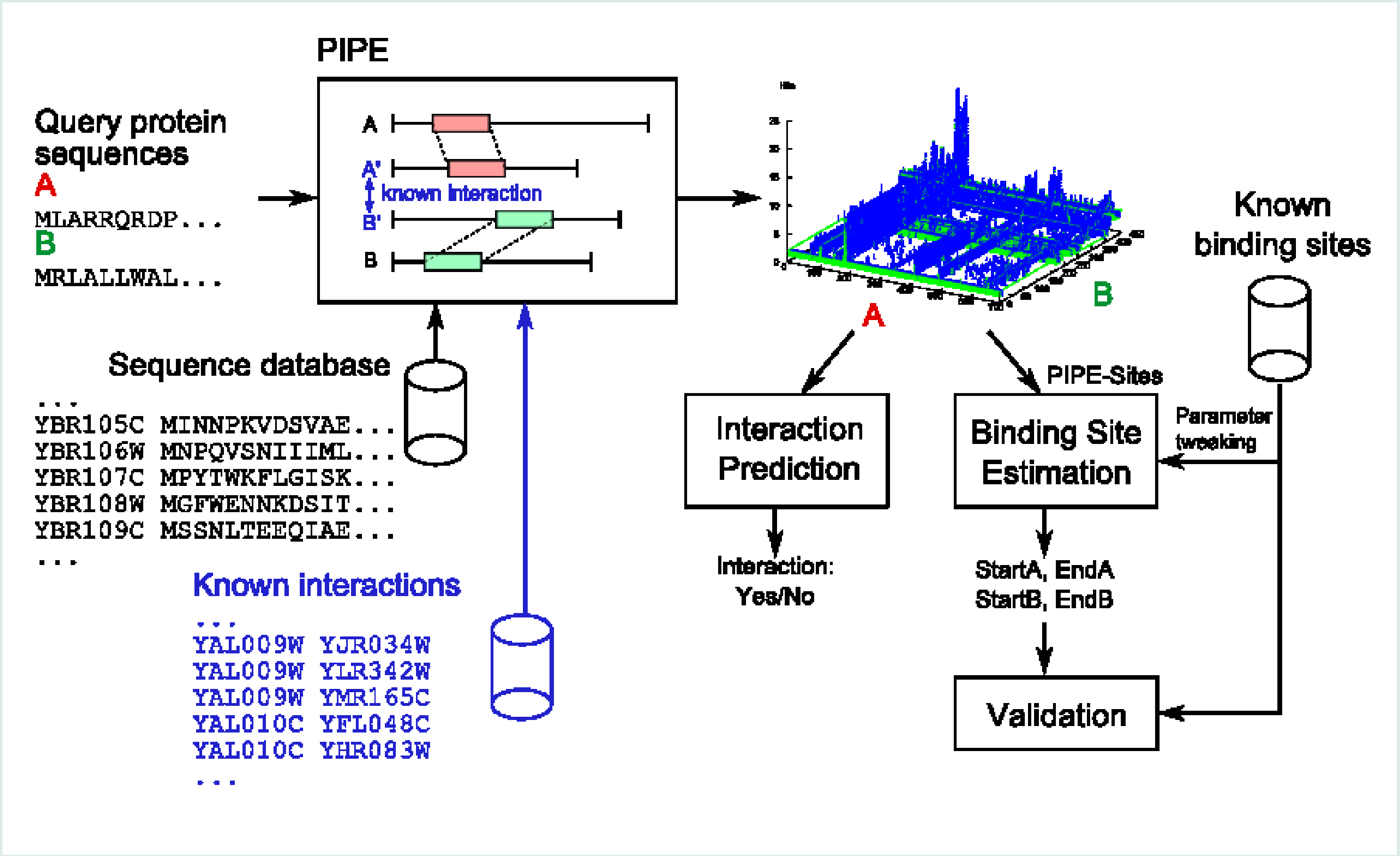
- Valuable for understanding protein function
- Costly to determine experimentally

PPI Prediction @ CU


Pair of query protein sequences
Known interactions, sequences
PIPE, PIPE-Sites
Interaction prediction (yes/no)
Binding sites (amino acid ranges)

- “PIPE”: Protein Interaction Prediction Engine
  - Best observed performance at high specificity (99.95%), crucial for proteome-wide prediction
  - 22K known human proteins: still $(22K)^2 / 2 \times 0.05\% = 121K$ false positives
- PIPE-Sites: binding sites
  - Identifies actual site of protein-protein interface
  - Accuracy confirmed using databases of experimentally determined binding sites

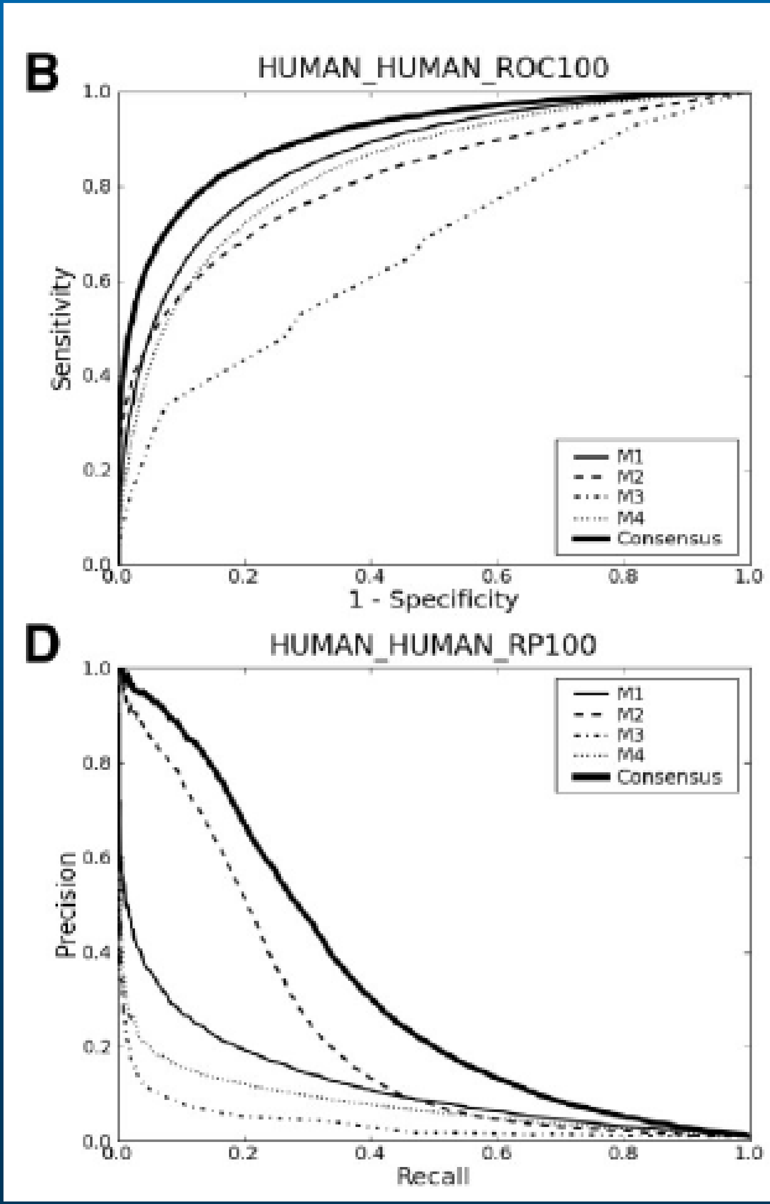
PIPE Detail


PIPE
Query protein sequences
A
MLARRQRDP...
B
MRLALLWAL...
A
A'
known interaction
B'
B
Sequence database
...
YBR105C MINNPKVDSVAE...
YBR106W MNPQVSNIIIML...
YBR107C MPYTWKFLGISK...
YBR108W MGFWENNKDSIT...
YBR109C MSSNLTEEQIAE...
...
Known interactions
...
YAL009W YJR034W
YAL009W YLR342W
YAL009W YMR165C
YAL010C YFL048C
YAL010C YHR083W
...
A
B
Known binding sites
PIPE-Sites
Interaction Prediction
Binding Site Estimation
Parameter tweaking
Interaction: Yes/No
StartA, EndA
StartB, EndB
Validation

PIPE Performance

ROC

Precision-Recall



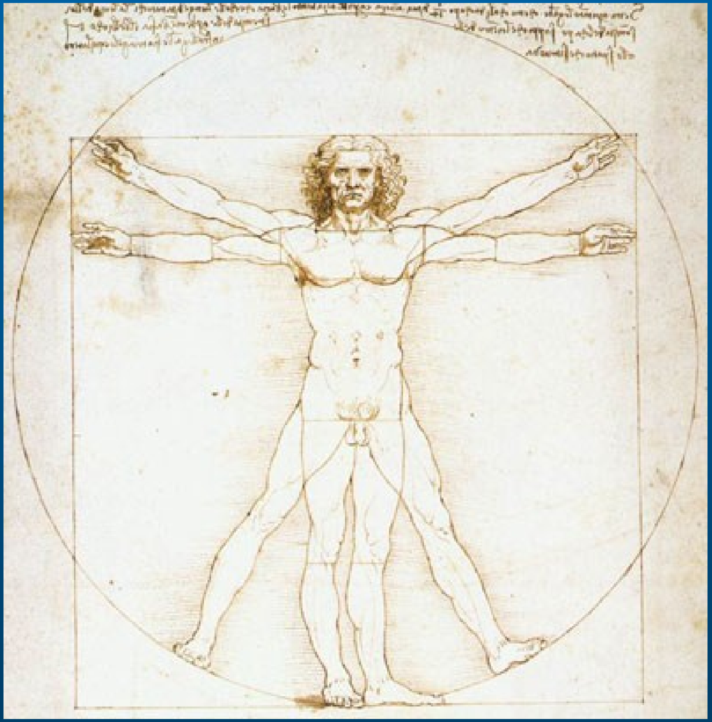
From: Park, BMC Bioinformatics, 2009, 10:419

PIPE: Yeast Global Scan

- PIPE has been used to do a **global scan of yeast**:
  - **29,589 interactions** detected (**14,438 novel** at the time of the experiment, some interactions were later confirmed by other traditional experiments).
- Using up-to-date data in 2013, a new global scan of the yeast genome resulted in **~87,000 PPIs**, more than yeast was expected to include.

