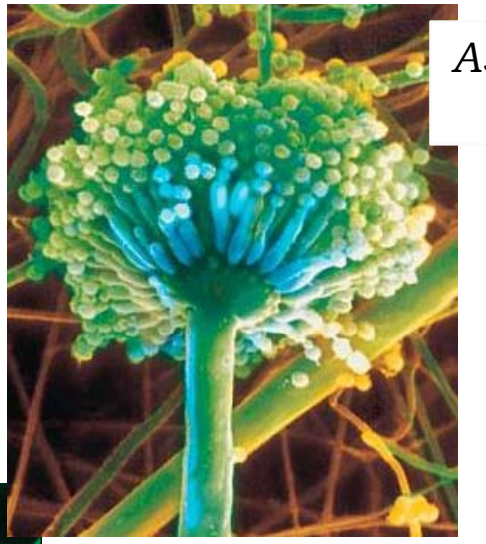
Kernel methods for integrating biological data

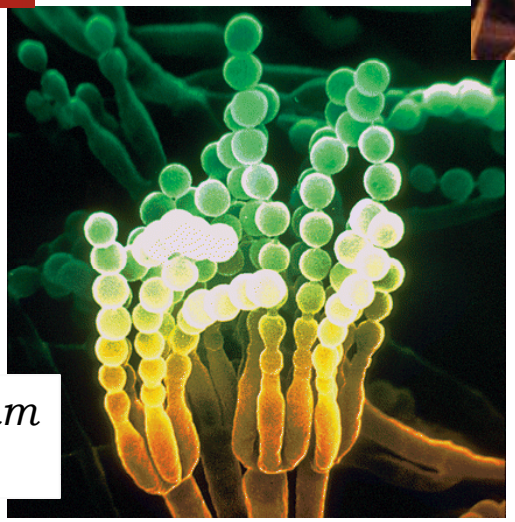
Dick de Ridder, Marc Hulsman & Bastiaan van den Berg
The Delft Bioinformatics Lab, Delft University of Technology

Biotechnology

Saccharomyces cerevisiae
alcohol

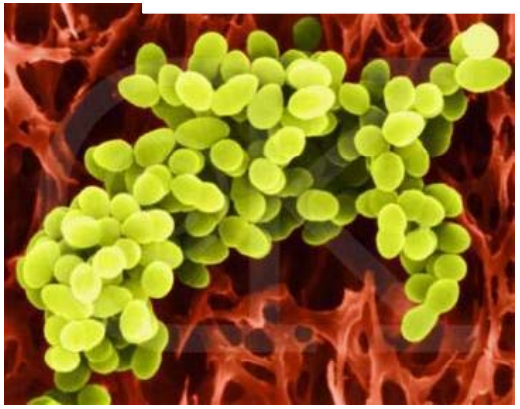


Aspergillus niger
citric acid



Penicillium chrysogenum
penicillin

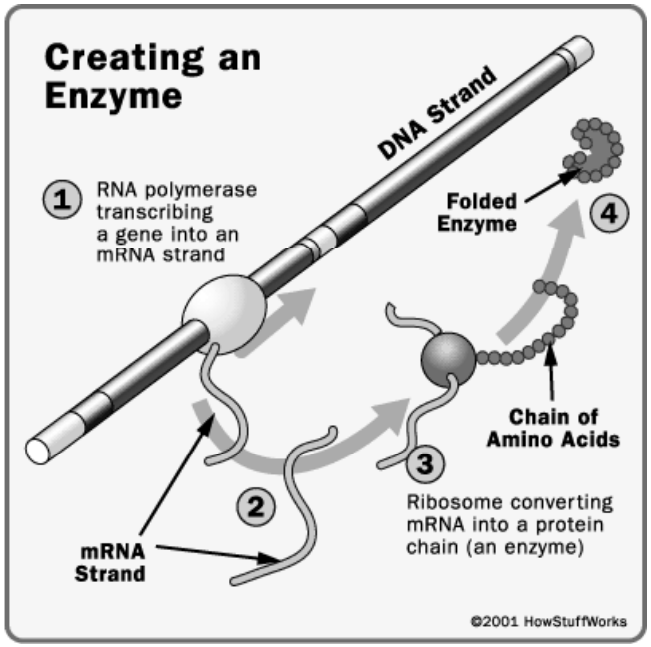
Lactococcus lactis
cheese



Industrial protein production

- Micro-organisms can be used as “cell factories”, genetically modified to produce e.g. specific proteins

- For industrial application, proteins should be:
  - (highly) expressed
    - introduce gene in genome
    - place a strong promoter sequence in front of gene
  - secreted



Industrial protein production (2)

