Machine Learning for Bioinformatics & Systems Biology

3. Feature selection and extraction

Lodewyk Wessels *The Netherlands Cancer Institute*
Marcel Reinders *Delft University of Technology*
Perry Moerland *Amsterdam UMC, University of Amsterdam*

Some material courtesy of Robert Duin and David Tax

Overview

- Feature extraction:
  - Linear:
    - PCA,
    - Fisher
  - Non-linear
    - MDS
- Feature selection:
  - Criteria
  - search algorithms
    - forward,
    - backward,
    - branch & bound.
- Regularized classifiers:
  - PAM (shrunken centroids)
  - Ridge,
  - LASSO

BioSB

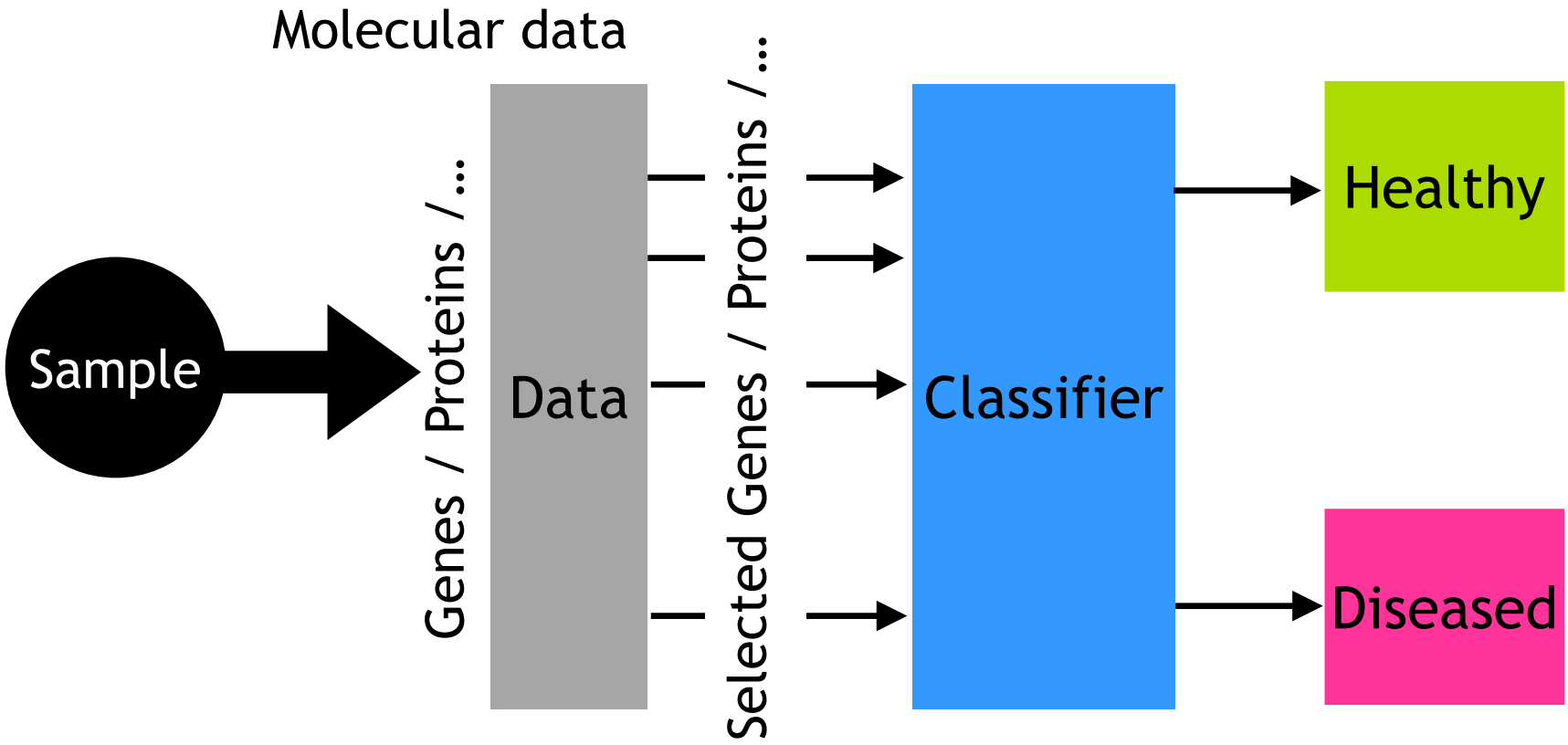
Dimensionality reduction

Use of dimensionality reduction:

1. **Fewer parameters**: faster, easier to estimate – possibly better performance
2. **Explain** which measurements (genes) are useful and which are not (reduce redundancy)
3. **Visualisation**

BioSB

Example: molecular diagnostic classifiers



Molecular data (e.g. RNAseq data)

- Curse of dimensionality:
  - as number of features *increases*, number of parameters increases;
  - for fixed sample size, performance *decreases*.

BioSB

Molecular data (e.g. RNAseq data)

- Curse of dimensionality:
  - as number of features *increases*, number of parameters increases;
  - for fixed sample size, performance *decreases*.
- Traditional assumption in pattern recognition:
  - need 5-10 times as many samples as there are parameters;
  - With regularization we can do with fewer.

BioSB

Molecular data (e.g. RNAseq data)

- Curse of dimensionality:
  - as number of features *increases*, number of parameters increases;
  - for fixed sample size, performance *decreases*.
- Traditional assumption in pattern recognition:
  - need 5-10 times as many samples as there are parameters;
  - With regularization we can do with fewer.
- But genomic data (e.g. RNAseq) is extreme:
  - 100-1000 times *fewer* samples than parameters!

BioSB

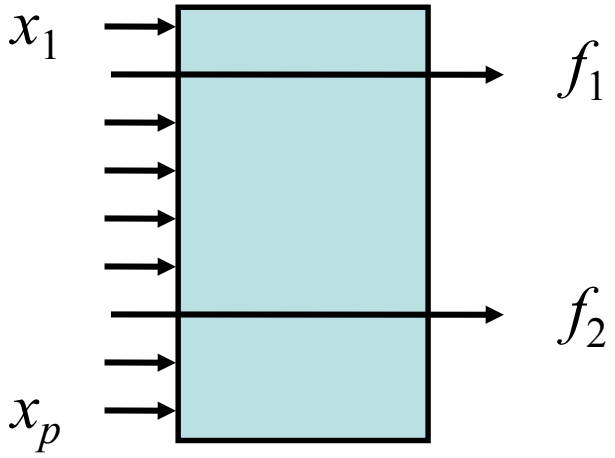
Molecular data (e.g. RNAseq data)

- Curse of dimensionality:
  - as number of features *increases*, number of parameters increases;
  - for fixed sample size, performance *decreases*.
- Traditional assumption in pattern recognition:
  - need 5-10 times as many samples as there are parameters;
  - With regularization we can do with fewer.
- But genomic data (e.g. RNAseq) is extreme:
  - 100-1000 times *fewer* samples than parameters!
- For example: nearest mean classifier on Golub data
  - $p = 3051$, $k = 2 \rightarrow$ number of parameters = 6102
  - Number of samples, $n = 38$

BioSB

Feature selection vs. extraction

- **Feature selection:**
  select d out of p measurements

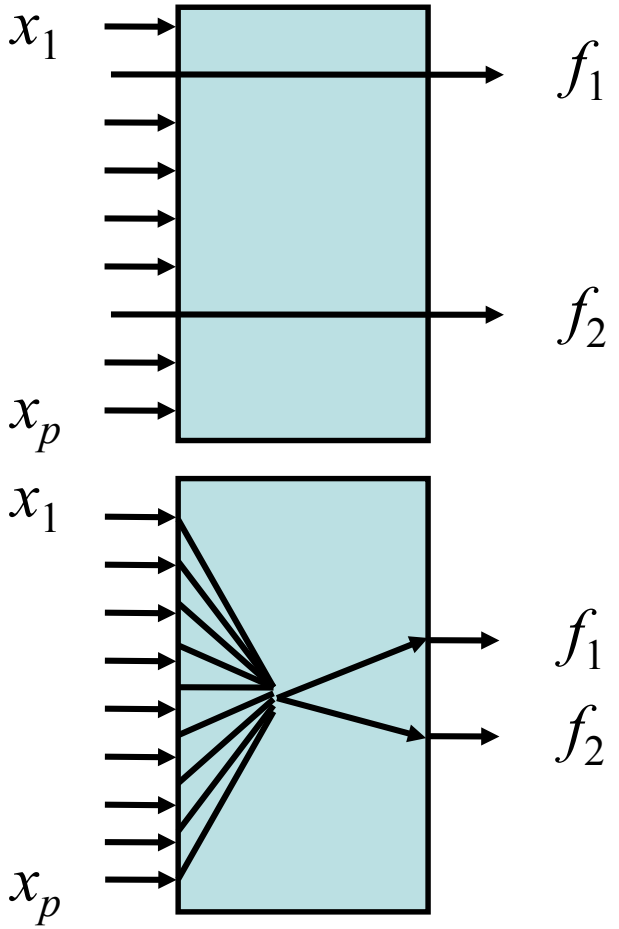


BioSB

Feature selection vs. extraction

- **Feature selection:**
  select d out of p measurements

- **Feature extraction:**
  map p measurements
  to d measurements
  (e.g. PCA, CCA, LDA, MDS)



BioSB

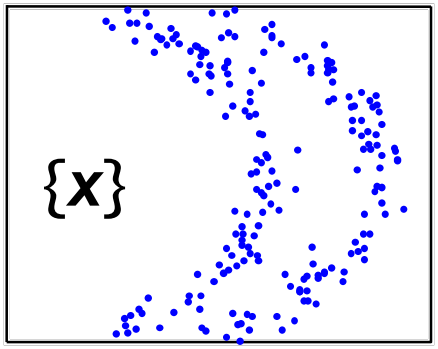
Feature selection v extraction (2)

	Advantage	Disadvantage
Selection	cut in measurements	expensive
	easy interpretation	often approximate
Extraction	cheap	need all measurements
	can be nonlinear	criterion sub-optimal
	not axis aligned	

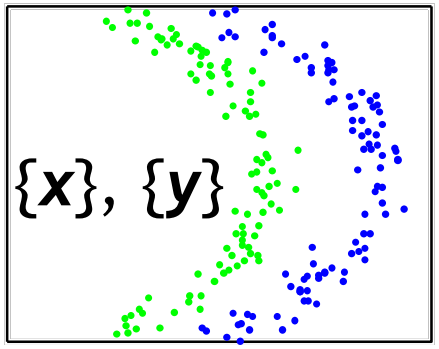
BioSB

Feature extraction (1)

Main types: supervised/unsupervised



Unsupervised

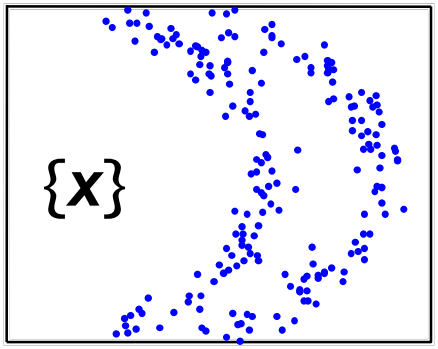


Supervised

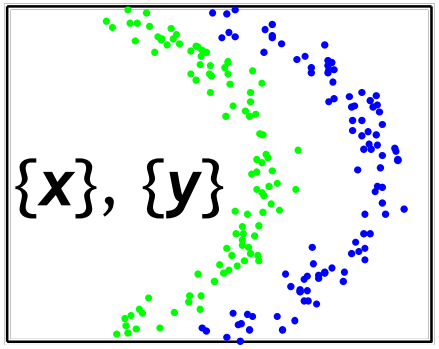
BioSB

Feature extraction (1)

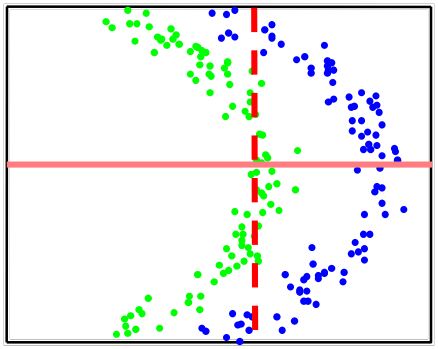
Main types: Linear / Nonlinear



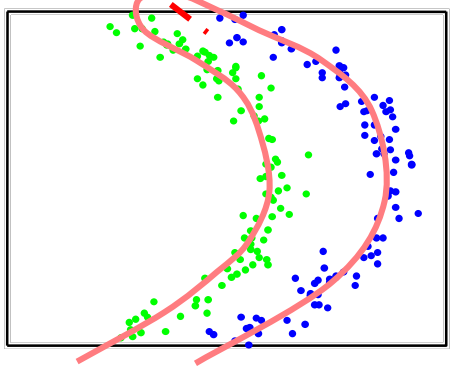
Unsupervised



Supervised



Linear



Nonlinear

BioSB

Feature extraction (2)

- Subjects:
  - Linear, unsupervised:
    - Principal Component Analysis (PCA)
  - Linear, supervised:
    - Linear Discriminant Analysis (LDA)

- PCA is perhaps *the* most important mapping technique

BioSB

Principal component analysis (Unsupervised feature extraction)

- Principal component analysis (PCA, 1901):
  find directions in data...
  - which retain as much *variation* as possible
  -

Principal component analysis (Unsuperrvised feature extraction)

- Principal component analysis (PCA, 1901):
  find directions in data...
  - which retain as much *variation* as possible
  - which make projected data *uncorrelated*
  -

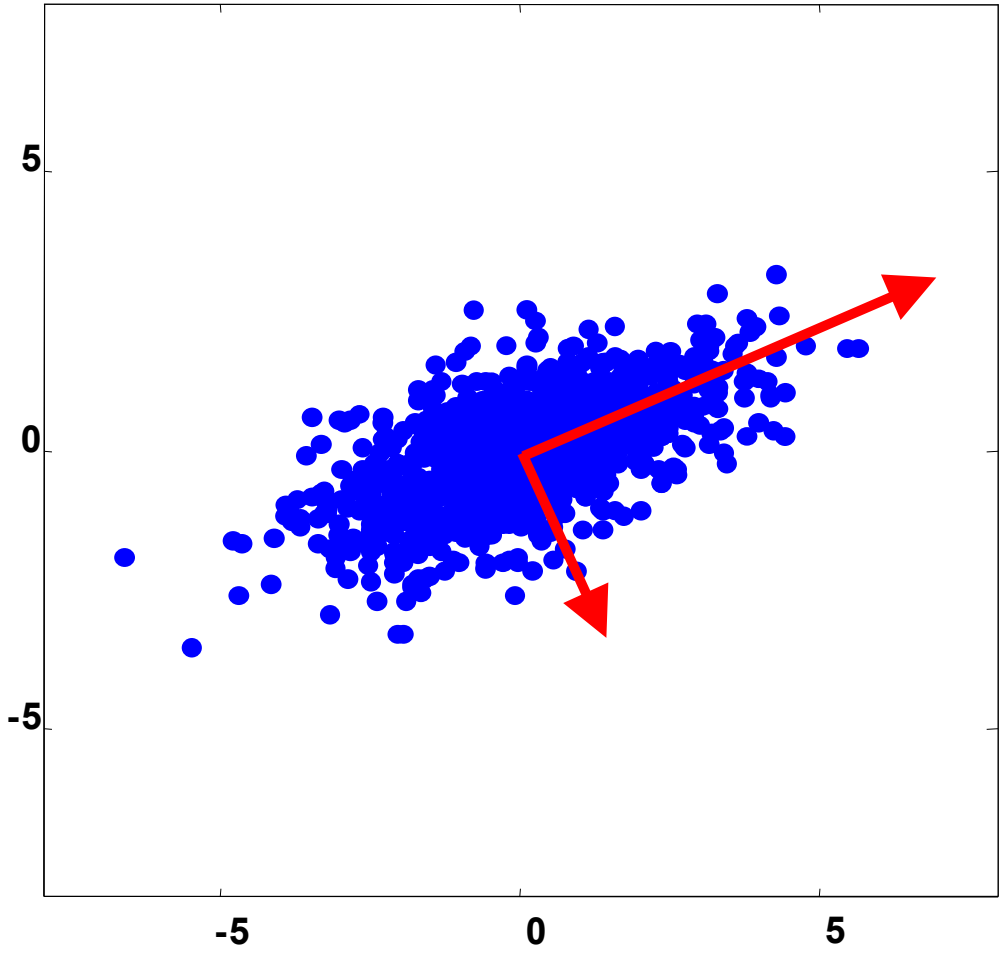
BioSB

Principal component analysis (Unsuperrvised feature extraction)

- Principal component analysis (PCA, 1901):
  find directions in data...
  - which retain as much *variation* as possible
  - which make projected data *uncorrelated*
  - which minimise squared *reconstruction error*

BioSB

Principal component analysis (Unsuperrvised feature extraction)



Steps:

1. Center data
2. Compute covariance, C
3. Perform PCA on C

Output:

1. Eigenvectors: $\boldsymbol{e}_1$, $\boldsymbol{e}_2$
2. Eigenvalues: λ_1, λ_2

Reducing dimensions:

Choosing '*d*'

BioSB

Choosing reduced dimensionality

- To choose d inspect the retained variance,

$$\sum_{i=1}^{d} \lambda_i$$

- or the ratio of retained variance,

$$\sum_{i=1}^{d} \lambda_i \Big/ \sum_{j=1}^{p} \lambda_j$$

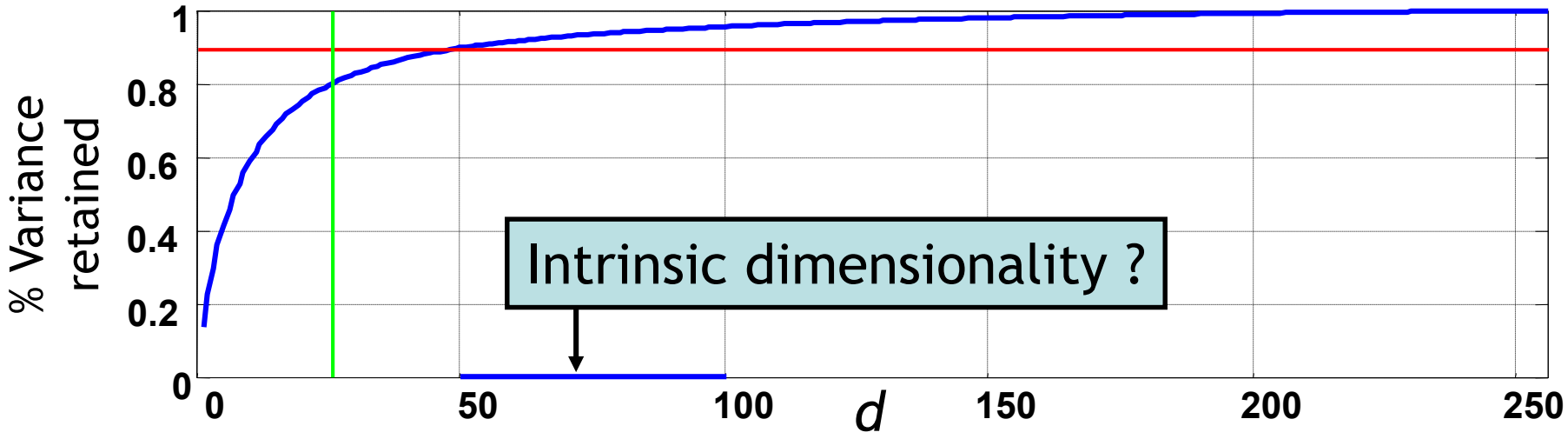
- 'Intrinsic dimensionality':
  d for which 90%-95% variance is retained (rule-of-thumb)

- Reduced dimensionality data set
  - $[\boldsymbol{x}_1^T; \boldsymbol{x}_2^T ; \dots ; \boldsymbol{x}_2^T][\boldsymbol{e}_1, \boldsymbol{e}_2 , \dots , \boldsymbol{e}_d]$

BioSB

PCA example

- *e.g.* NIST digits: 2000 samples, $p = 256$

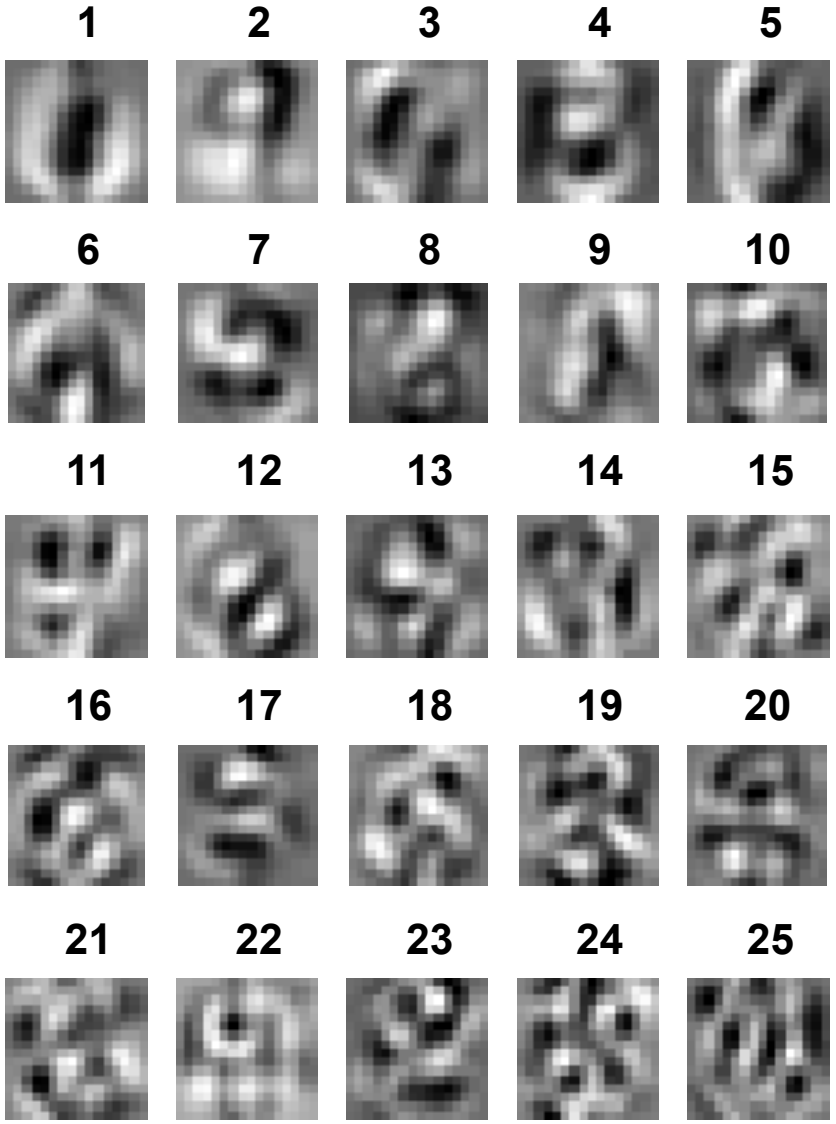


BioSB

PCA example (2)

- For image data, principal components can also be interpreted...

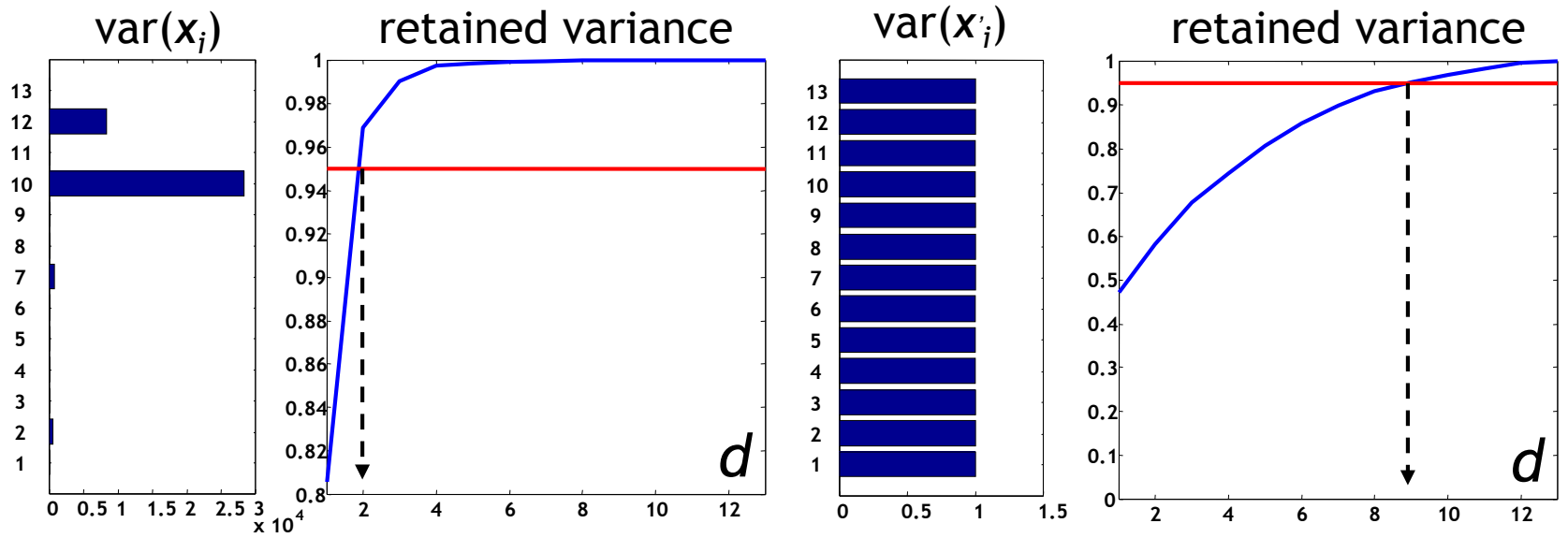
  most often occuring variations between digits



BioSB

PCA tips

- Derivation supposes mean of data is zero, so it should be removed: $x' \leftarrow (x - \mu)$
- PCA is sensitive to scaling (*e.g.* length in cm has a much larger variance than length in m), so it's best to standardise: $x' \leftarrow (x - \mu) / \sigma$

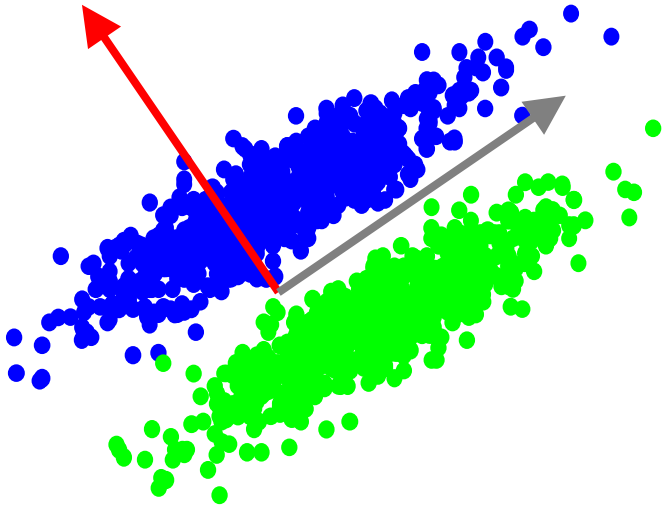


BioSB

PCA conclusions

- PCA:
  - Is **global** and **linear** (but mixtures can be used)
  - Is **unsupervised** (but we can do PCA on each class)
  - Needs a **lot of data** to estimate Σ well.

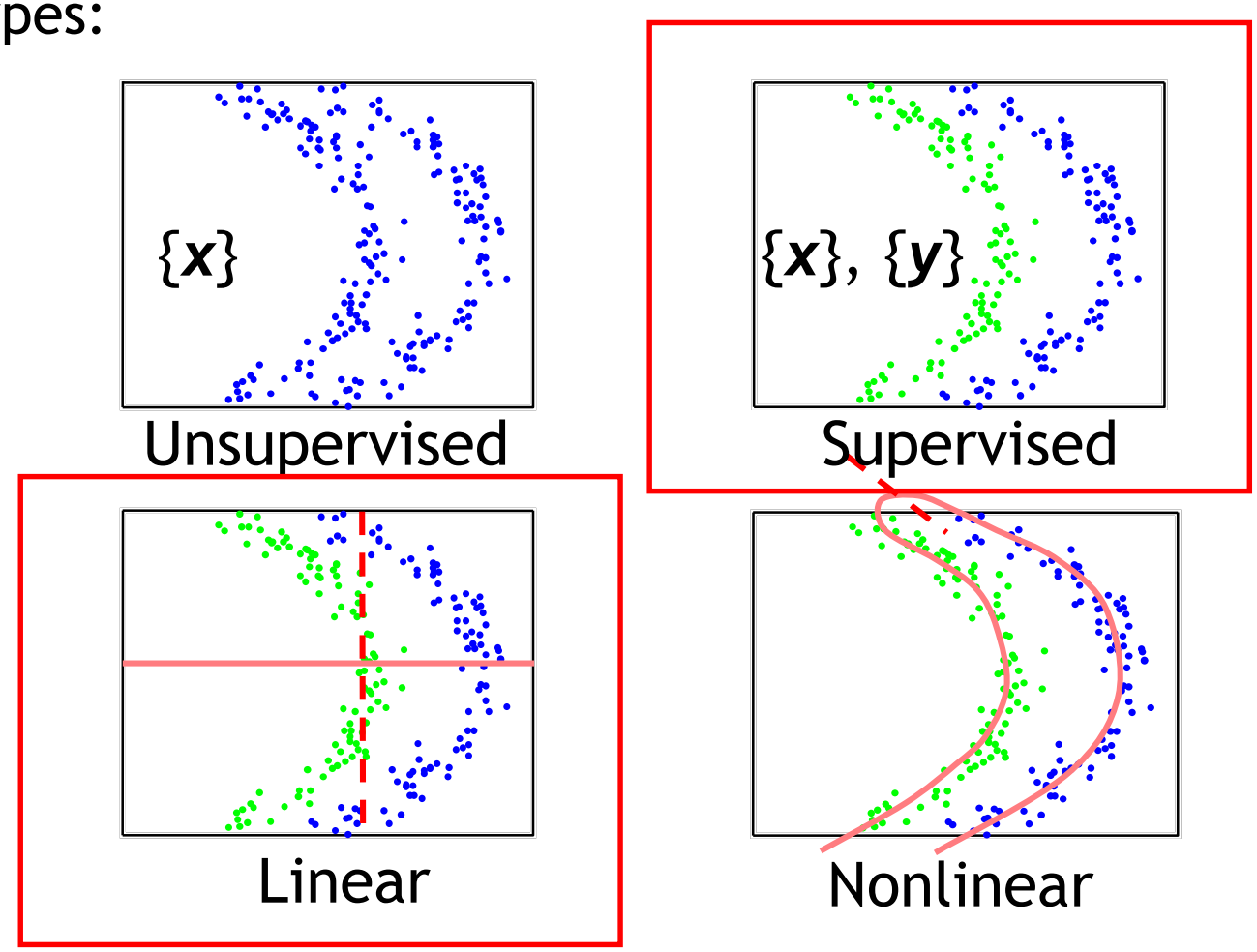
- Danger:
  criterion is not necessarily related to the goal; might discard important directions



BioSB

Feature extraction overview

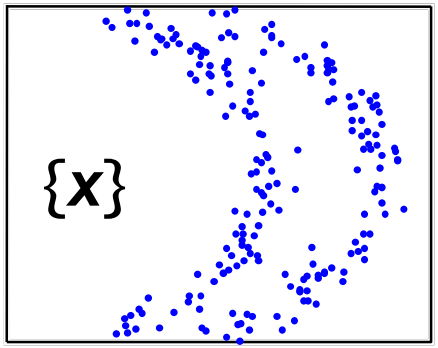
Main types:



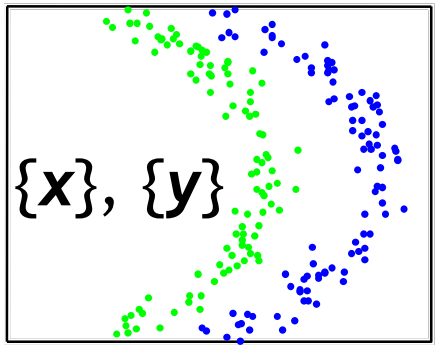
BioSB

Supervised feature extraction

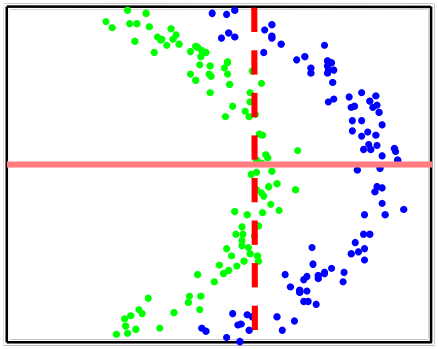
- If class label ω (or y) is given, supervised extraction
- Examples: Fisher mapping; Linear Discriminant Analysis (Day 2)



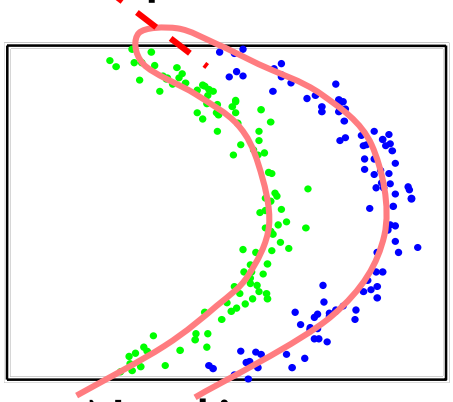
Unsupervised



Supervised



Linear



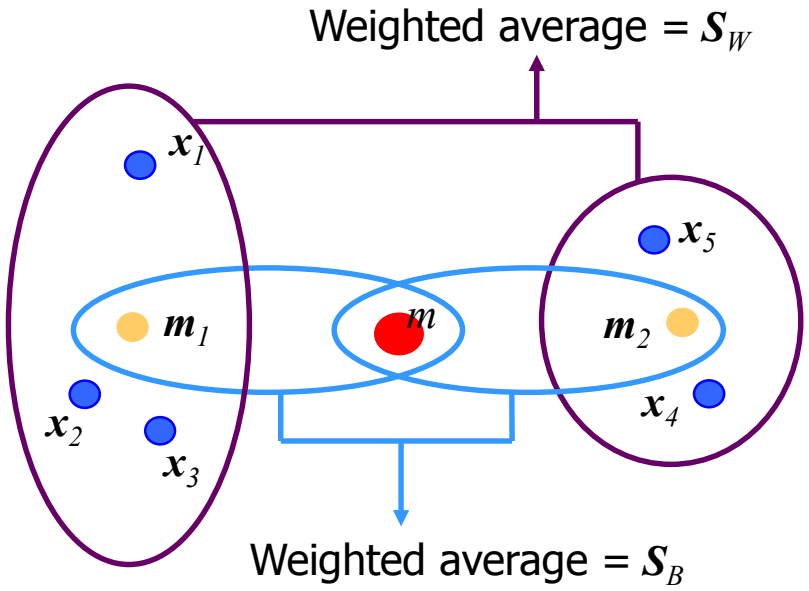
Nonlinear

BioSB

Supervised feature extraction (2)

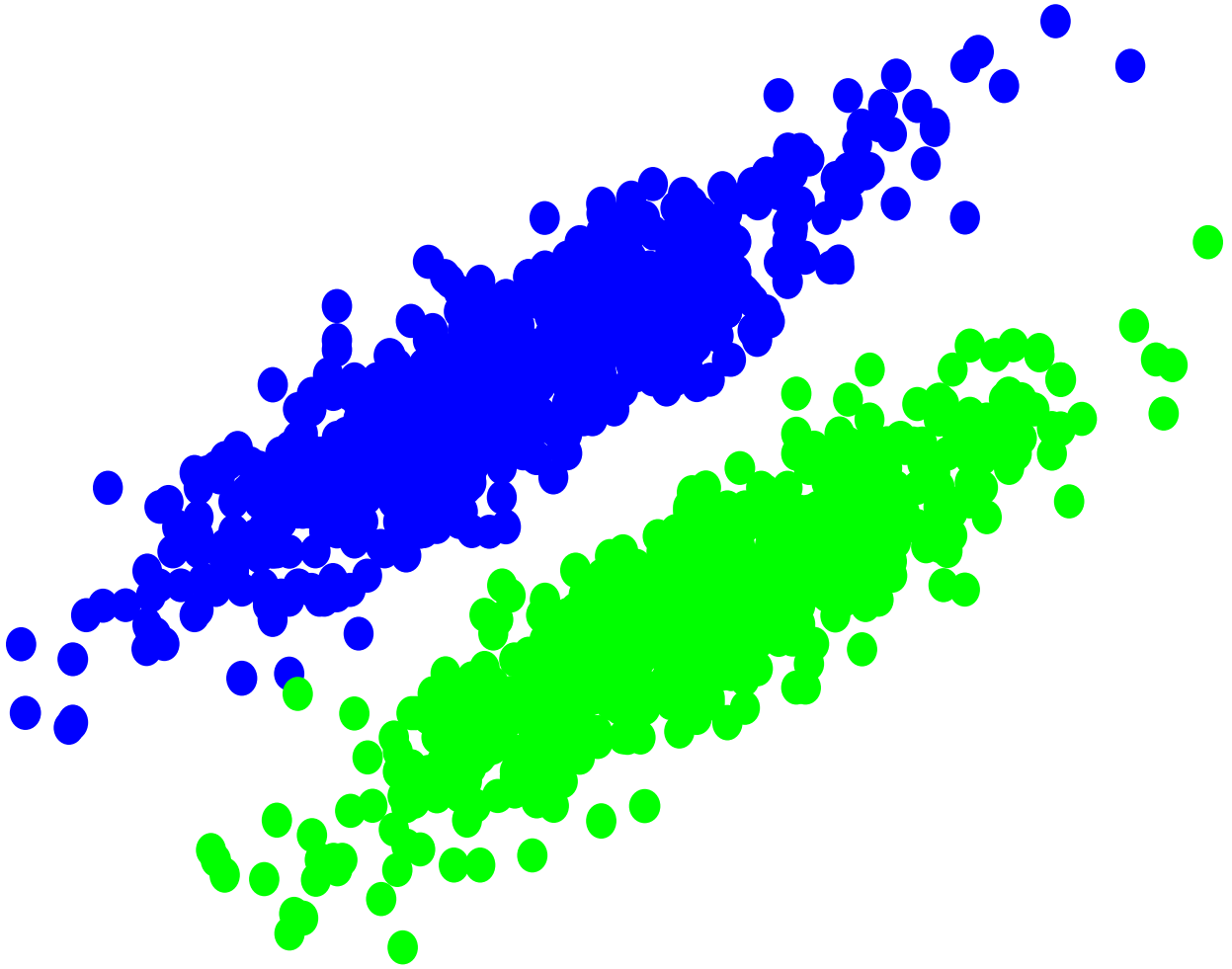
Within-class and between-class scatter matrices:

- With-class: $\boldsymbol{S}_w = \sum_{i=1}^{C} \frac{n_i}{n} \boldsymbol{\Sigma}_i$
- Between-class: $\boldsymbol{S}_B = \sum_{i=1}^{C} \frac{n_i}{n} (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$