PIPE: *Homo Sapiens* Global Scan

- First ever "complete" human interactome!
  - Other methods can only examine ~25% of protein pairs
    - Computational complexity (PIPE <1s per pair)
    - Availability of input features (e.g. structure)
  - Now applying network analysis
    - (e.g. pathways)



Homo Sapiens (Human)*

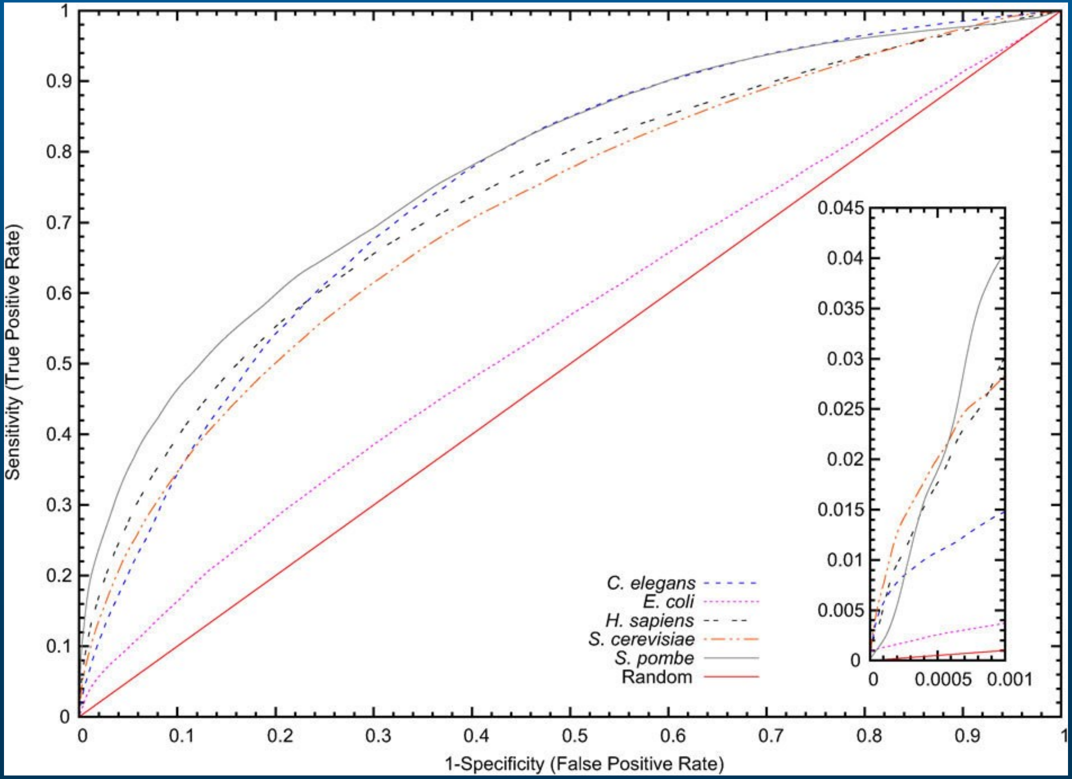
* Image from BrainMaps.org

PIPE: Global Human Results

- Human genome is believed to code for 20,000–40,000 protein-coding genes & contain between 154,000 and 600,000 interactions.
- Online Predicted Human Interaction Database contains 47,221 interactions involving 10,579 unique proteins (8-31% of estimated total).
- We conducted a global scan of all possible human protein pairs which resulted in over **170,000 PPIs** → 4x increase in knowledge
- The experiments were conducted on HPCVL's Victoria Falls cluster.
  - 1168 Sun UltraSparc T2+ cores.
  - Total runtime: **three months**.

Cross-Organism Predictions

- One of the nice features of PIPE is the ability to predict new interactions in one organism by using known interactions in another.
- This makes it possible to predict PPI in a **newly sequenced organism**, something most other methods can't do.

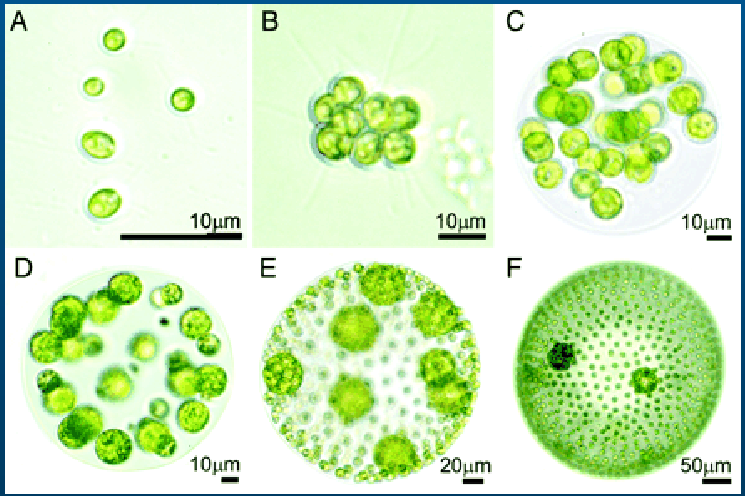


PIPE: Seasonal Allergic Rhinitis (SAR)

- Collaborative project with:
  - Department of Pediatrics, Gothenburg University, Gothenburg, Sweden.
  - The Centre for Individualized Medication, Linköping University. Linköping, Sweden.
  - Banting and Best Department of Medical Research, Donnelly Centre, University of Toronto, Toronto, Canada.
- “Hay fever”
- Study to find new biomarkers to identify SAR in patients.
- Results were supported by patient data.

PIPE: Volvox/Chlamy/Gonium

- Collaborative project with:
  - Bradley Olson (Olson Lab, Kansas State)
  - Pierre Durand (Wits University, South Africa)
  - Jonathan Featherston (Agricultural Research Council, South Africa)
  - Richard E. Michod (University of Arizona)
- Chlamydomonas (*C. reinhardtii*)
  - Unicellular (undifferentiated cells).
- Goniaceae (*G. pectorale)*
  - Unicellular, but forms colonies.
- Volvocaceae *(V. carteri)*
  - Multicellular.

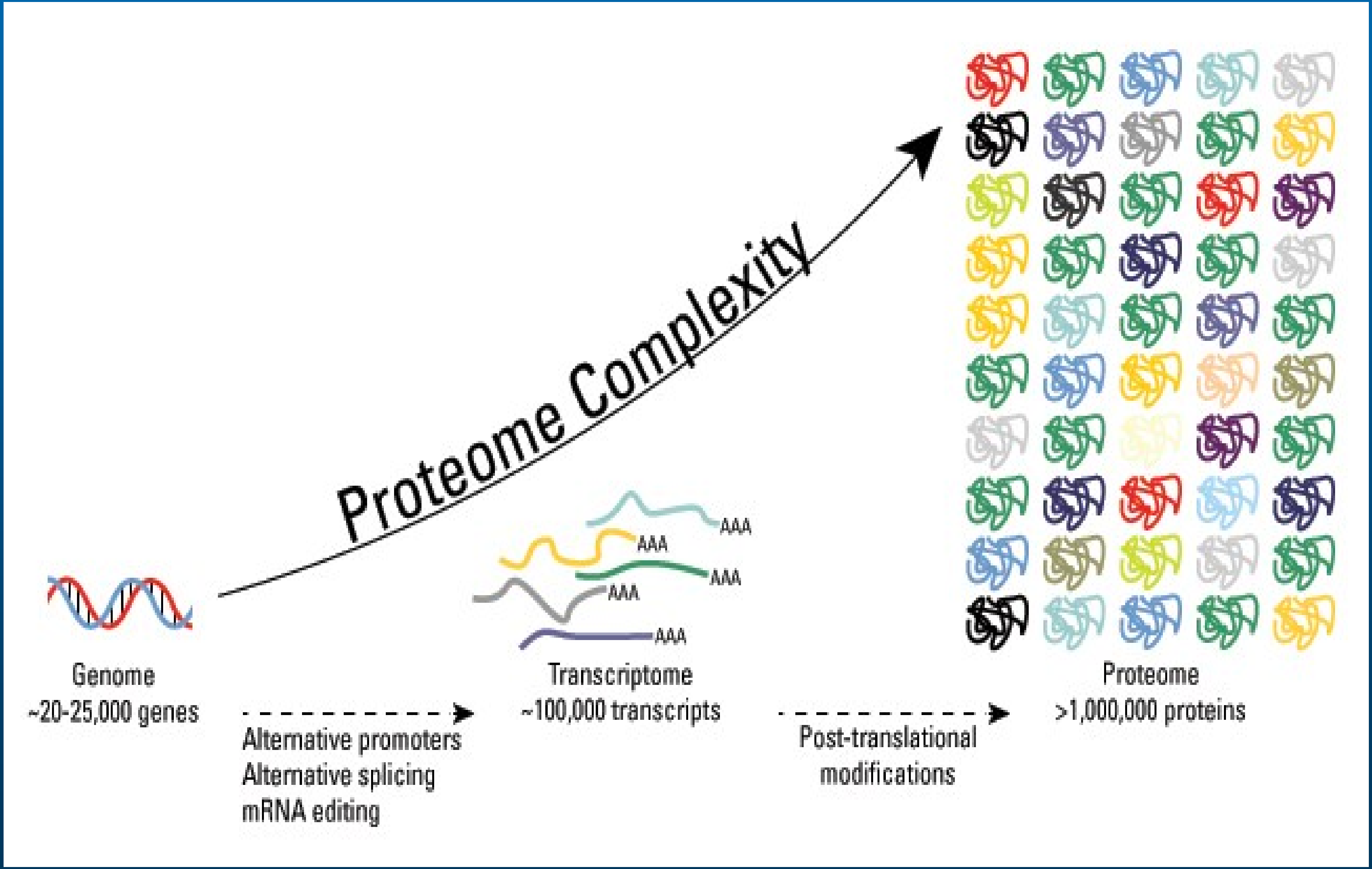


Richard E. Michod, Evolution of individuality during the transition from unicellular to multicellular life, PNAS, 2007

Other Results

- PIPE has also be used to predict interactions between organisms and viruses such as:
  - Influenza (H1N1)
  - HIV
  - Hepatitis B, C
- An obstacle to predicting Human-Virus interactions is the small number of known interactions.