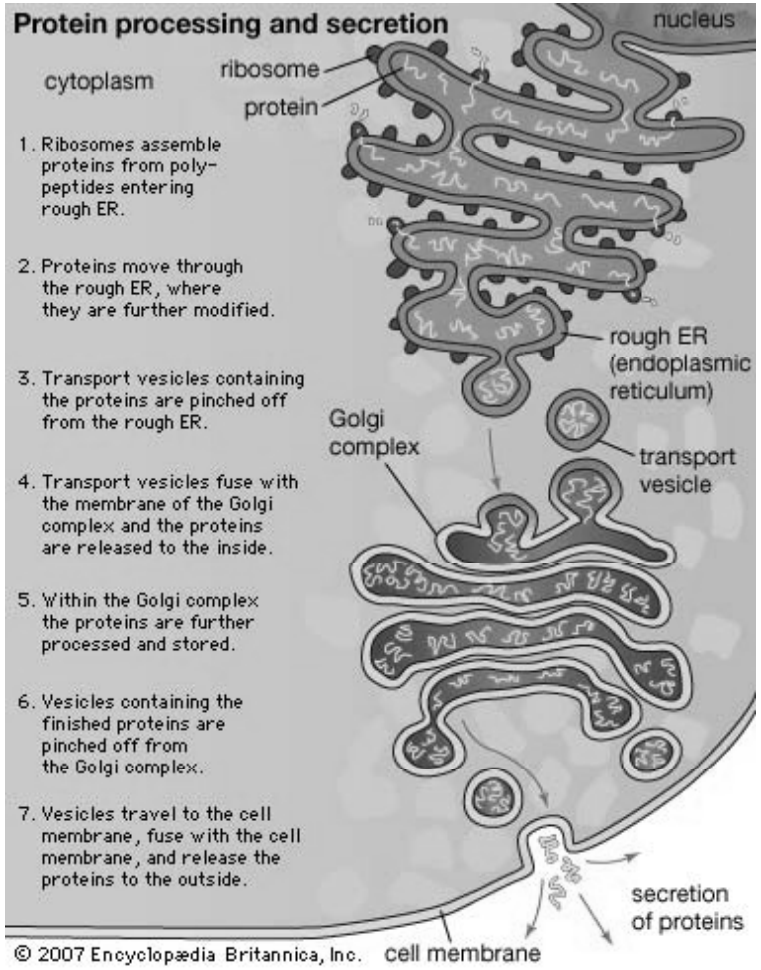
- Billion-dollar industry
- Continuous search for new products: from lab to plant

- Test phase is tedious and costly
- Can we predict what proteins can be successfully expressed?

A bioinformatics problem

- Expression is relatively easy, secretion is hard to get right

- Basic machinery is known, but...
  - signal sequences are not unique
  - many alternative routes
  - lack of knowledge on *heterologous* protein expression

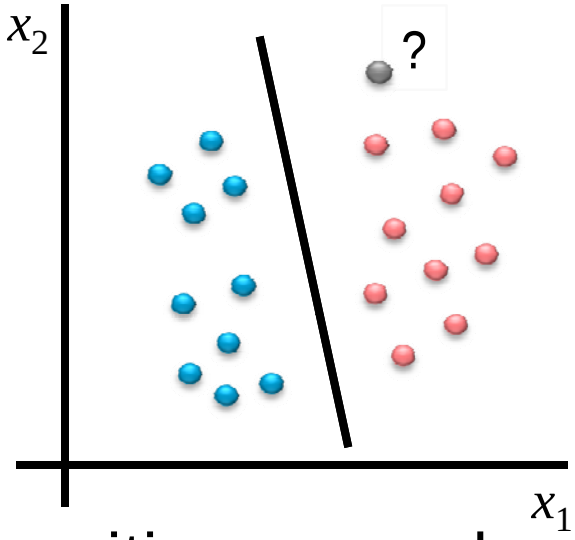


A pattern recognition problem

- When all else (model) fails, turn to pattern recognition

A pattern recognition problem (2)

- Learn from experience:
  dataset of 683 proteins for which secretion was attempted

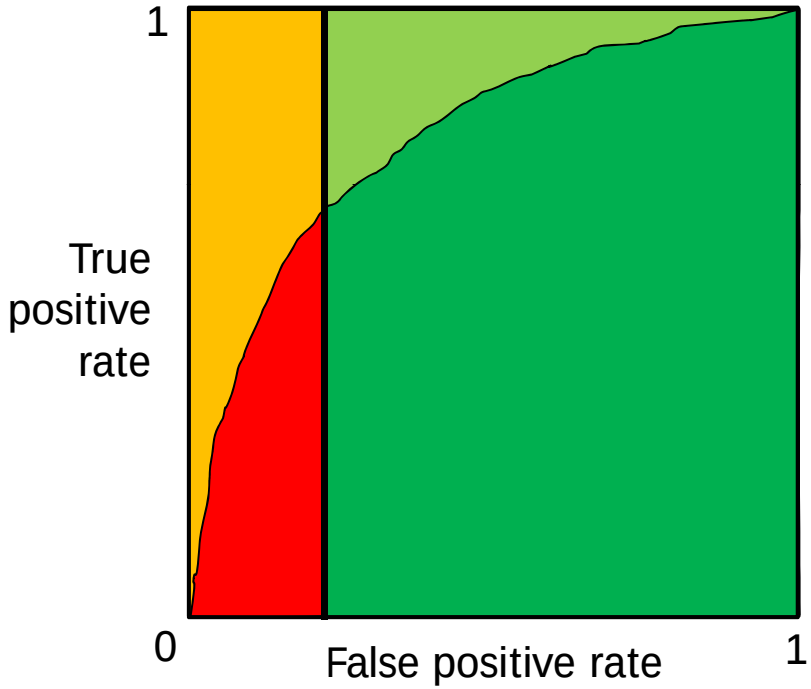


- Required elements for a pattern recognition approach:
  - Objects: proteins
  - Labels: detected secretion at relatively high level (gel)
  - Target: ...?
  - Features: ...?