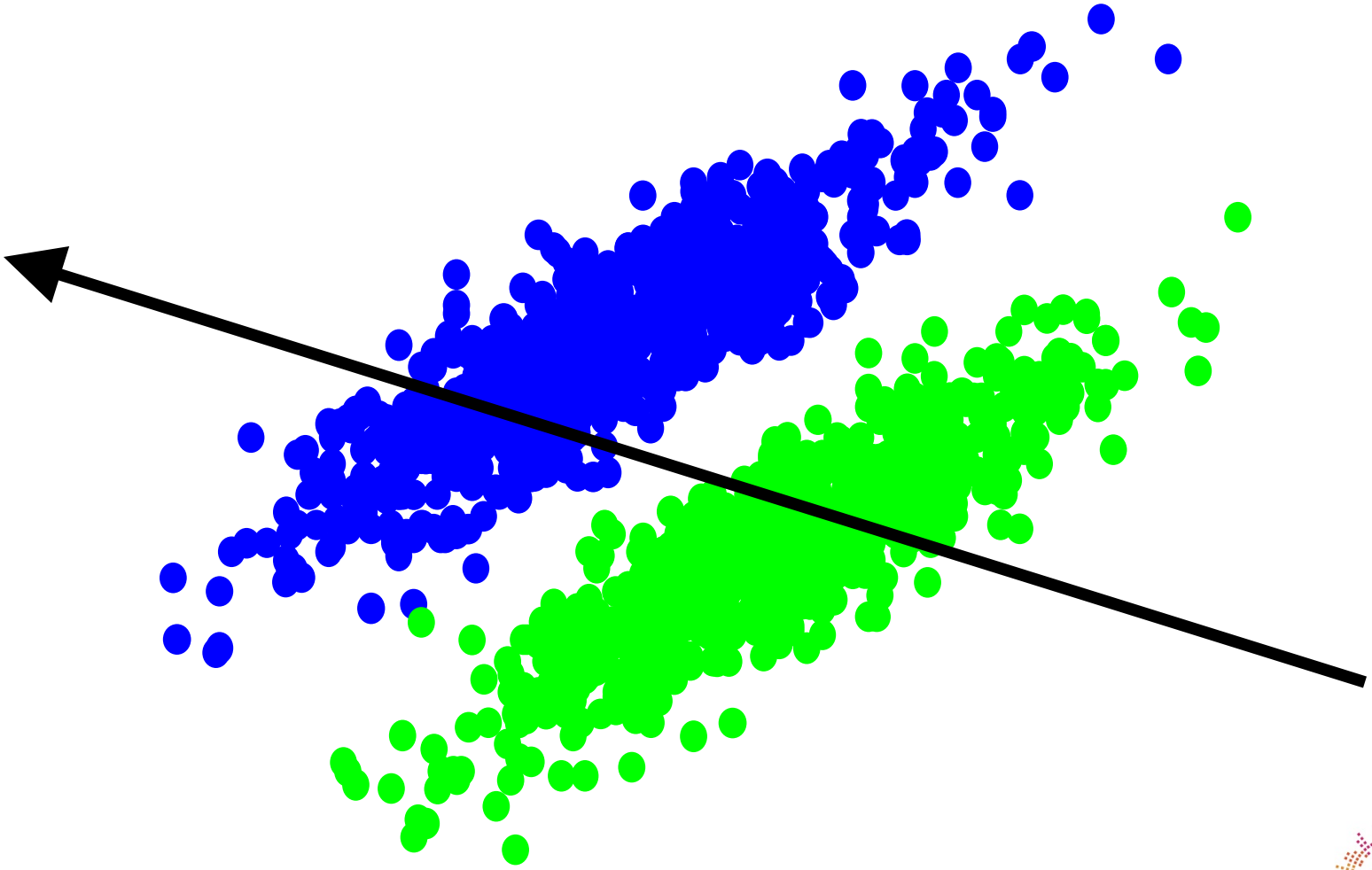
BioSB

Fisher mapping



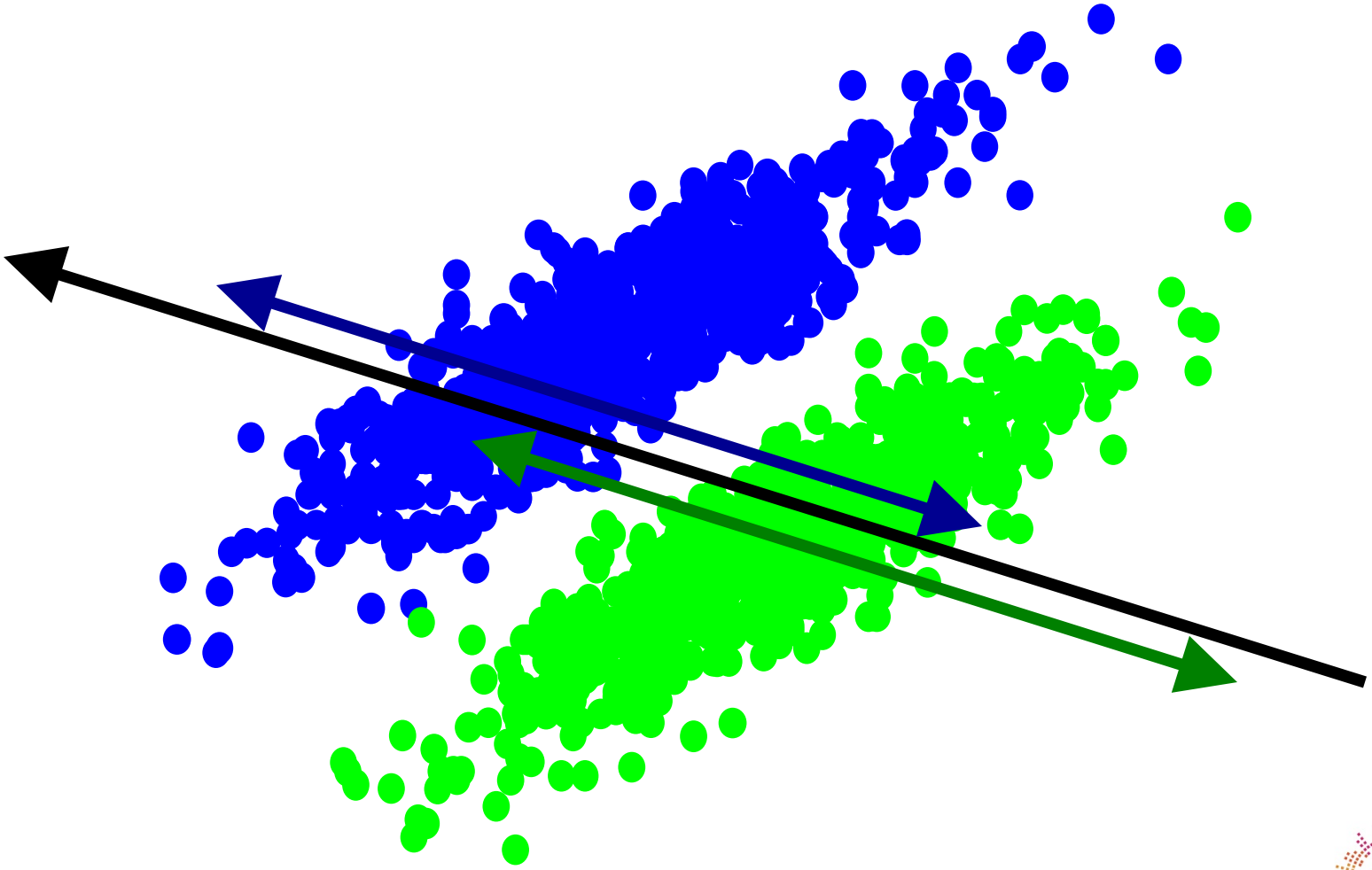
BioSB

Fisher mapping



BioSB

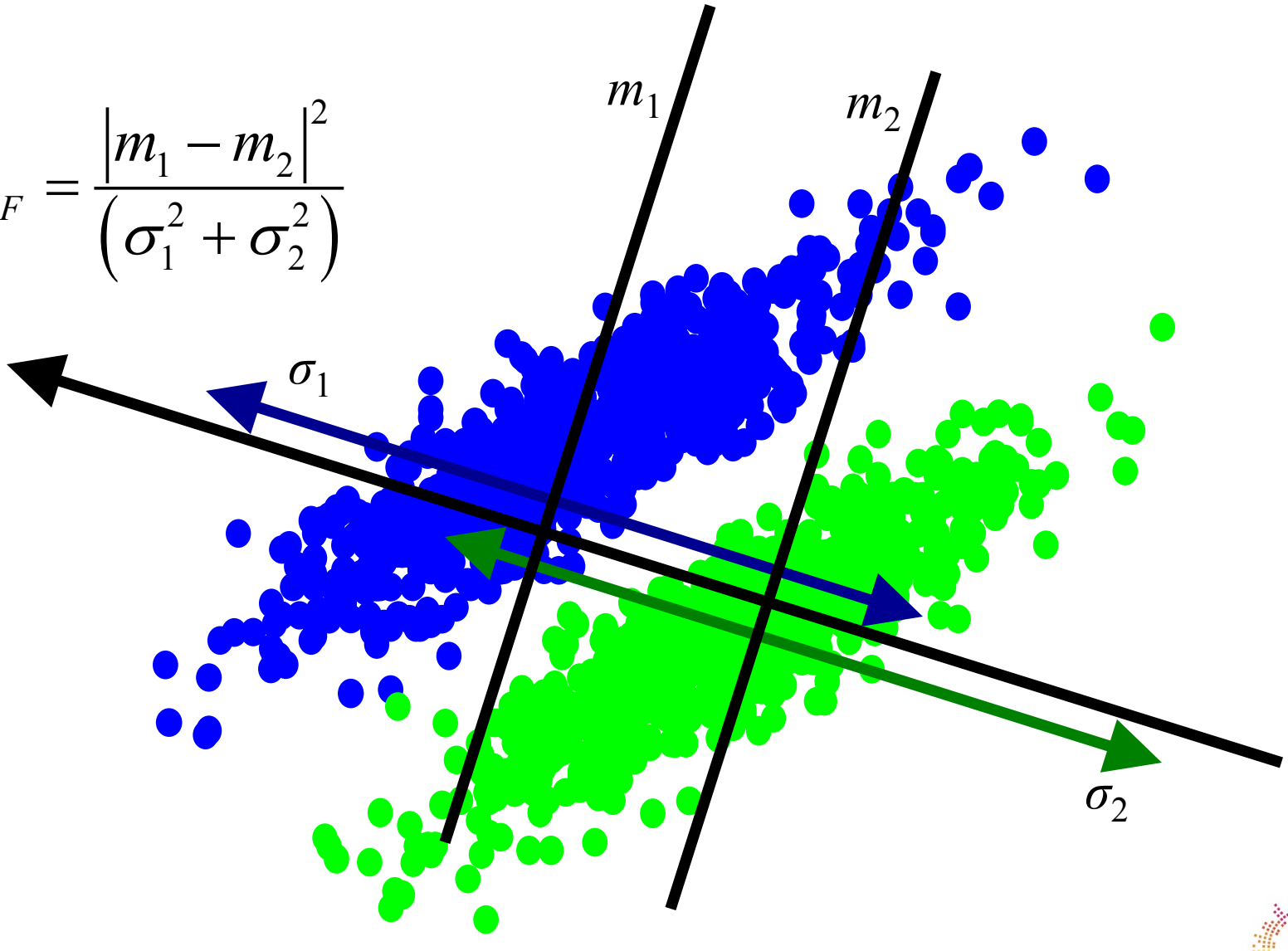
Fisher mapping



BioSB

Fisher mapping

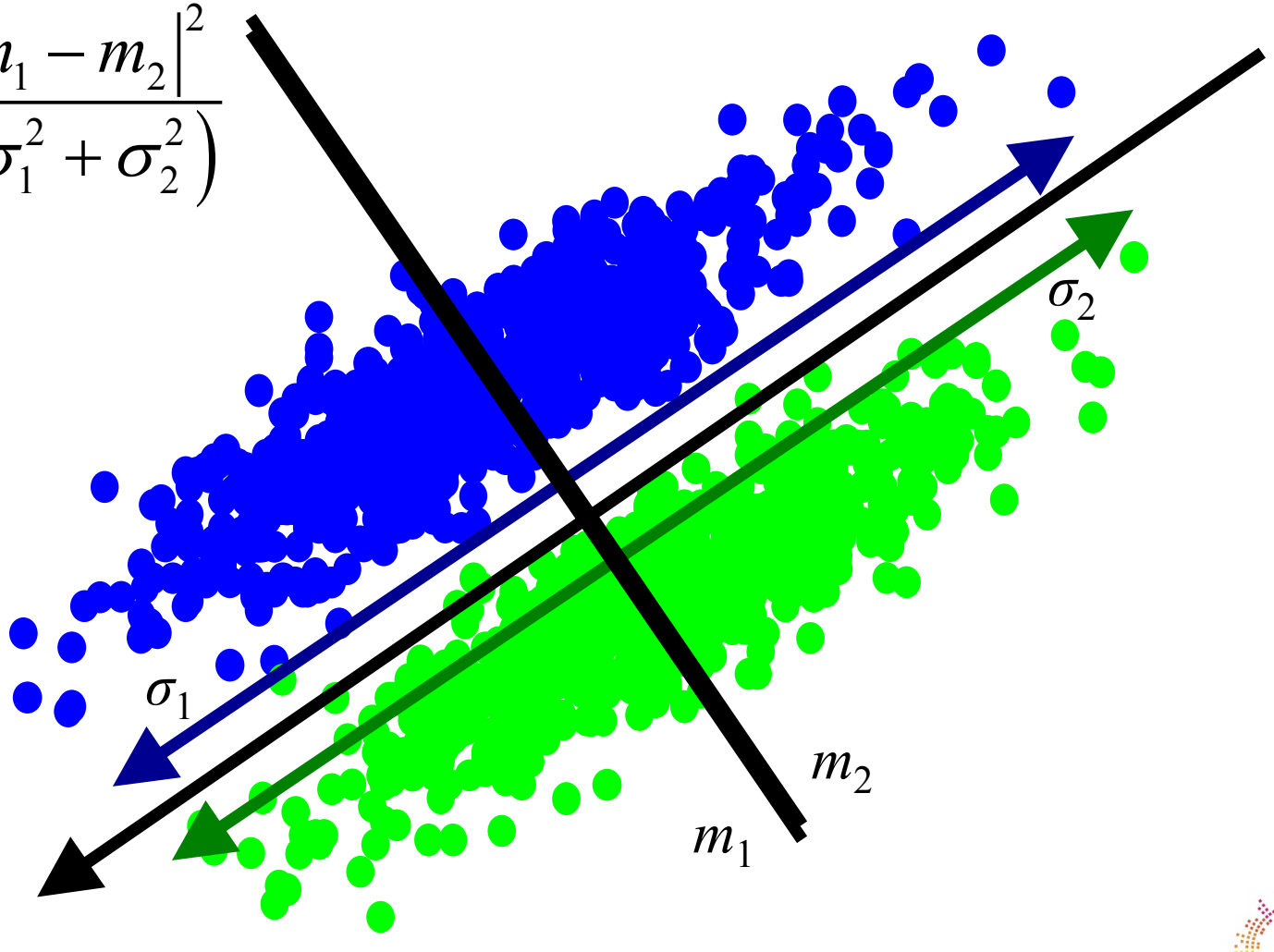
$$J_F = \frac{\left|m_1 - m_2\right|^2}{\left(\sigma_1^2 + \sigma_2^2\right)}$$



BioSB

Fisher mapping

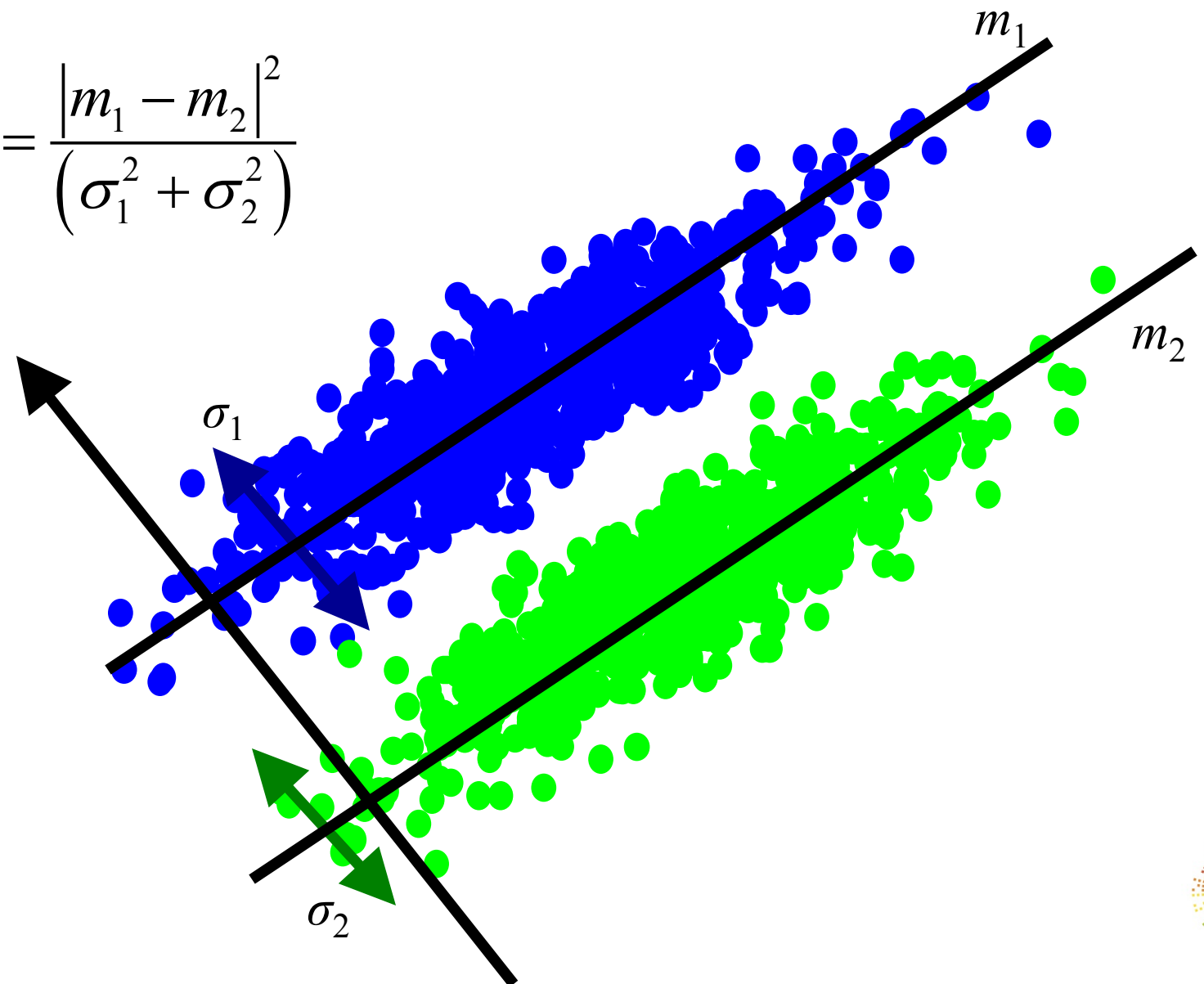
$$J_F = \frac{|m_1 - m_2|^2}{\left(\sigma_1^2 + \sigma_2^2\right)}$$



BioSB

Fisher mapping

$$J_F = \frac{|m_1 - m_2|^2}{\left(\sigma_1^2 + \sigma_2^2\right)}$$



BioSB

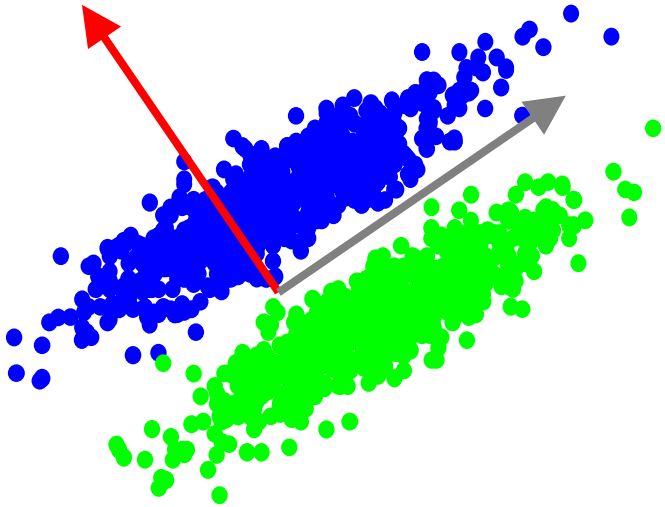
Fisher mapping

- Find basis vector $\boldsymbol{a}_1$ for $\{\boldsymbol{x}\}$ such that in the projections, the classes are maximally separated
- Choose $\boldsymbol{a}_1$ to maximise Fisher criterion:

$$J_F(\boldsymbol{a}_1) = \frac{\boldsymbol{a}_1^T \boldsymbol{S}_B \boldsymbol{a}_1}{\boldsymbol{a}_1^T \boldsymbol{S}_W \boldsymbol{a}_1}$$

- Maximize between class variance

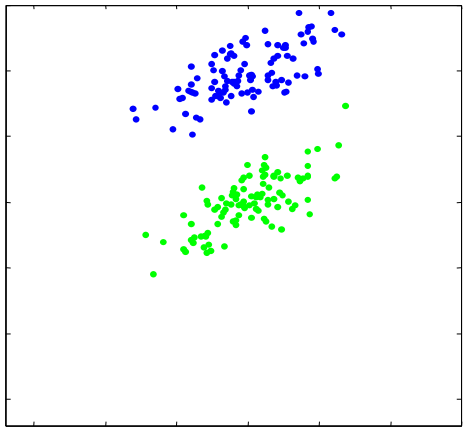
  Minimize within class variance



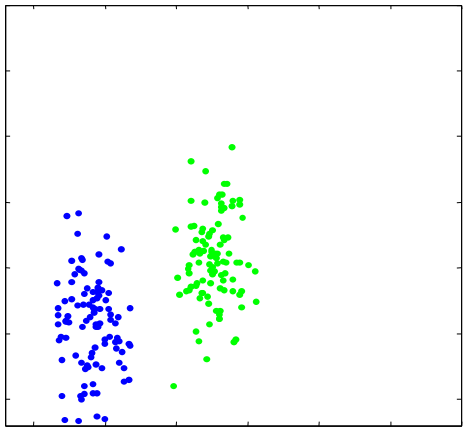
- Solution: eigen-analysis on $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$

BioSB

Fisher mapping (2)



Original



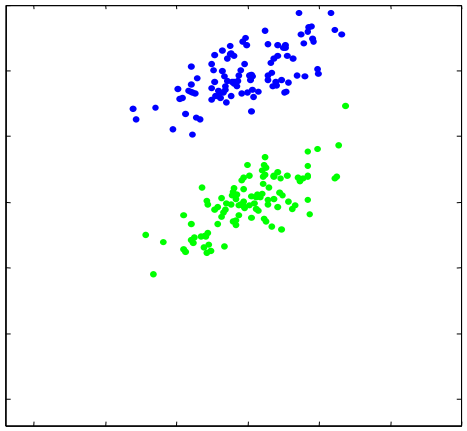
Decorrelated

Decorrelation:

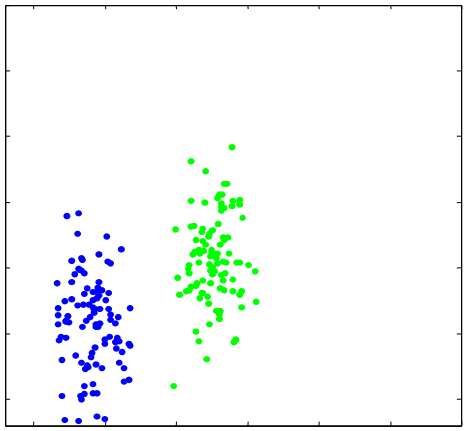
1. we assume that the classes have a similar and normal distribution
2. we ROTATE the dataset such that the class conditional densities are uncorrelated (decorrelated)

BioSB

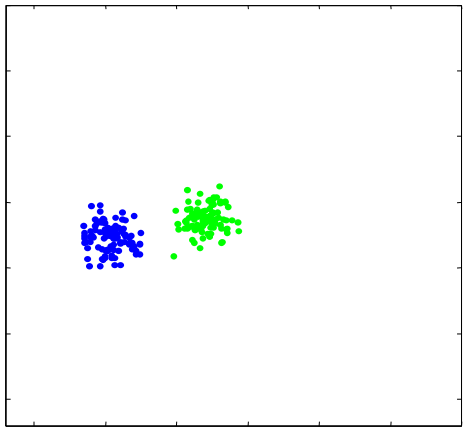
Fisher mapping (2)



Original



Decorrelated



Sphered ($\boldsymbol{S}_W^{-1}$)

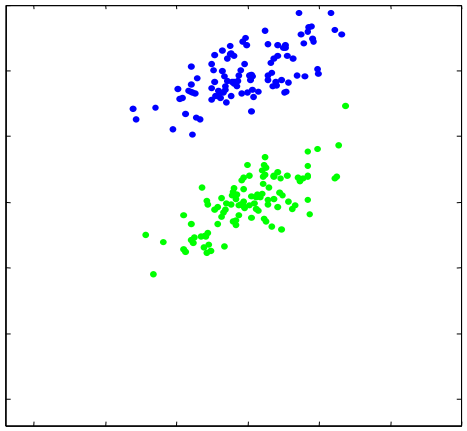
Sphering:

1. we assume that the classes have a similar and normal distribution
2. use the within-class scatter matrix to sphere the class conditional distributions

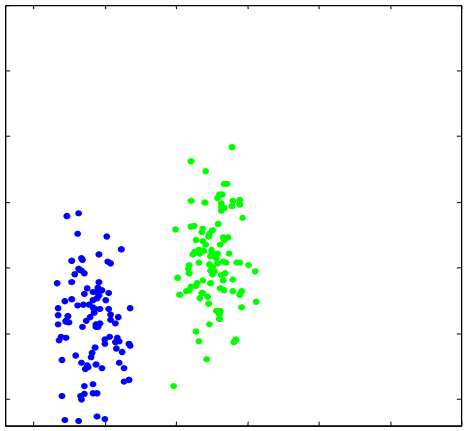
(Class-conditional effects have been removed now)

BioSB

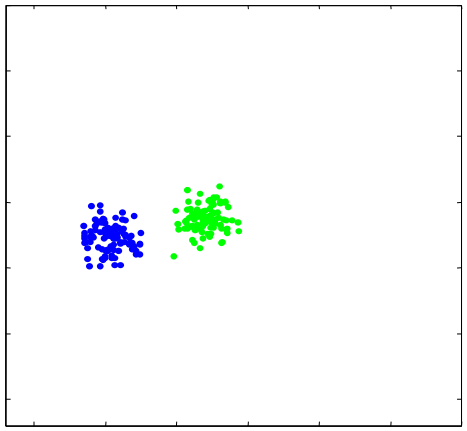
Fisher mapping (2)



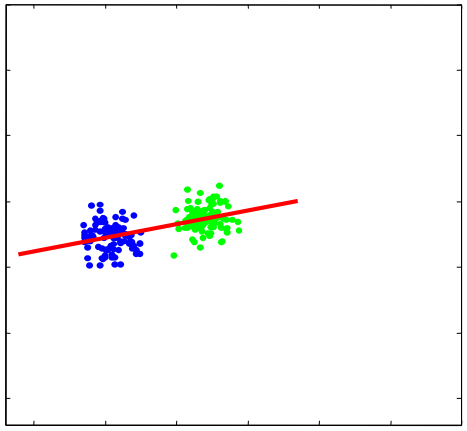
Original



Decorrelated



Sphered (S_W^{-1})



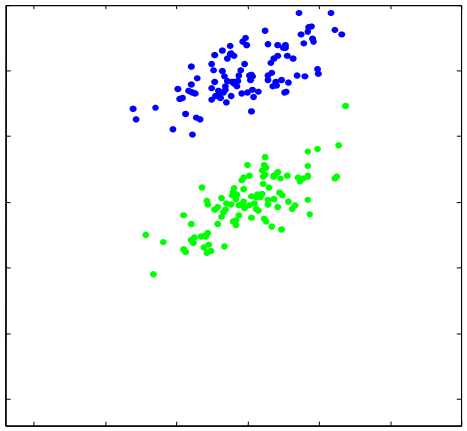
PCA on means (S_B)

PCA on means:

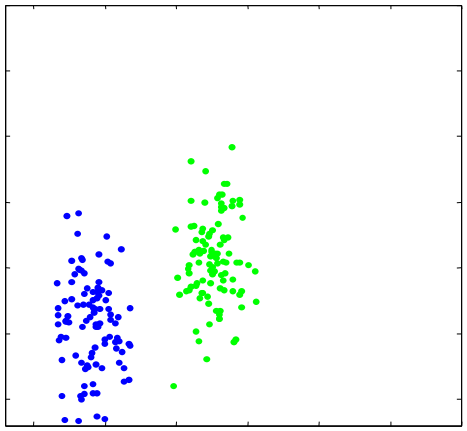
1. Perform PCA on the means (S_B)
2. For c classes, extract c-1 dimensions

BioSB

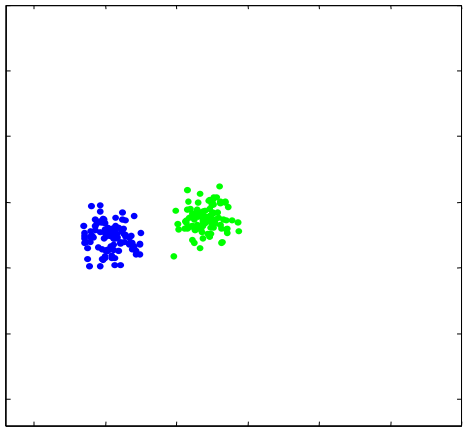
Fisher mapping (2)



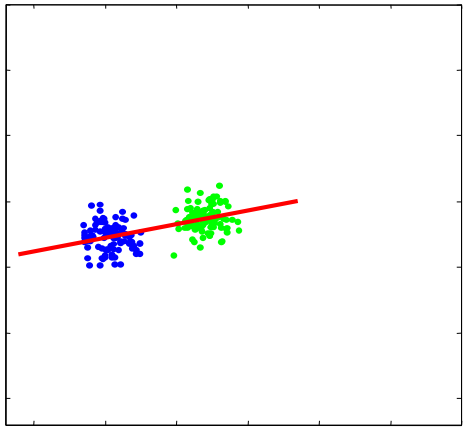
Original



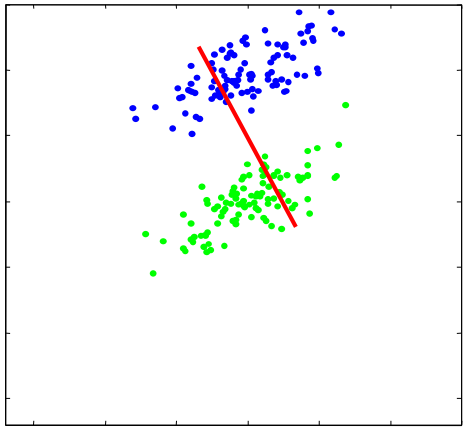
Decorrelated



Sphered (S_W^{-1})



PCA on means (S_B)



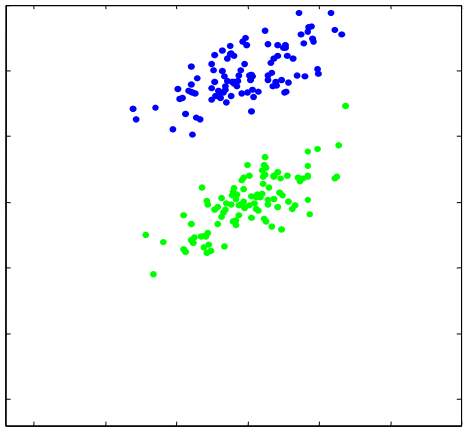
Restored

Restoring:

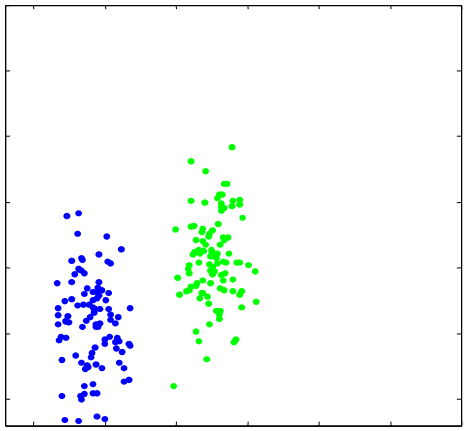
Project this direction back in the original space

BioSB

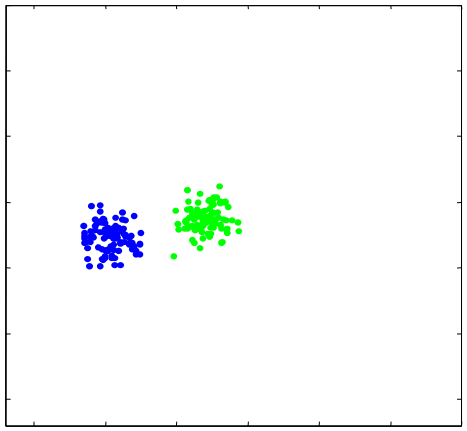
Fisher mapping (2)



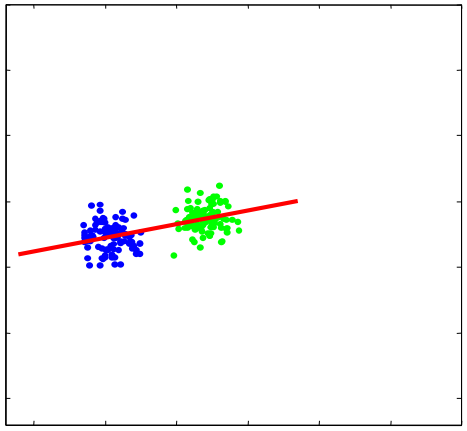
Original



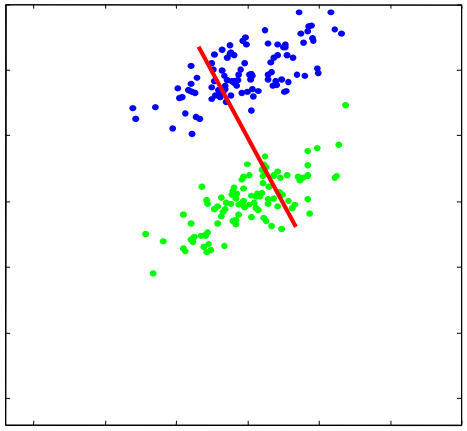
Decorrelated



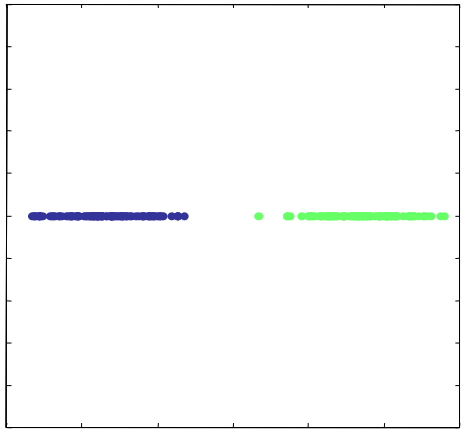
Sphered (S_W^{-1})



PCA on means (S_B)



Restored



Projected data

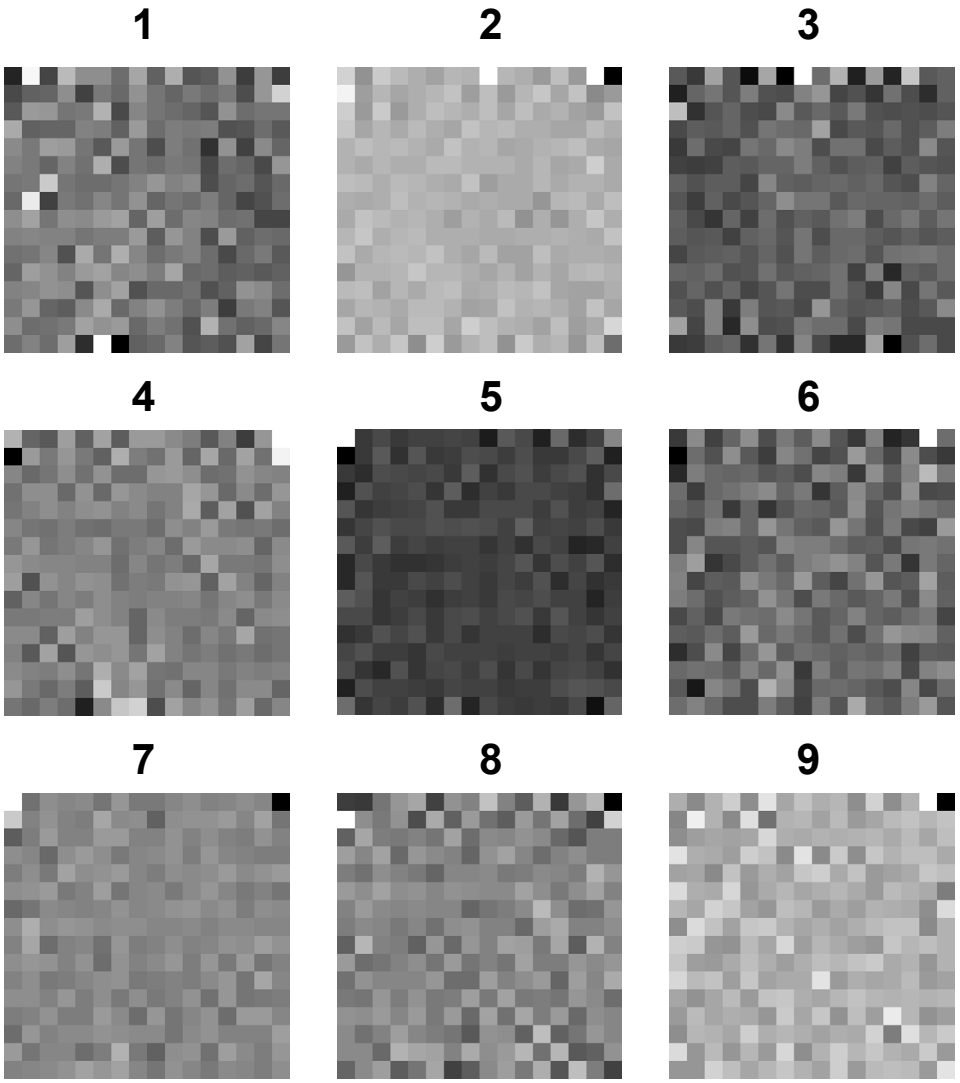
Project:

Project data on projected direction

BioSB

Fisher mapping (3)