PTM Prediction

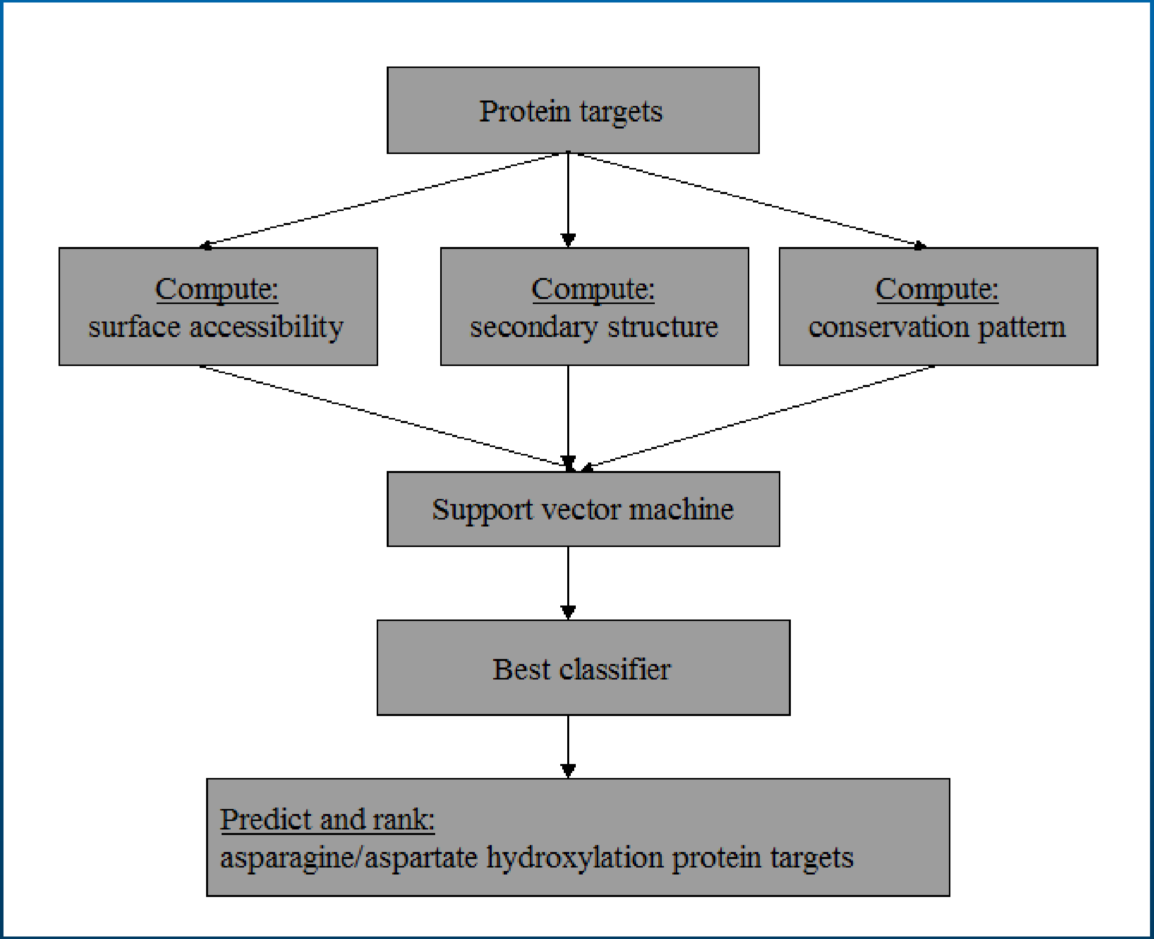


http://www.design4evolution.net/efficient-genome-building

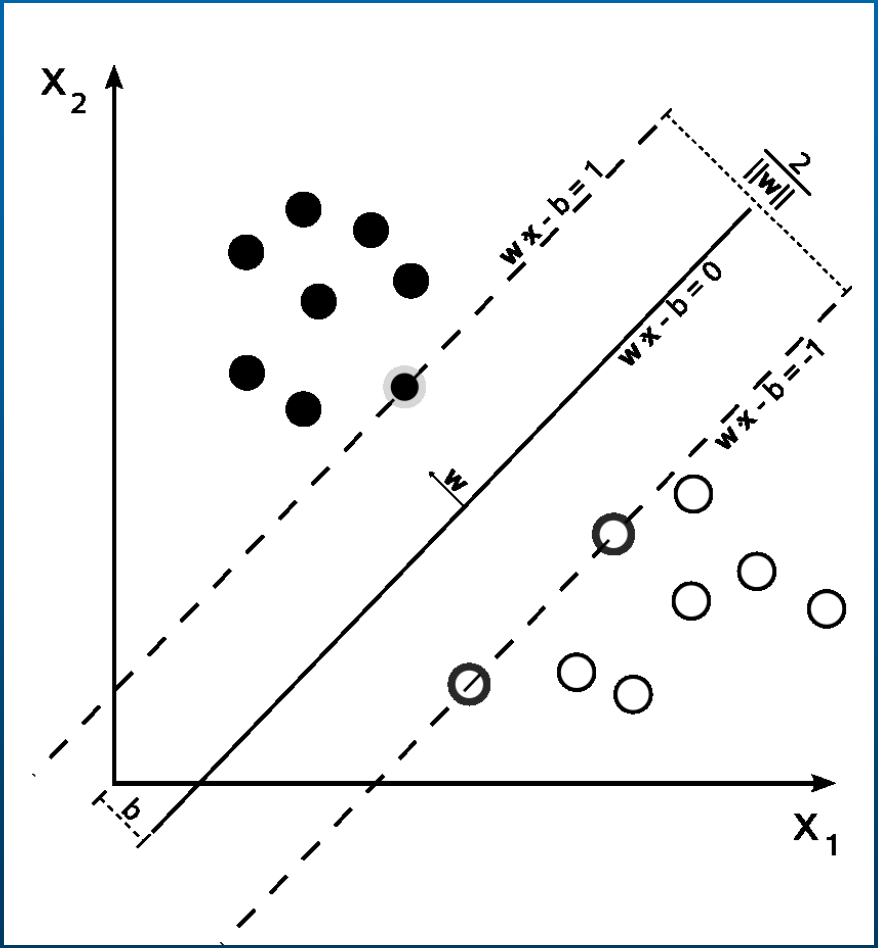
PTM Prediction

- Known N/D hydroxylation data limited
  - Identified 40 known hydroxylation targets
    - dbPTM & literature review
    - 22 possess EGF domain, 16 ankyrin repeat domain
  - 60 positives sites, 1980 (*presumed*) negatives
    - Extracted windows of ±7 AAs around N/D
    - Eliminated duplicate windows: 47+, 1223-
  - Trained/evaluated SVM using LOO test
    - 92.7% recall; 61.45% precision
- Applied to all 1.3M N/D in human proteins
  - *Now what?*

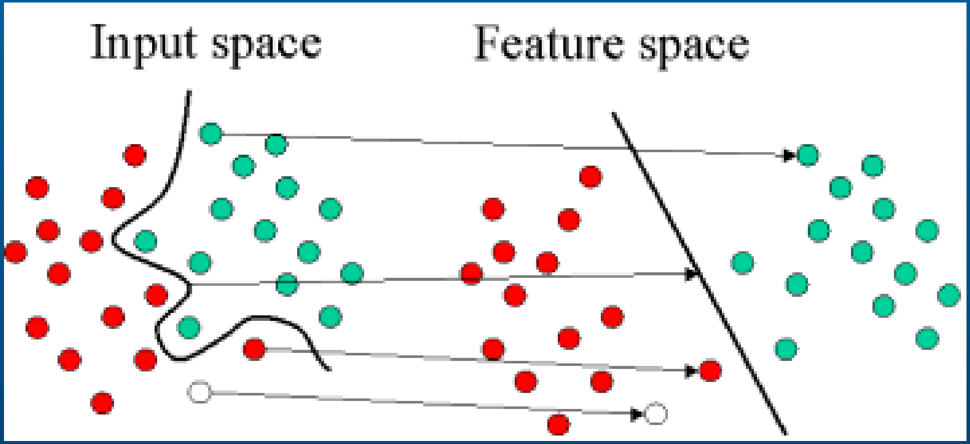
PTM Prediction


Protein targets
Compute: surface accessibility
Compute: secondary structure
Compute: conservation pattern
Support vector machine
Best classifier
Predict and rank: asparagine/aspartate hydroxylation protein targets

PTM Prediction - SVM



x_2
x_1
$\mathbf{w} \cdot \mathbf{x} - b = 1$
$\mathbf{w} \cdot \mathbf{x} - b = 0$
$\mathbf{w} \cdot \mathbf{x} - b = -1$
$\frac{2}{\|\mathbf{w}\|}$
$\mathbf{w}$
b

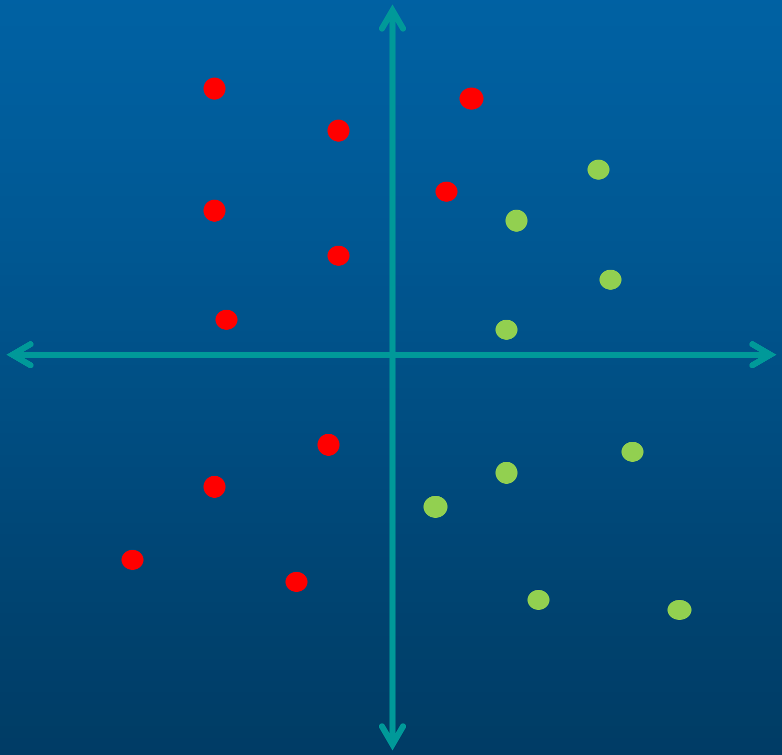


PTM Prediction

- Known N/D hydroxylation data limited
  - Identified 40 known hydroxylation targets
    - dbPTM & literature review
  - 60 positives sites, 1980 (*presumed*) negatives
    - Extracted windows of ±7 AAs around N/D
    - Eliminated duplicate windows: 47+, 1223-
  - Trained/evaluated SVM using LOO test
    - 92.7% recall; 61.45% precision
- Applied to all 1.3M N/D in human proteins
  - *Now what?*

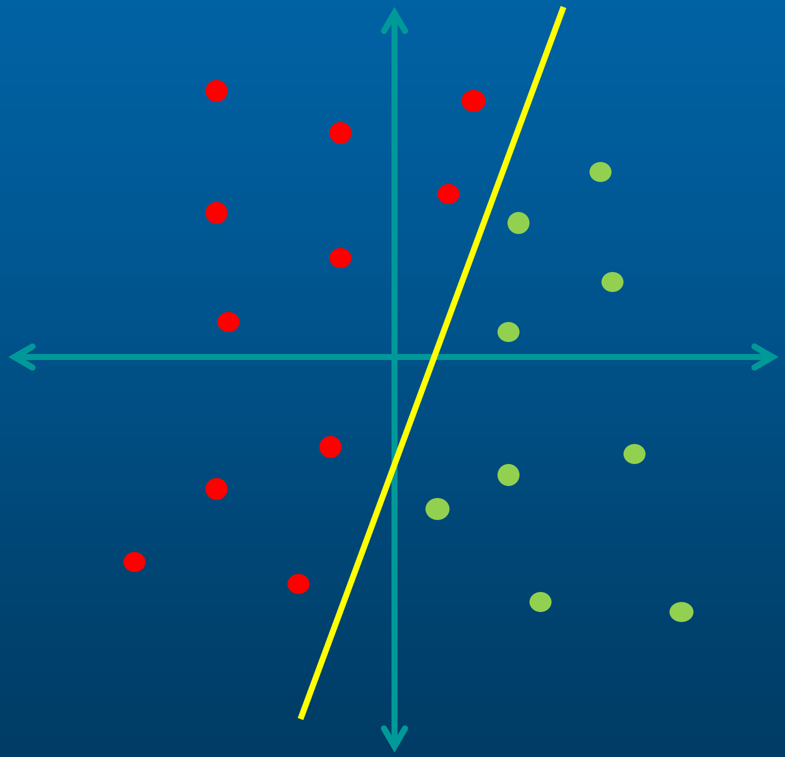
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