Bastiaan van den Berg
TB1, Thu 11.15

What is the target?

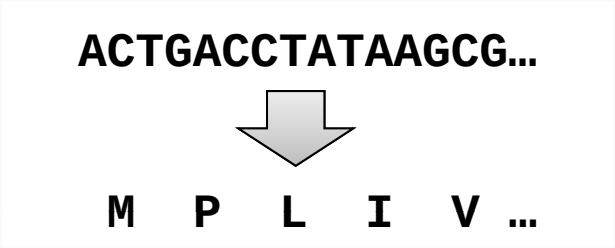
- Predictions need to be experimentally tested: *prioritize*

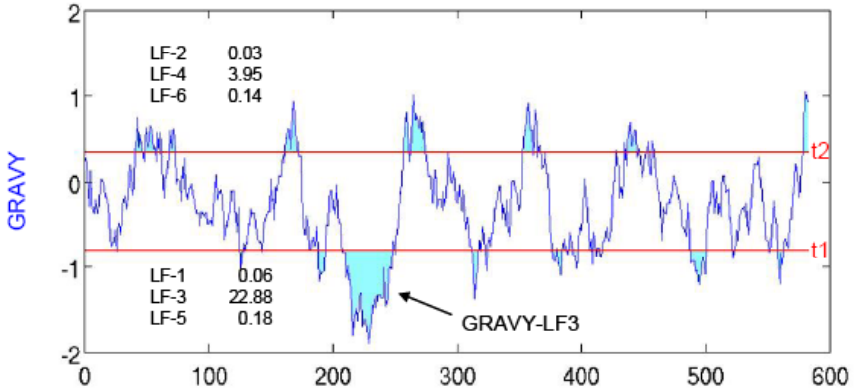


- Criterion: (partial) area under the ROC curve

What is a good representation?

- Available: DNA sequences





- Sequence characteristics
  - Composition:
    - length
    - nucleotide/amino acid composition
    - amino acid subset composition (basic, charged, ...)
  - Derived:
    - codon adaptation index (DNA)
    - hydrophobicity/philicity (protein)

What is a good representation (2)?

- Protein characteristics (predicted)
  - presence of signal sequence
  - subcellular localization
  - protein function
  - isoelectric point
- Heterologous proteins
  - relation to host organism
- … (whatever works)

- Heterogeneous sources of information (prior knowledge) and data (measurements) need to be integrated