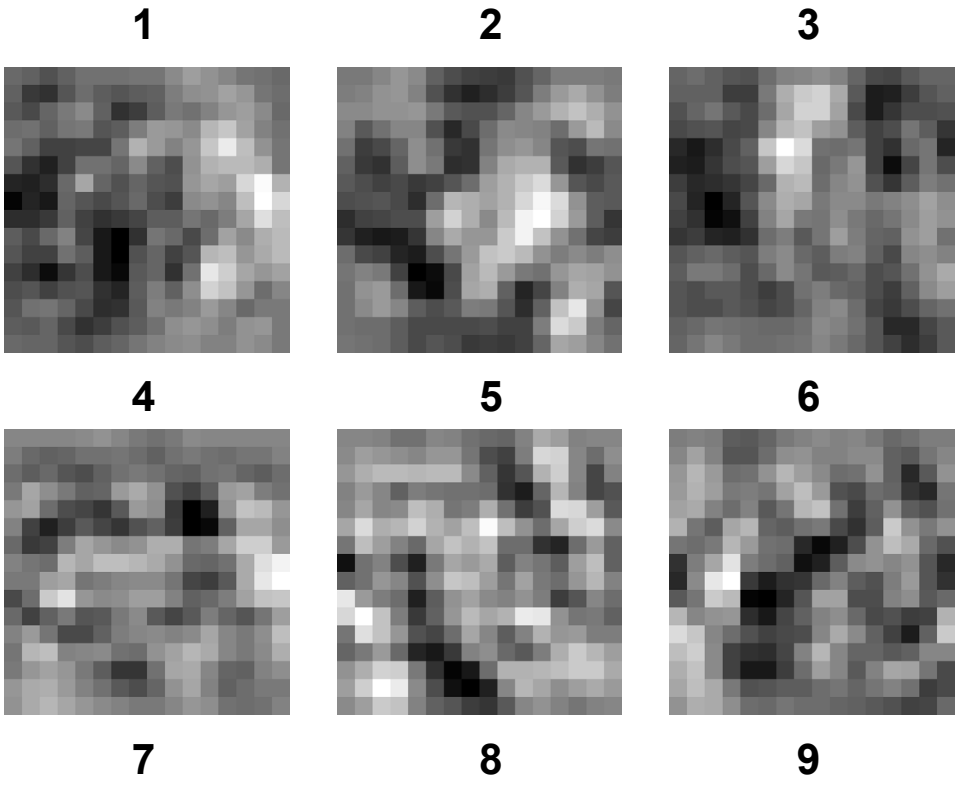
- Map down to a maximum of c - 1 dimensions (why?)
- Example: NIST digits



BioSB

Fisher mapping (4)

- To avoid fitting noise, can do PCA first
- If system underdetermined ($n \leq p$), first doing PCA is required



BioSB

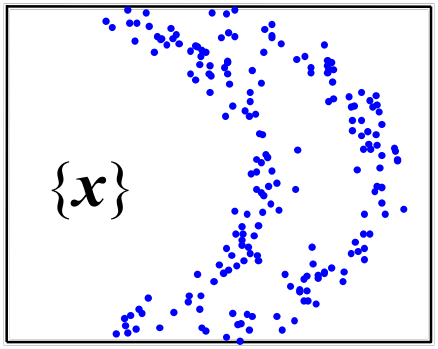
Recapitulation

- Discussed:
  - Linear feature extraction
  - Unsupervised: principal component analysis (PCA), widely used linear feature extraction method
  - Supervised: Fisher mapping

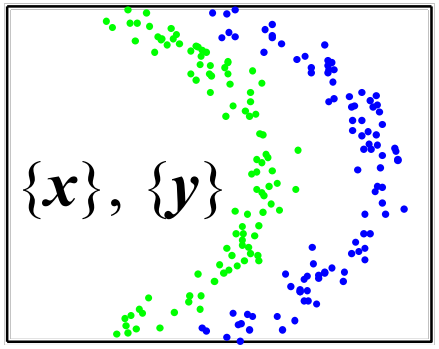
BioSB

Feature extraction

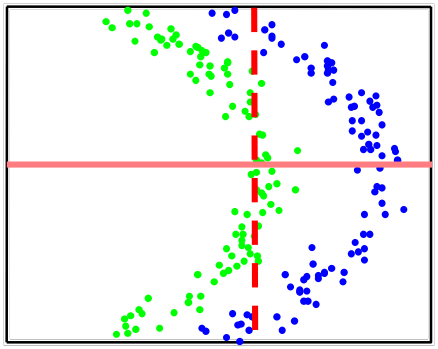
- Main types:



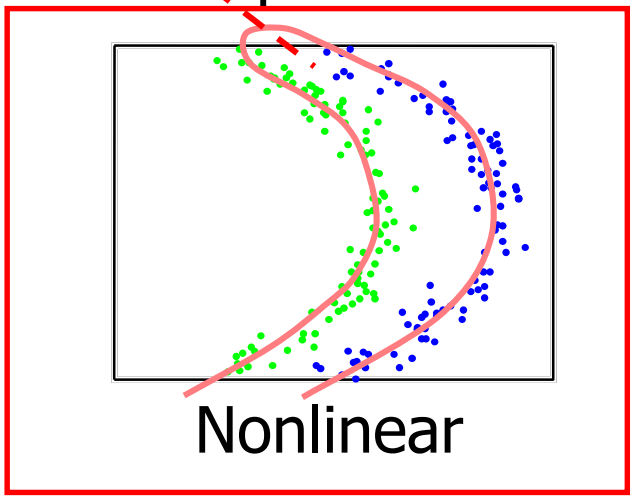
Unsupervised



Supervised



Linear



Nonlinear

BioSB

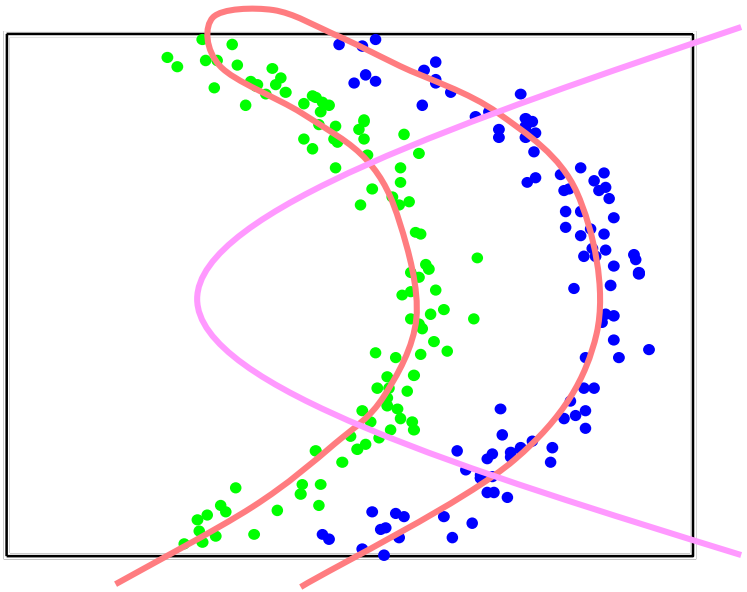
Overview

- Nonlinear feature extraction

- Multidimensional scaling (MDS):
  - Linear: classical scaling
  - Nonlinear:
    - Sammon mapping
    - t-SNE

Nonlinear feature extraction

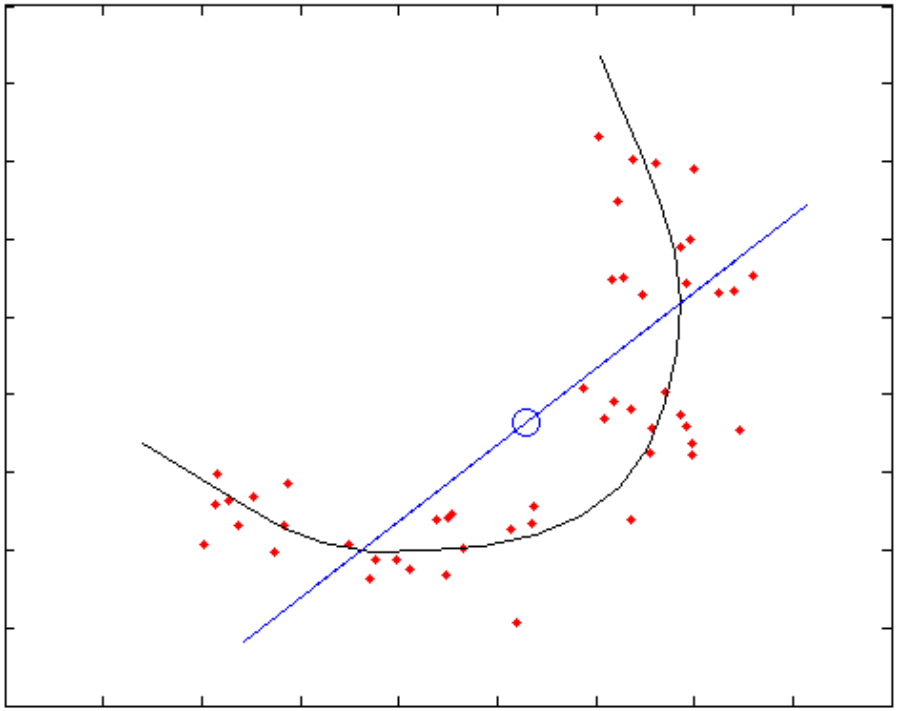
- Nonlinear mappings:
  a large collection of possible mappings,
  but not all applicable to all problems
- Usually needs an *optimisation algorithm*



BioSB

Nonlinear feature extraction (2)

- Example:
  principal curves



- Principal component: minimise deviation from line
- Principal curve: minimise deviation from a curve
  (from a given family of curves)

BioSB

Nonlinear feature extraction (3)

Example: embedding

- Find new representation such that distances between samples are preserved as well as possible

BioSB

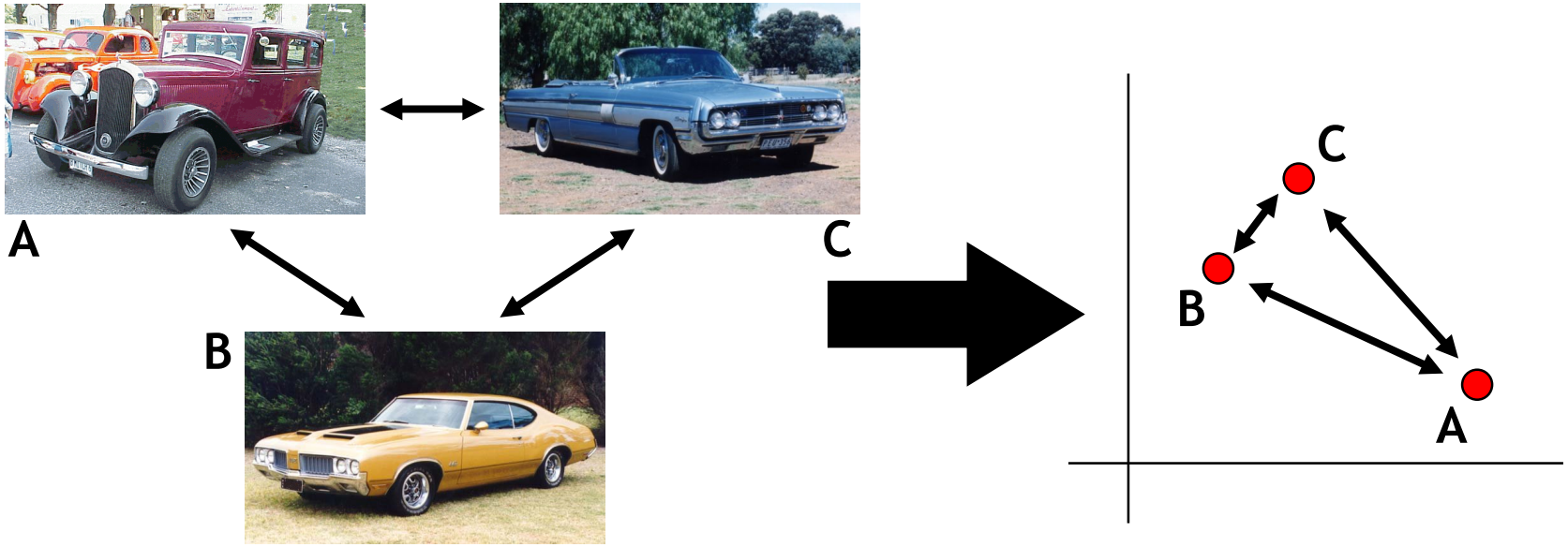
Multidimensional scaling (MDS)

- Criterion: preserve all inter-sample distances
- Needed: *n* x *n* distance matrix between all points
- Map samples to a new (lower dimensional) space

BioSB

Multidimensional scaling (MDS)

- Criterion: preserve all inter-sample distances
- Needed: *n* x *n* distance matrix between all points
- Map samples to a new (lower dimensional) space



BioSB

MDS (2)

- Advantages of using distances:
  - do not necessarily need original data

MDS (2)

- Advantages of using distances:
  - do not necessarily need original data
  - allows for inclusion of **knowledge on objects** (e.g. characteristics of amino acids when comparing proteins)

MDS (2)

- Advantages of using distances:
  - do not necessarily need original data
  - allows for inclusion of **knowledge on objects** (e.g. characteristics of amino acids when comparing proteins)
  - allows for inclusion of **knowledge of relations** (e.g. invariances) in distance measure (e.g. Pearson correlation being shift and scale independent when comparing expression profiles)

MDS (2)

- Advantages of using distances:
  - do not necessarily need original data
  - allows for inclusion of **knowledge on objects** (e.g. characteristics of amino acids when comparing proteins)
  - allows for inclusion of **knowledge of relations** (e.g. invariances) in distance measure (e.g. Pearson correlation being shift and scale independent when comparing expression profiles)
  - easy to introduce nonlinearity

MDS (2)

- Advantages of using distances:
  - do not necessarily need original data
  - allows for inclusion of **knowledge on objects** (e.g. characteristics of amino acids when comparing proteins)
  - allows for inclusion of **knowledge of relations** (e.g. invariances) in distance measure (e.g. Pearson correlation being shift and scale independent when comparing expression profiles)
  - easy to introduce nonlinearity

- Algorithms should find:
  - new, low-dimensional coordinates for each object
  - the number of dimensions to embed the data in

MDS: Non-linear mappings (5)

- d_{ij} : distance $\| \boldsymbol{x}_i - \boldsymbol{x}_j \|$ in original space (? - dimensional)
- δ_{ij} : distance $\| \boldsymbol{y}_i - \boldsymbol{y}_j \|$ in new space (d - dimensional)

MDS: Non-linear mappings (5)

- d_{ij} : distance $\| \boldsymbol{x}_i - \boldsymbol{x}_j \|$ in original space (? - dimensional)
- δ_{ij} : distance $\| \boldsymbol{y}_i - \boldsymbol{y}_j \|$ in new space (d - dimensional)

$$Stress(\boldsymbol{y}) = \frac{1}{\sum_i \sum_{j>i} d_{ij}^{(q+2)}} \sum_i \sum_{j>i} d_{ij}^{q} (\delta_{ij} - d_{ij})^2$$

- weight factor $q = \ldots, -2, -1, 0, 1, 2, \ldots$

  $q < 0$: emphasise short distances

  $q > 0$: emphasise large distances

  Sammon mapping: $q = -1$

BioSB

MDS: Non-linear mappings (5)

- d_{ij} : distance $\| \boldsymbol{x}_i - \boldsymbol{x}_j \|$ in original space (? - dimensional)
- δ_{ij} : distance $\| \boldsymbol{y}_i - \boldsymbol{y}_j \|$ in new space (d - dimensional)

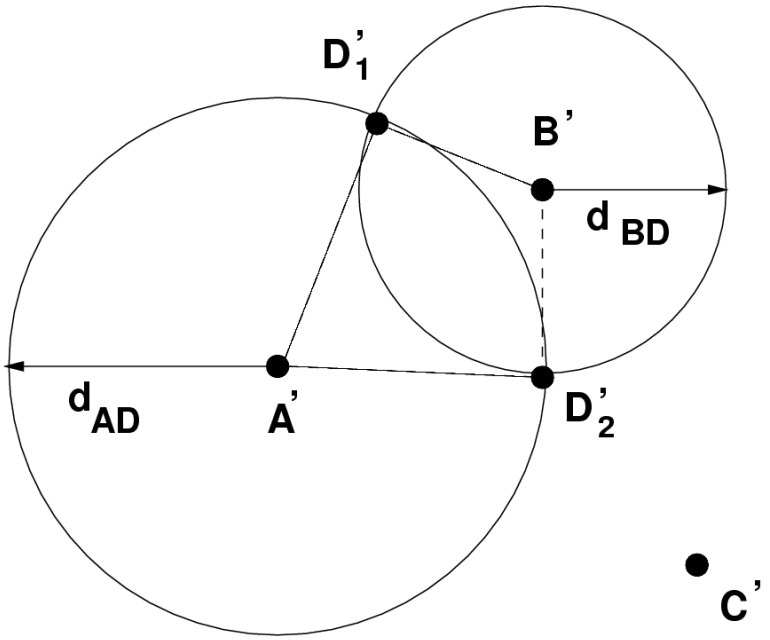
$$Stress(\boldsymbol{y}) = \frac{1}{\sum_i \sum_{j>i} d_{ij}^{(q+2)}} \sum_i \sum_{j>i} d_{ij}^{q} (\delta_{ij} - d_{ij})^2$$

- weight factor $q = \ldots, -2, -1, 0, 1, 2, \ldots$
  $q < 0$: emphasise short distances
  $q > 0$: emphasise large distances
  Sammon mapping: $q = -1$
- Minimise as function of $\boldsymbol{y}$, using your favourite minimizer, e.g. gradient descent:

$$\boldsymbol{y}' = \boldsymbol{y} - \alpha \frac{\partial Stress(\boldsymbol{y})}{\partial \boldsymbol{y}}$$

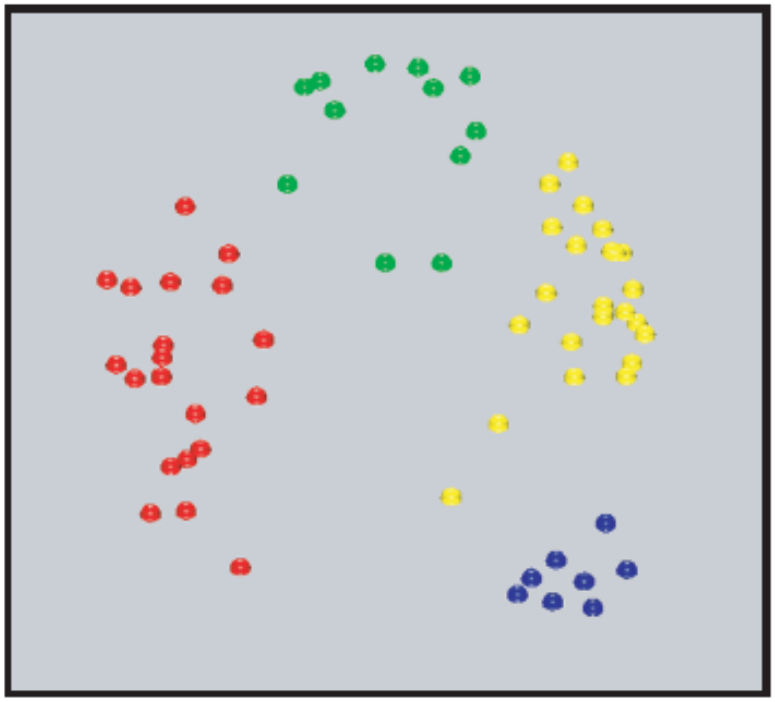
Embedding new points

- Problematic: re-run entire algorithm...
- Sub-optimal solution: triangulation
  - Embed new point **D**
  - **D** has **A** and **B** as neighbors in original space
  - Preserve distance to two embedded neigbours **A'**, **B'** exactly
  - Use **C'** to decide which of the two candidates D_1', D_2' to use

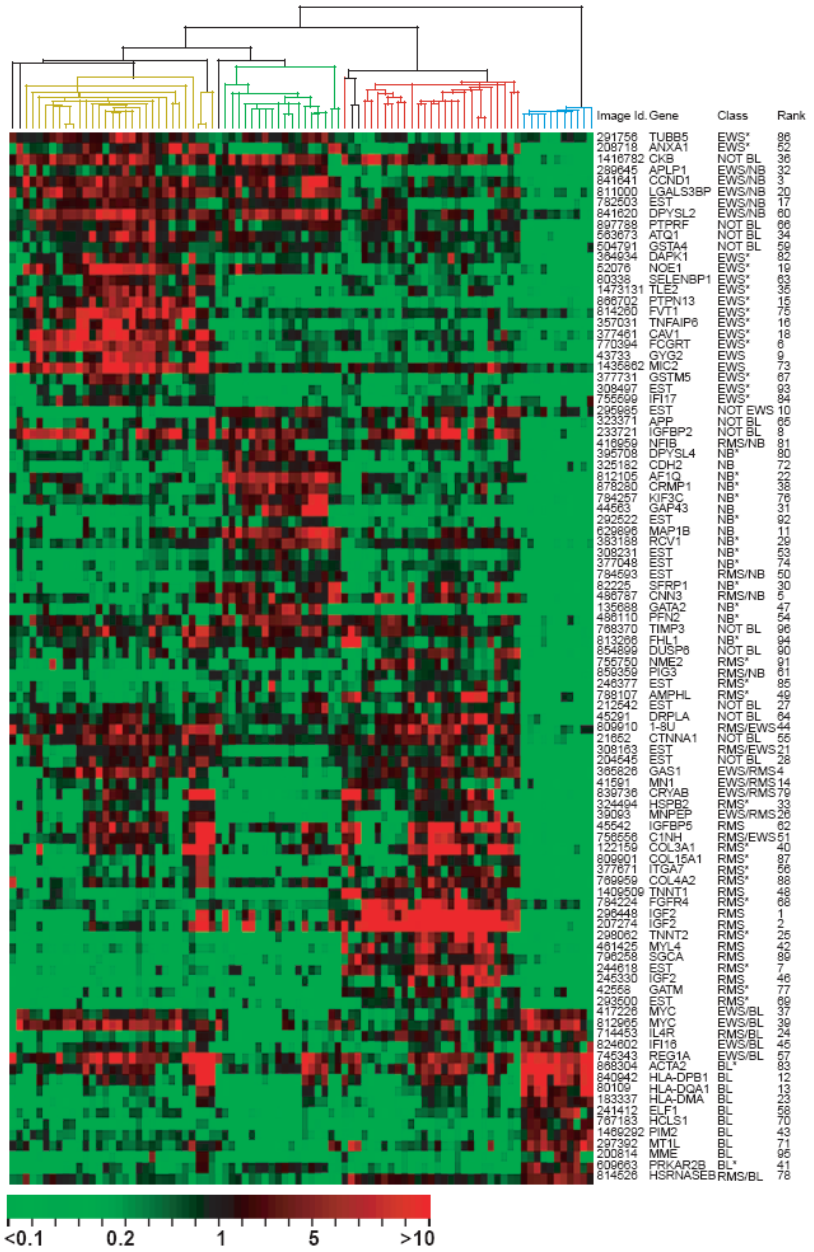


BioSB

MDS example



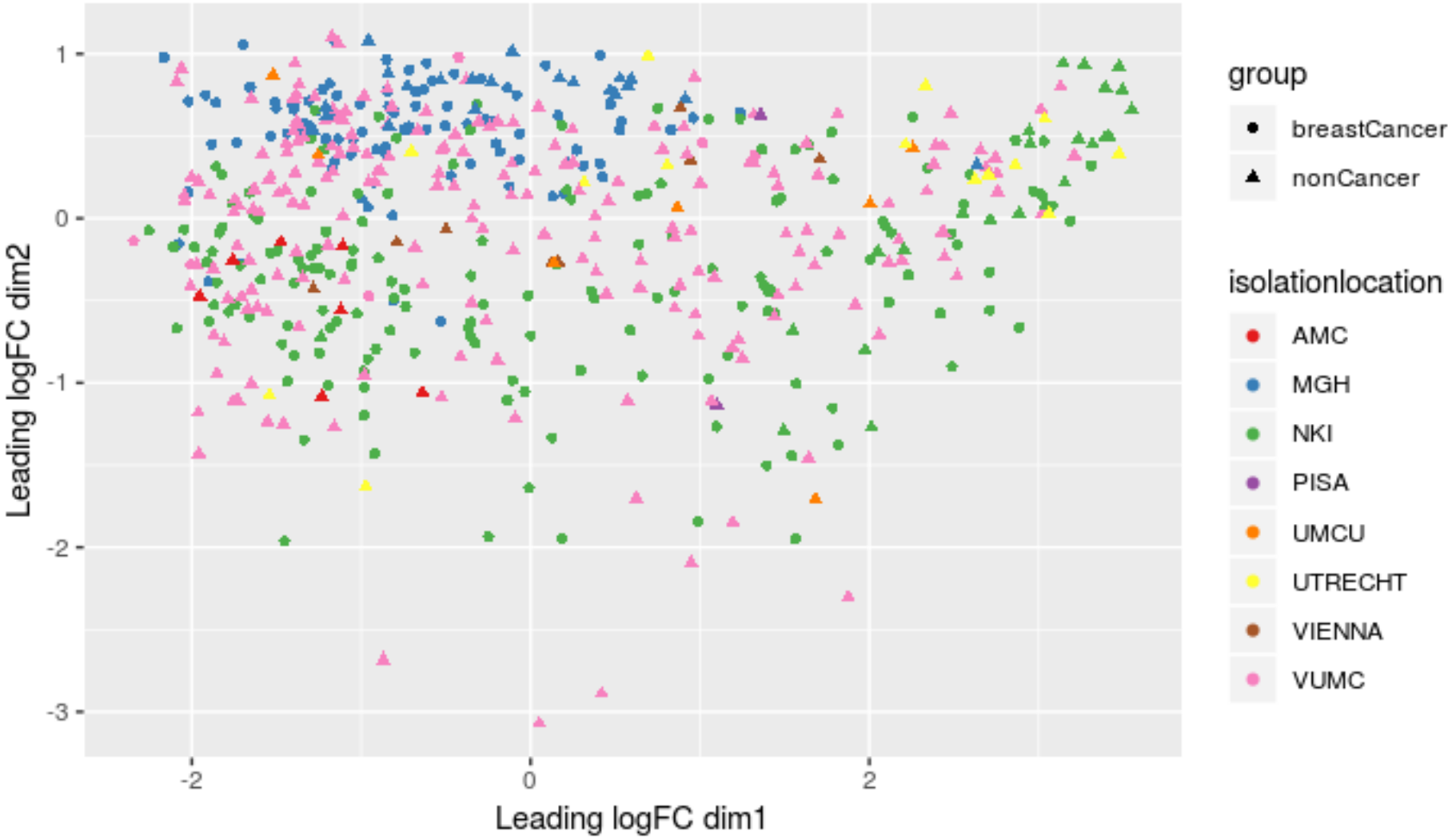
- Neuroblastoma (NB)
- Rhabdomyosarcoma (RMS)
- Burkitt lymphoma (BL)
- Ewing family of tumors (EWS),



Khan et al, Nature Medicine, 2001

BioSB

MDS Example: detecting batch effects



t-SNE (t-distributed stochastic neighbor embedding) (van der Maaten et al, 2008)

- **Definitions:**
  - A **data point** is a point x_i in the original data space R^D
  - A **map point** is a point y_i in the map space R^2
  - This space will contain our final representation of the dataset
  - There is a bijection between the data points and the map points
  - Every map point represents one of the data points.

BioSB

t-SNE (3)

- **How do we choose the positions of the map points?**
  - We want to conserve the structure of the data
  - data points close $\Longrightarrow$ corresponding map points close

BioSB

t-SNE (3)

- **How do we choose the positions of the map points?**
  - We want to conserve the structure of the data
  - data points close $\Longrightarrow$ corresponding map points close
- **Similarity of data points**
  - Let $|x_i - x_j|$ be the Euclidean distance between two data points
  - The conditional similarity between the two data points is:

$$p_{j|i} = \frac{\exp\left(-|x_i - x_j|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-|x_i - x_j|^2 / 2\sigma_i^2\right)}$$

BioSB

t-SNE (3)

- **How do we choose the positions of the map points?**
  - We want to conserve the structure of the data
  - data points close $\Longrightarrow$ corresponding map points close
- **Similarity of data points**
  - Let $|x_i - x_j|$ be the Euclidean distance between two data points
  - The conditional similarity between the two data points is:

$$p_{j|i} = \frac{\exp\left(-|x_i - x_j|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-|x_i - x_j|^2 / 2\sigma_i^2\right)}$$

  - This represents the likelihood of x_j given x_i
  - Assuming a Gaussian distribution around x_i with variance σ_i^2

BioSB

t-SNE (4)

- The conditional similarity between the two data points is:

$$p_{j|i} = \frac{\exp\left(-|x_i - x_j|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-|x_i - x_j|^2 / 2\sigma_i^2\right)}$$

- This represents the likelihood of x_j given x_i
- Assuming a Gaussian distribution around x_i with variance σ_i^2
- σ_i^2 is different for every point
  - dense areas are given a smaller variance
  - sparse areas are given a larger variance
- Similarity is symmetrized form:

$$p_{ij} = (p_{j|i} + p_{i|j})/2N$$

t-SNE (5)

- **Similarity of map points:**
  - Let $|y_i - y_j|$ be the Euclidean distance between the map points

$$q_{ij} = \frac{f(|y_i - y_j|)}{\sum_{k \neq i} f(|y_i - y_k|)}$$

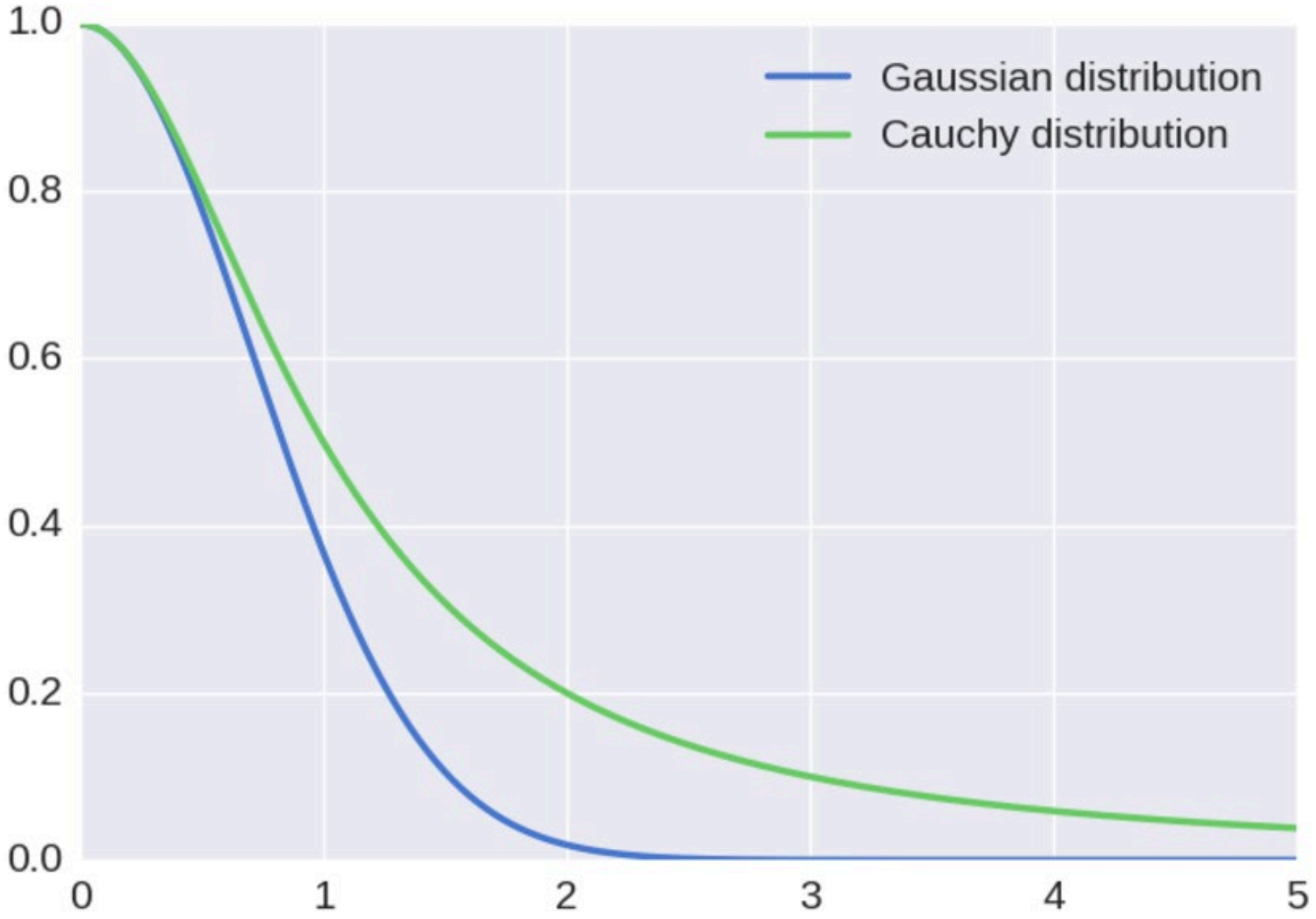
    with

$$f(z) = 1/(1 + z^2)$$

  - Same as for the data points, but different distribution:
  - t-Student with one degree of freedom, or Cauchy distribution

BioSB

t-SNE (10): Cauchy and Gaussian distribution



BioSB

t-SNE (6)

- **How do we choose the positions of the map points?**
  - Data similarity matrix (p_{ij}) is fixed
  - Map similarity matrix (q_{ij}) depends on the map points
  - Get two matrices as similar as possible
  - Idea: similar data points yield similar map points

BioSB

t-SNE (7)

- **Algorithm**
  - Minimizing the Kullback-Leibler divergence between p_{ij} and q_{ij}:

$$KL(P,Q) = \sum_{i,j} p_{ij} \log(p_{ij}/q_{ij})$$

  - $KL(P,Q)$ measures the distance between our two distributions

t-SNE (7)

- **Algorithm**
  - Minimizing the Kullback-Leibler divergence between p_{ij} and q_{ij}:

$$KL(P,Q) = \sum_{i,j} p_{ij} \log(p_{ij}/q_{ij})$$

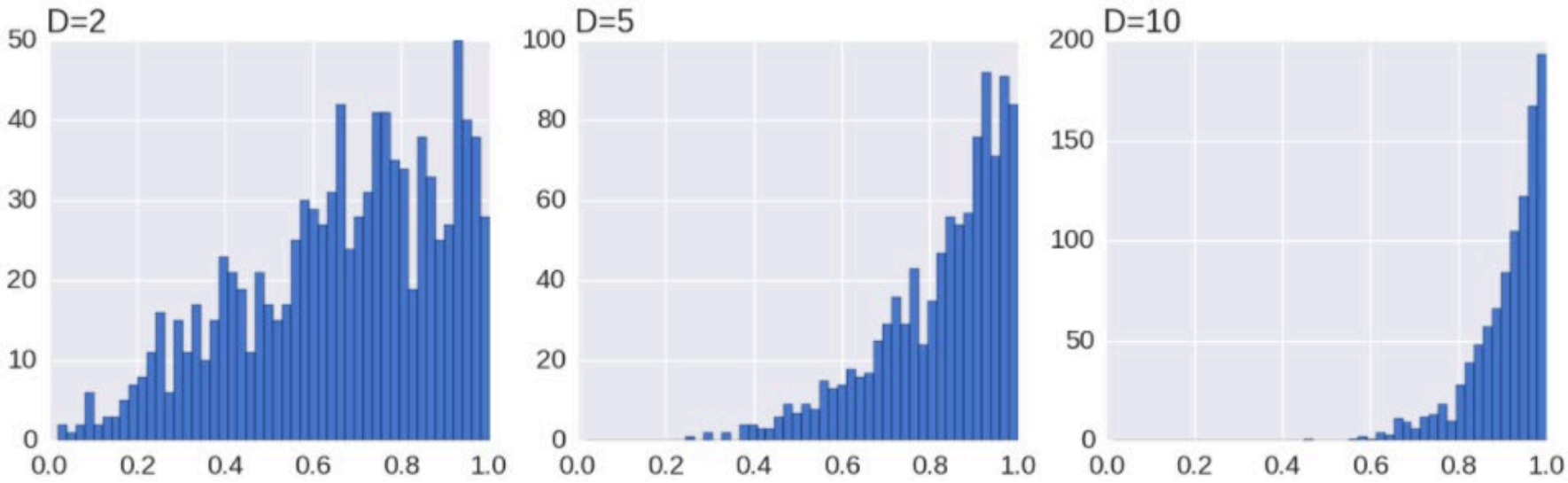
  - $KL(P,Q)$ measures the distance between our two distributions
  - To minimize this score, we perform a gradient descent
  - The gradient can be computed analytically:

$$\frac{\partial KL(P,Q)}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij}) f(|y_i - y_j|)$$

    where $f(z) = z/(1+z^2)$
  - The gradient expresses the sum of all spring forces applied to map point i.

t-SNE (8)

- **Why the t-Student distribution?**
  - The volume of the N-dim. ball of radius r scales as r^N
  - For large N, random points are close to the surface, not in center



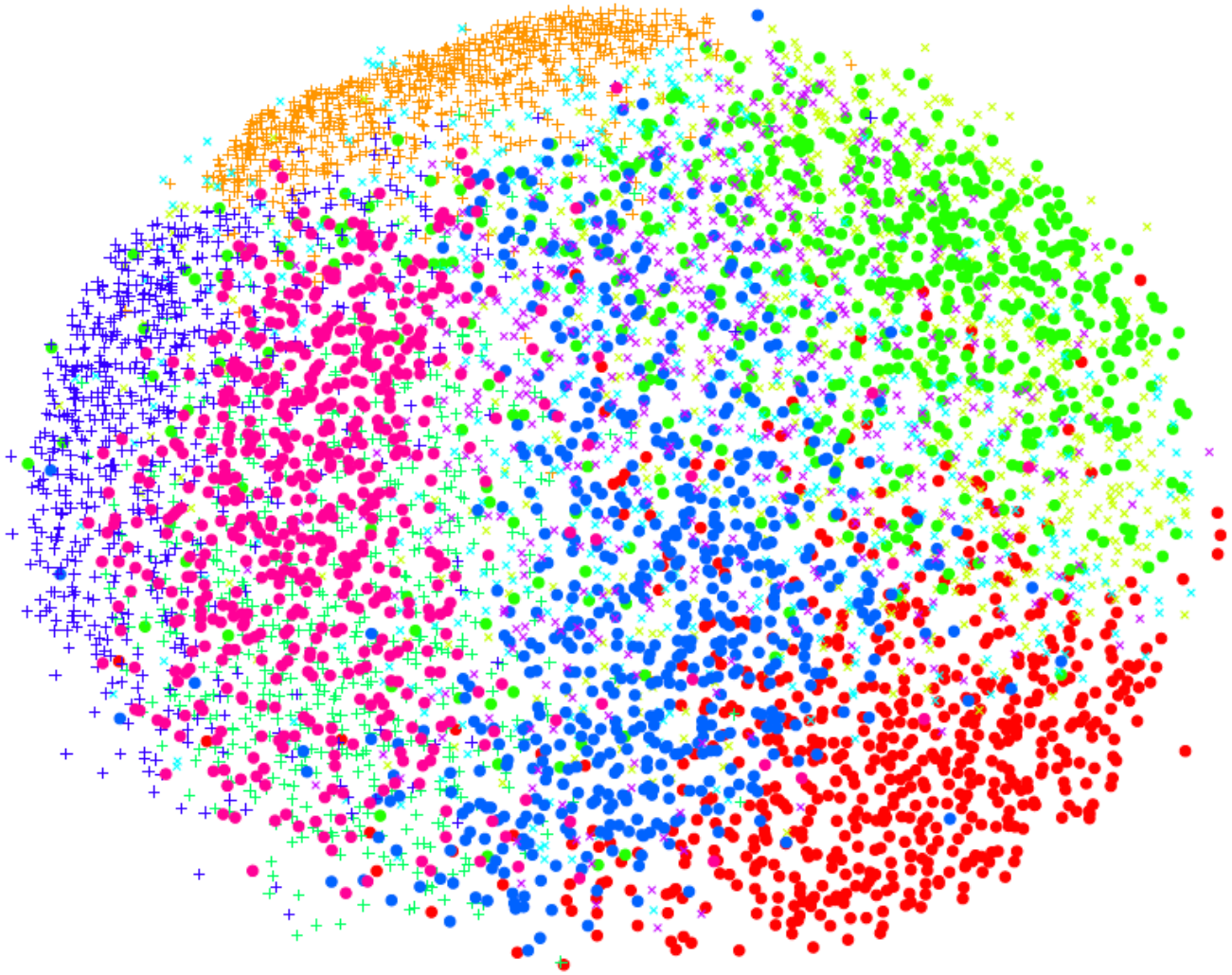
BioSB

t-SNE (9)