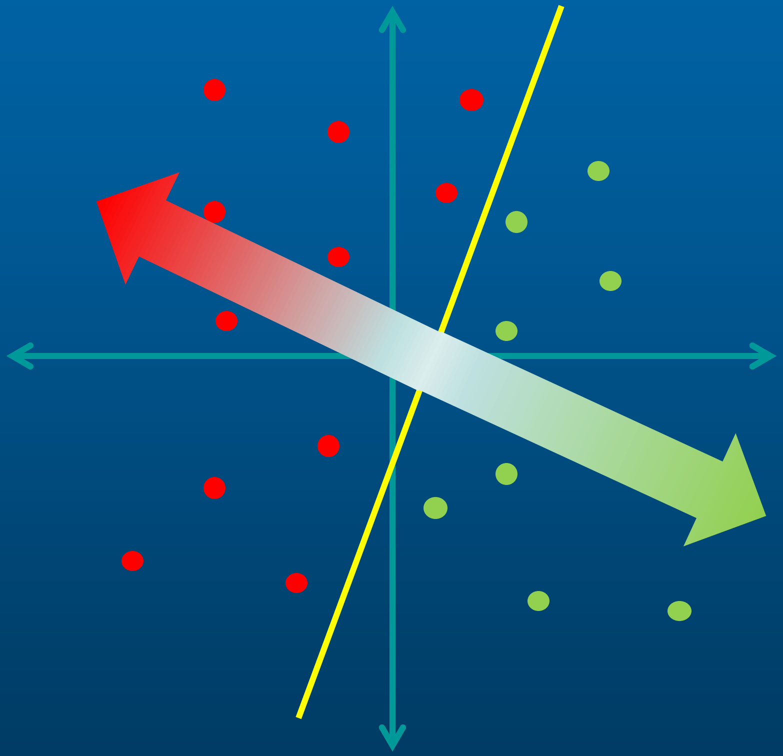
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



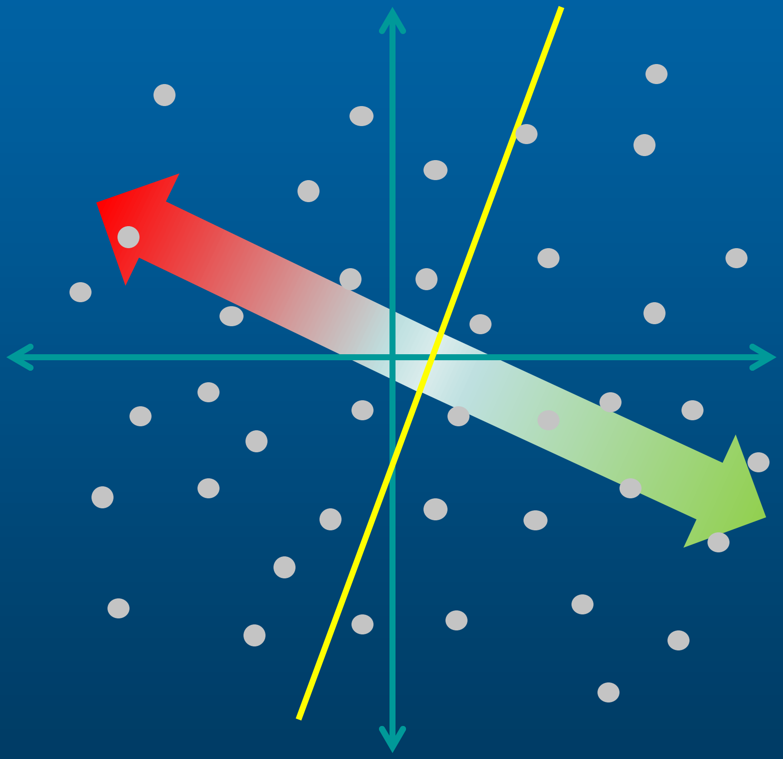
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



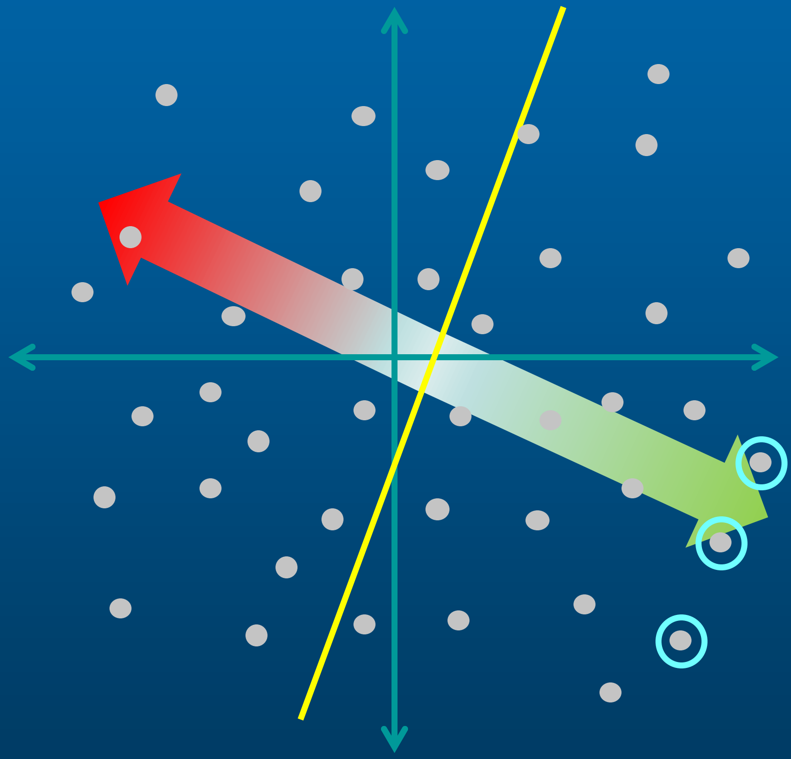
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



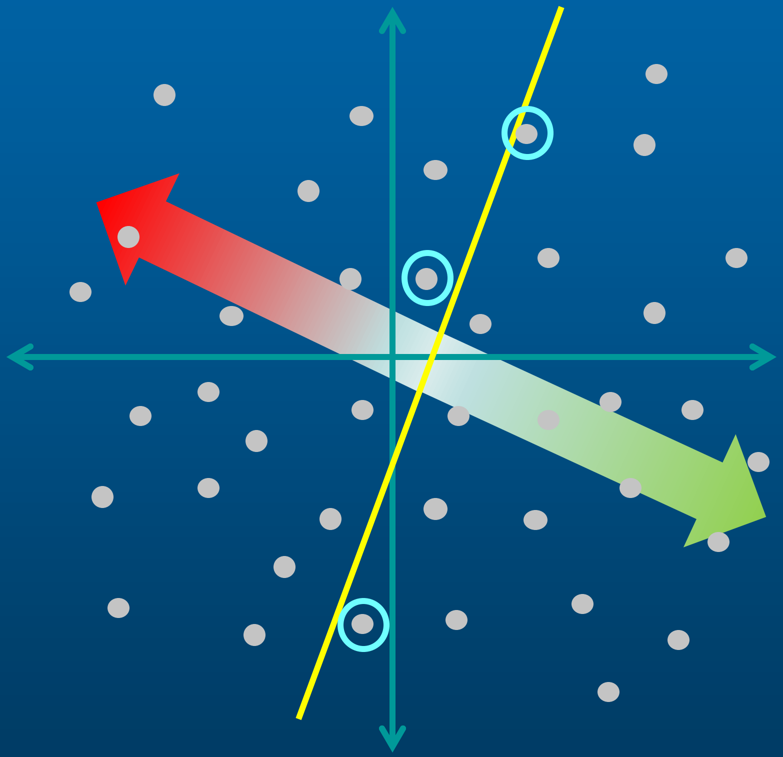
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain Classifier



Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



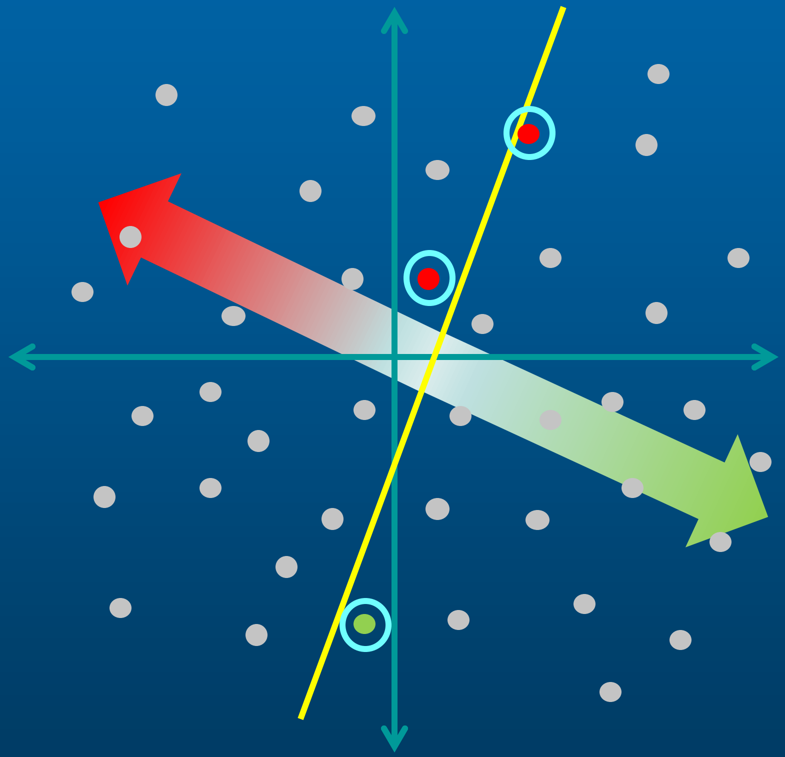
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifer

?