Integrative bioinformatics

Kernel-based algorithms

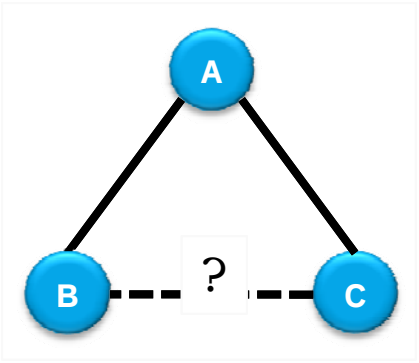
Kernels

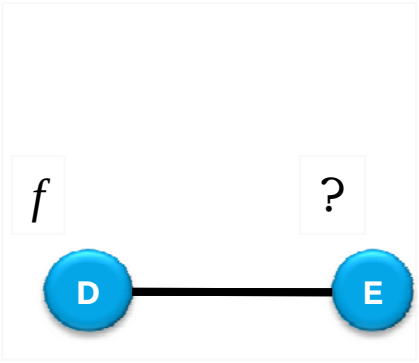
Kernel combination

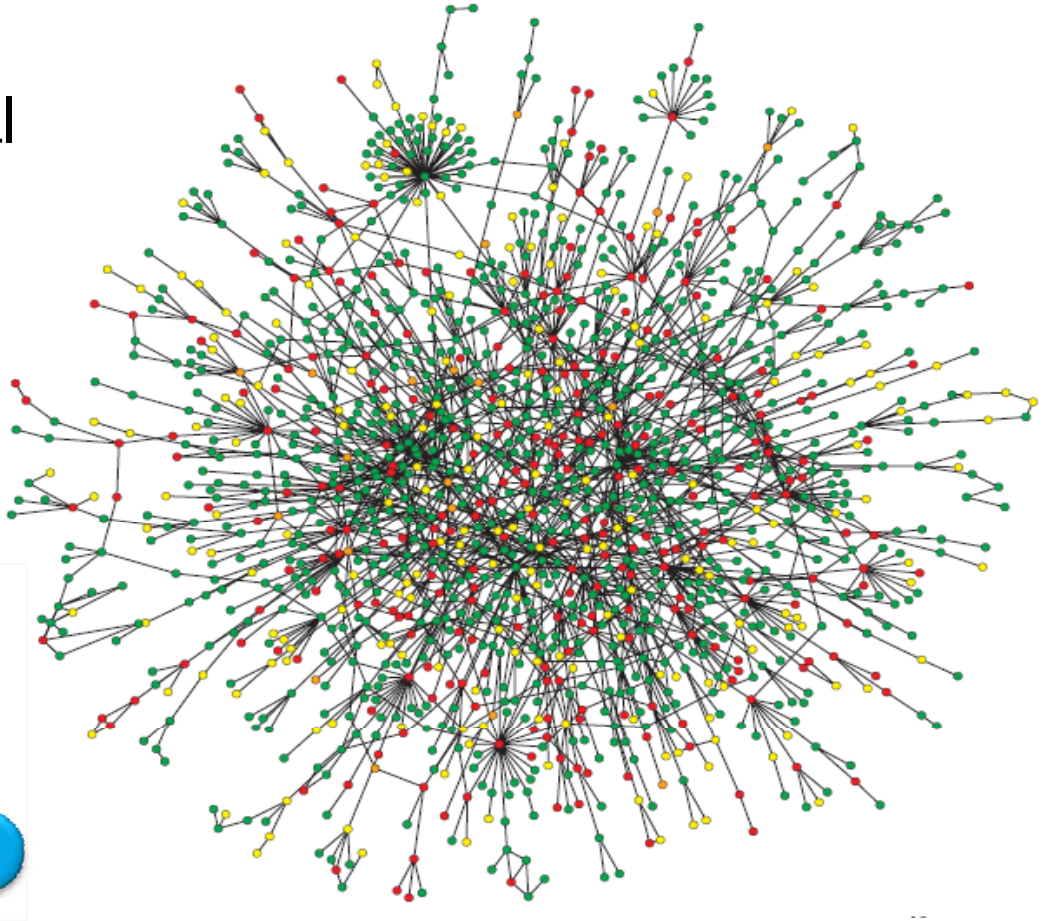
Example applications

Integrative bioinformatics

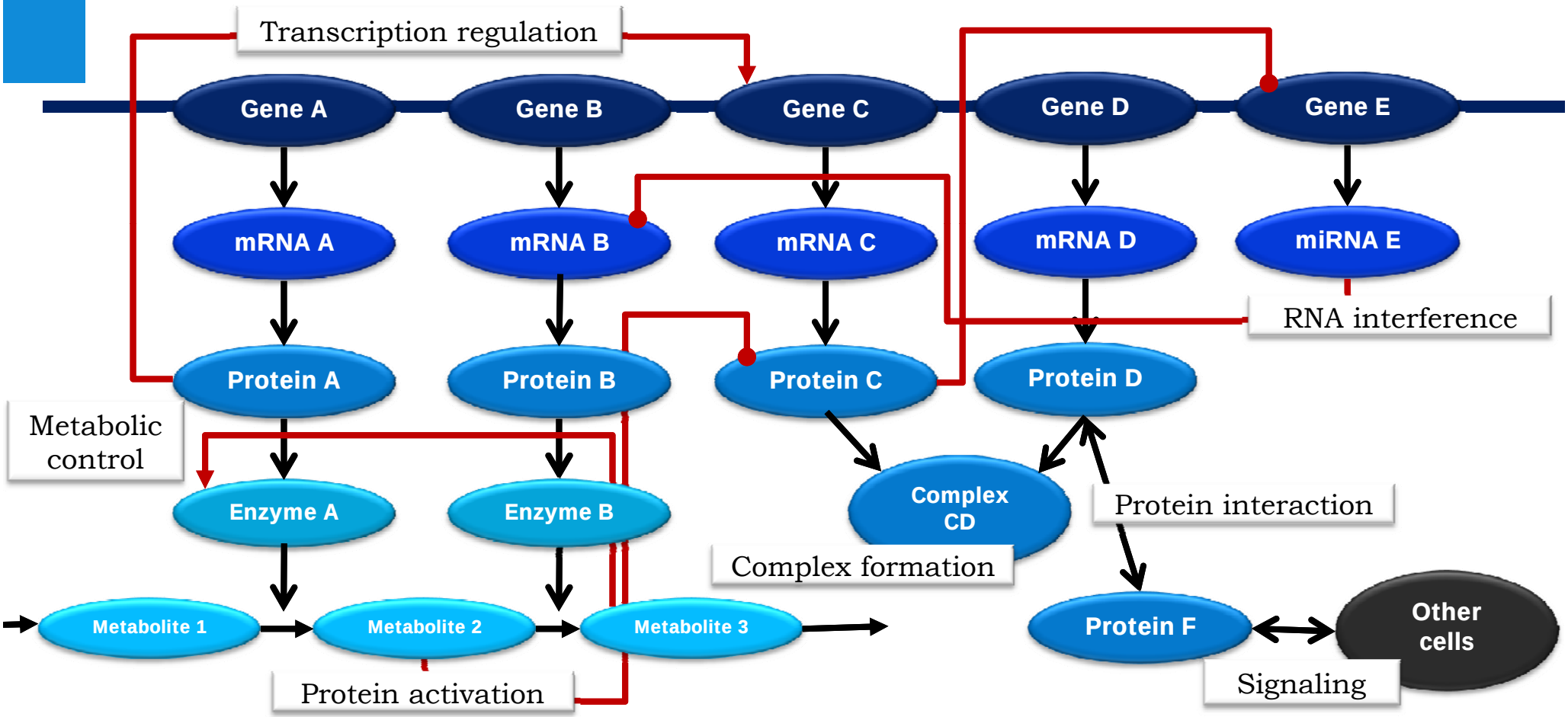
- Construct and interpret networks of biochemical interactions in a living cell, making use of *all available data and prior knowledge*