- **Why the t-Student distribution?**
  - We go from high dimensionality to low dimensionality
  - If we use the same distribution in data and map space
  - There is an imbalance in the distribution of the distances of a point's neighbors.
  - As distances are so different between a high-D and low-D space
  - Yet, t-SNE tries to reproduce the same distances in the two spaces
  - Result: excess of 'attraction forces' (gradient) that move map space points
  - t-SNE compensates this by the heavier tail of the t-distribution in low-D
  - Larger distances in low-D get more weight so match high-D

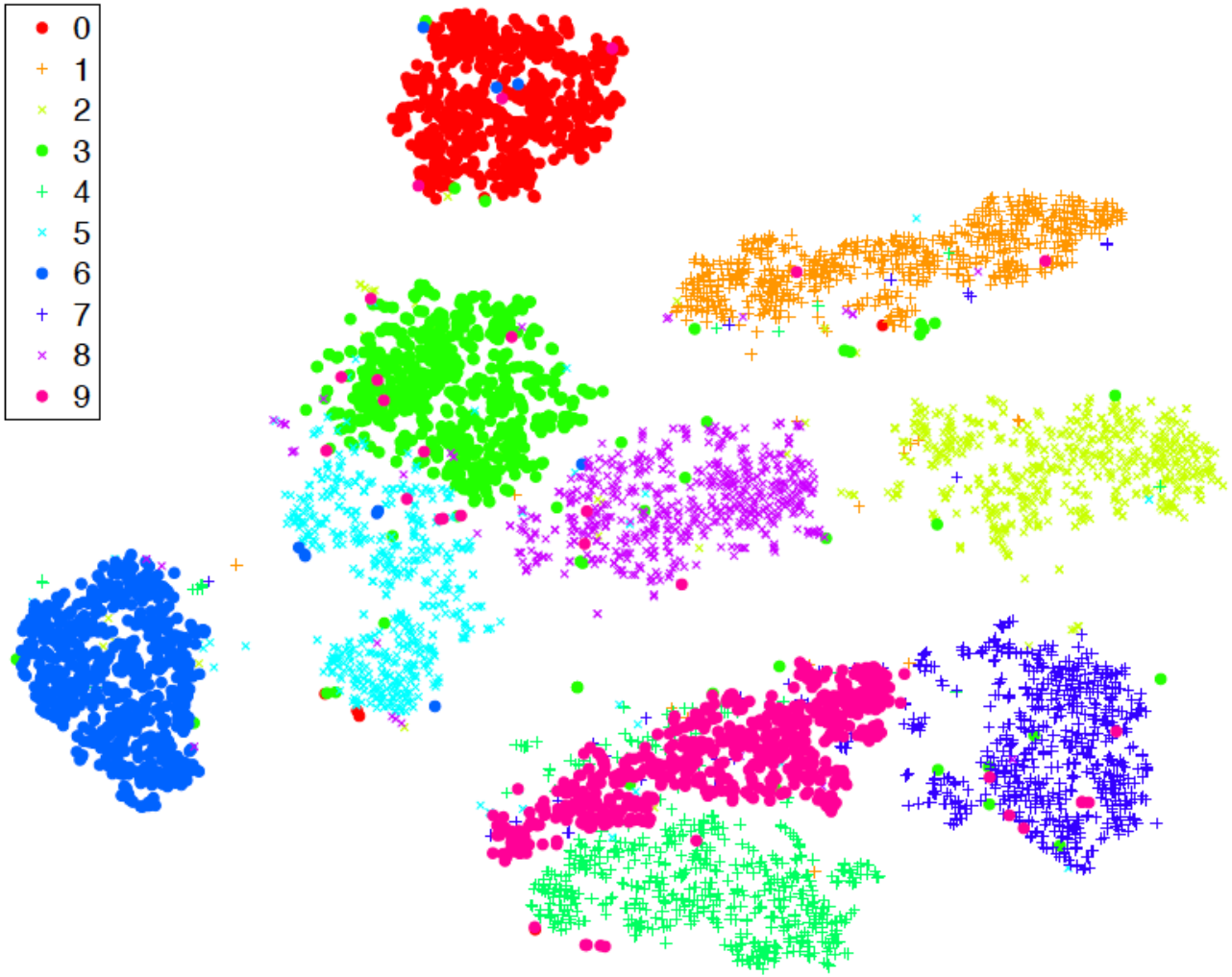
BioSB

t-SNE (11): Sammon map of digit data



BioSB

t-SNE (12): t-SNE map of digit data



BioSB

MDS conclusions

- Experts or measurements give distances
- Optimise a *stress-function* (MDS) or KL distance (t-SNE)
- Important:
  - *the distance measure used*: is it representative?
  - *the weighting of distances* (q): can influence outcome heavily.
  - t-SNE run with defaults is quite reliable
- Largest risk:
  seeing structure in the data that is not really there
- Remaining problem: embedding new data points

Feature selection

- For feature selection, we need:
  - A **criterion** function
    e.g. error, class overlap, information loss
  - A **search algorithm**
    e.g. pick the best single feature at each time

Criteria

1. **Wrapper**: exact performance measure
    - base performance estimate on classifier;
    - estimate performance using cross-validation:
    - very expensive!

Criteria

1. **Wrapper**: exact performance measure
    - base performance estimate on classifier;
    - estimate performance using cross-validation:
    - very expensive!

Note:
we should never use the training set to calculate performance; this will give a biased estimate!

Criteria

1. **Wrapper**: exact performance measure
   - base performance estimate on classifier;
   - estimate performance using cross-validation:
   - very expensive!

Note:
we should never use the training set to calculate performance; this will give a biased estimate!

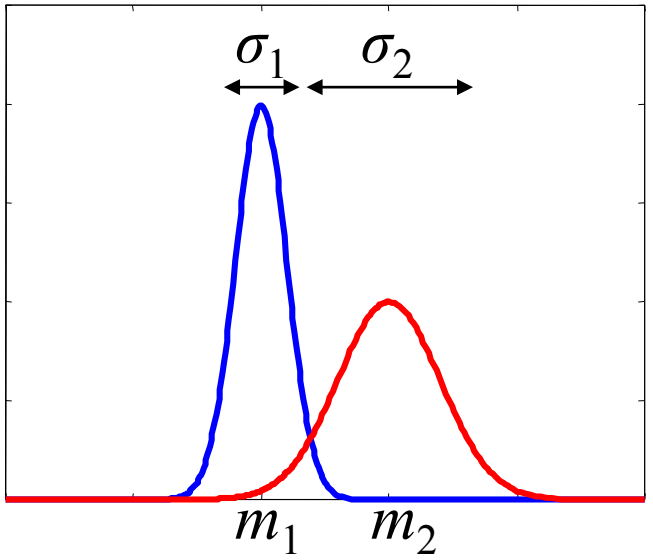
2. **Filter**: approximate performance predictors:
   - calculate the performance of an easy-to-use model;
   - this gives an indication of how well a more powerful model may perform but
   - is much faster to compute.

BioSB

Criteria (2)

- Example
  - Simple measure of the ‘separability’ of classes given a feature
  - 1D case: Signal-to-Noise Ratio (SNR) or Fisher criterion:

$$J_F = \frac{|m_1 - m_2|^2}{\left(\sigma_1^2 + \sigma_2^2\right)}$$

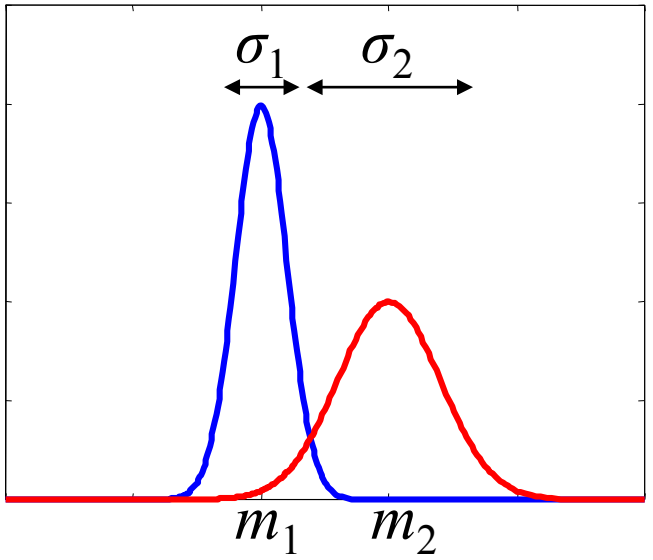


BioSB

Criteria (3)

- Example
  - Simple measure of the ‘separability’ of classes given a feature
  - 1D case: Signal-to-Noise Ratio (SNR) or Fisher criterion:

$$J_F = \frac{|m_1 - m_2|^2}{\left(\sigma_1^2 + \sigma_2^2\right)}$$



  - If J_F is large: good separability
  - If J_F is small: poor separability

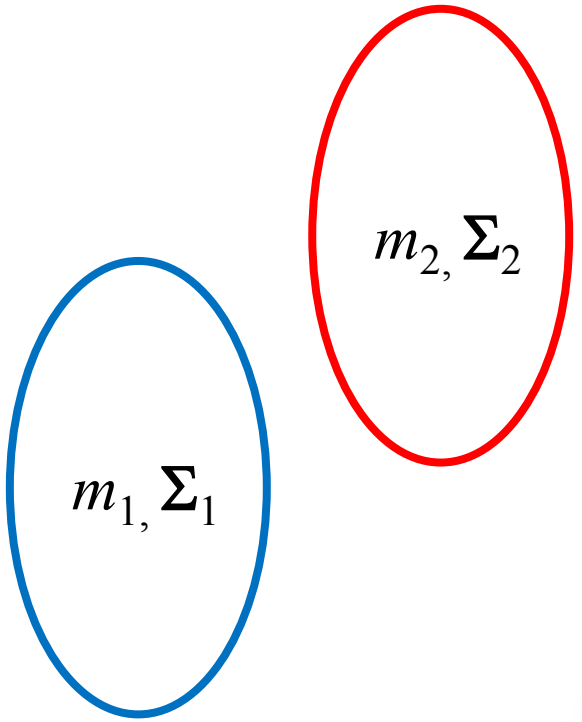
BioSB

Criteria (4)

- The multi-variate equivalent of Fisher criterion is the
- Mahalanobis distance:
  - assumes
    - Gaussian distributions with
    - equal covariance matrix Σ:

$$D_M = (m_1 - m_2)^T \Sigma^{-1} (m_1 - m_2)$$

$$\Sigma = \sum_{i=1}^{C} \frac{n_i}{n} \Sigma_i$$



BioSB

Criteria (5)

- Recall

$$\boldsymbol{S}_w = \sum_{i=1}^{C} \frac{n_i}{n} \boldsymbol{\Sigma}_i \qquad \boldsymbol{S}_B = \sum_{i=1}^{C} \frac{n_i}{n} (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

- Scatter-based classification performance indicators:

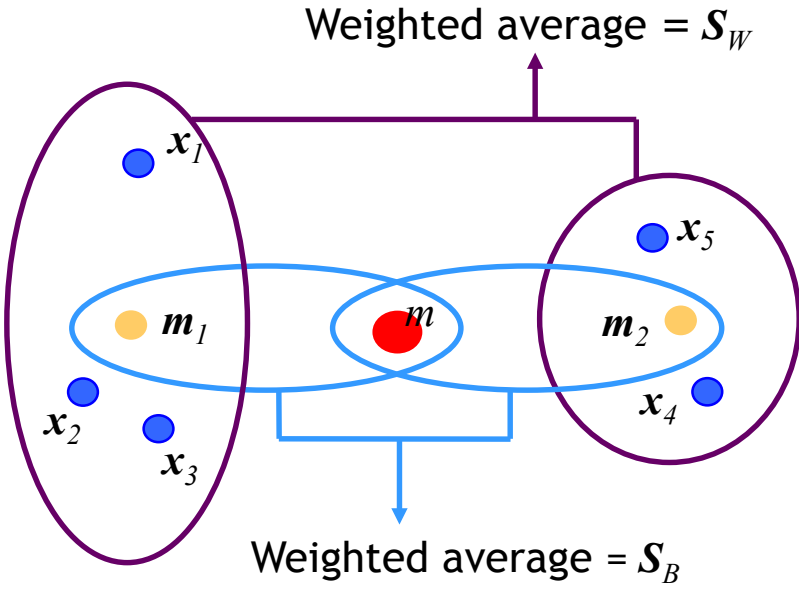
$J_1 = \text{trace}\,(\boldsymbol{S}_W + \boldsymbol{S}_B) = \text{trace}\,(\boldsymbol{\Sigma})$

$J_2 = \text{trace}\,(\boldsymbol{S}_B / \boldsymbol{S}_w)$

$J_3 = \det\,(\boldsymbol{\Sigma}) / \det\,(\boldsymbol{S}_w)$

$J_3 = \text{trace}\,(\boldsymbol{S}_W) / \text{trace}\,(\boldsymbol{S}_B)$

(trace = sum of diagonal elements)



- These are all just approximations!

BioSB

Search algorithms

- **Feature selection**: select a subset of d out of p measurements which optimises the criterion

- Simplest solution: look at all possible subsets

- Problem: there are $\binom{p}{d} = \frac{p!}{(p-d)!d!}$ subsets

- *e.g.* $p = 50$ measurements, $d = 2$: 1225 subsets
  $d = 5$: 2.1×10^6 subsets
  $d = 25$: 1.3×10^{14} subsets

BioSB

Search algorithms (2)

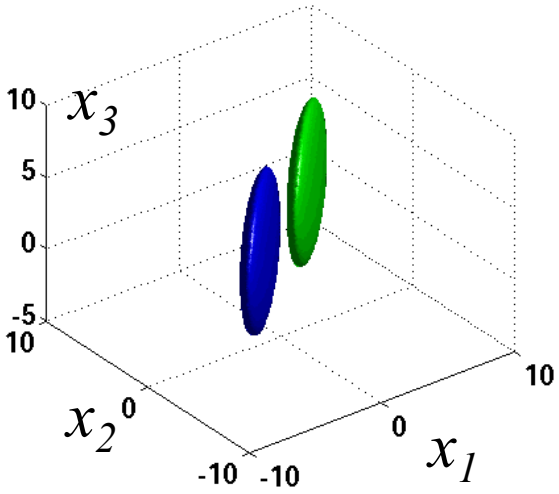
- Sub-optimal algorithms: select or deselect one feature (or a few features) at a time

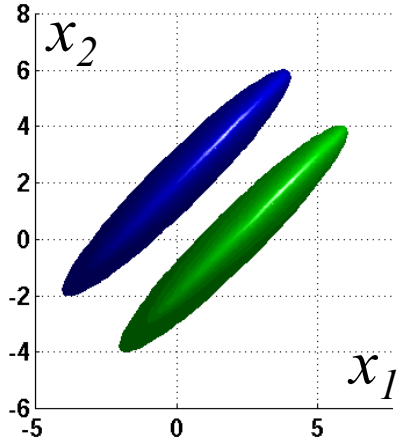
- Simplest: best individual d
  but these are not necessarily the best d !

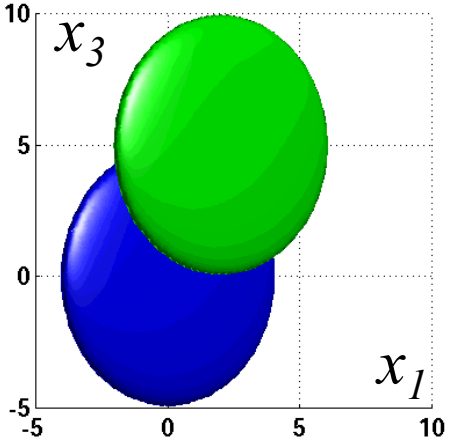
- Demonstration: two Gaussians;
  select 2 features out of 3 for classification

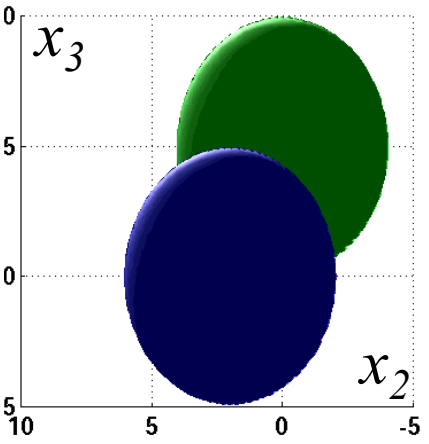
BioSB

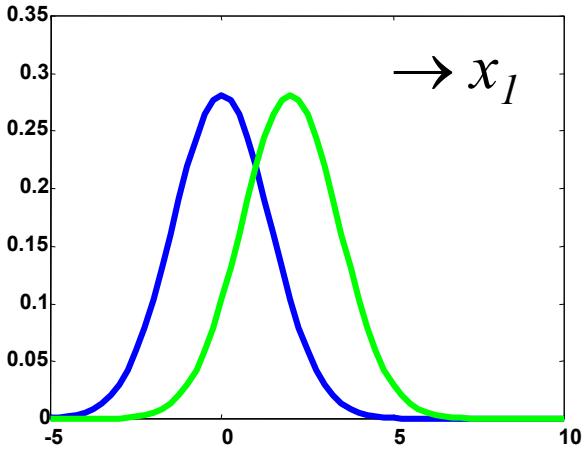
Search algorithms (3)

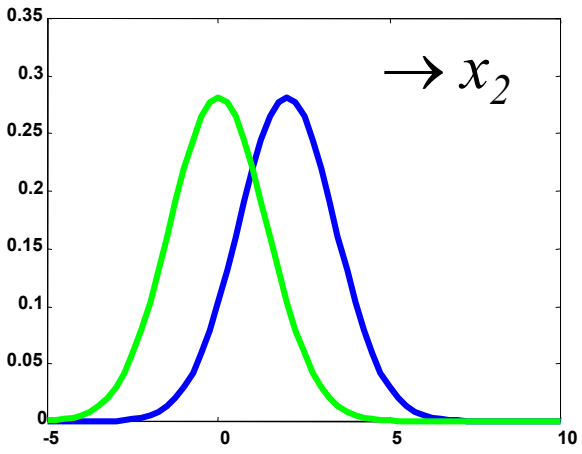


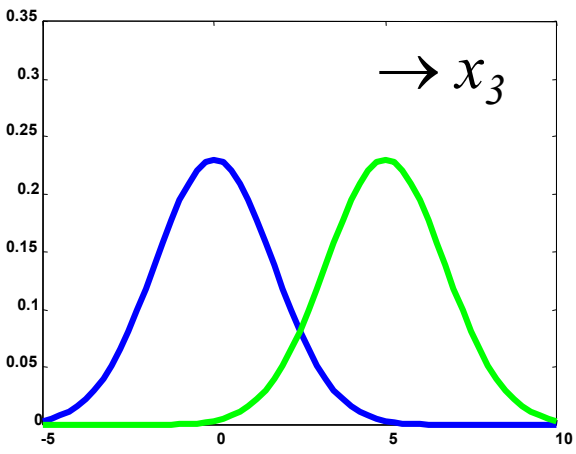






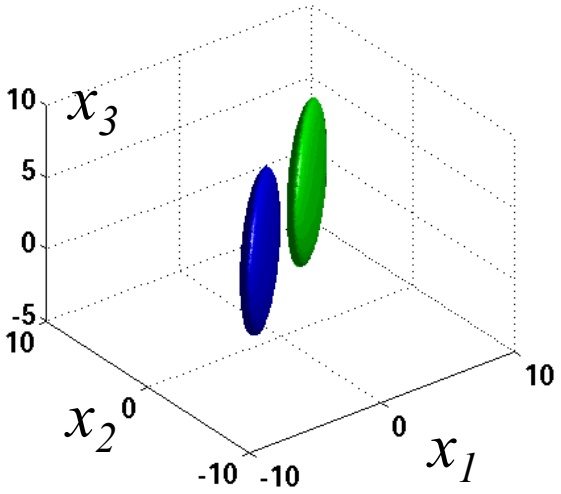




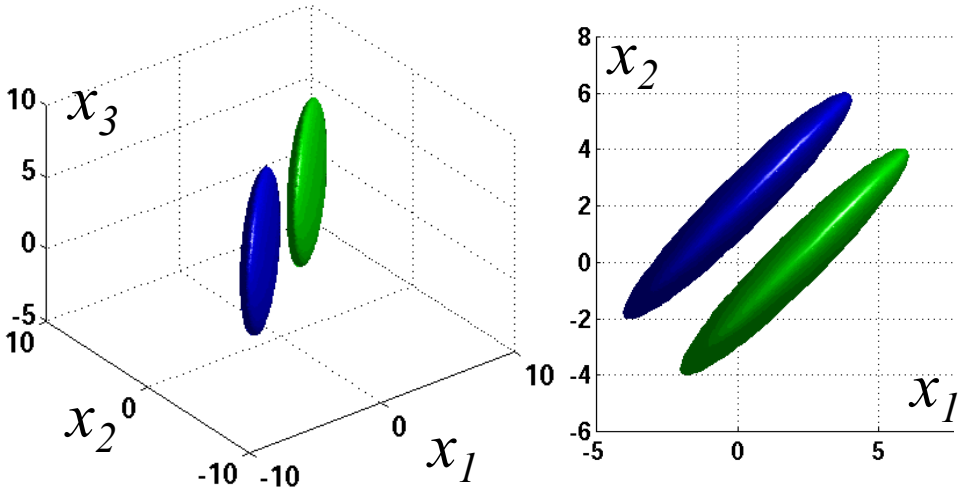


BioSB

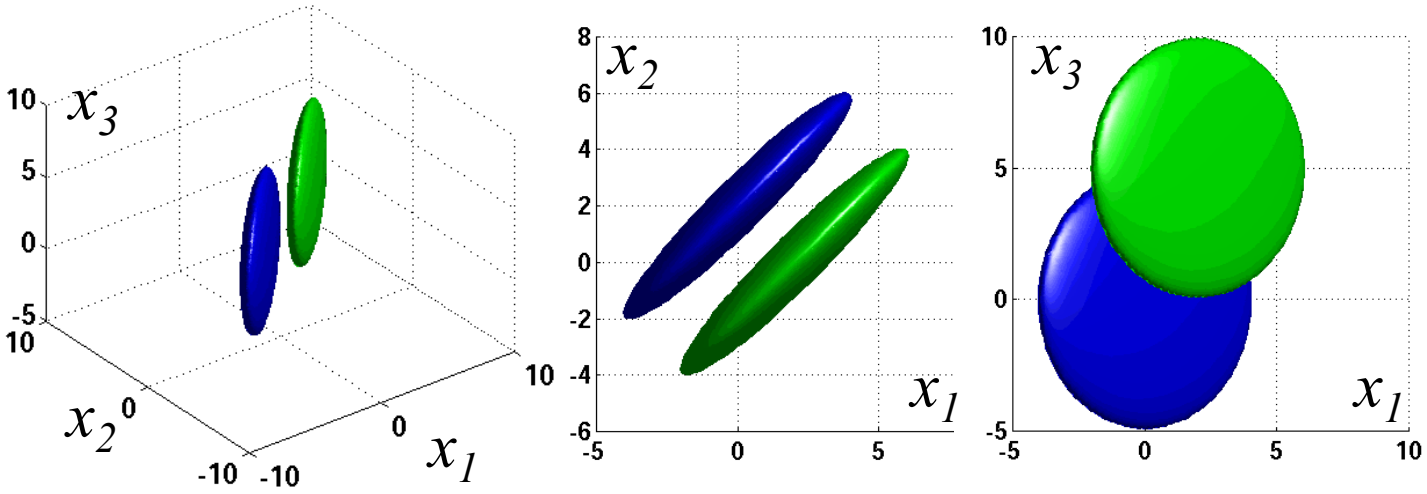
Search algorithms (3)



Search algorithms (3)

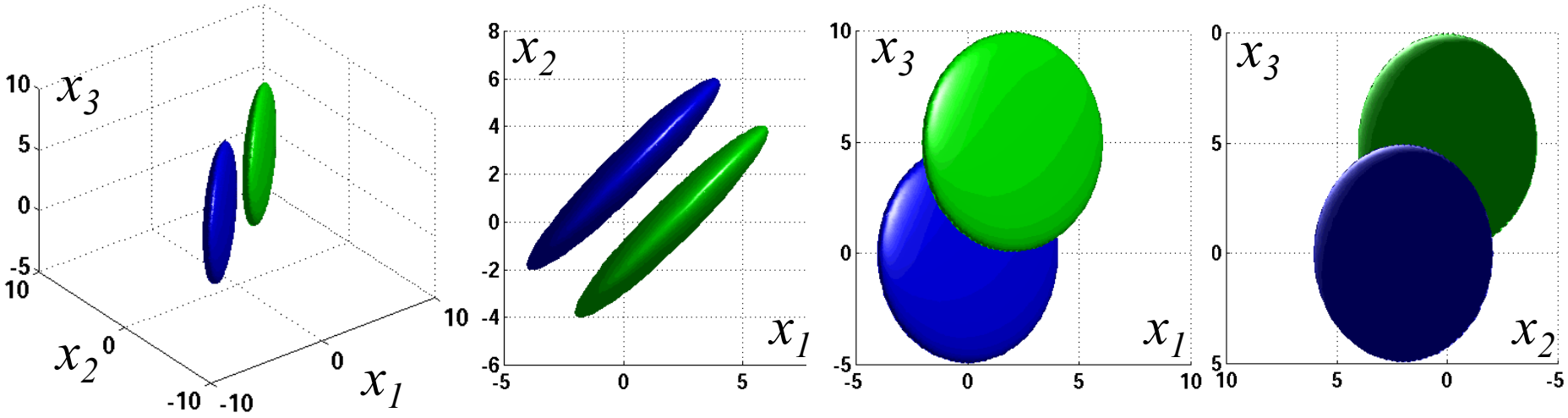


Search algorithms (3)

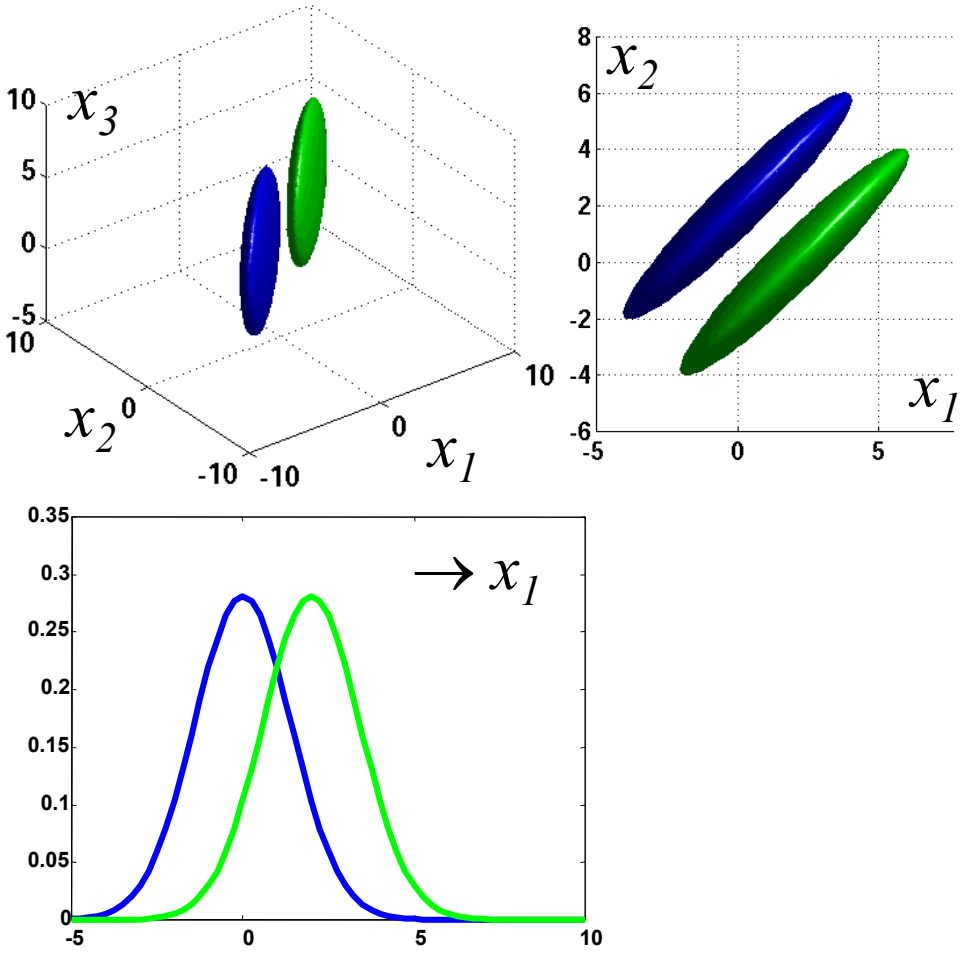


BioSB

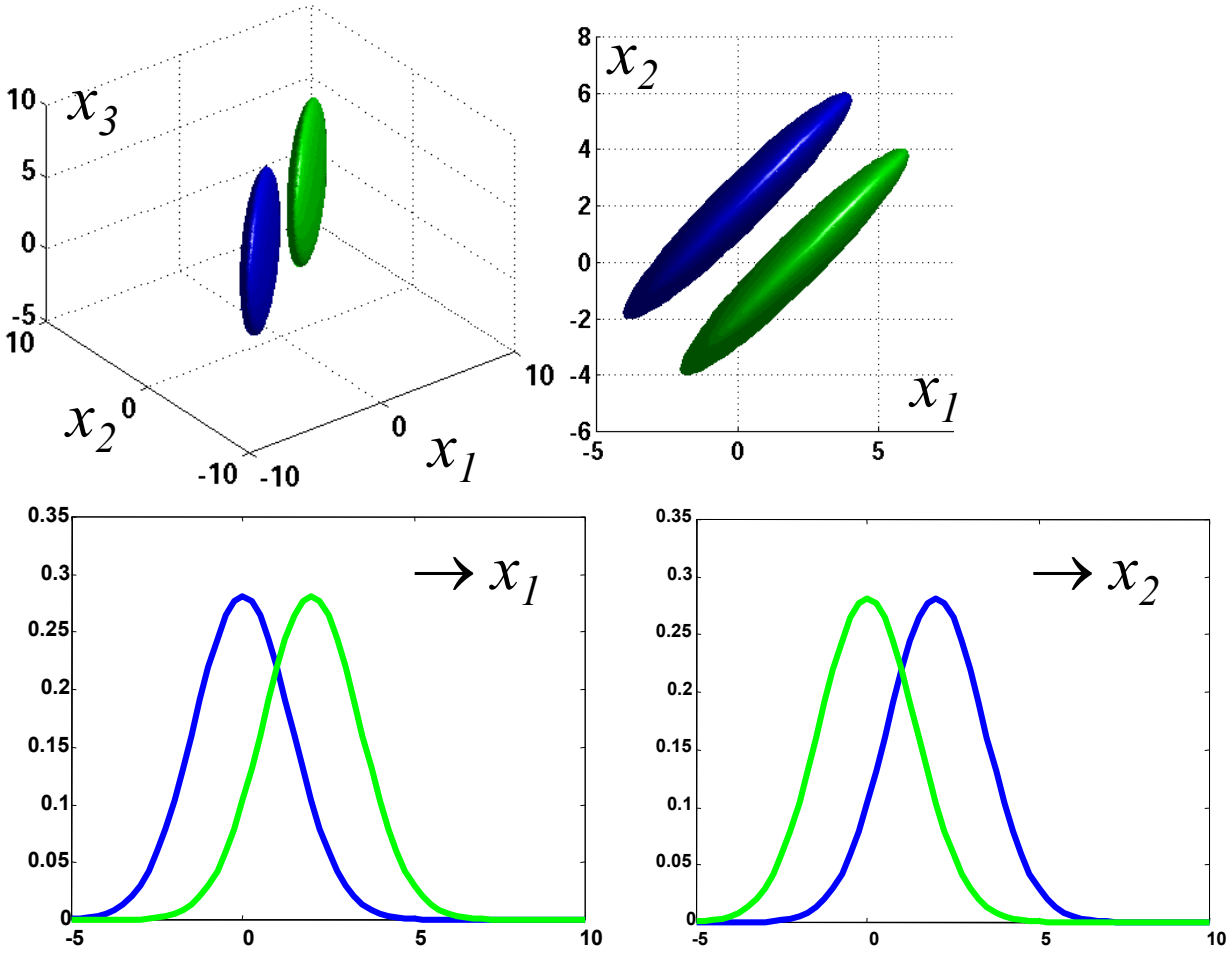
Search algorithms (3)



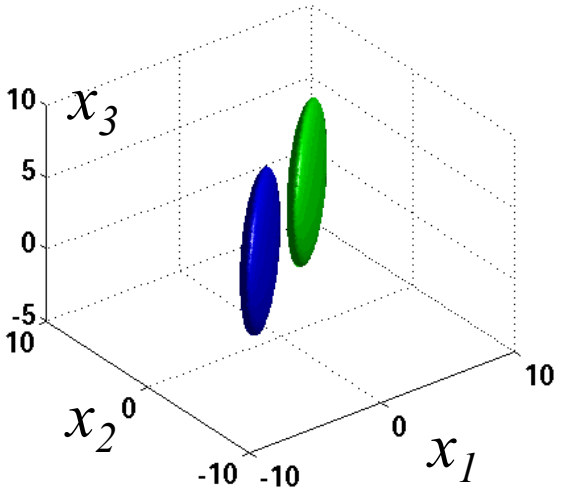
Search algorithms (3)

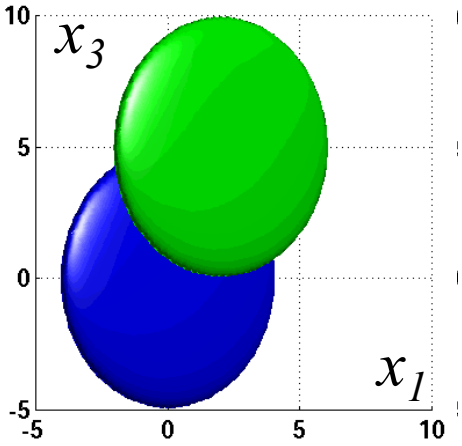


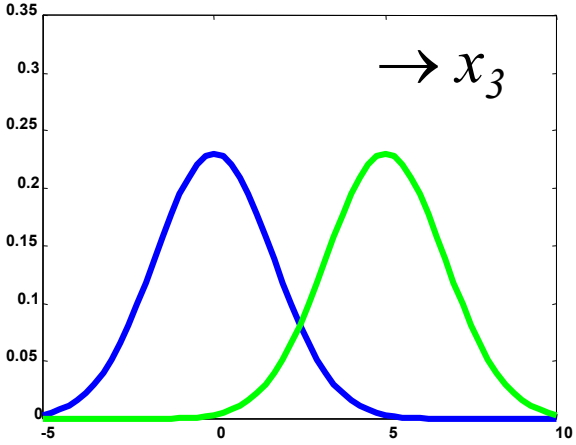
Search algorithms (3)



Search algorithms (3)

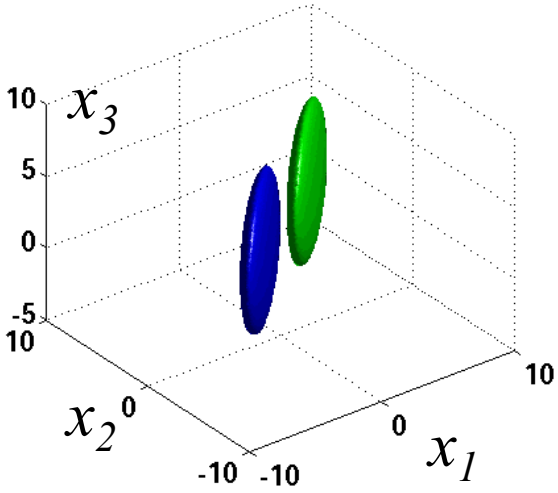


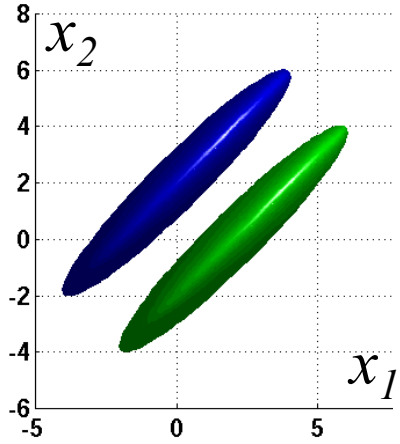


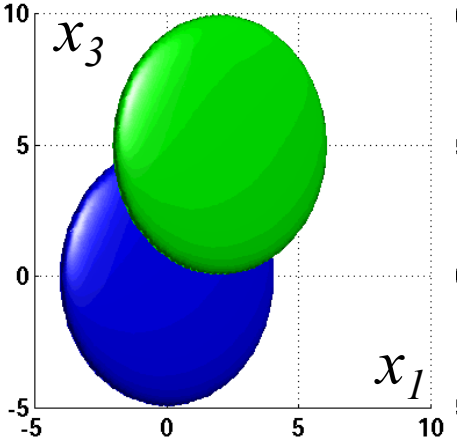


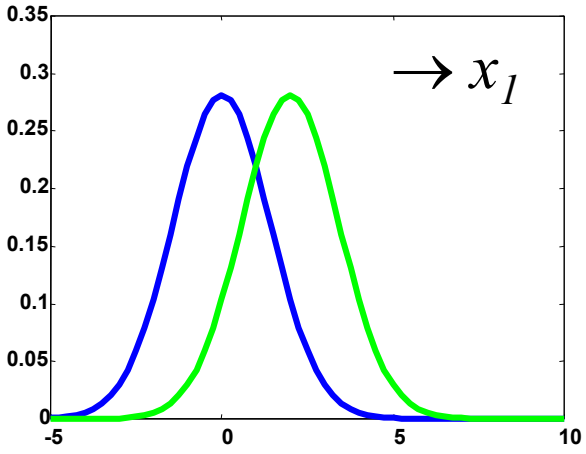
BioSB

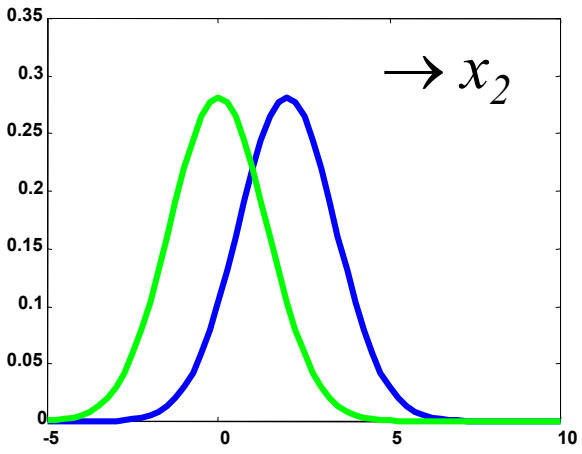
Search algorithms (3)