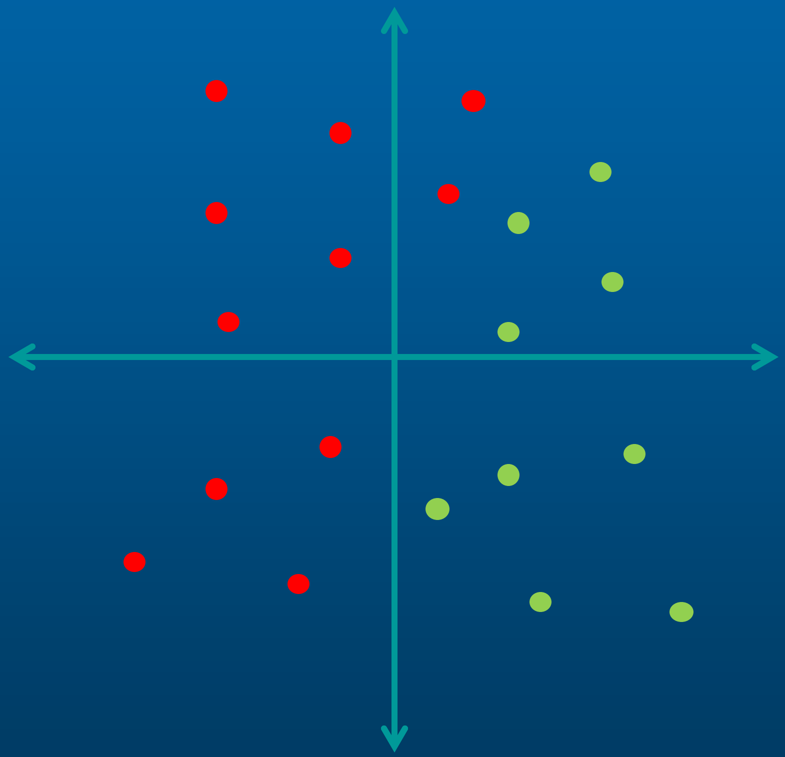
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



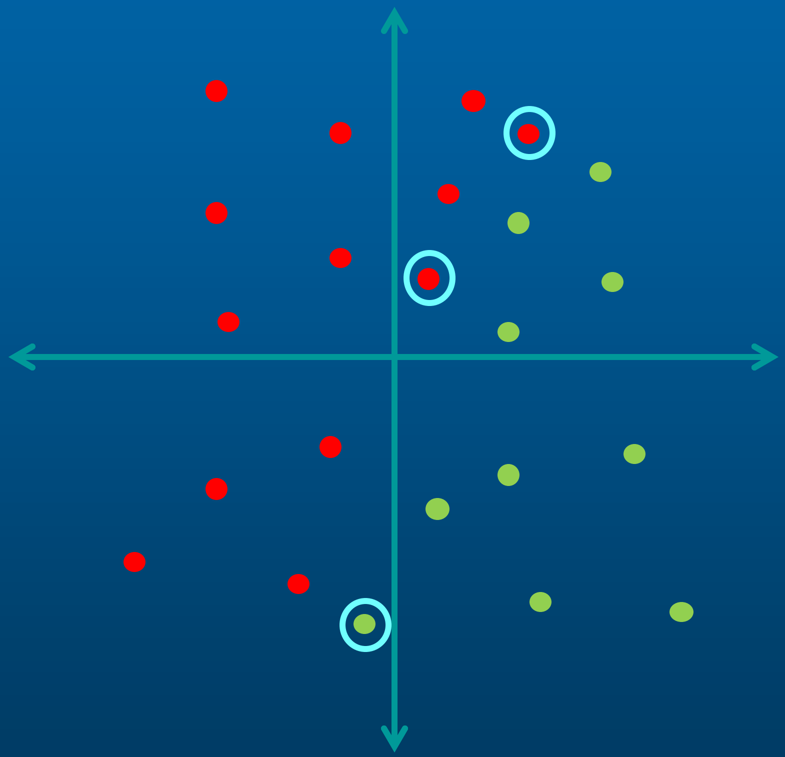
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifer



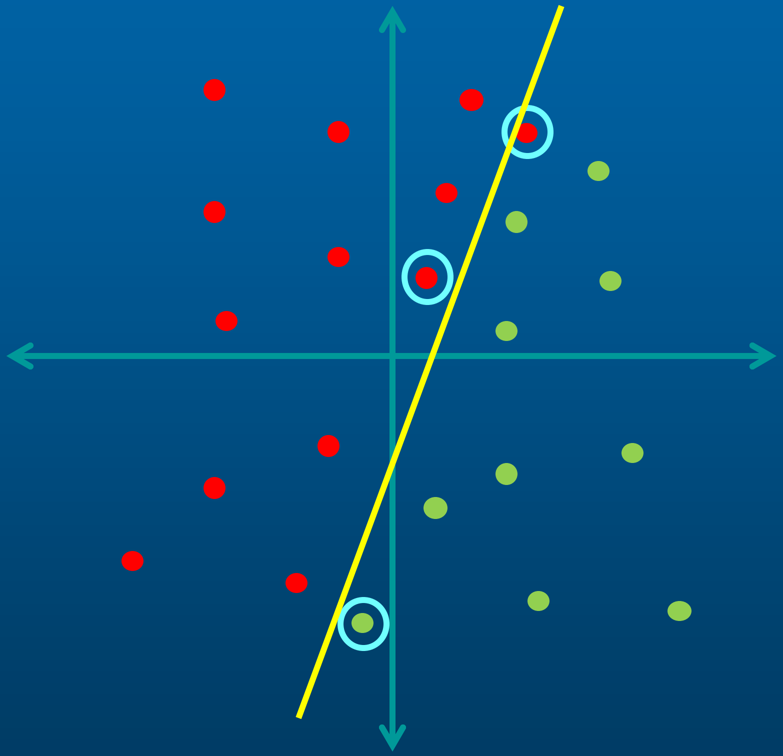
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifier



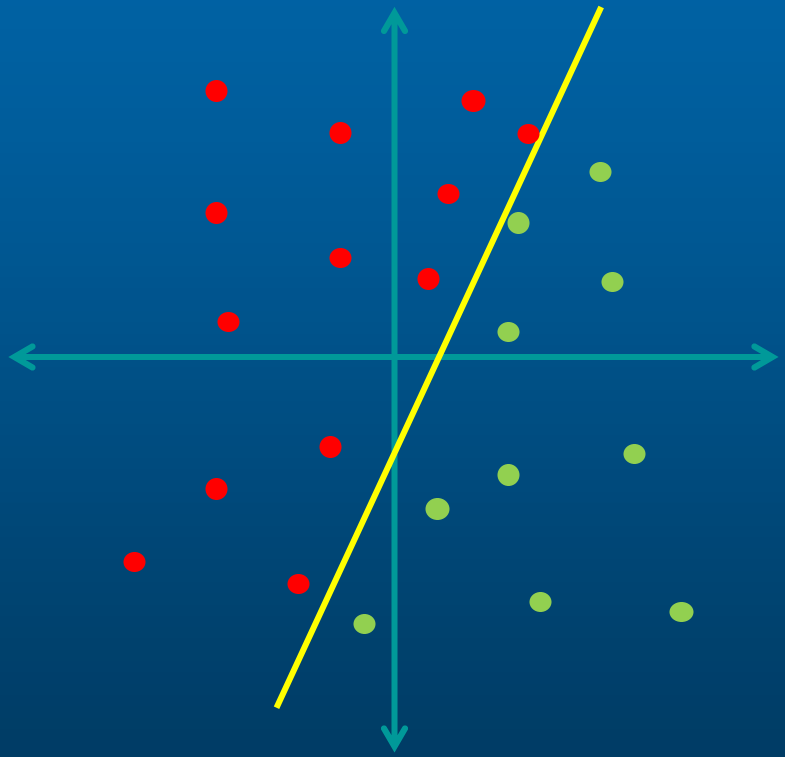
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifer

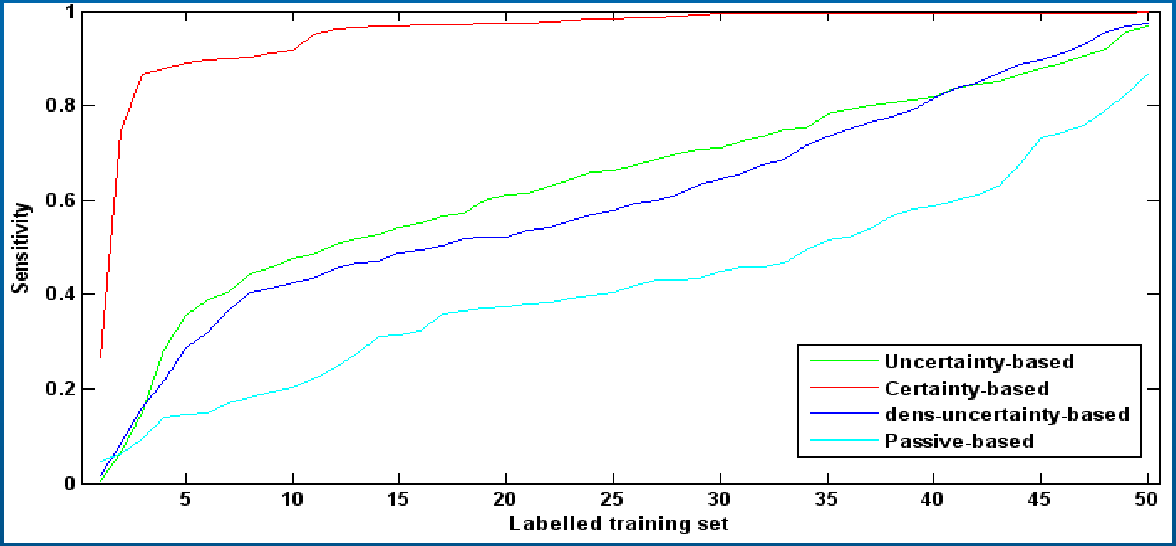


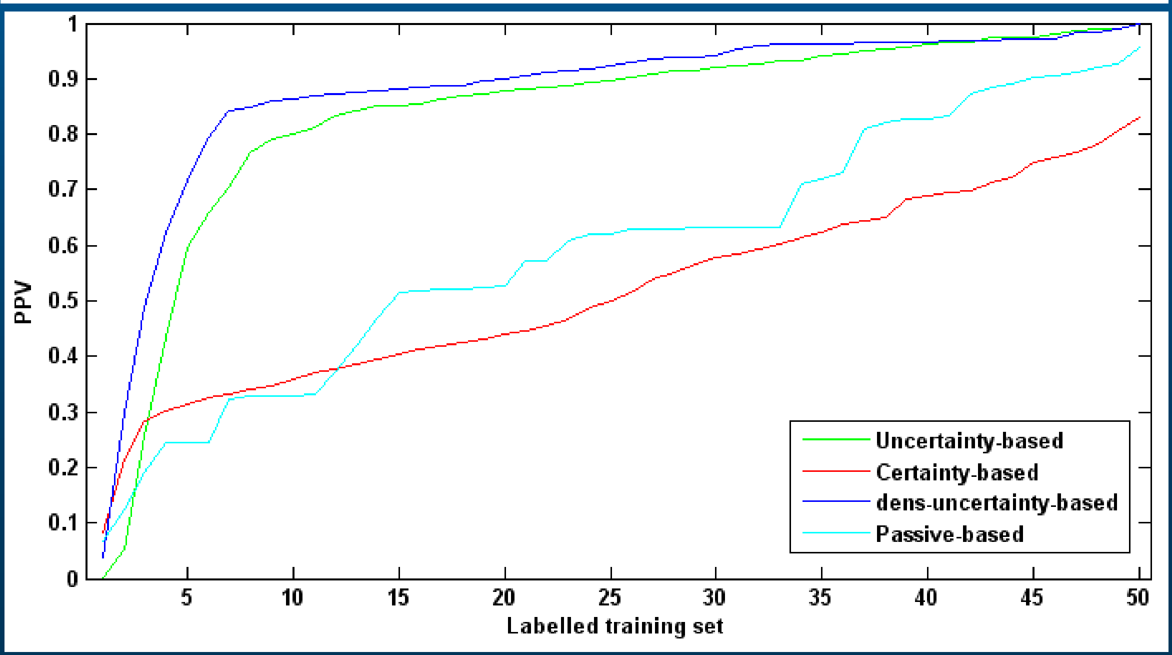
Active Learning

1. Collect labelled training data
2. Train a classifier
3. Apply to unlabelled data
4. Select points to validate
5. Perform wetlab validation
6. Add newly labelled samples to training data
7. Retrain classifer