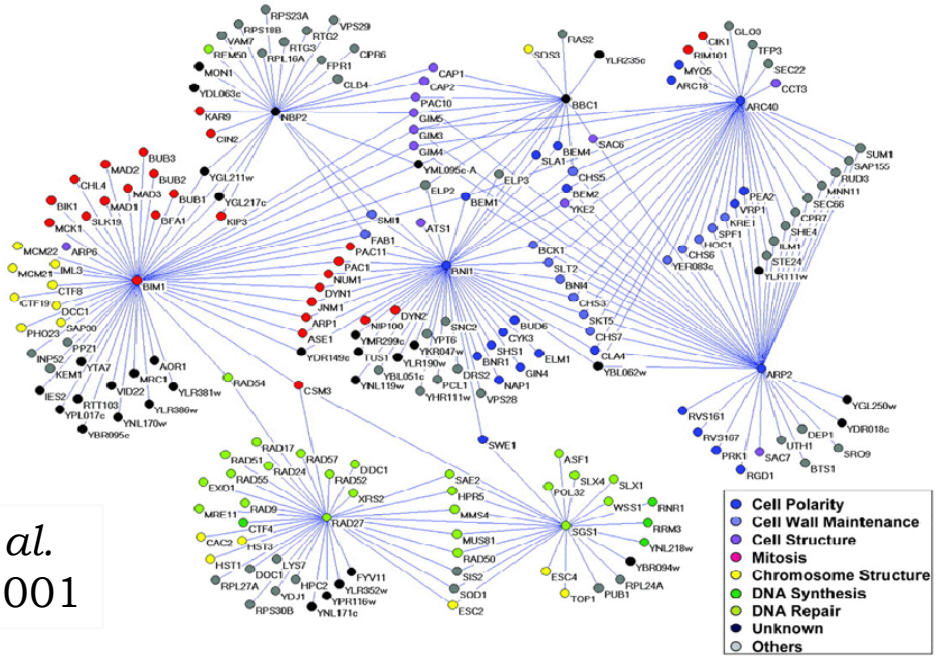
Molecular interaction networks



Focus

- Genetic interactions:
  - Transcription regulation etc. (genes cannot interact)
  - Catch-all term used in functional genomics

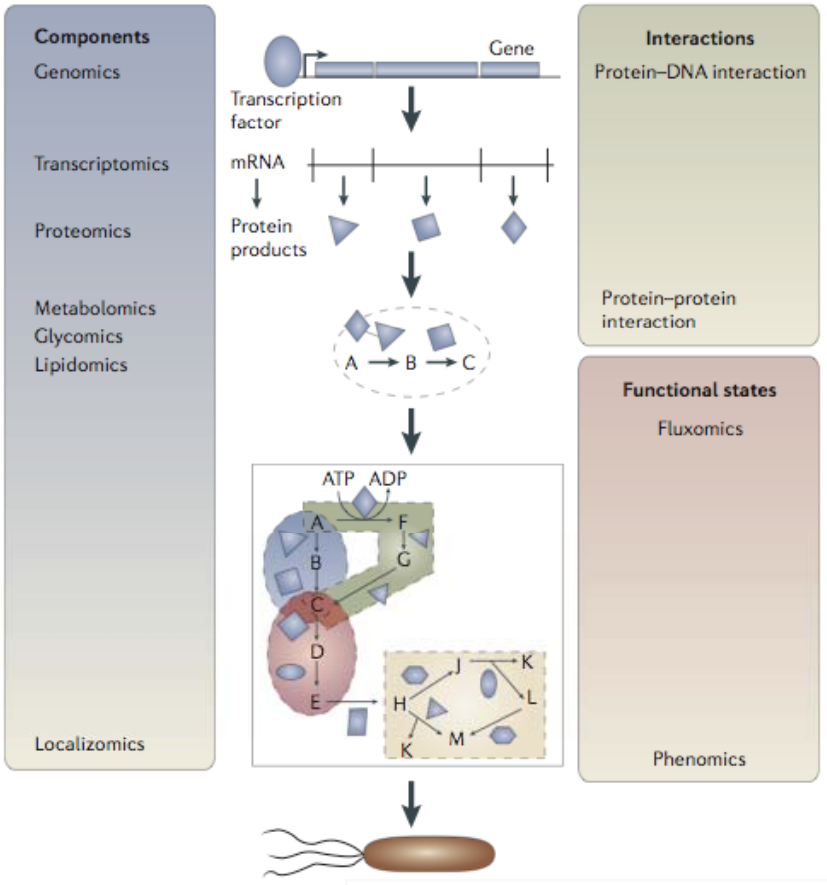
- Protein-protein interaction:
  - Signalling
  - Transport
  - Complex formation