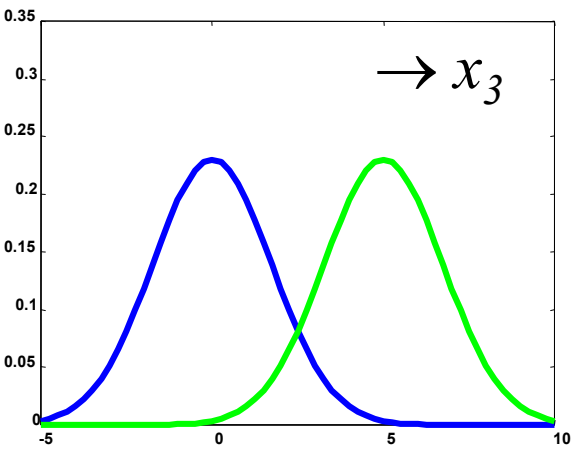












Search algorithms (4)

- Other sub-optimal algorithms:
  - Forward selection (for when d is low)
    - start with empty set
    - keep adding one feature at a time
      so that the entire subset so far performs best

Search algorithms (4)

- Other sub-optimal algorithms:
  - Forward selection (for when d is low)
    - start with empty set
    - keep adding one feature at a time
      so that the entire subset so far performs best
  - Backward selection (for when d is high)
    - start with entire set
    - keep removing one feature at a time
      so that the entire subset so far performs best

BioSB

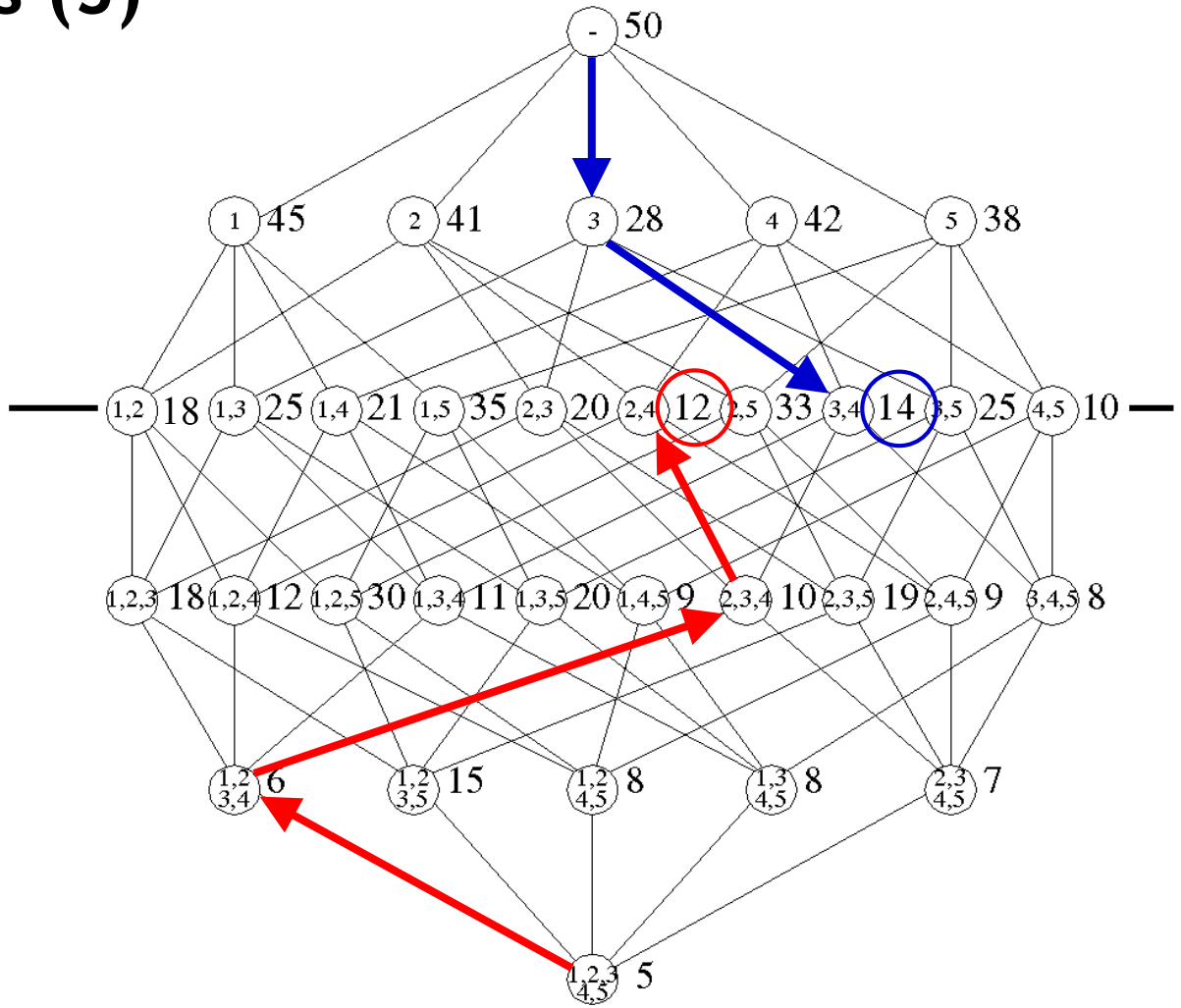
Search algorithms (4)

- Other sub-optimal algorithms:
  - Forward selection (for when d is low)
    - start with empty set
    - keep adding one feature at a time
      so that the entire subset so far performs best
  - Backward selection (for when d is high)
    - start with entire set
    - keep removing one feature at a time
      so that the entire subset so far performs best
  - Plus-l-takeaway-r (may be slightly better)
    - start with empty set (if $l > r$) or entire set (if $l < r$)
    - keep adding best l and removing worst r

BioSB

Search algorithms (5)

- Select $d = 2$ out of $p = 5$ features
- Sub-optimality illustrated:
  - forward
  - backward



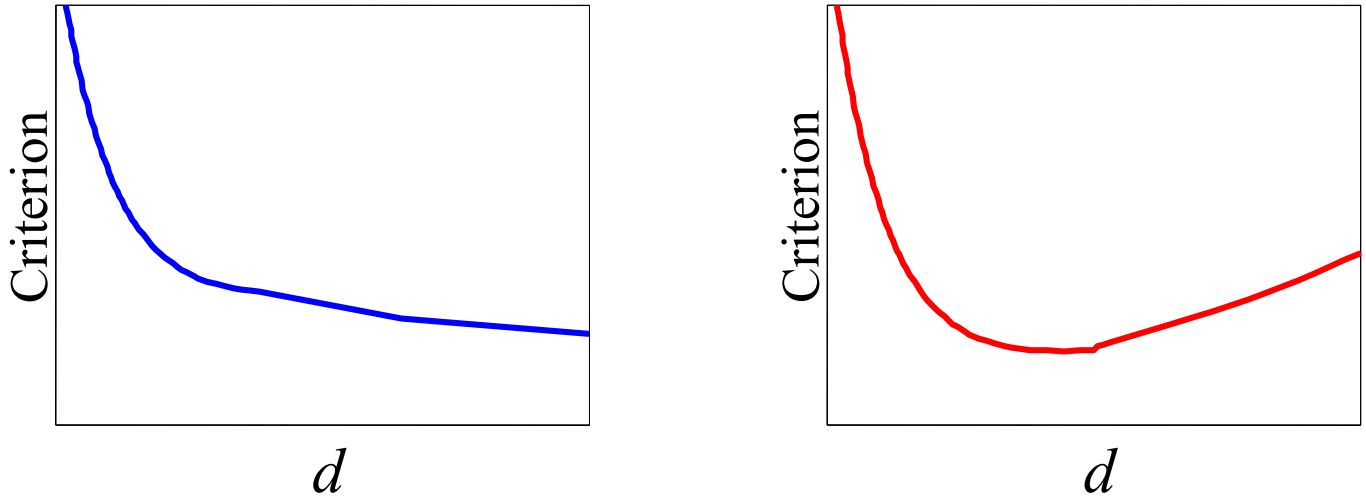
Feature subset

Criterion value

BioSB

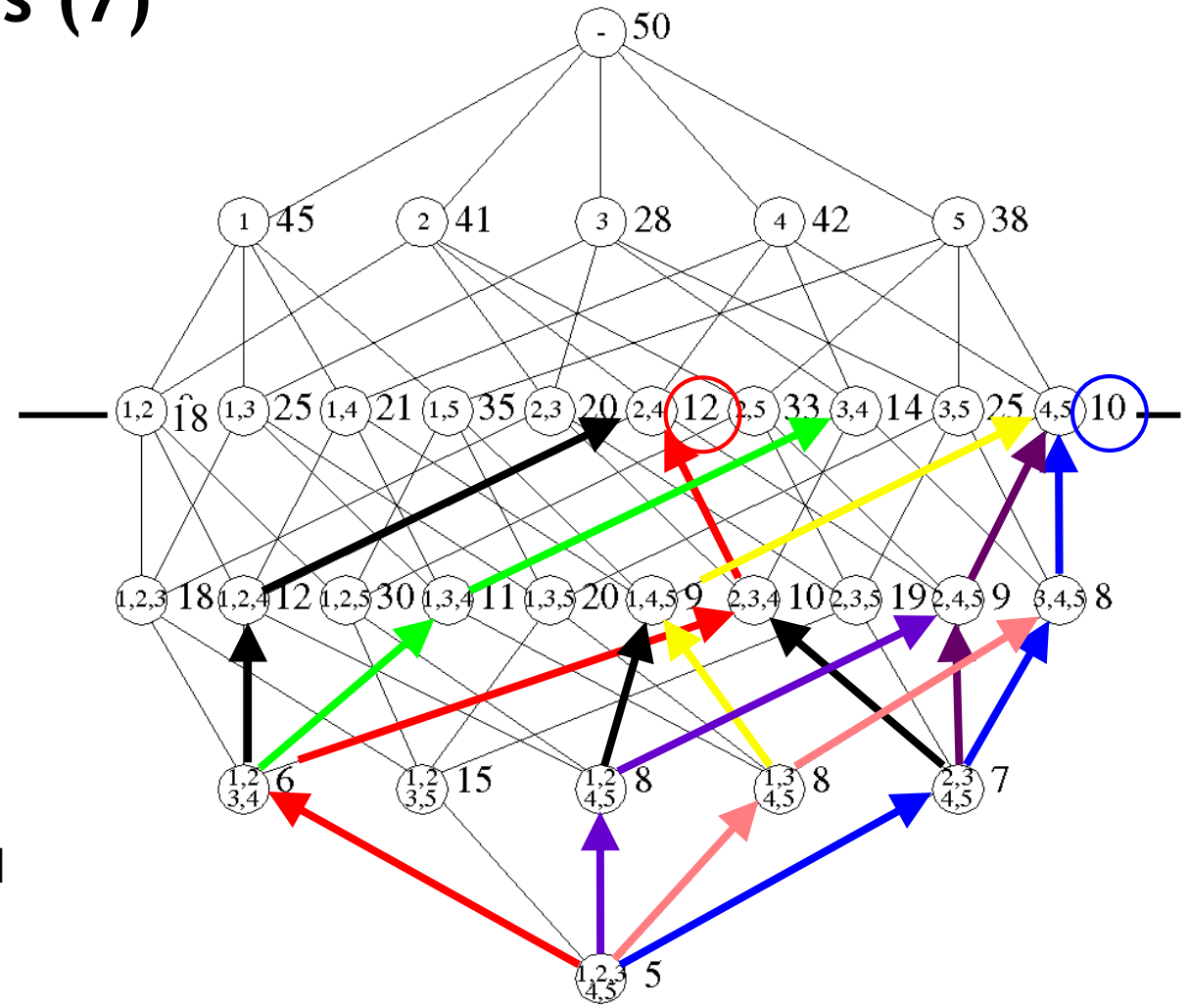
Search algorithms (6)

- Branch & bound: backtracking
- Optimal when criterion is monotonic in the number of features d



Search algorithms (7)

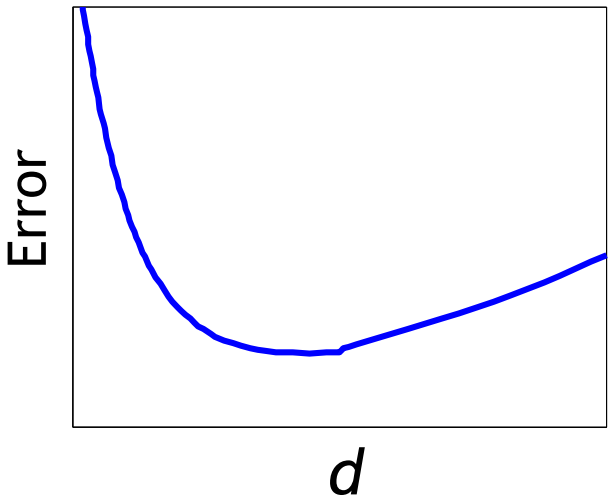
- Branch & bound
  - Use backward search to find preset number of features
  - Set *bound*, backtrack (*branch*) and use backward search again, considering just sets with criterion values better than the bound



BioSB

Search algorithms (8)

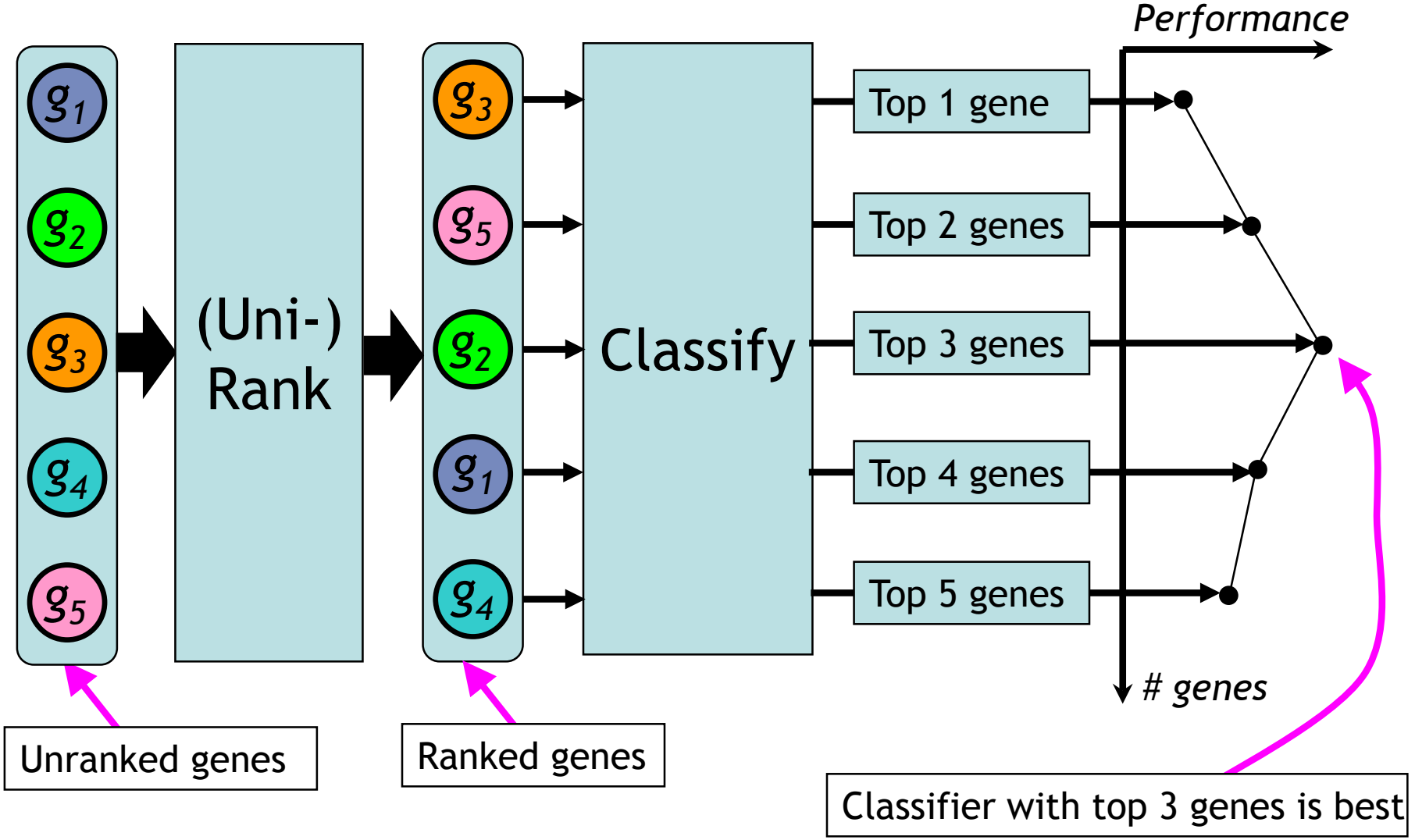
- When should we stop?
  - Due to estimation problems (*e.g.* covariance matrix), we may be overtraining on training set (in cross validation)
  - This is revealed by increasing error on the test set



  - Otherwise (with very large sample sizes), we will have to specify a desired number of measurements

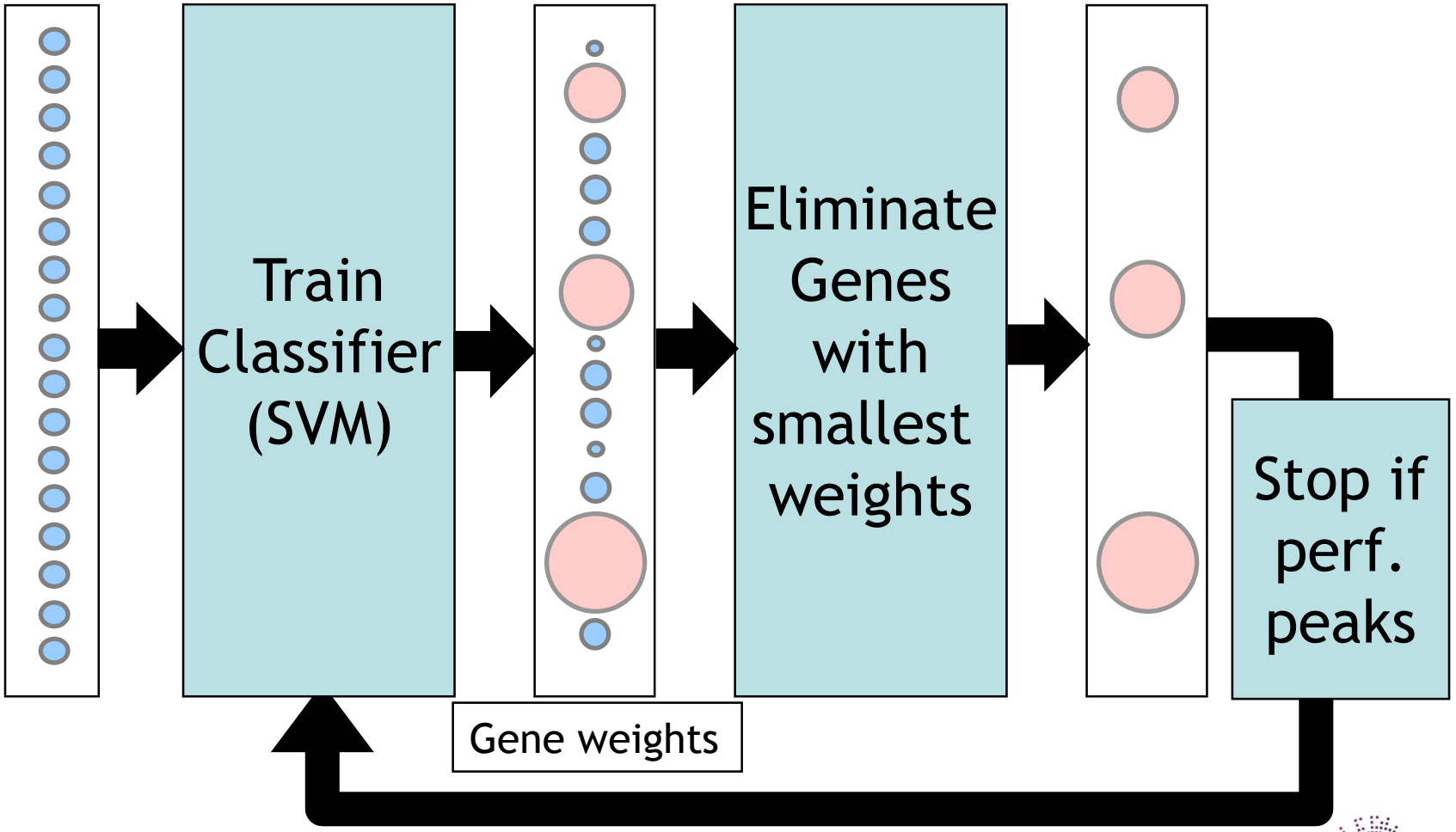
BioSB

Example: Filtering (forward selection)



Example: Recursive feature elimination (RFE)

Wrapper, Backward search



BioSB

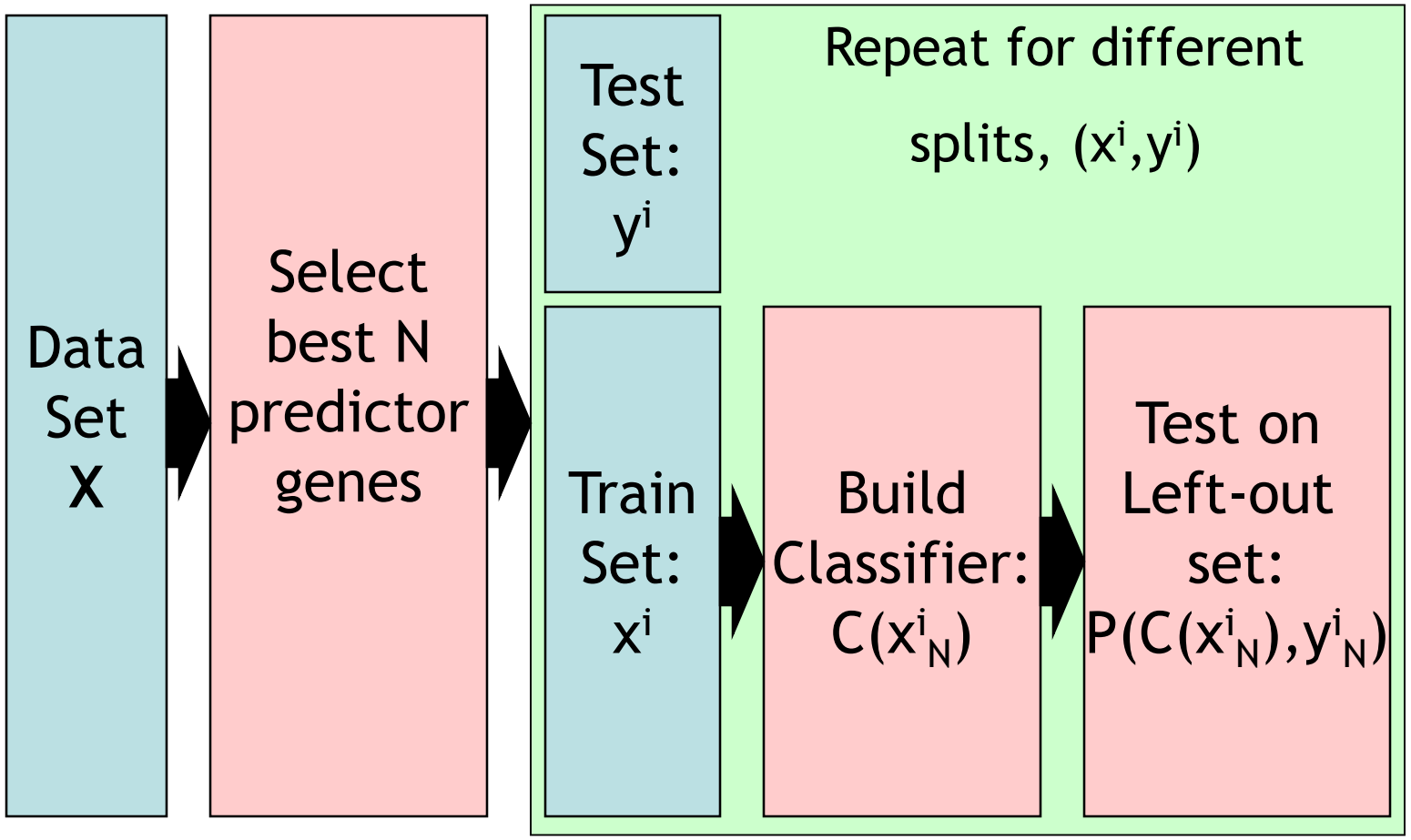
What can go wrong?

Selection bias...

- Guyon et al. (2002). Machine Learning **46**, 389 – 422.
- Ambroise and McLachlan (2002). PNAS **99**, 6562-6566.

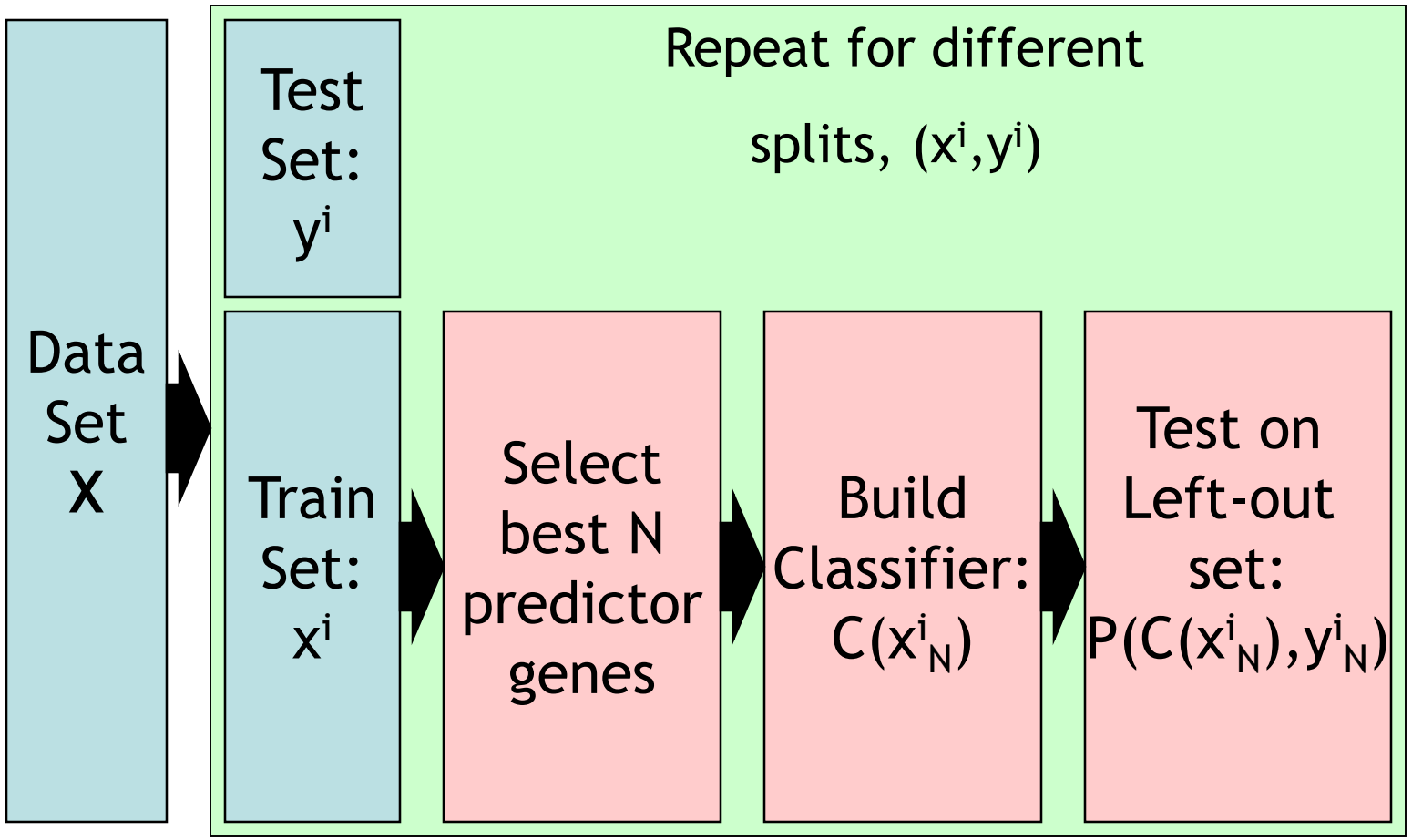
BioSB

Biased selection



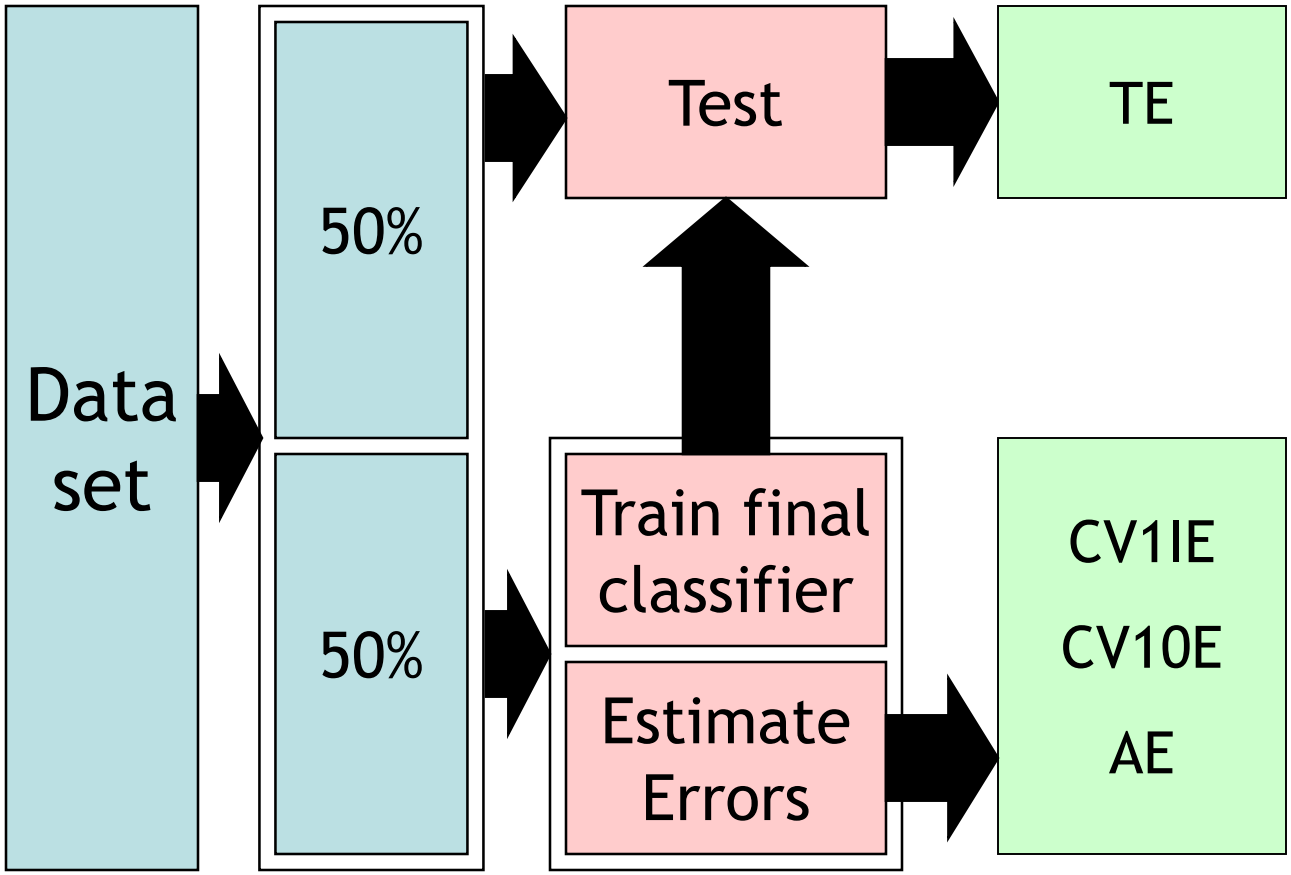
BioSB

Unbiased selection



BioSB

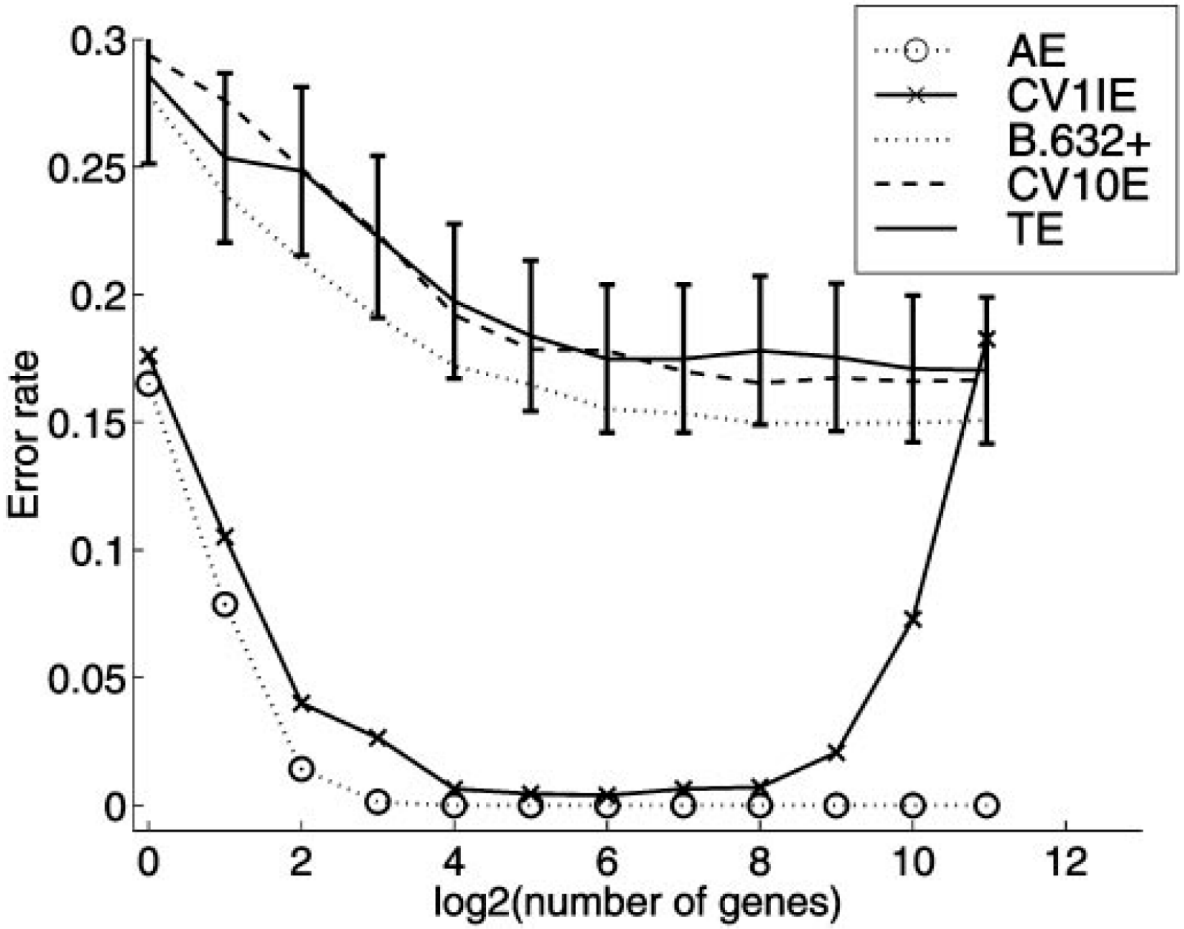
Ambroise & McLachlan experiments



BioSB

Ambroise & McLachlan experiments

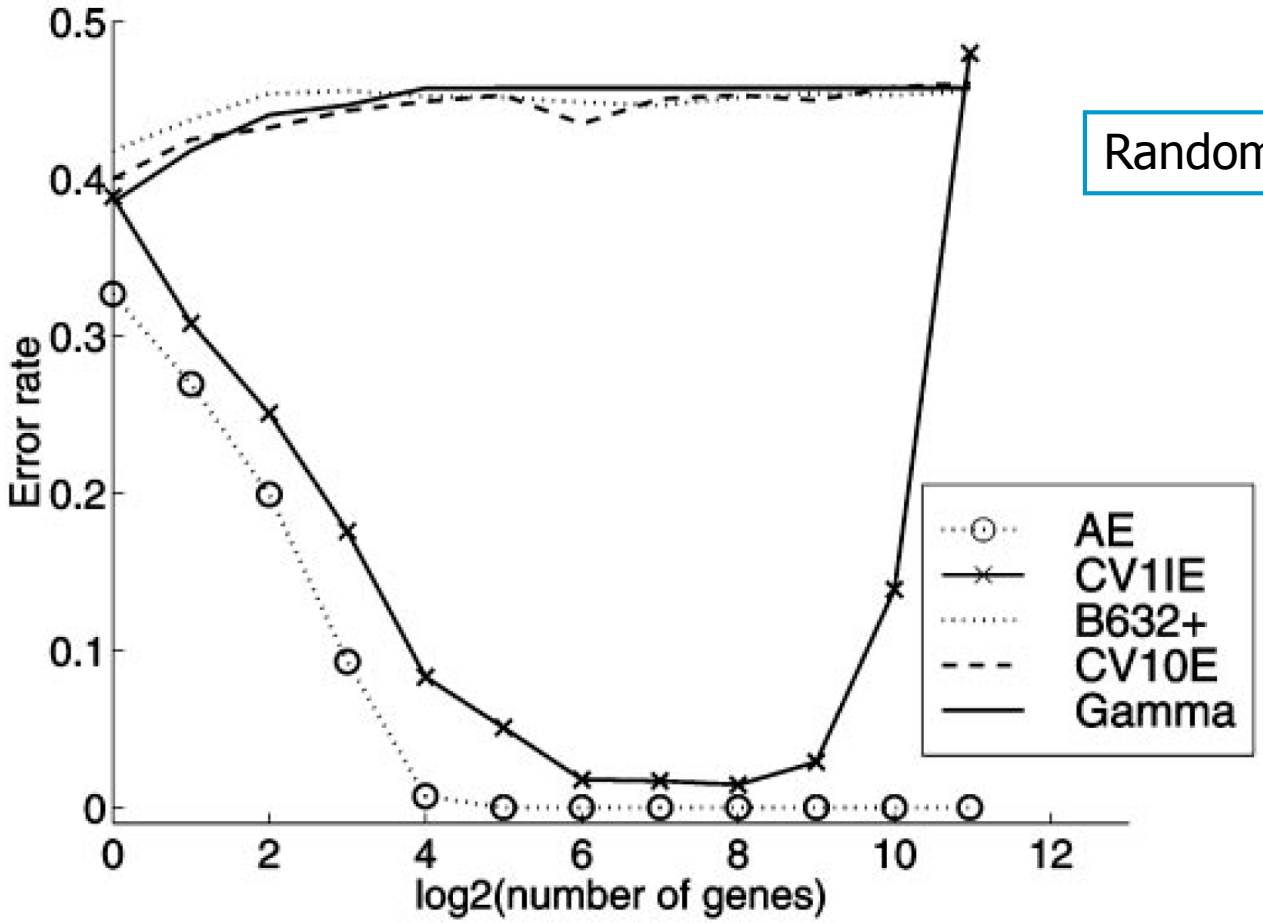
Colon vs. normal data



BioSB

Ambroise & McLachlan experiments

Random data



Random data

Cross-validation

- Remember:

Note:
we should never use the training set to calculate performance; this will give a biased estimate!