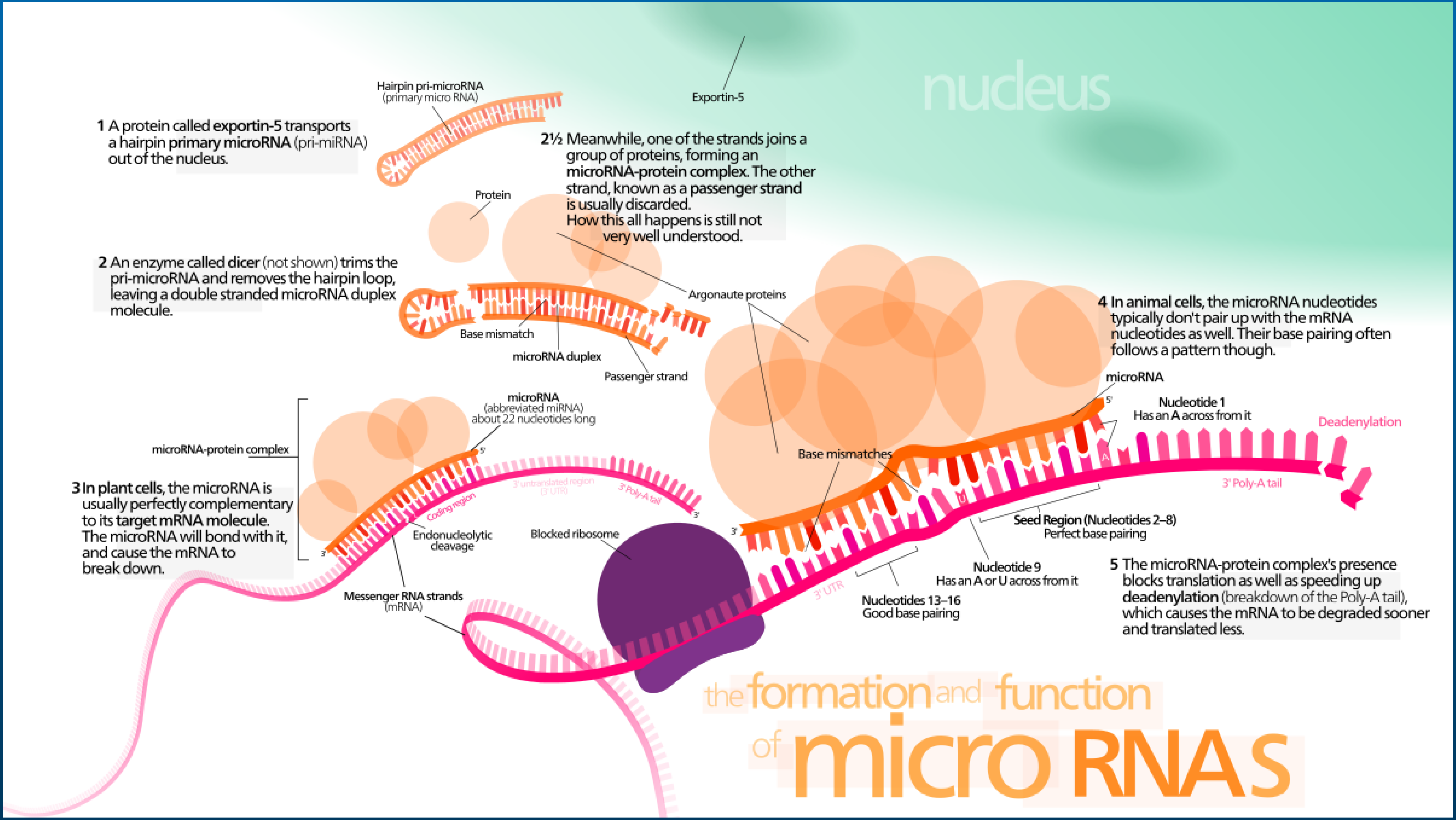


Active Learning





miRNA


nucleus
Hairpin pri-microRNA (primary micro RNA)
Exportin-5
Protein
Argonaute proteins
Base mismatch
microRNA duplex
Passenger strand
microRNA (abbreviated miRNA) about 22 nucleotides long
microRNA-protein complex
Coding region
3' untranslated region (3' UTR)
3' Poly-A tail
Endonucleolytic cleavage
Blocked ribosome
Messenger RNA strands (mRNA)
Base mismatches
microRNA
Nucleotide 1 Has an A across from it
Deadenylation
3' Poly-A tail
Seed Region (Nucleotides 2–8) Perfect base pairing
Nucleotide 9 Has an A or U across from it
Nucleotides 13–16 Good base pairing
3' UTR
the formation and function of micro RNAs

1 A protein called **exportin-5** transports a hairpin **primary microRNA** (pri-miRNA) out of the nucleus.

2 An enzyme called **dicer** (not shown) trims the pri-microRNA and removes the hairpin loop, leaving a double stranded microRNA duplex molecule.

2½ Meanwhile, one of the strands joins a group of proteins, forming an **microRNA-protein complex.** The other strand, known as a **passenger strand** is usually discarded. How this all happens is still not very well understood.

3 **In plant cells,** the microRNA is usually perfectly complementary to its **target mRNA molecule.** The microRNA will bond with it, and cause the mRNA to break down.

4 **In animal cells,** the microRNA nucleotides typically don't pair up with the mRNA nucleotides as well. Their base pairing often follows a pattern though.

5 The microRNA-protein complex's presence blocks translation as well as speeding up **deadenylation** (breakdown of the Poly-A tail), which causes the mRNA to be degraded sooner and translated less.

Wikipedia Commons

miRNA Prediction

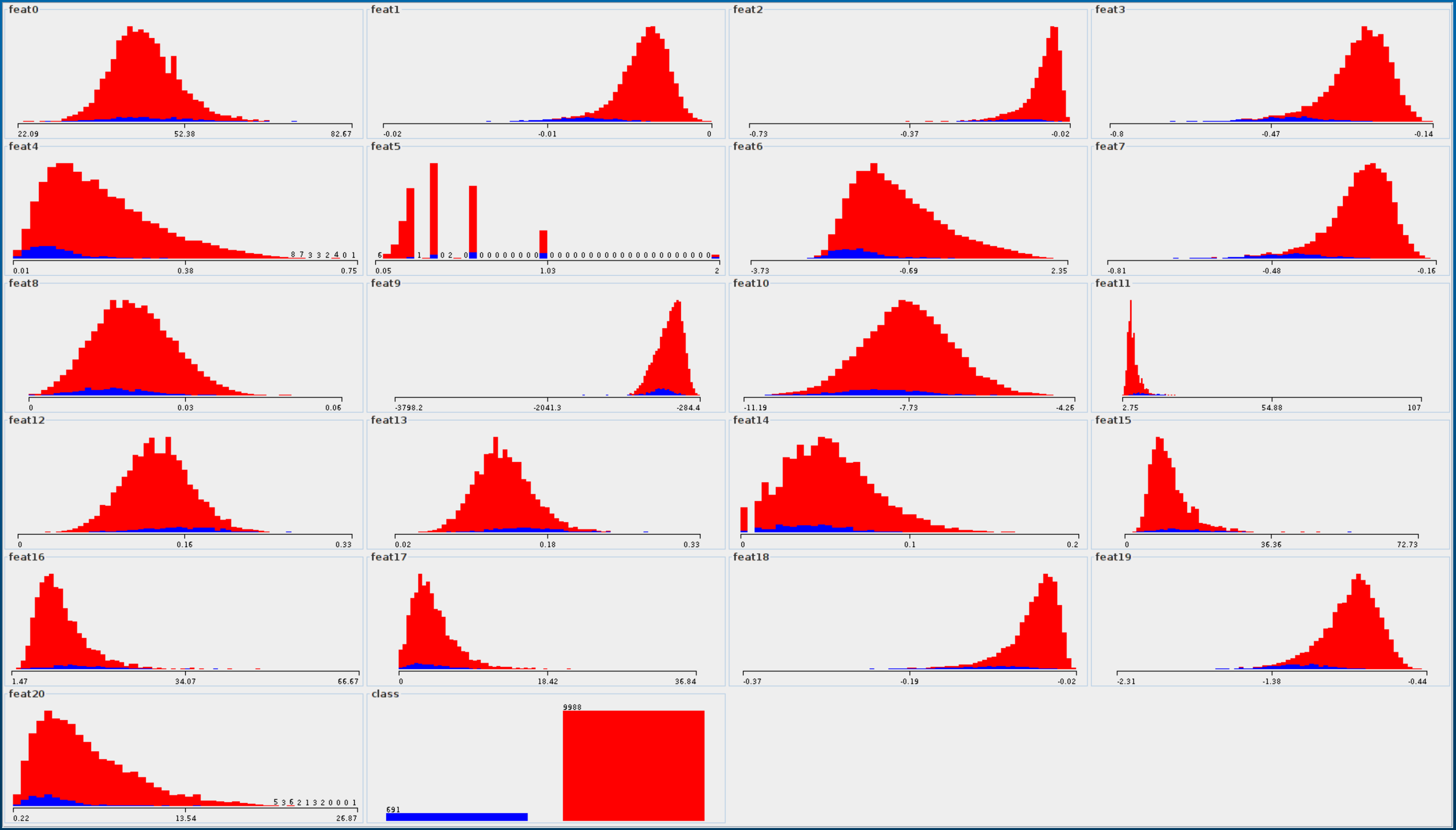
- microPred most widely used miRNA prediction tool
  - Trained on human known miRNAs
  - Uses 21 features, 5 of which relate to secondary structure free energy
    - Problem?
  - Accuracy evaluated using geometric mean
    - What are they failing to account for?
  - Tested on other species, <u>sensitivity maintained</u>
    - What is missing?