Tong *et al.*
Science 2001

Measurements

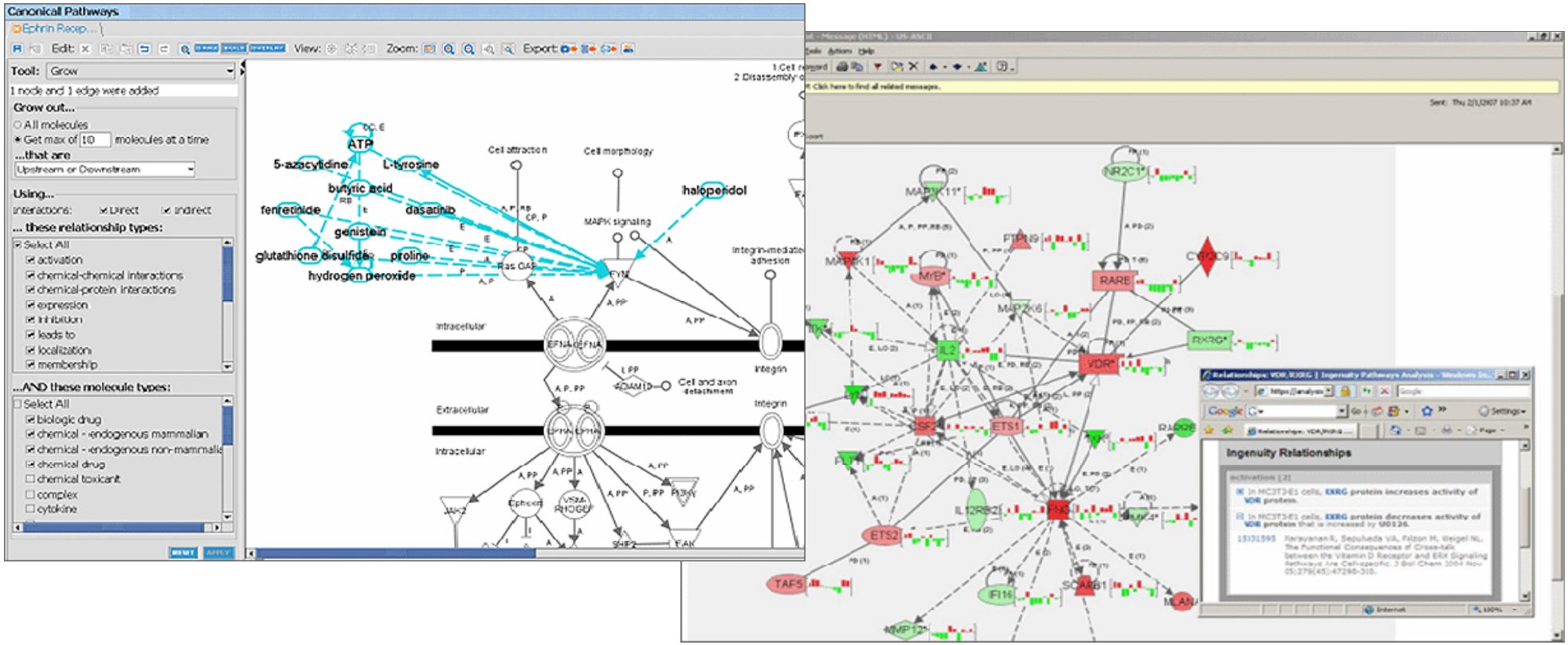
- -omics:
  - sequences
  - transcripts (mRNA)
  - metabolic fluxes
  - protein/metabolite levels
  - protein location
  - protein-protein interaction
  - protein-DNA interaction
  - synthetic sick-or-lethal
  - phenome (conditions)
- Much in (curated) databases



Joyce & Palsson,
Nature Rev Mol Cell Biol 2006

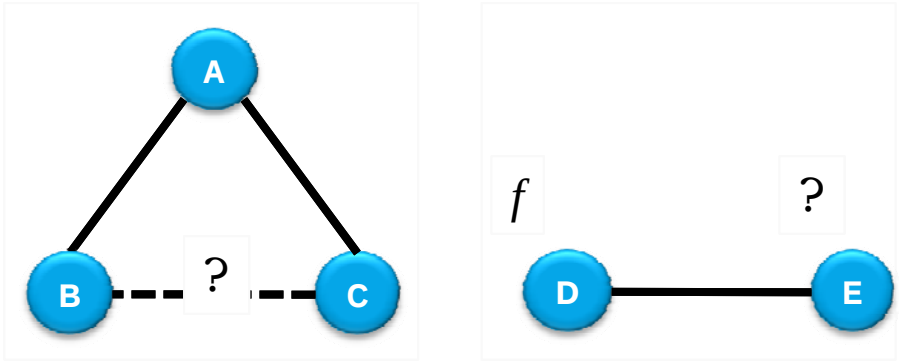
Prior knowledge

- Names, annotations, pathways, reactions, literature...