  - for small sample size: use cross-validation
- Cross-validation should be applied to *every choice* made, including:
  - the number of features to use
  - the features to use
  - the type of classifier to use
  - …

BioSB

Feature selection: summary

- Feature selection can improve performance and help interpretation
- Requirements: a criterion and a search algorithm
- Methodology (cross-validation) is very important, especially for RNAseq data (‘p >> n’)
- There seems to be some evidence that the simplest methods (individual selection) work best

BioSB

Shrinkage

- Feature selection: selects a subset of features (1/0)
- Feature extraction: combinations of features are constructed based on variance and accuracy arguments
- Regularization 1: 'shave off' genes based on individual quality and control degree of 'shaving' with error
- Regularization 2: combines accuracy (error) and penalty on large weights (= simple models) in one criterion.

BioSB

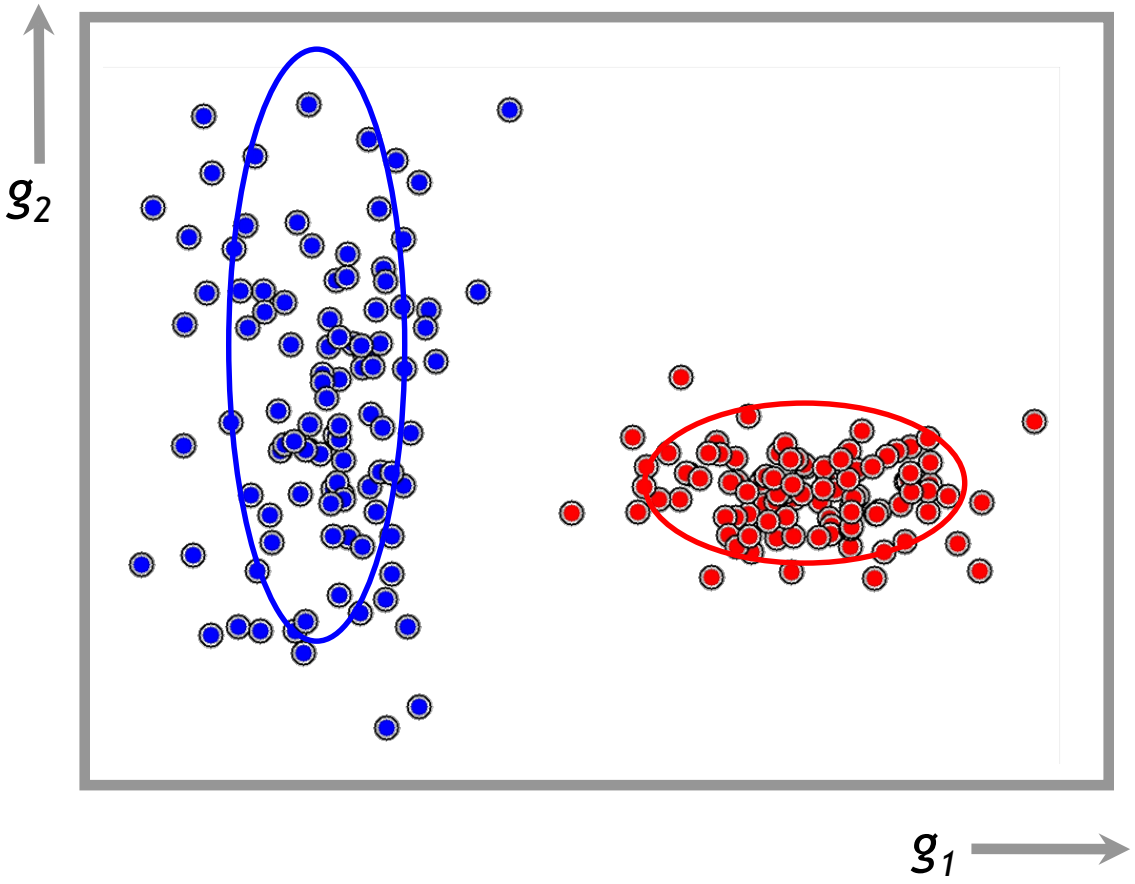
Shrunken centroids

- Same principle as forward filtering
- Genes are evaluated *individually*
- BUT, do not start with the best and keep adding;
- RATHER, start removing worst genes from the back
- In PAM* genes can participate ‘partially’, in forward filtering a gene is either 100% in or out.

* PAM: Prediction analysis of micro-arrays
R. Tibshirani, T. Hastie, B. Narasimhan and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS 99(10):6567 6572, 2002.

BioSB

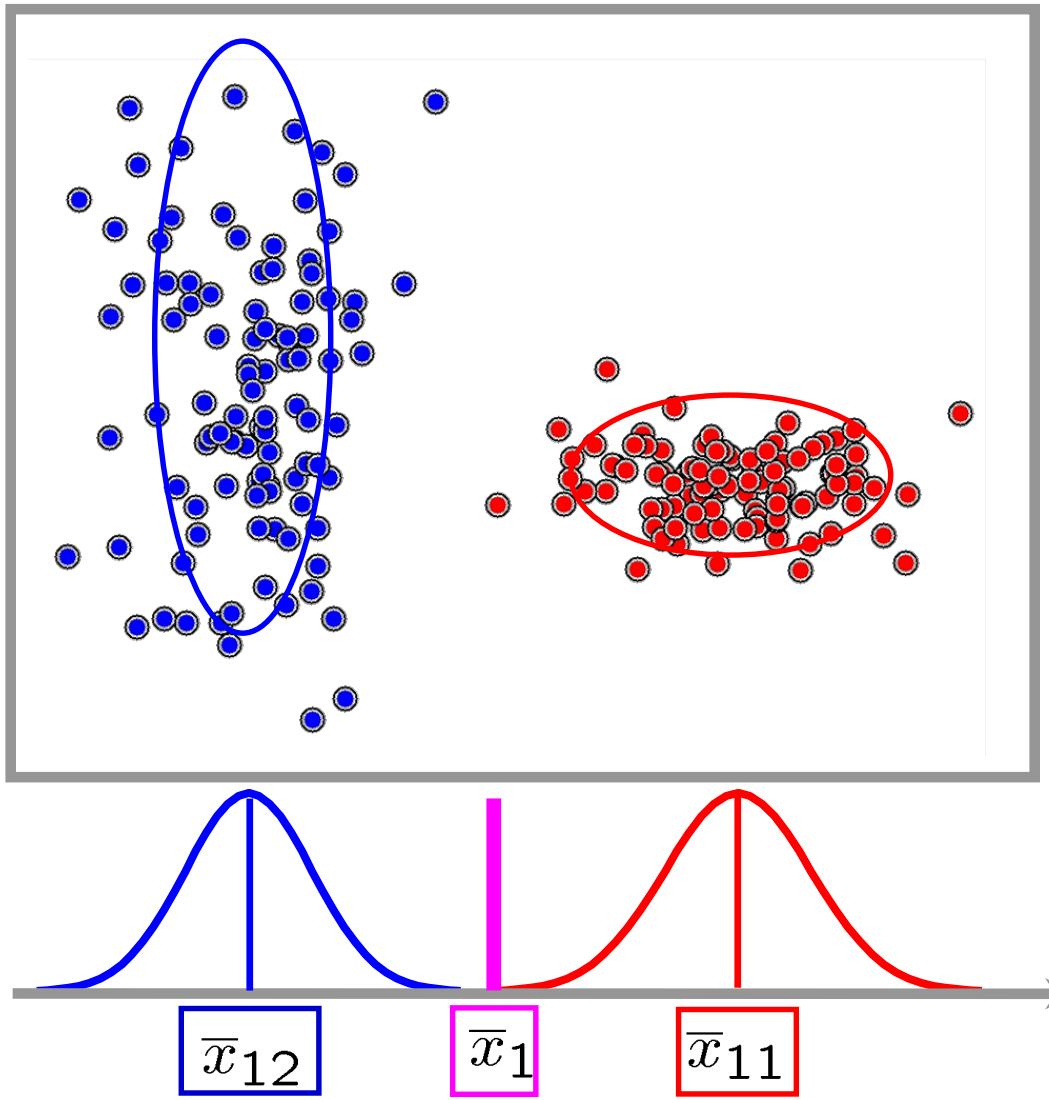
Shrunken centroids in 2D (1)



Which is the best marker gene?

BioSB

Shrunken centroids in 2D (1)

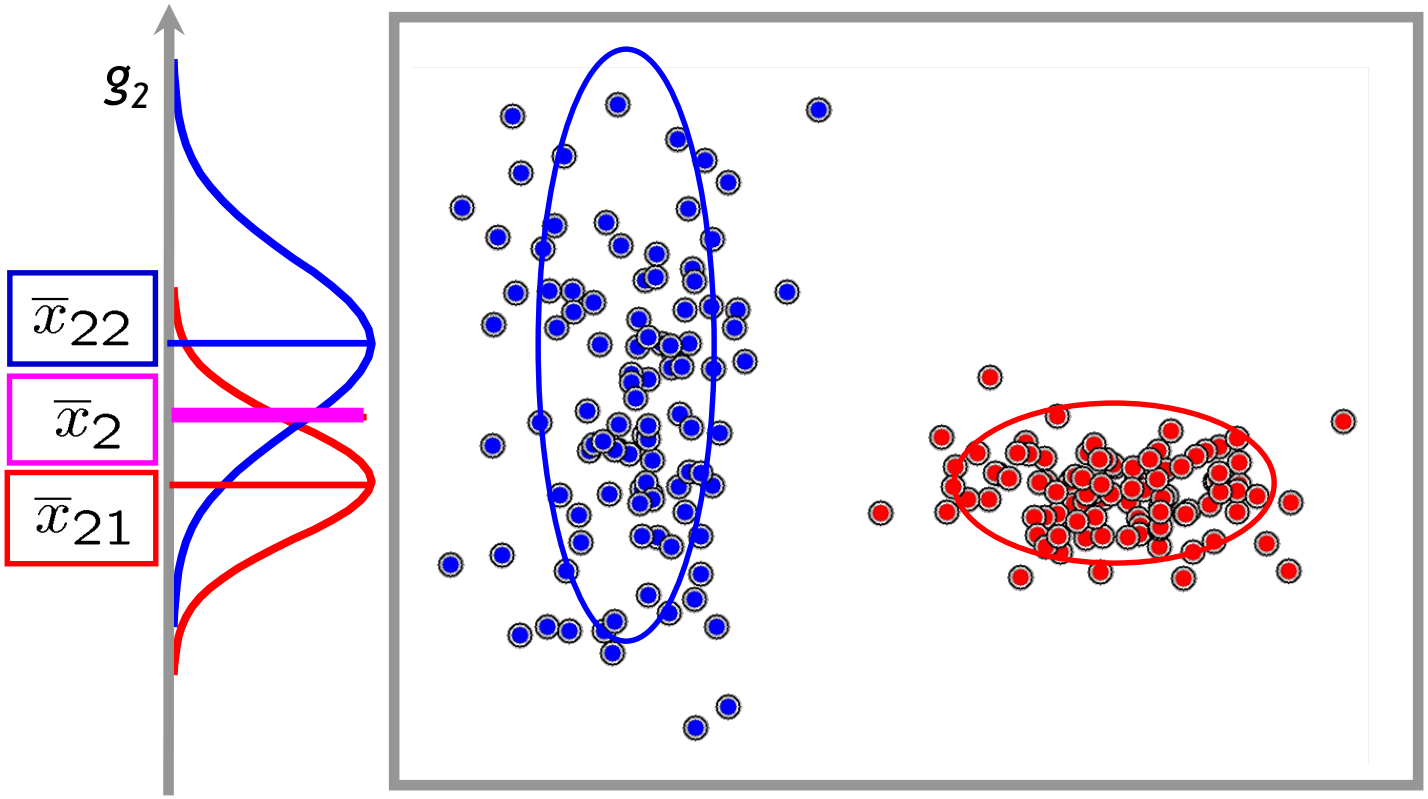


$$\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n$$

$$\overline{x}_{ik} = \sum_{j \in C_k} x_{ij}/n_k$$

BioSB

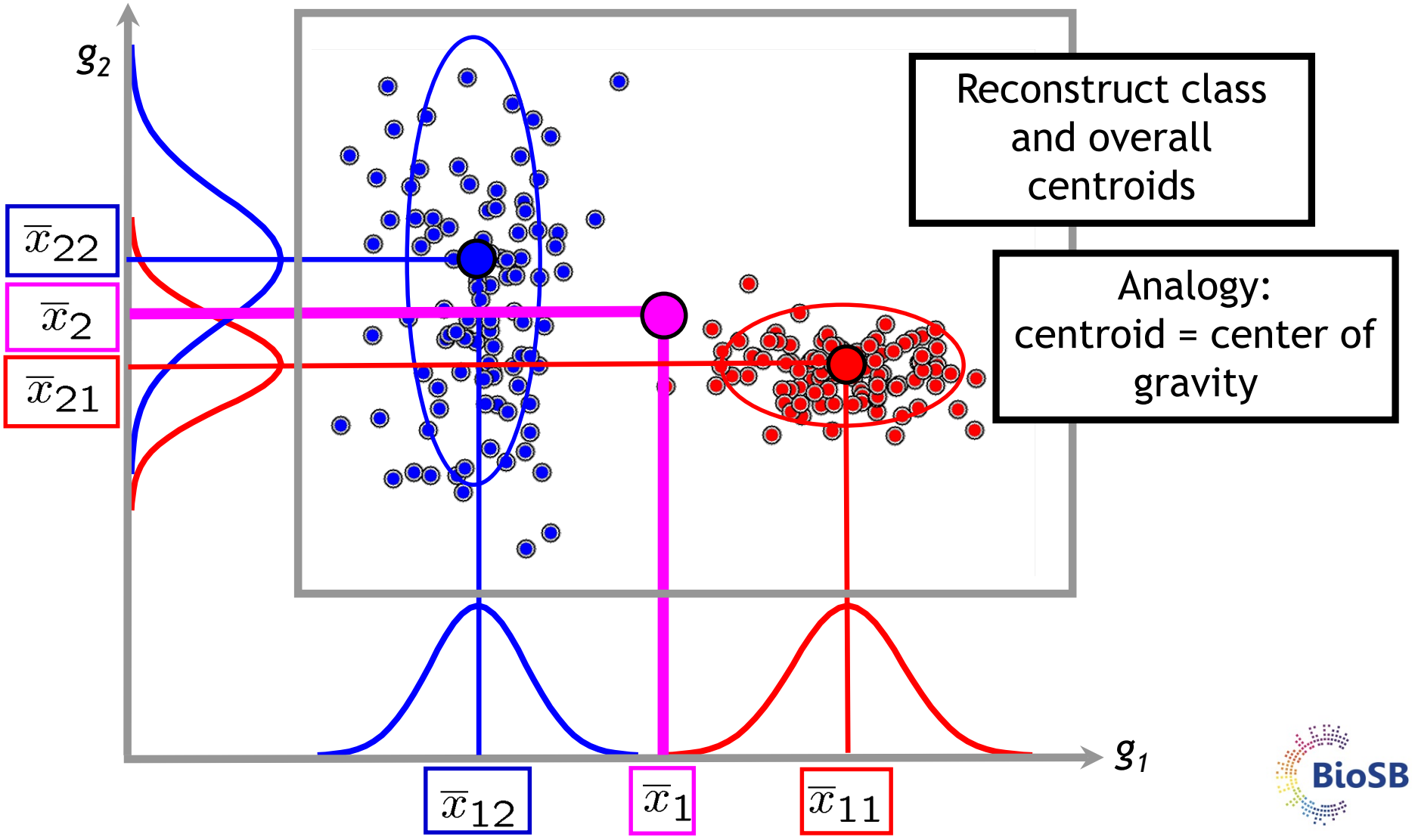
Shrunken centroids in 2D (2)



$$\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n \qquad \overline{x}_{ik} = \sum_{j \in C_k} x_{ij}/n_k$$

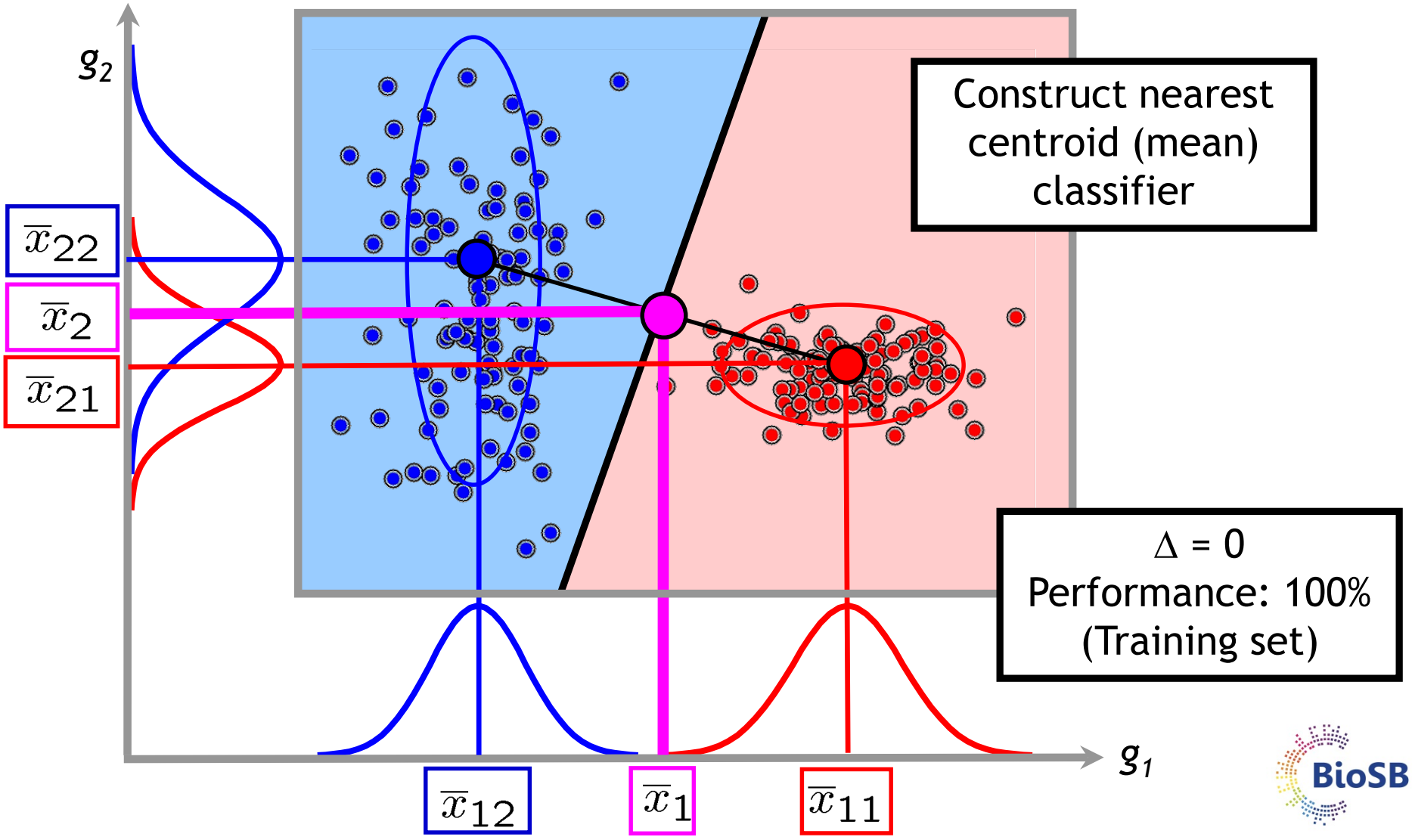
BioSB

Shrunken centroids in 2D (3)



BioSB

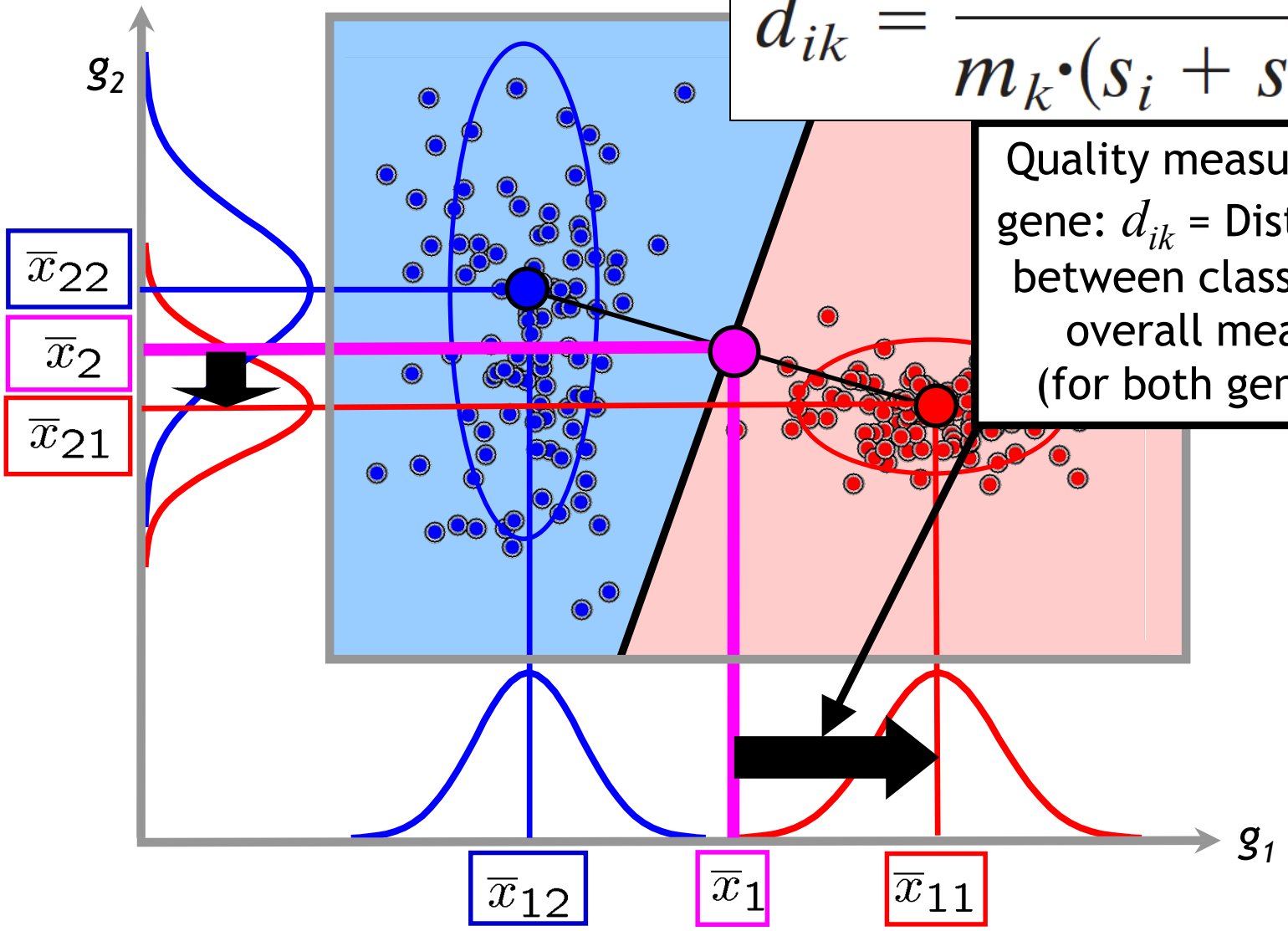
Shrunken centroids in 2D (4)



BioSB

Shrunken centroids in 2D (5)

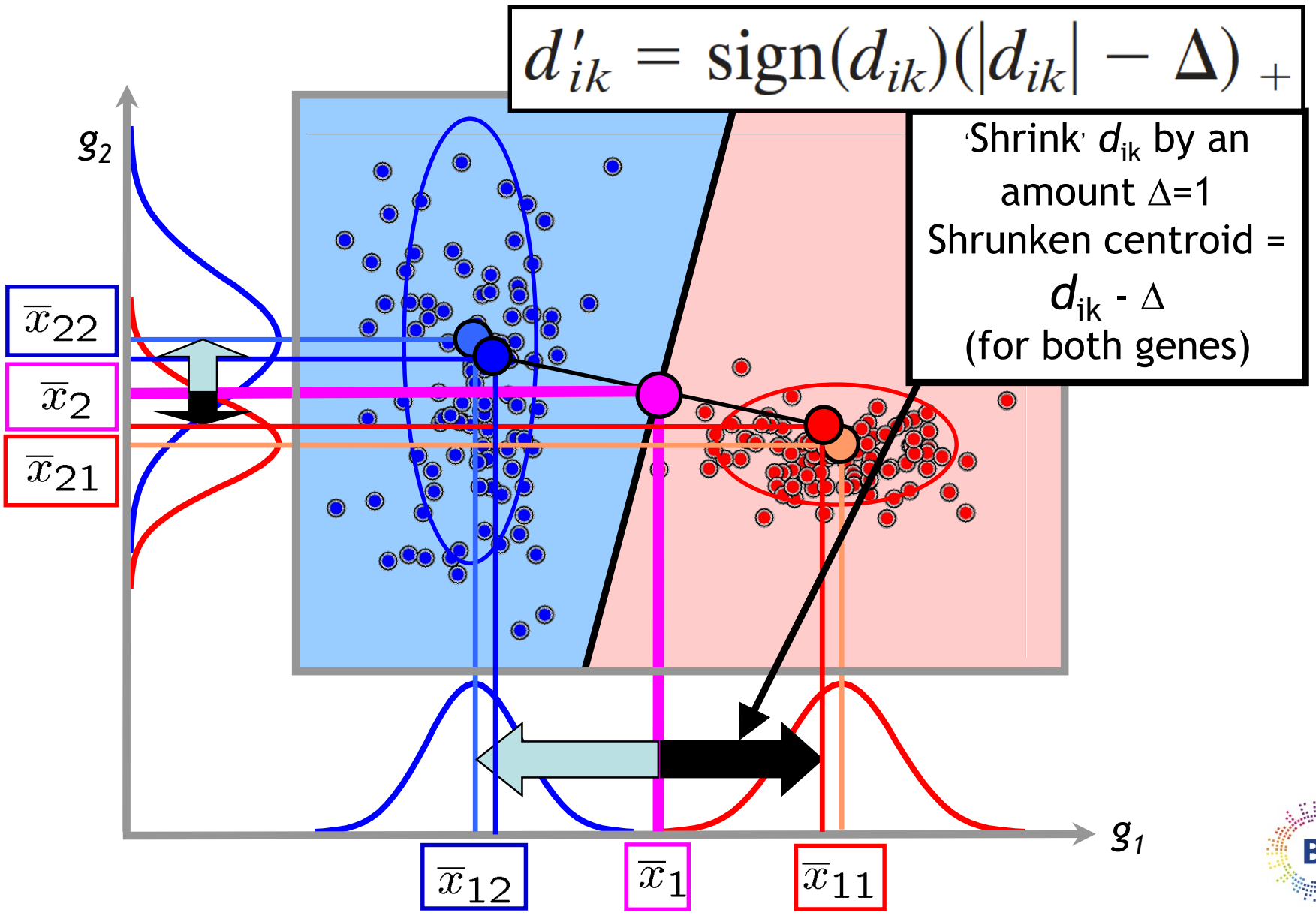
$$d_{ik} = \frac{\bar{x}_{ik} - \bar{x}_i}{m_k \cdot (s_i + s_0)}$$



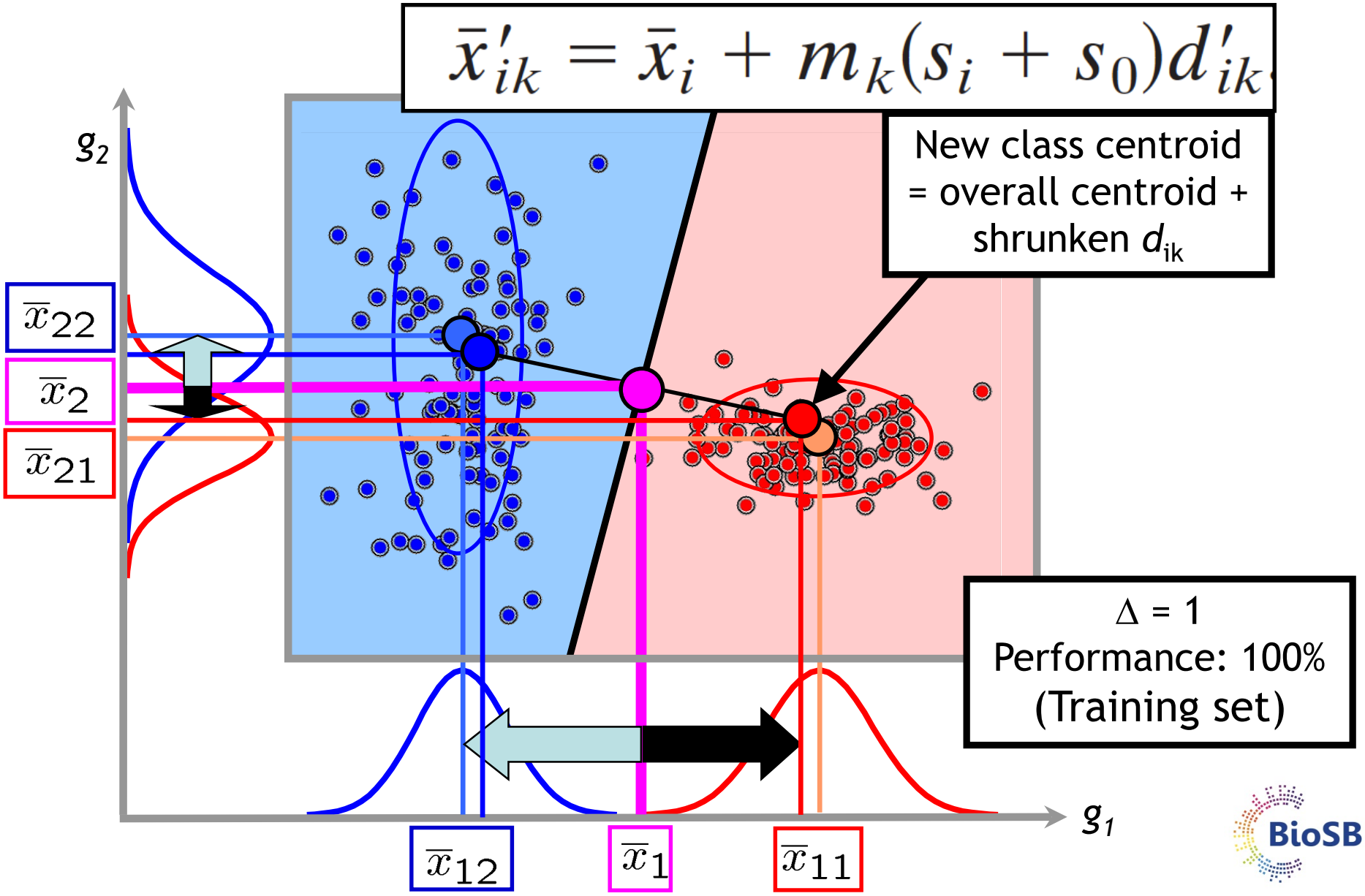
Quality measure of gene: d_{ik} = Distance between class and overall mean (for both genes)

BioSB

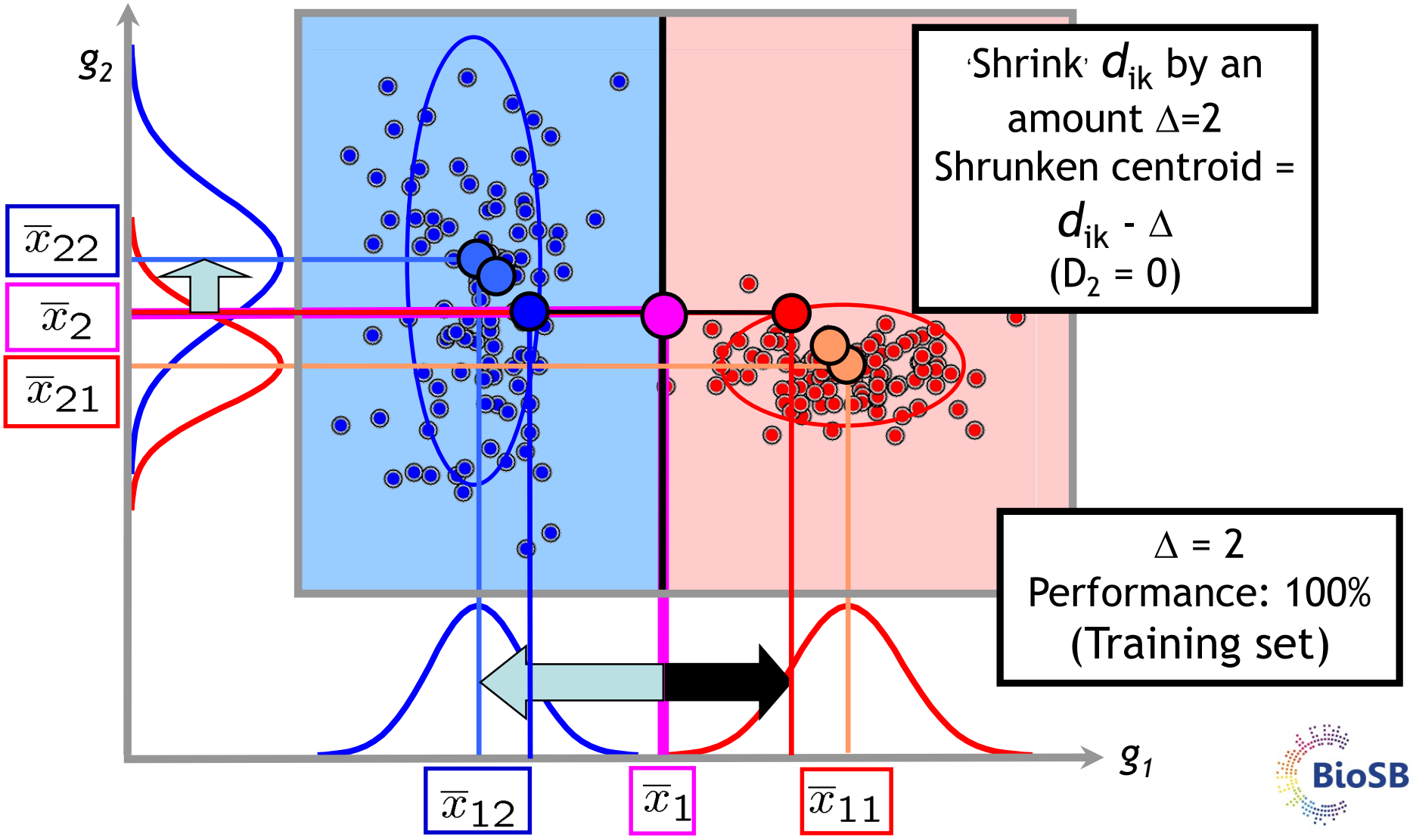
Shrunken centroids in 2D (6)



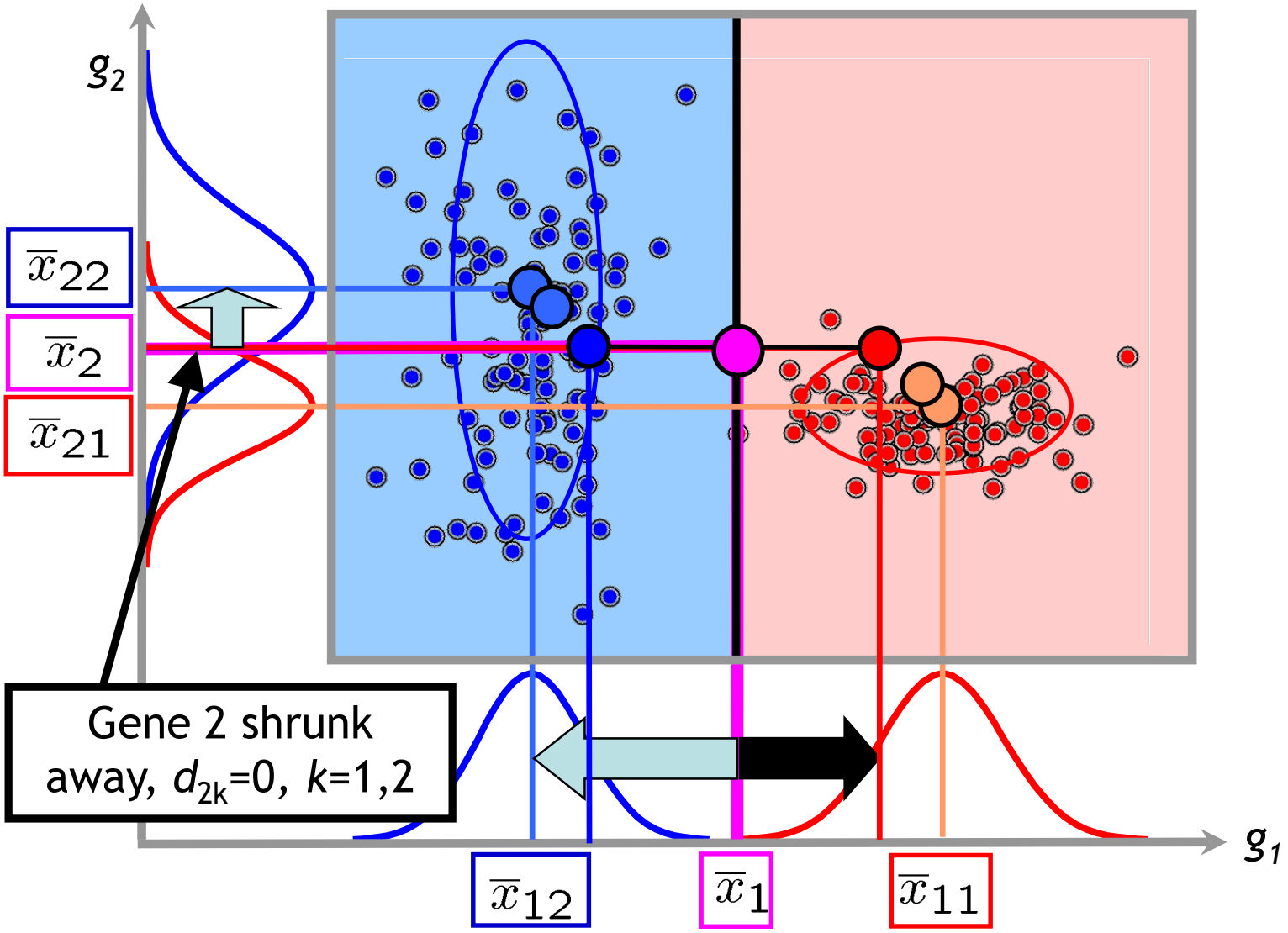
Shrunken centroids in 2D (6)