Redundant Features

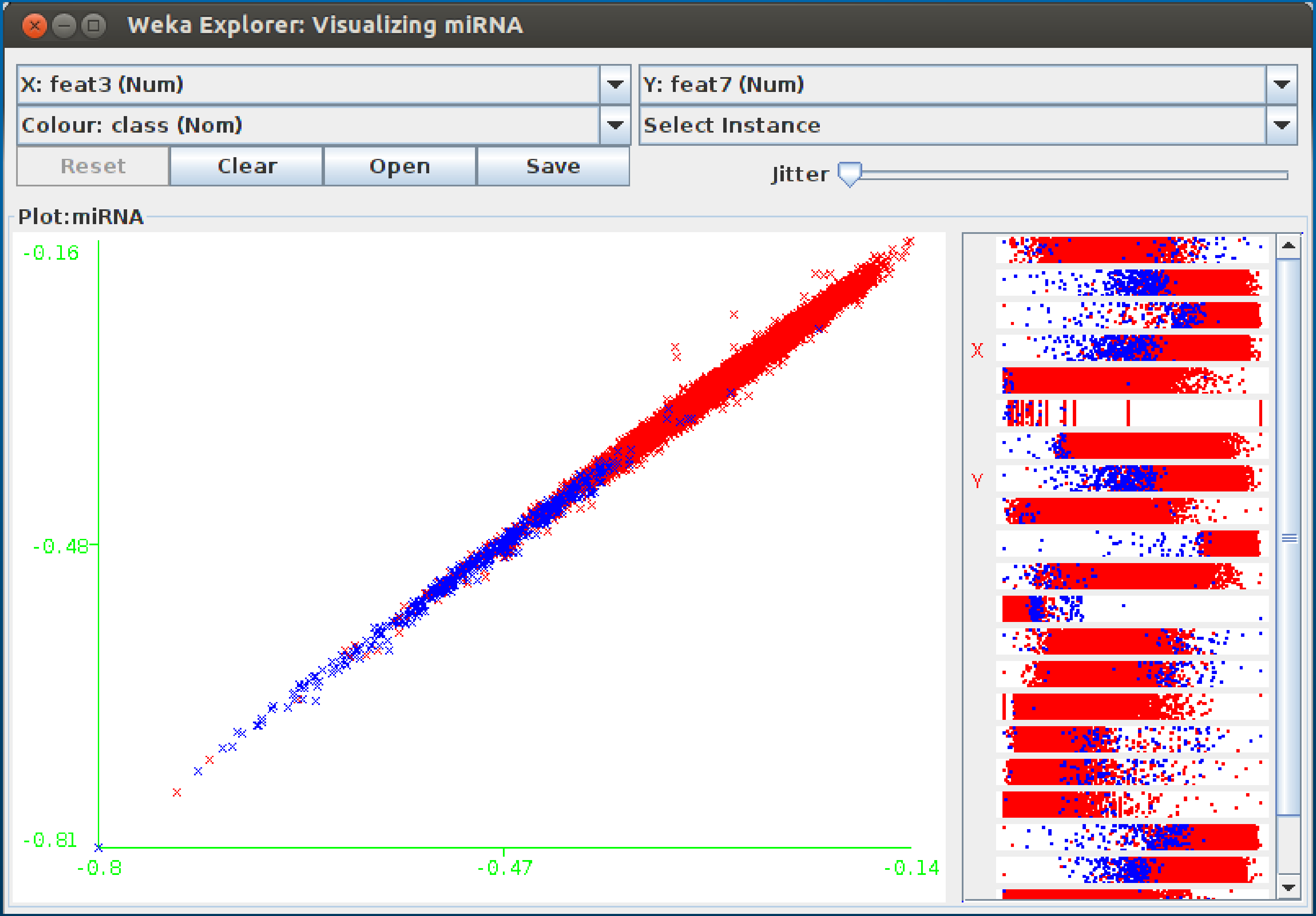
- We observed that specificity of method varied wildly
  - Depended on negative set used (hairpins from 100 random coding regions)

- Recall that:
  - ➢ *Some methods will actually do worse with more features*
    - *May be overly sensitive to noisy features*
    - *May overweight redundant features*

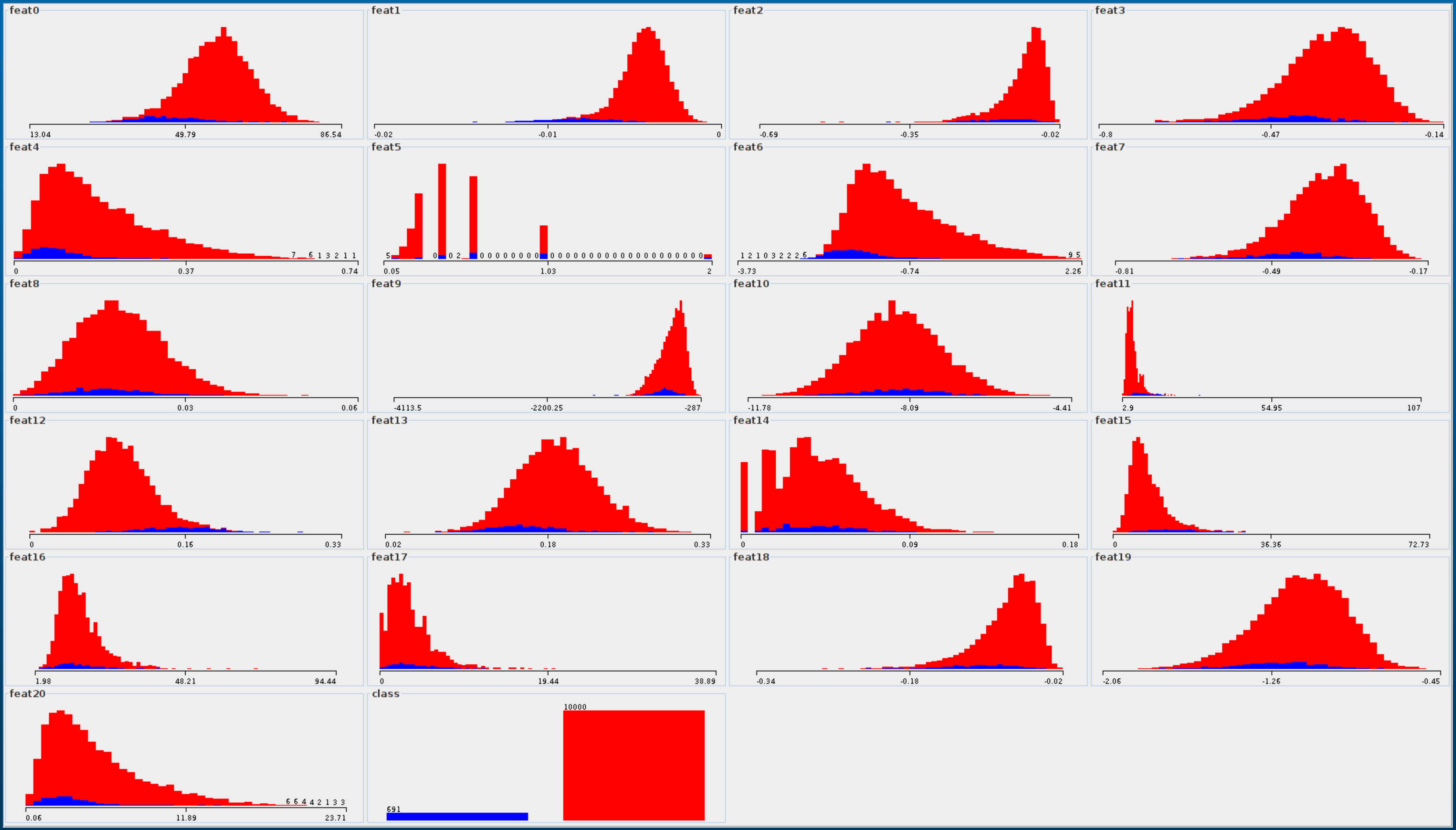
21 Features (*easy set*)



Redundant Features



21 Features *(hard set)*



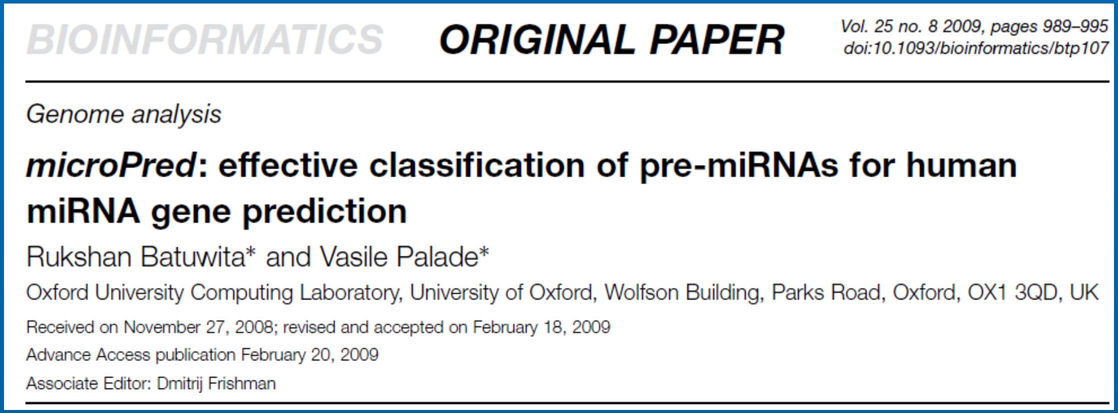
Effect of Class Imbalance

- Batuwida & Palade could achieve either:

	Sn	Sp	G-mean
Approach A	83.36%	99.0%	90.84%
Approach B	90.02%	97.28%	**93.58%**

- However considering class imbalance of 1000 negatives per positive:

	Sn	Sp	G-mean	Precision
Approach A	83.36%	99.0%	90.84%	7.6%
Approach B	90.02%	97.28%	**93.58%**	**3.2%**


BIOINFORMATICS ORIGINAL PAPER

Vol. 25 no. 8 2009, pages 989–995
doi:10.1093/bioinformatics/btp107

Genome analysis

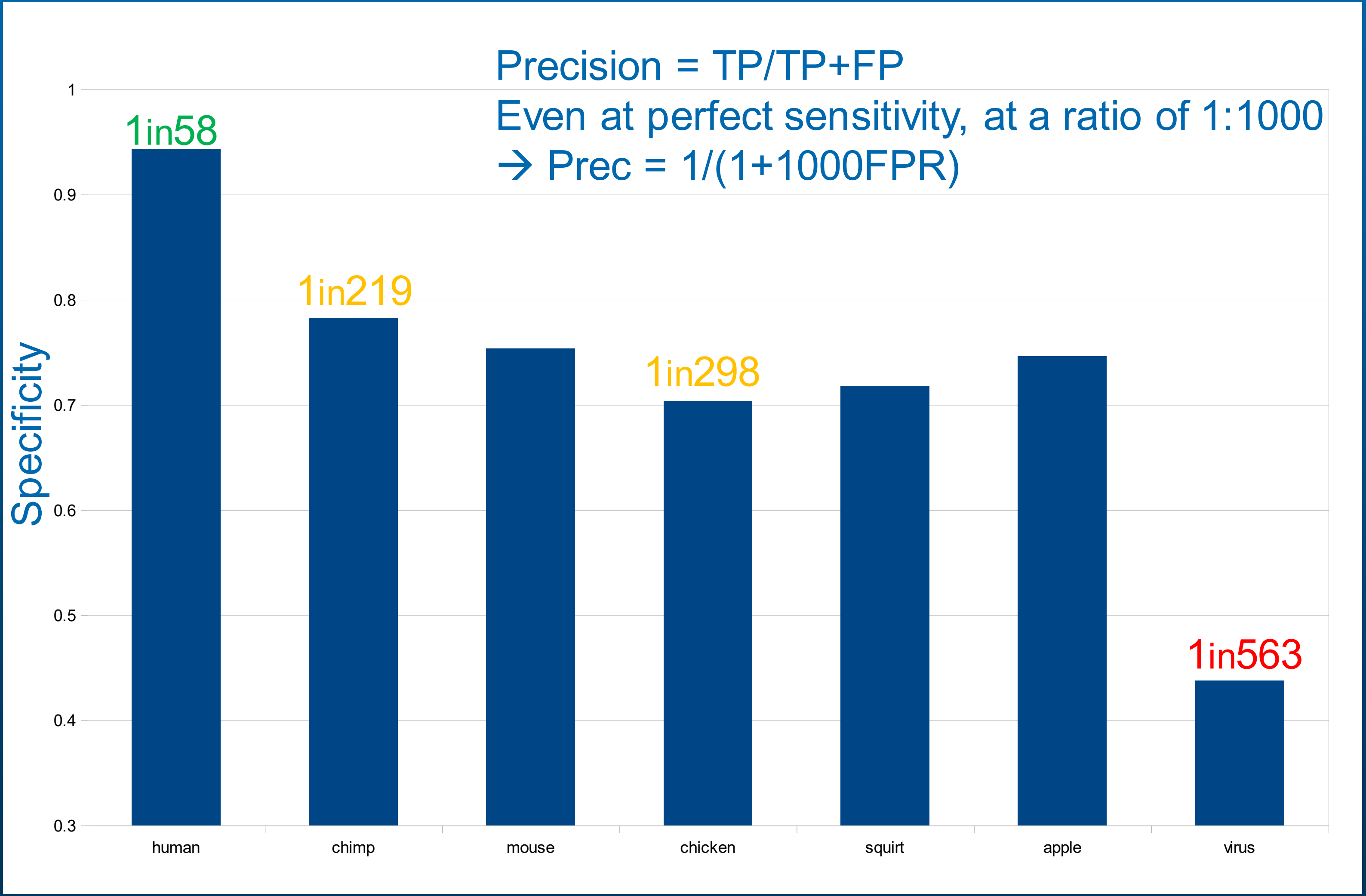
***microPred*: effective classification of pre-miRNAs for human miRNA gene prediction**