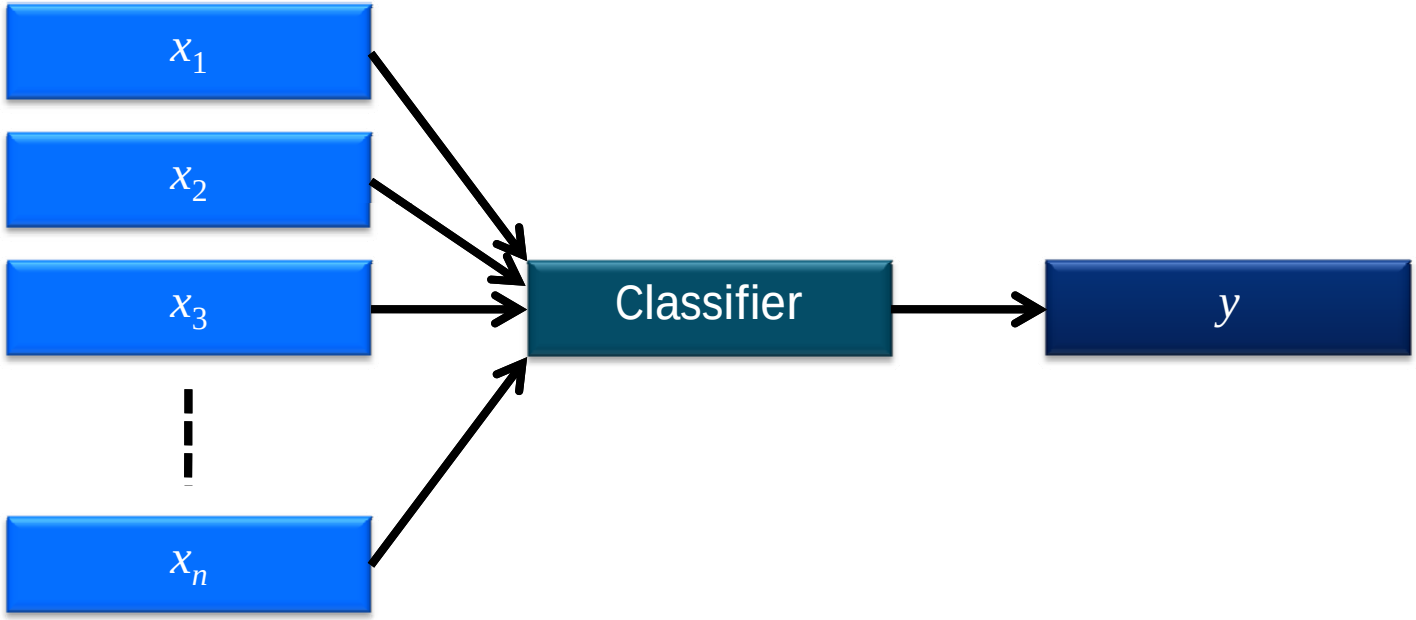
Pattern recognition for integration

- Predicting interactions is really a classification problem
  - input: measurements/data $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
  - output: presence of interaction /function $y \in \{0,1\}$