Shrunken centroids in 2D (7)

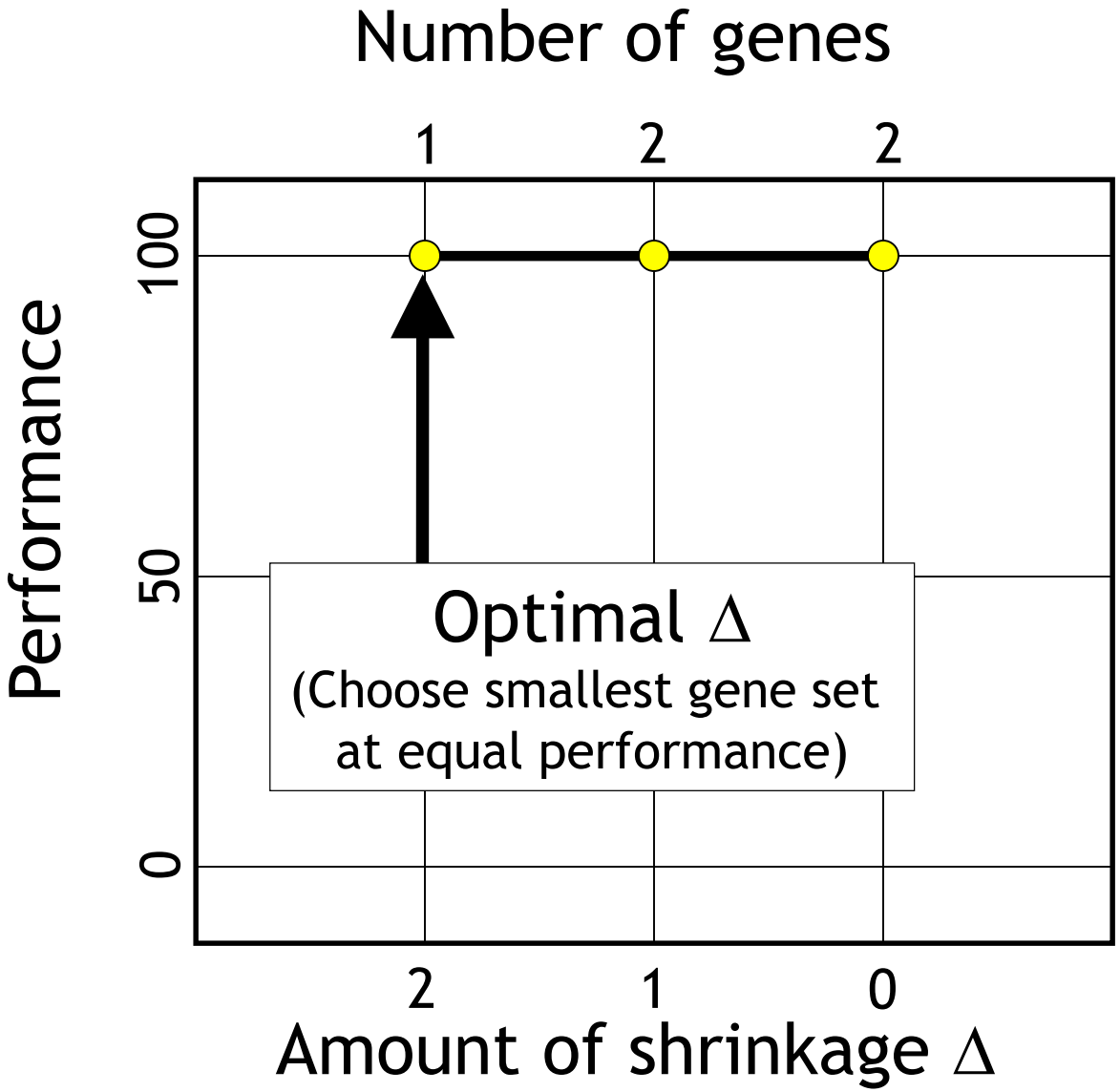


Shrunken centroids in 2D (8)



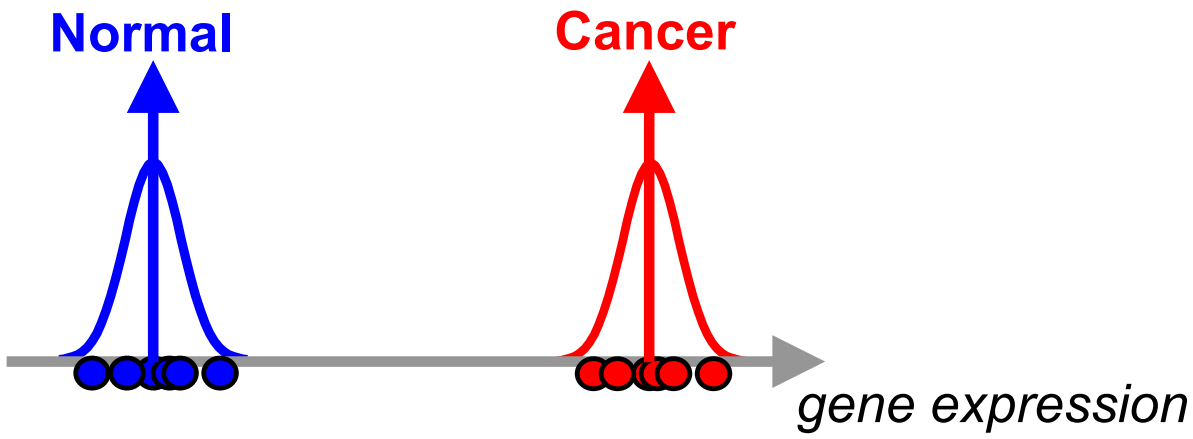
BioSB

Results: number of genes selected

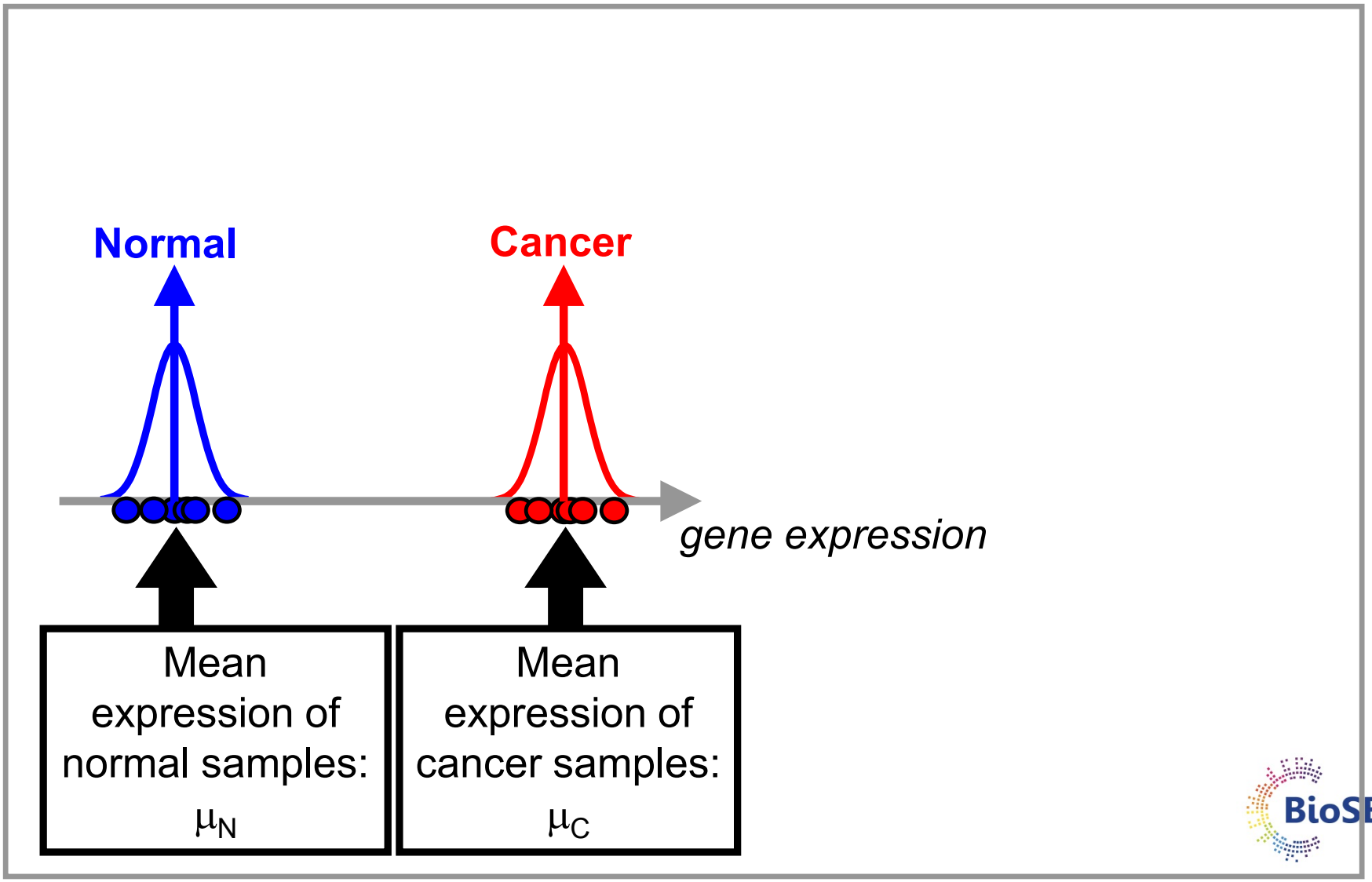


Shrunken centroids: alternative view

Shrunken centroids: D-measure (1)

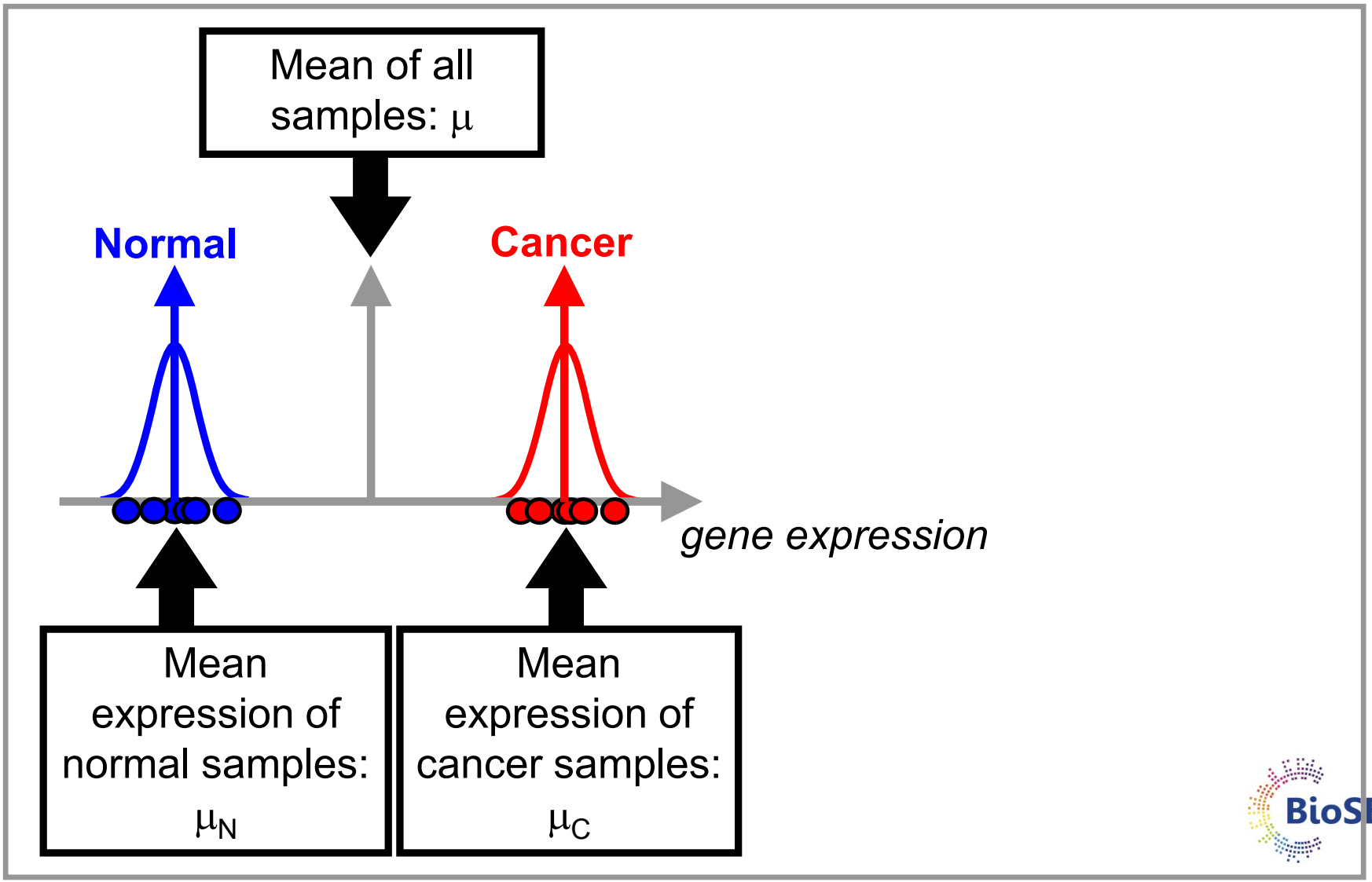


Shrunken centroids: D-measure (2)



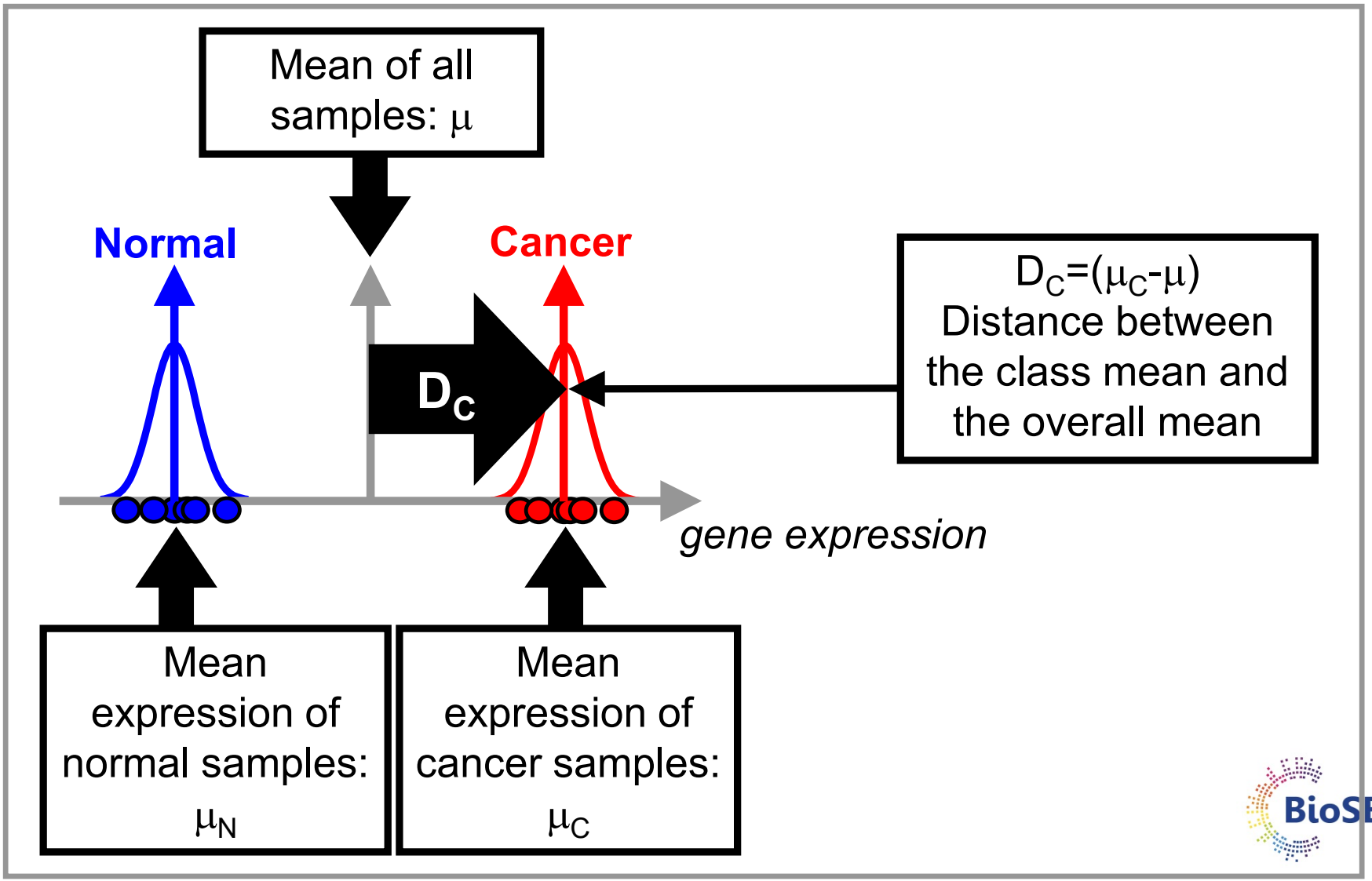
BioSB

Shrunken centroids: D-measure (3)

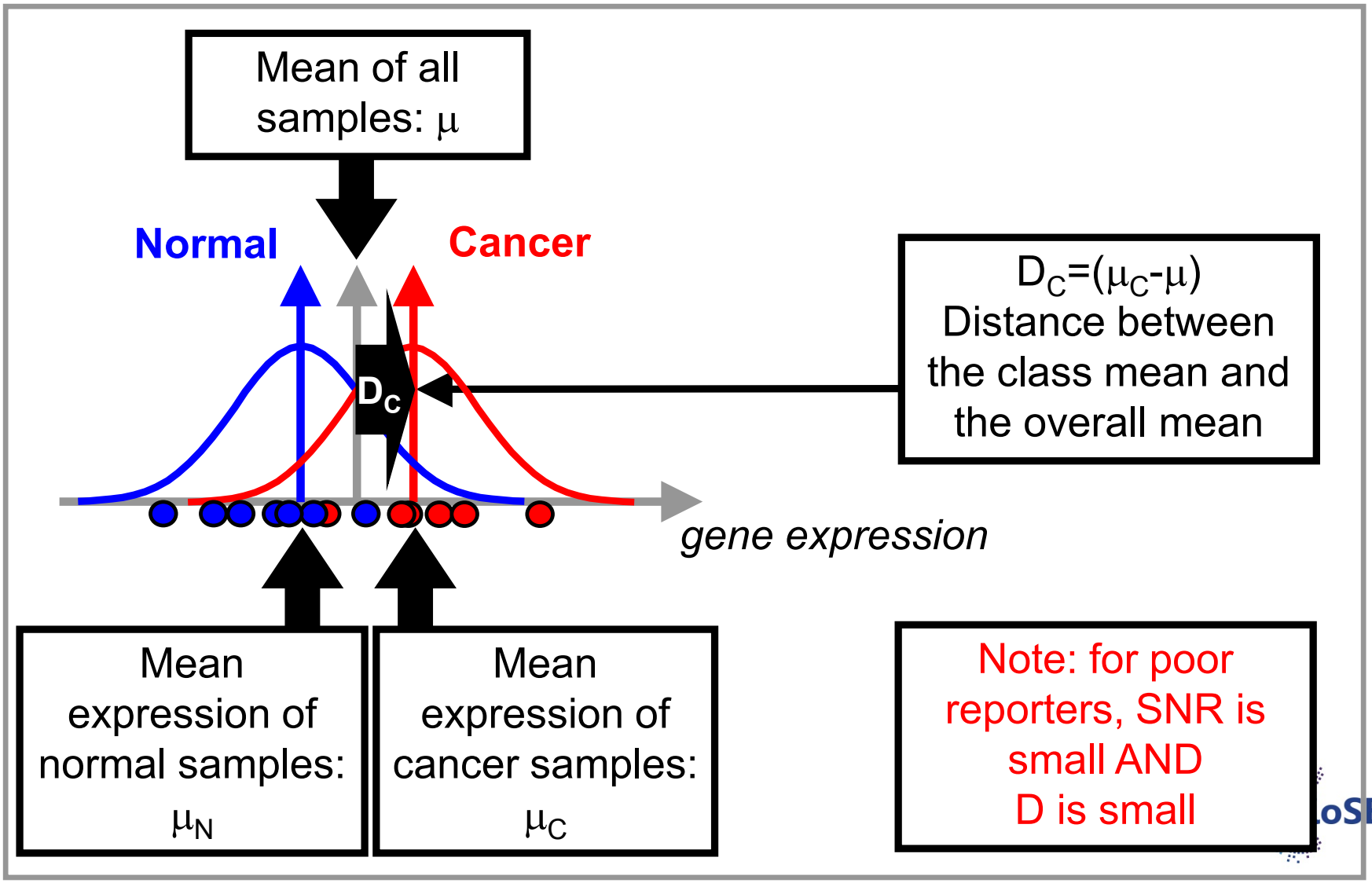


BioSB

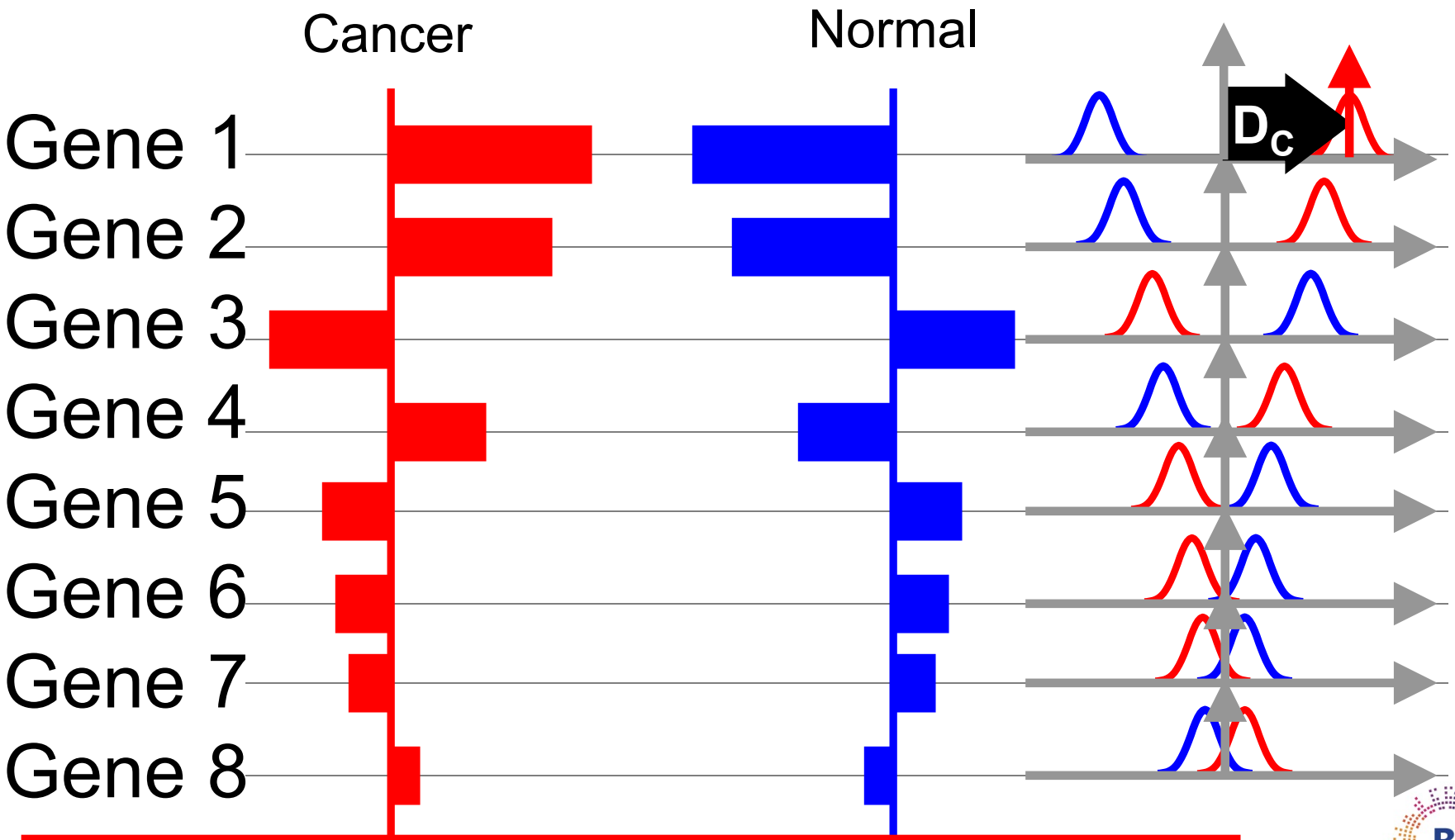
Shrunken centroids: D-measure (4)



Shrunken centroids: D-measure (5)



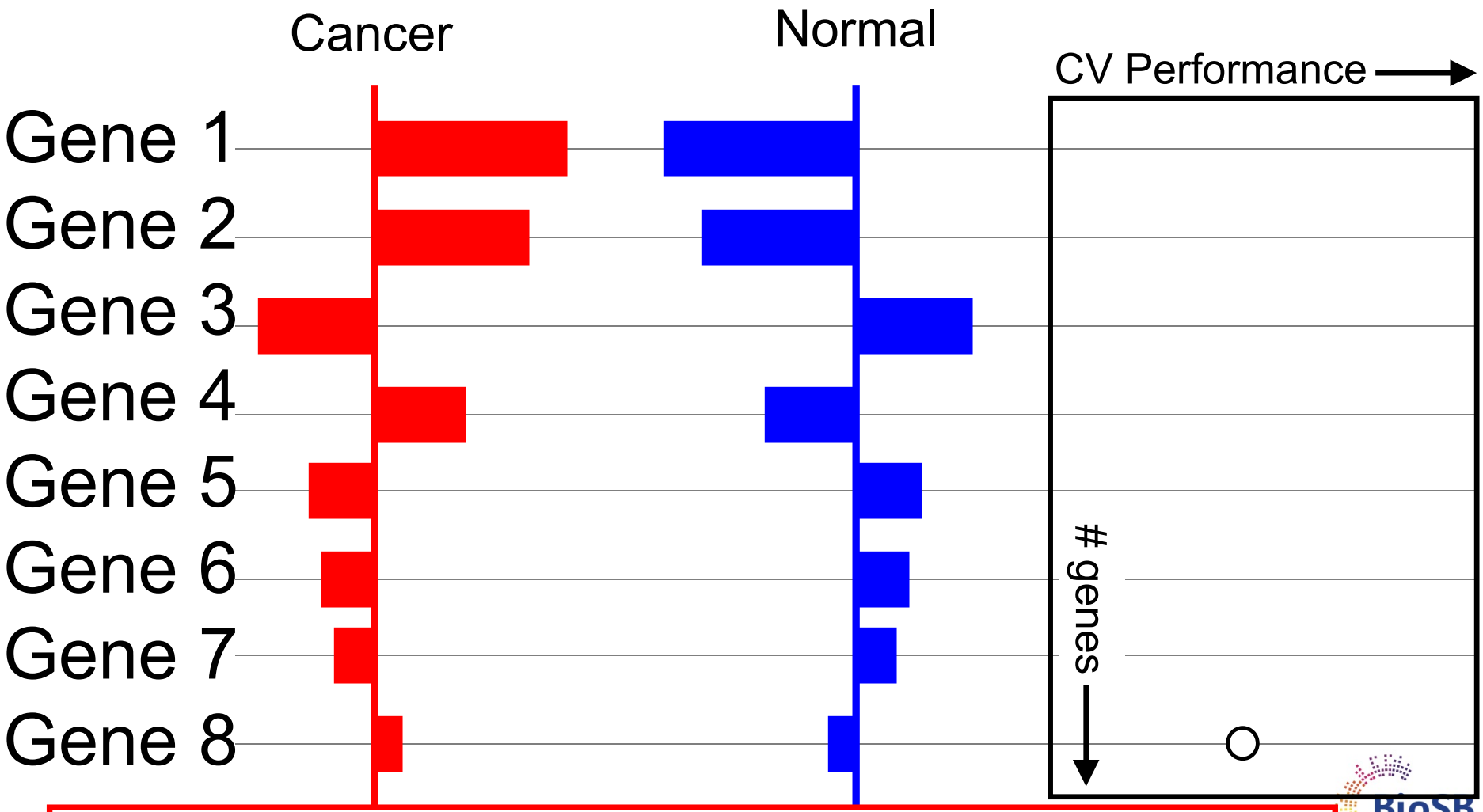
Shrunken centroids: selecting the genes



Genes sorted based on D-measure: best to worse

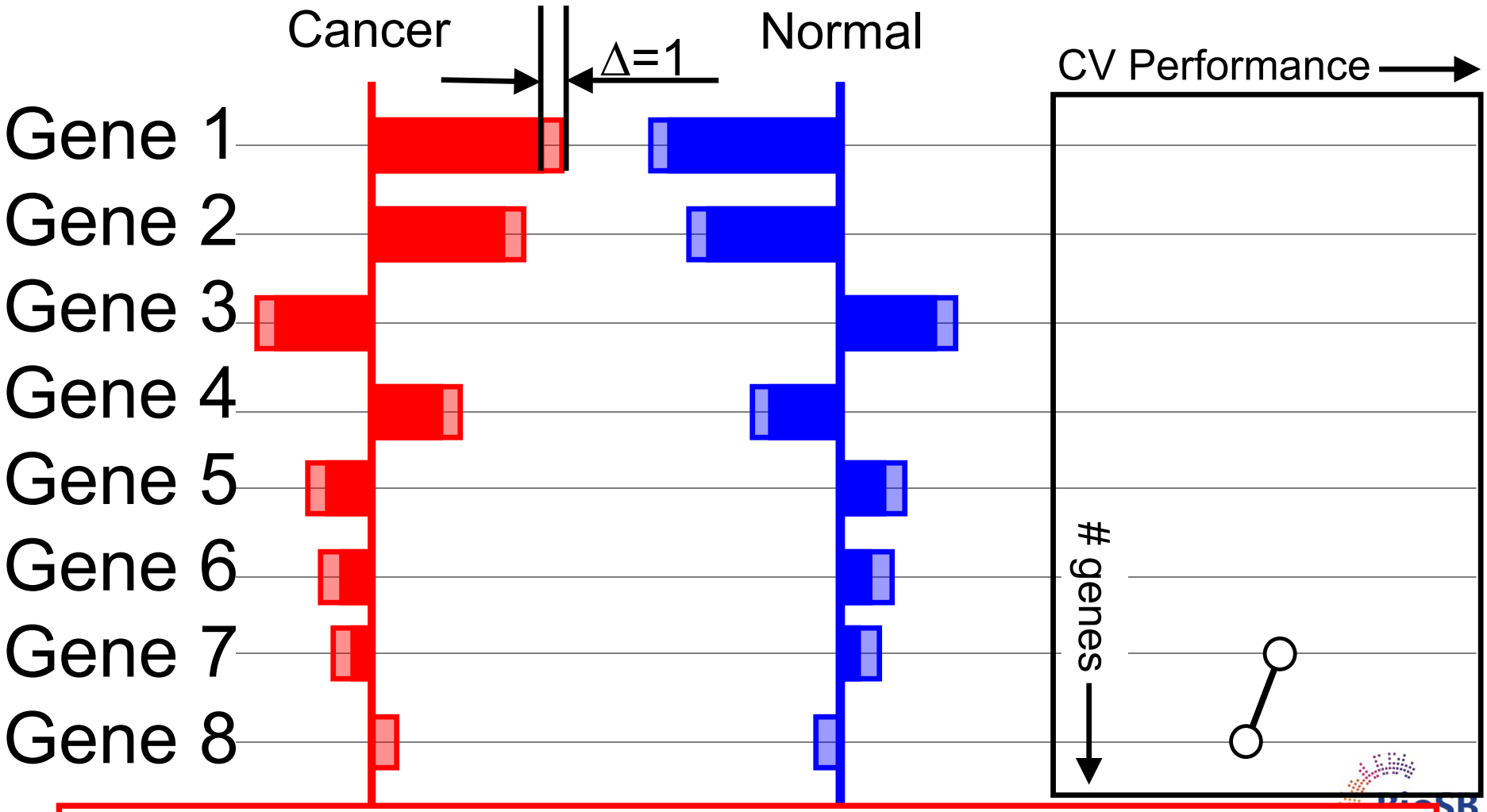
BioSB

Shrunken centroids: selecting the genes



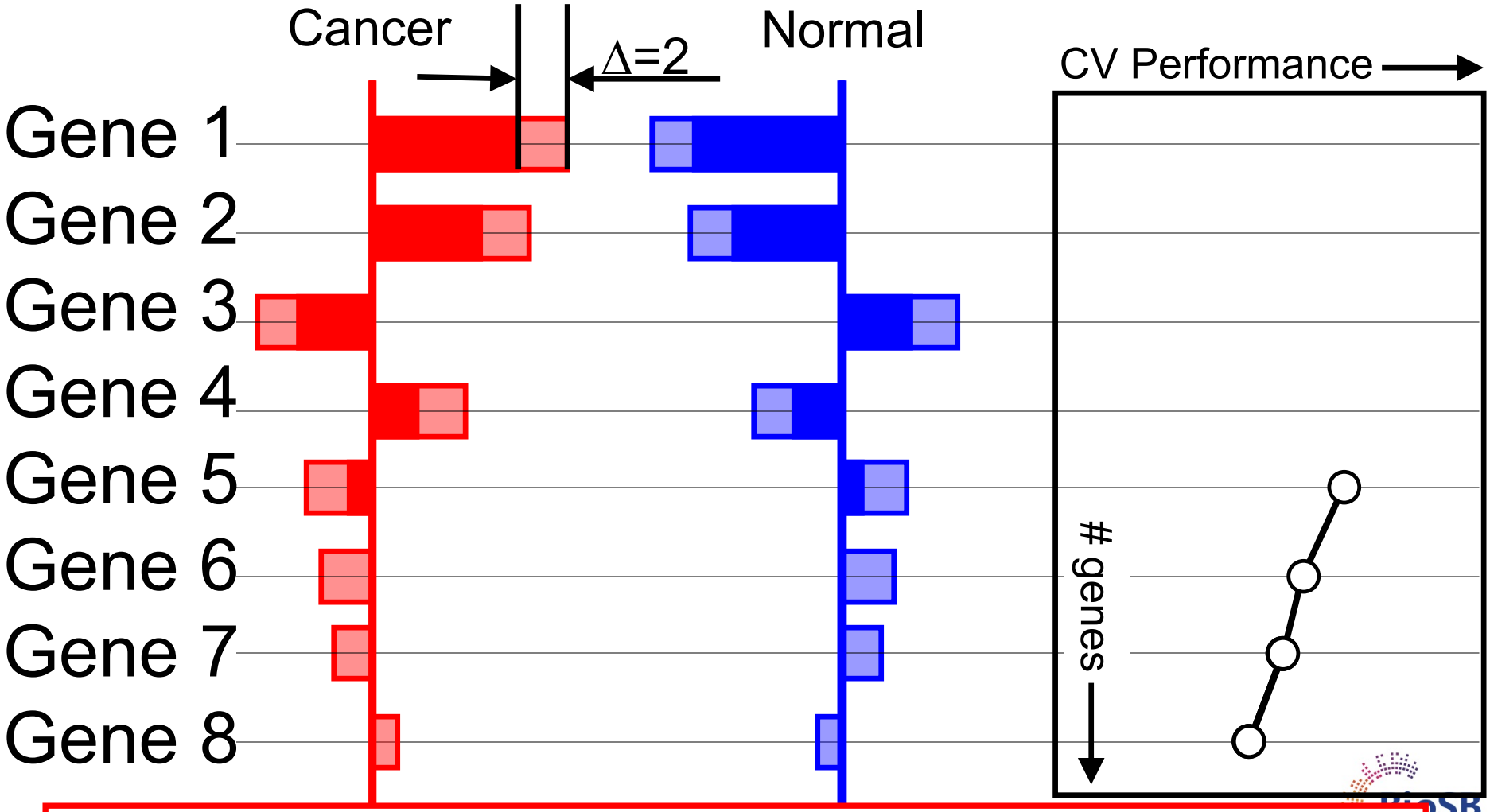
Train classifier on all 8 genes; estimate CV performance