Rukshan Batuwita* and Vasile Palade*
Oxford University Computing Laboratory, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

Received on November 27, 2008; revised and accepted on February 18, 2009
Advance Access publication February 20, 2009
Associate Editor: Dmitrij Frishman

- "We validated the *microPred* predictions on the other animal (non-human) and viral pre-miRNAs published in the *miRBase12*, and obtained a high sensitivity. Out of 6095 other animal pre-miRNAs across 49 species, *microPred* identified 5651 correctly with 92.71% of recognition rate. Out of 139 viral pre-miRNAs across 12 species, 131 were predicted correctly with 92.24% of recognition rate."

Specificity for non-human species

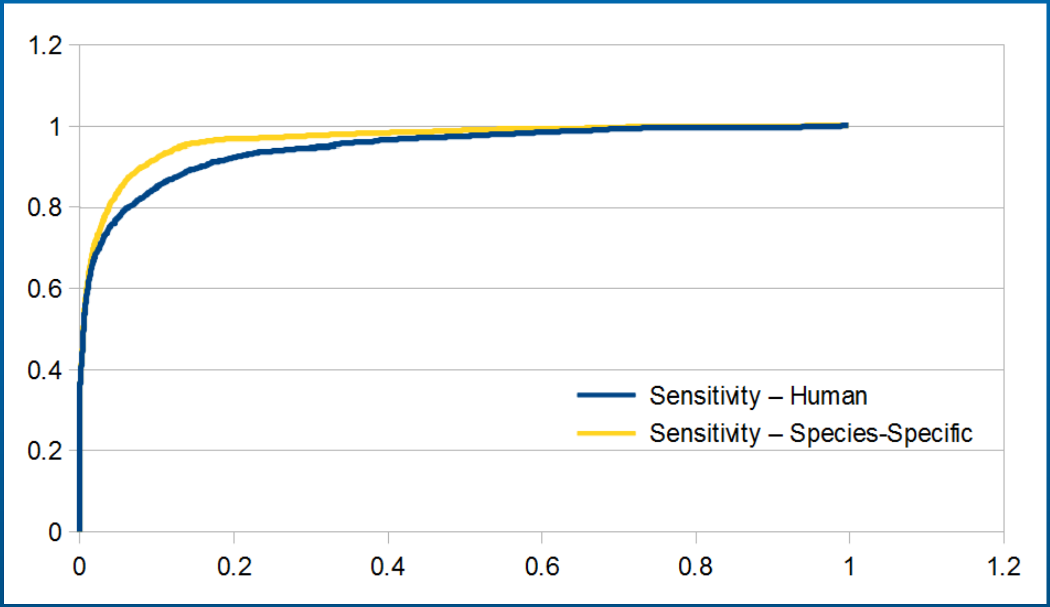


Our miRNA Prediction Approach

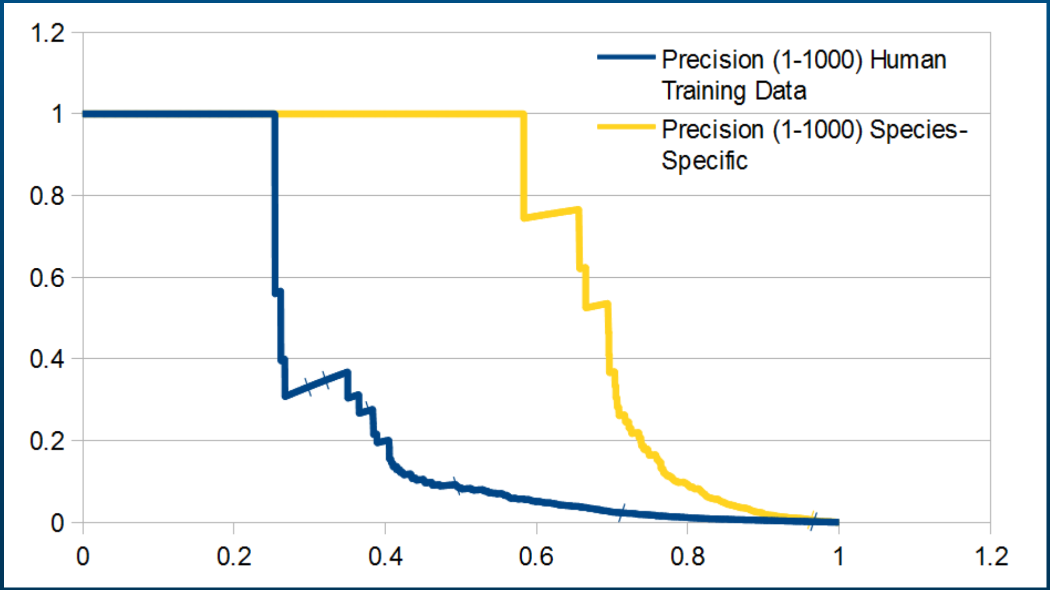
1. Cluster known miRNA from all species
2. Select largest N clusters
3. From each cluster, select representative closest to the target species → +ve training
   - Use SMOTE to generate synthetic minority data
   - Avoid redundant features
4. Get -ve training data from "related" species
   - Hairpin regions of known coding regions
5. Apply *leave-one-species-out* testing
6. Measure performance using precision-recall (1000:1 ratio)

Train: Xenopus Tropicalis
Test: Anolis Carolinensis

ROC



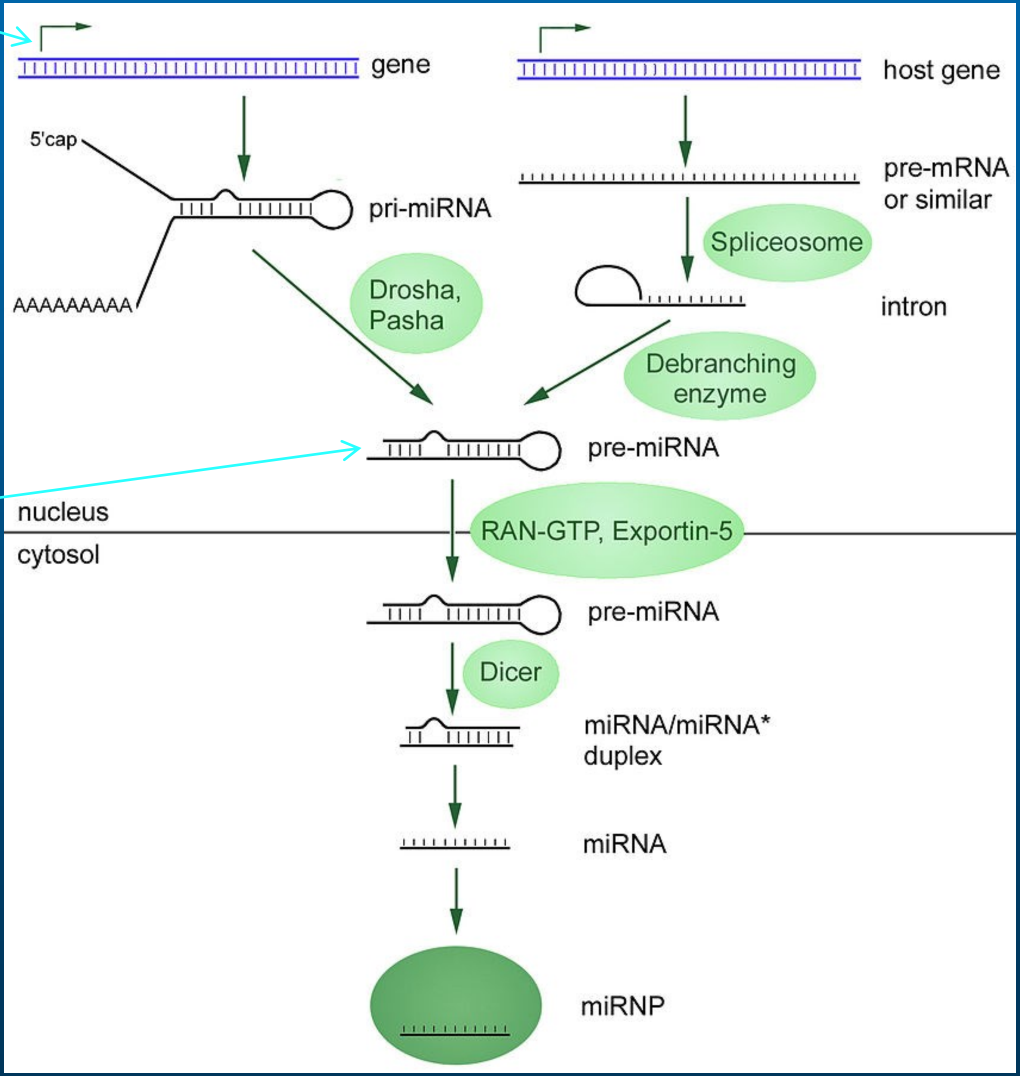
Prec-Recall



Scanning Mode

Test Data

Training Data



Wikipedia Commons

Acknowledgments

1. Funding: NSERC, CFI, ORF, MITACS, Carleton U
2. Graduate students
   1. microRNA prediction: Robert Peace
   2. Hydroxylation PTM: Zhen Liu, Festus Iyuke
   3. PIPE: Sylvain Pitre, Catalin Patulea, Andrew Schoenrock, Adam Amos-Binks, Allen Amos-Binks, Chris North, several biologists!
3. Collaborators
   1. microRNA prediction: Kyle Biggar, Ken Storey
   2. Hydroxylation PTM: Bill Willmore
   3. PIPE: Frank Dehne, Ashkan Golshani, Alex Wong, Michel Dumontier, several biologists!

Pattern Classification Challenges in Bioinformatics

James R. Green

Systems and Computer Engineering

Carleton University