Shrink all D by Δ=1: reduce length by 1



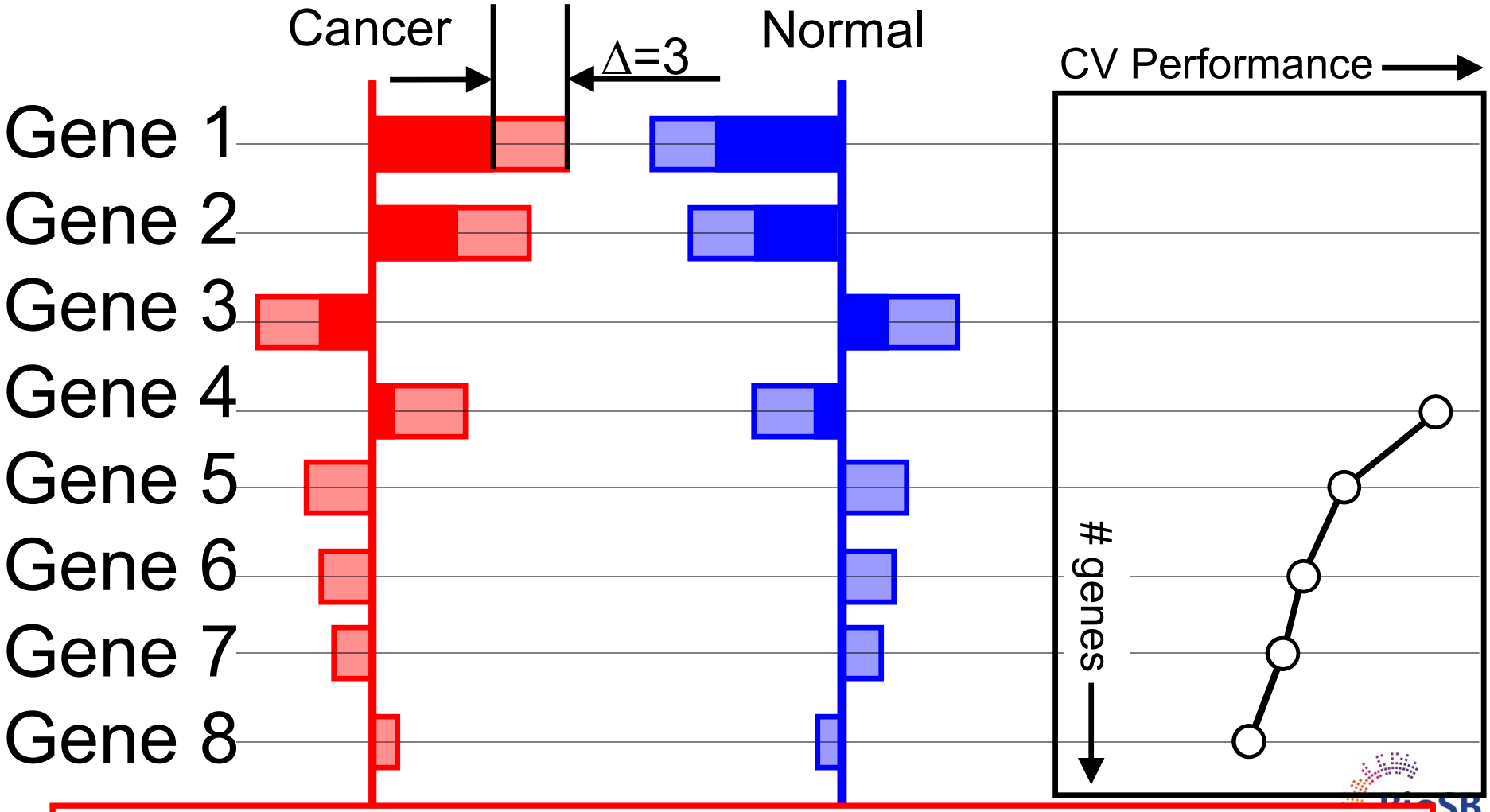
Train classifier on 7 genes (D>0); estimate CV performance

Shrink all D by Δ=2: reduce length by 2



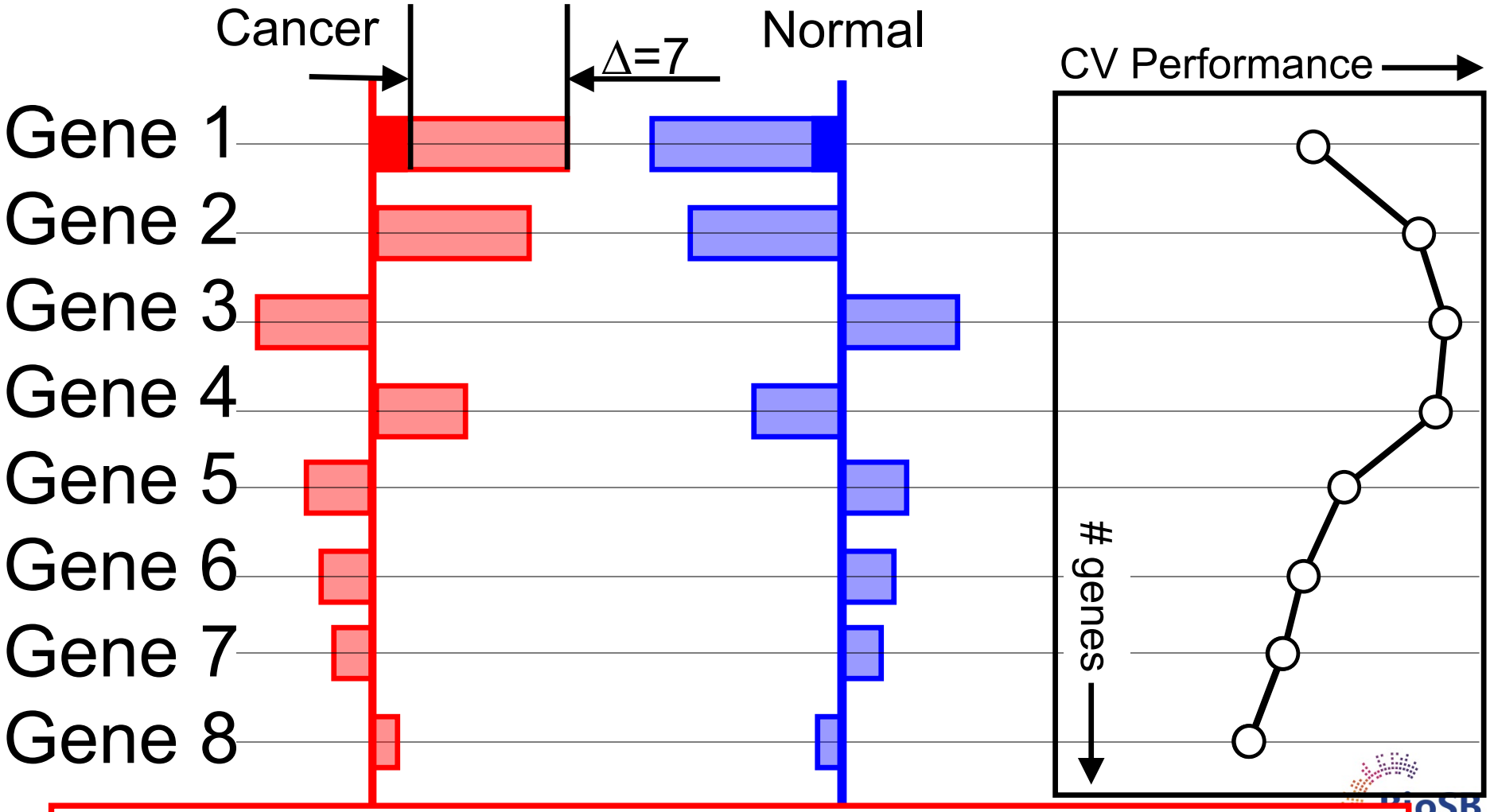
Train classifier on 5 genes (D>0); estimate CV performance

Shrink all D by Δ=3: reduce length by 3



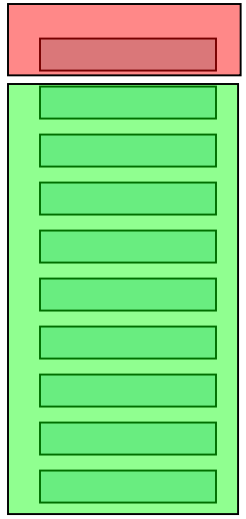
Train classifier on 4 genes (D>0); estimate CV performance

Shrink all D by Δ=7: reduce length by 7

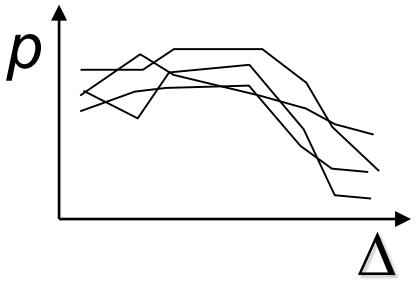


Train classifier on 1 gene (D>0); estimate CV performance

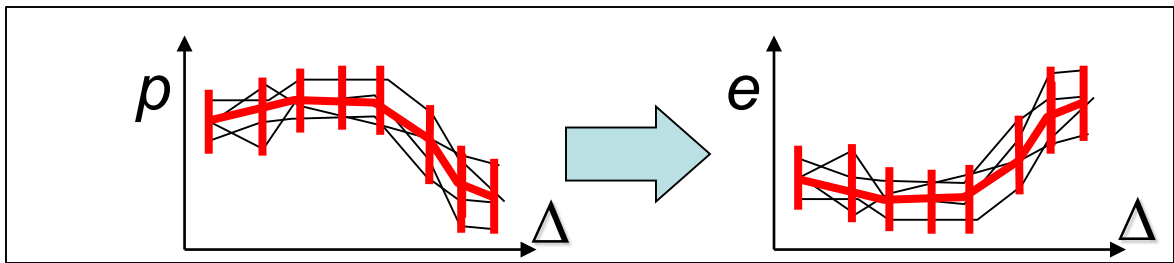
Determining the optimal Δ



1. Split the data (X) in 10 equal parts (x_1,…x_{10})
2. For each of the 10 folds (i=1,2,…,10)
3. On the training set ($X \backslash x_i$)
   1. Compute the class and overall centroids
   2. For a range of Δ (Δ = [0,0.5,…,7])
      i. Shrink D for all genes
      ii. Compute 'shrunken centroids' on training set
      iii. Test the resulting classifier on the test set (x_i)
   3. Result: 10 Curves of performance vs. Δ

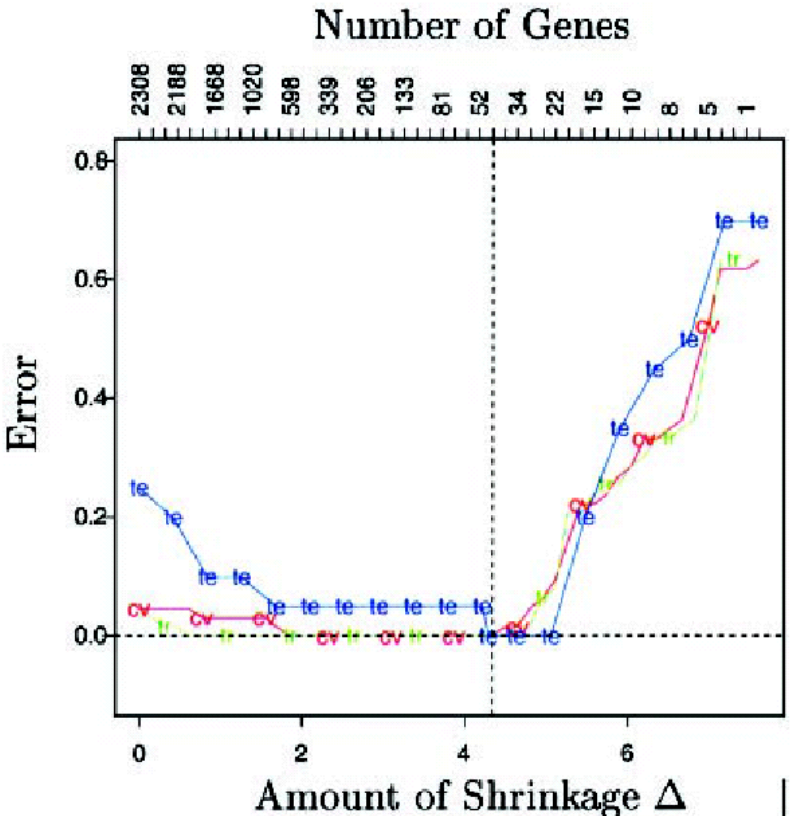


4. Average all 10 curves and compute std. dev. at each Δ
5. Pick the Δ where the performance is maximal (error min.)



PAM

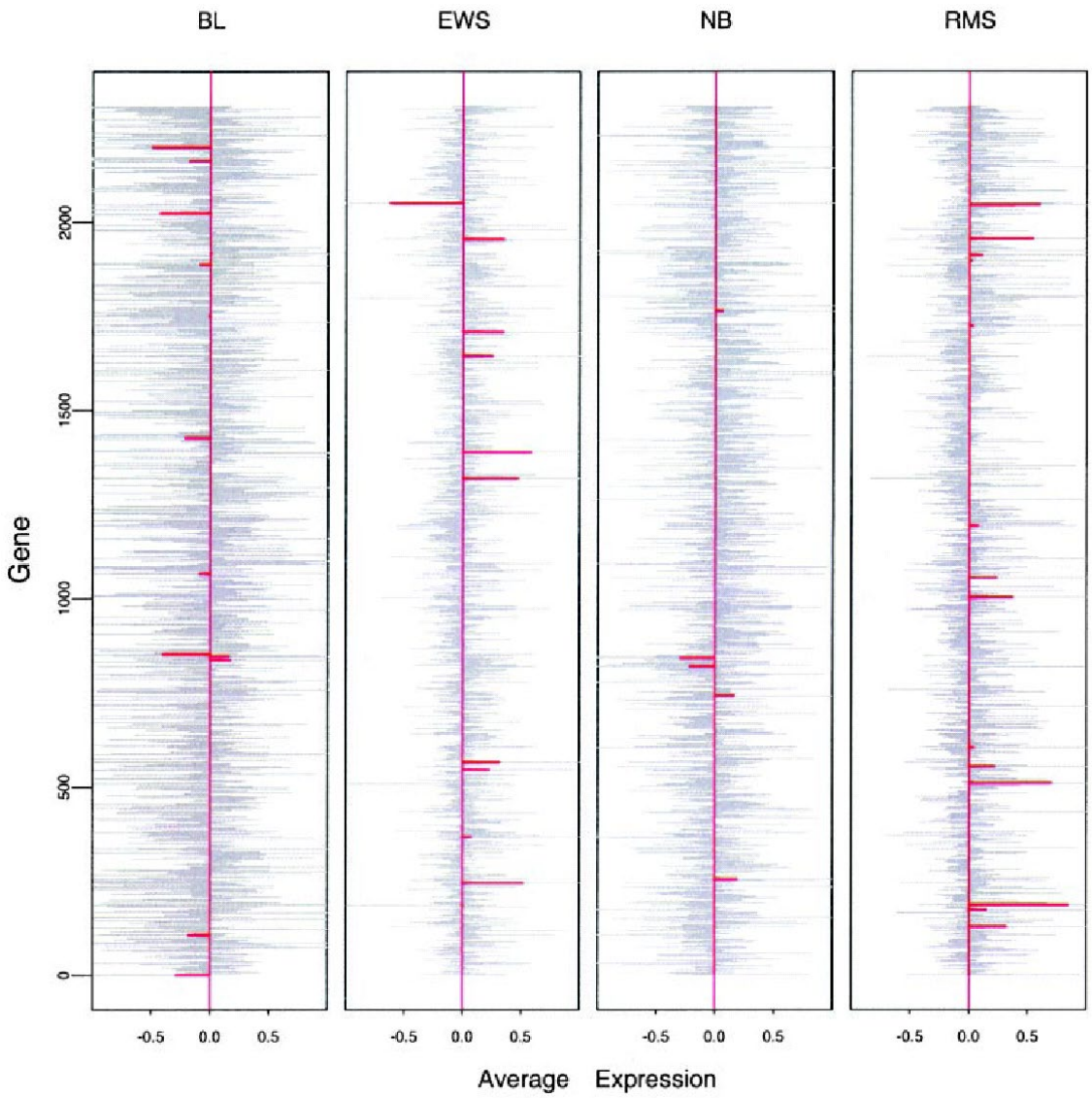
- For the Khan datatset; 4 classes: BL, EWS, NB, RMS
- At optimal Δ : 43 genes *not* shrunk away



Neuroblastoma (NB)
Rhabdomyosarcoma (RMS)
Burkitt lymphoma (BL)
Ewing family of tumors (EWS),

R. Tibshirani *et al.* (2002) PNAS 99(10):6567-6572, 2002.

PAM (2)



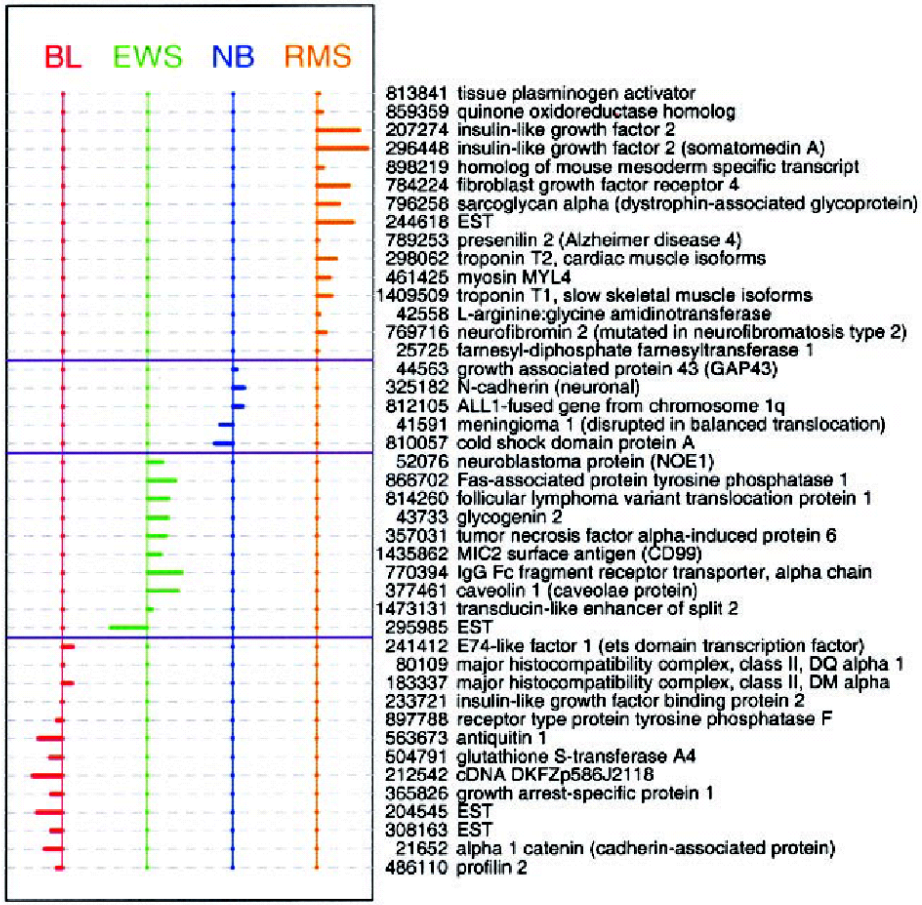
shrunk

unshrunk

BioSB

PAM (3)

At optimal Δ : 43 genes *not* shrunk away



- Neuroblastoma (NB)
- Rhabdomyosarcoma (RMS)
- Burkitt lymphoma (BL)
- Ewing family of tumors (EWS),

R. Tibshirani *et al*. (2002) PNAS 99(10):6567-6572, 2002.

BioSB

PAM (4)

- For a new sample x* (assuming normal class conditionals and applying Bayes' rule):

$$C(l^*) = l, \text{where } \delta_l(x^*) = \min_k(\delta_k(x^*))$$

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}'_{ik})^2}{(s_i + s_0)^2} - 2\log(\pi_k)$$

Discriminant for class *k*

PAM (4)

- For a new sample x* (assuming normal class conditionals and applying Bayes' rule):

$$C(l^*) = l, \text{where } \delta_l(x^*) = \min_k(\delta_k(x^*))$$

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}'_{ik})^2}{(s_i + s_0)^2} - 2\log(\pi_k) \qquad \left[\sum_{k=1}^{K} \pi_k = 1 \right]$$

Compensate for prior prob.

Discriminant for class *k*

BioSB

PAM (4)

- For a new sample x* (assuming normal class conditionals and applying Bayes' rule):

$$C(l^*) = l, \text{where } \delta_l(x^*) = \min_k(\delta_k(x^*))$$

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}'_{ik})^2}{(s_i + s_0)^2} - 2\log(\pi_k) \qquad \left[\sum_{k=1}^{K} \pi_k = 1 \right]$$

Discriminant for class *k*

Squared distances to (shrunken) Centroid of class *k*

Compensate for prior prob.

PAM (4)

- For a new sample x* (assuming normal class conditionals and applying Bayes' rule):

$$C(l^*) = l, \text{where } \delta_l(x^*) = \min_k(\delta_k(x^*))$$

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}'_{ik})^2}{(s_i + s_0)^2} - 2\log(\pi_k) \qquad \left[\sum_{k=1}^{K} \pi_k = 1 \right]$$

Discriminant for class *k*

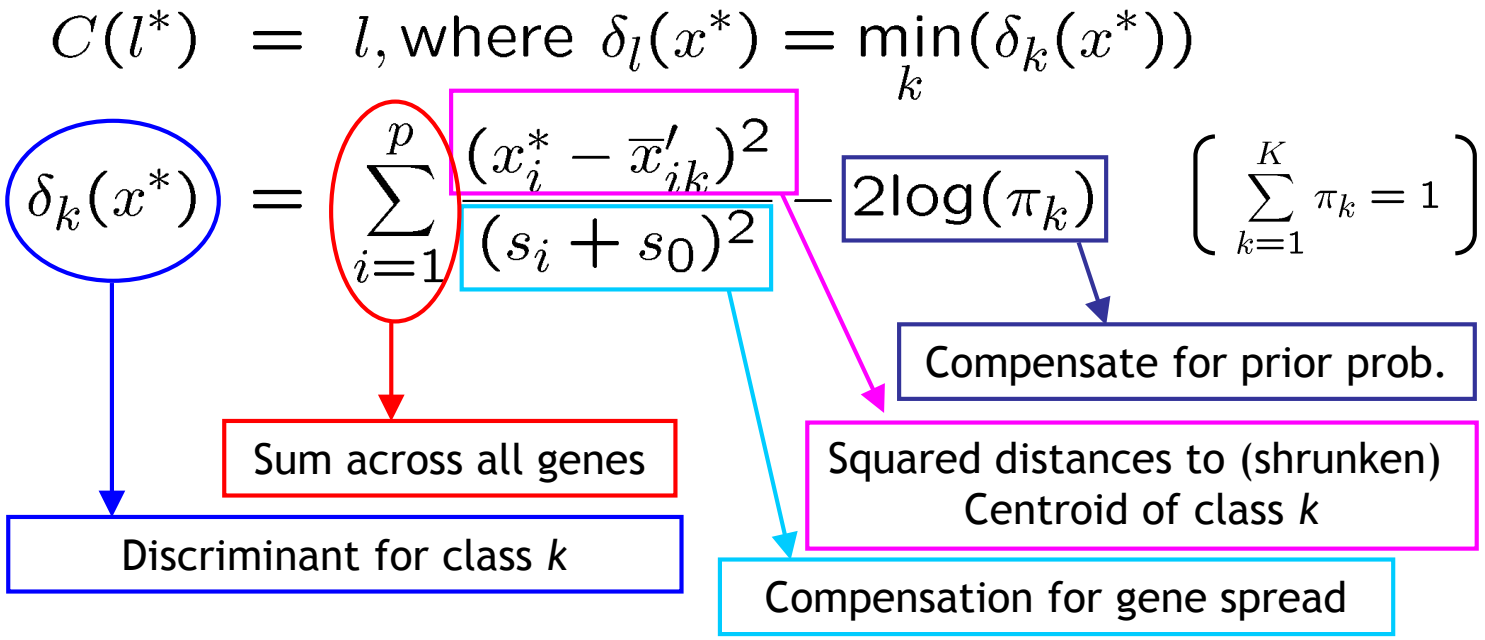
Compensate for prior prob.

Squared distances to (shrunken) Centroid of class *k*

Compensation for gene spread

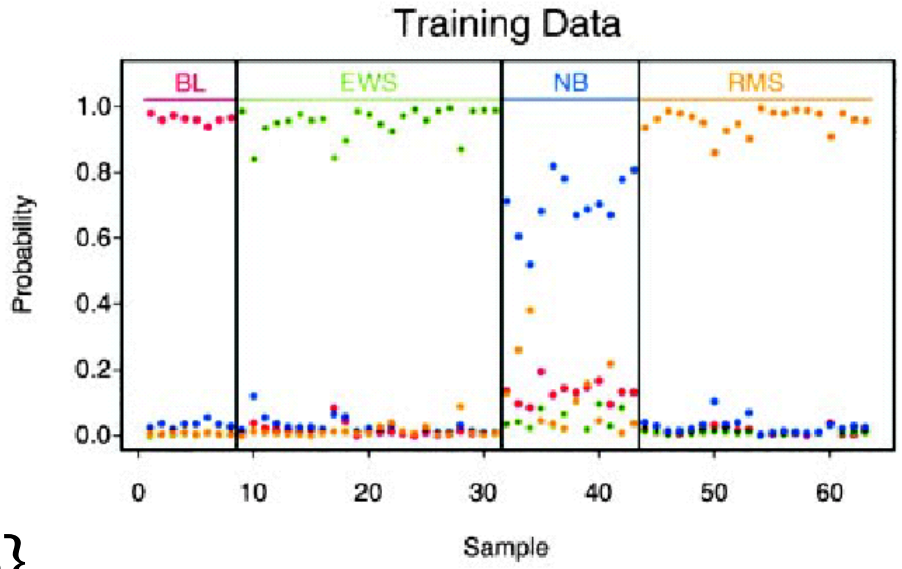
PAM (4)

- For a new sample x* (assuming normal class conditionals and applying Bayes' rule):



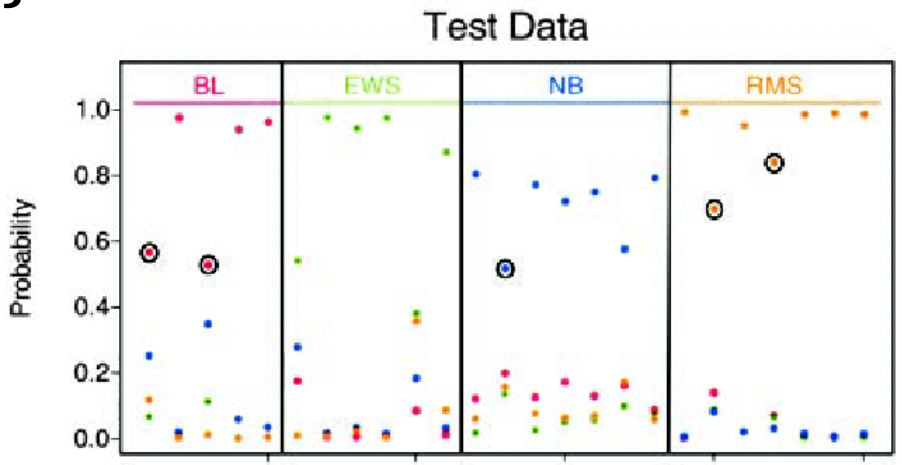
Scoring samples by posterior prob's

$$\hat{p}(k|x^*)$$



k = {BL,EWS,NB,RMS}

$$\hat{p}(k|x^*)$$



BioSB

Shrinkage

- PAM determines a weight for every gene based on the predictive capacity of the gene (type of t-statistic)
- This weight determines role of the gene in a DLDA classifier (weights can be zero, i.e. no participation)
- Weights of all genes are shrunk by the same amount (Δ)
- PAM computes effect of shrinkage on error rate, and chooses shrinkage (=number of genes) with lowest error
- PAM: implicit simultaneous error and complexity minimisation
- Other approach: regularisation, combine error and penalty for number of genes explicitly

BioSB

Shrinkage (2)

- Model: $y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \varepsilon$

- Penalised (*aka* regularised) least squares:
  - Ridge regression:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[\sum_{j=1}^{n} \left(y_j - \beta_0 - \sum_{i=1}^{p} \beta_i x_{j,i} \right)^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \right]$$

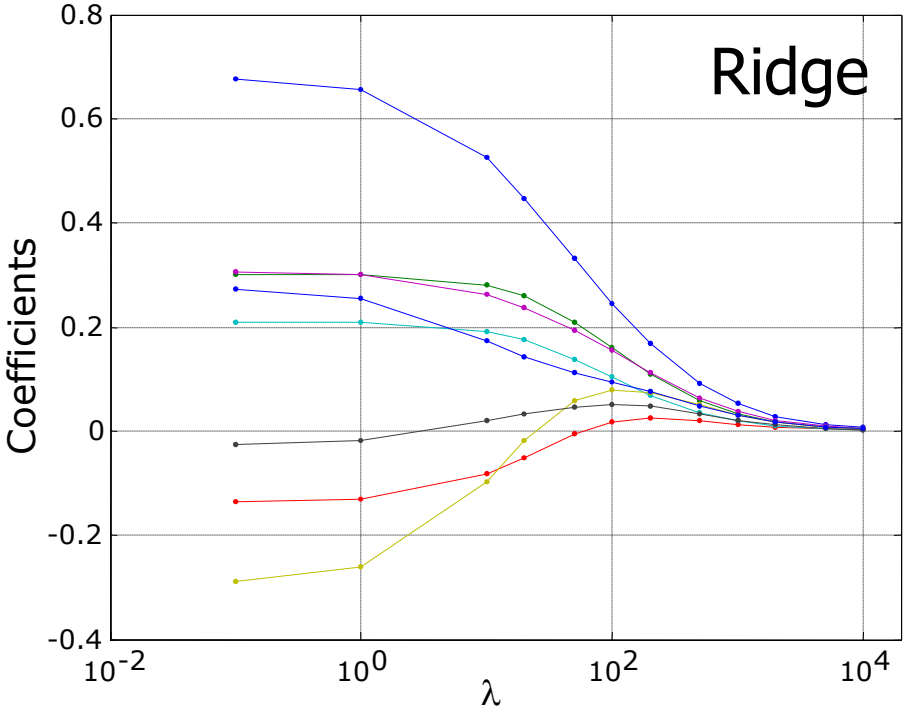
  - LASSO: minimise

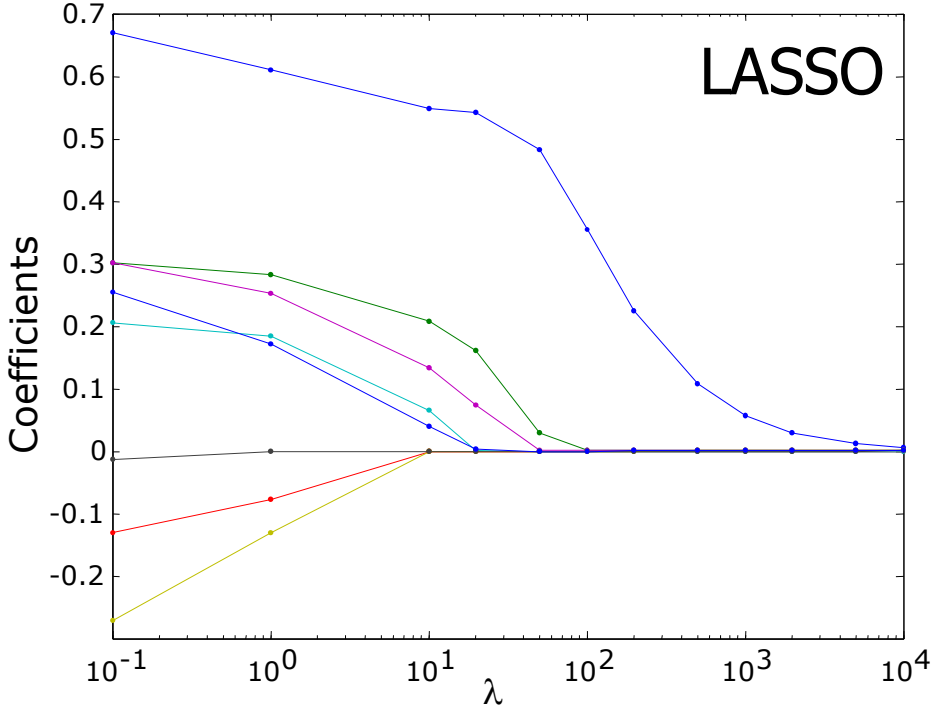
$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[\sum_{j=1}^{n} \left(y_j - \beta_0 - \sum_{i=1}^{p} \beta_i x_{j,i} \right)^2 + \lambda \sum_{i=1}^{p} |\beta_i| \right]$$

BioSB

LASSO

- Difference seems small, but effect of LASSO is that genes are no longer used (like in PAM!)

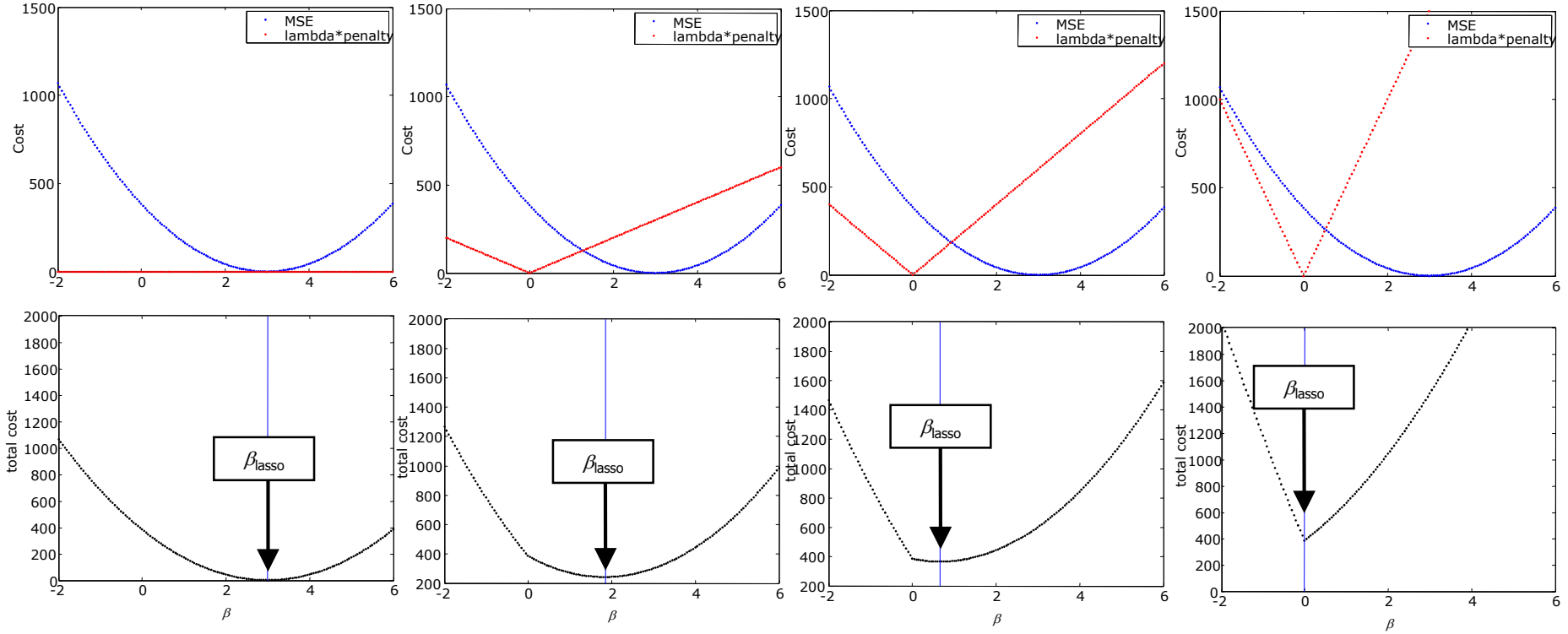




BioSB

LASSO (2)

- Example: true function y = 3x

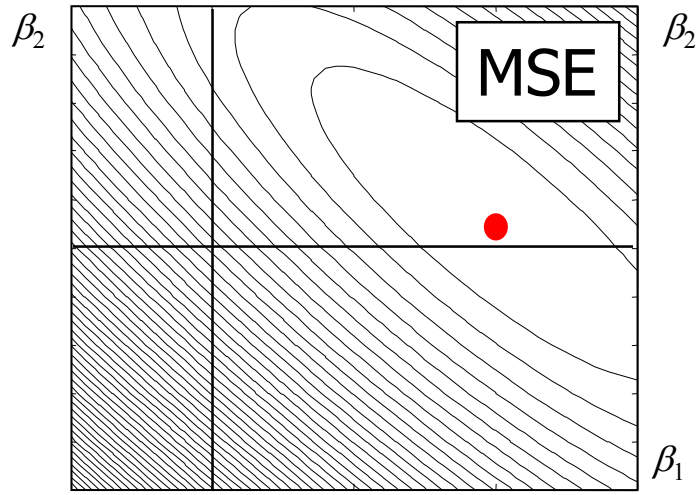


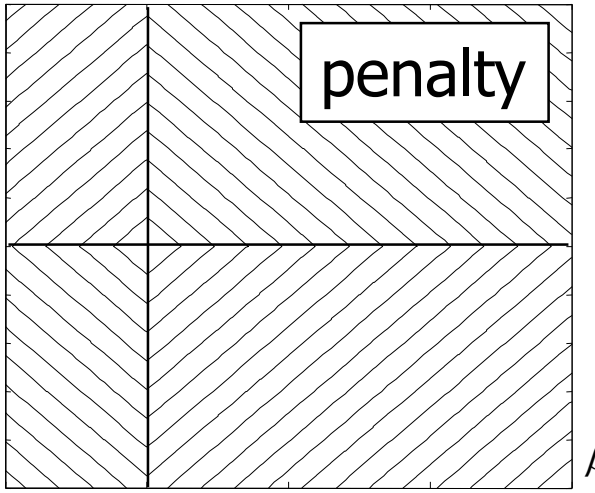
λ = 0 λ = 100 λ = 200 λ = 500

LASSO (3)

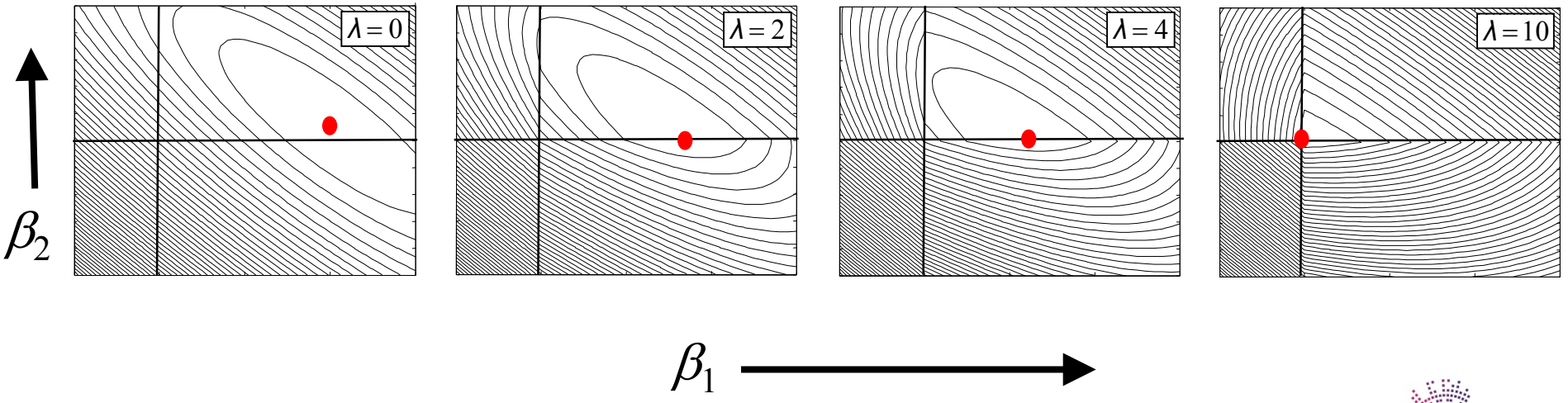
Example:

$y = 0.1x_1 + 0.01x_2$





Optimum



Final summary

- Feature extraction:
  - Linear:
    - PCA,
    - Fisher
  - Non-linear
    - MDS
- Feature selection:
  - Criteria
  - search algorithms
    - forward,
    - backward,
    - branch & bound.
- Sparse classifiers:
  - Ridge,
  - LASSO

BioSB