Early integration

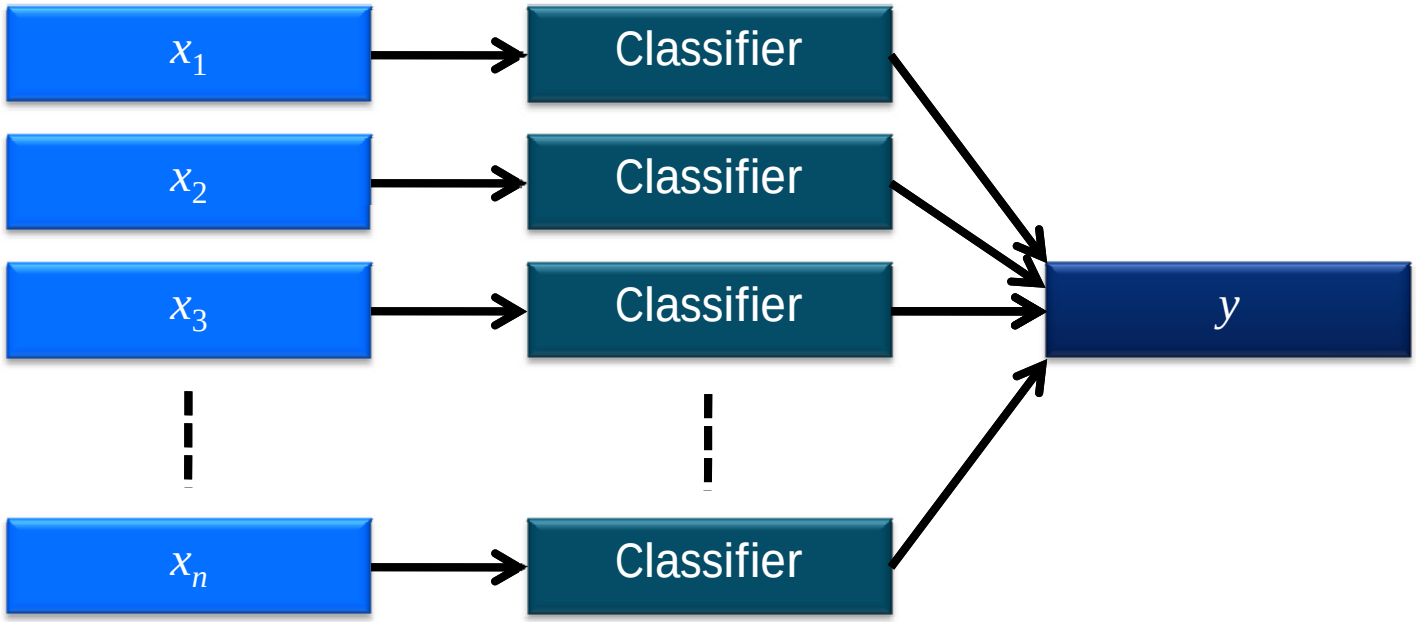
- Feature fusion: the standard approach



- Usually weighted (nonlinear) combination, optimised w.r.t. target

Late integration

- Classifier combination



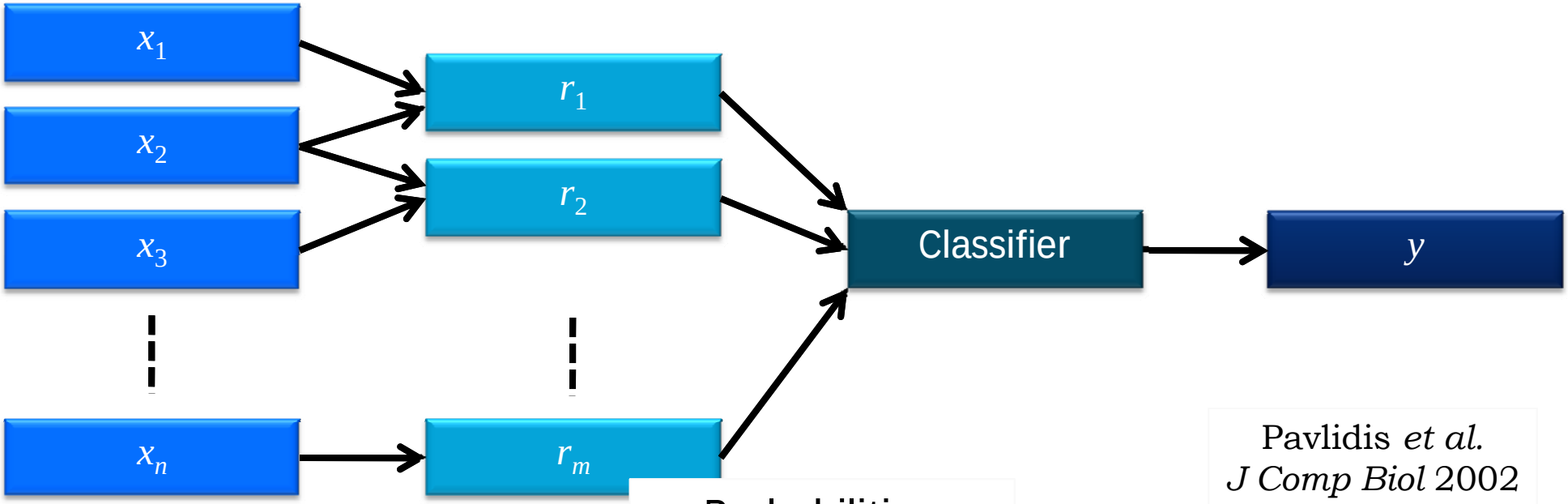
- By fixed rule (max, min, ...) or by trained combiner

Pitfalls

- Early integration:
  - need to convert all features to single representation, e.g. by binning (but: how to do this for sequences, graphs, ...?)
- Late integration:
  - choosing combination mechanism
  - integration of different classifiers not straightforward
  - knowledge of data heterogeneity unused

Intermediate integration

- Transform characteristics into a “common language”, an intermediate representation suitable for integration

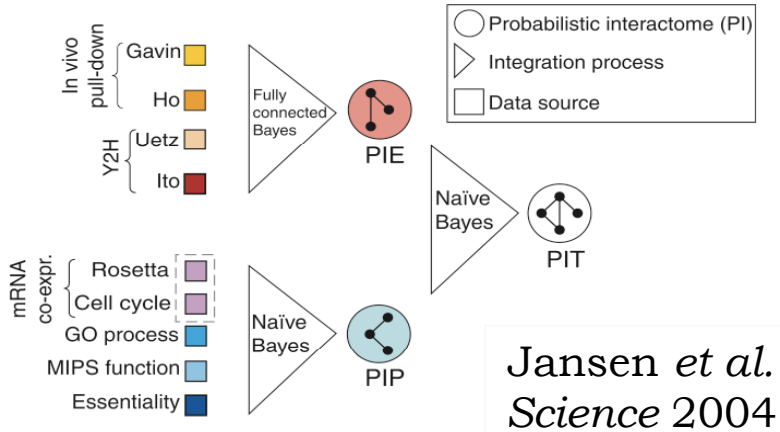


- Probabilities
- (Dis)similarities
- Kernels

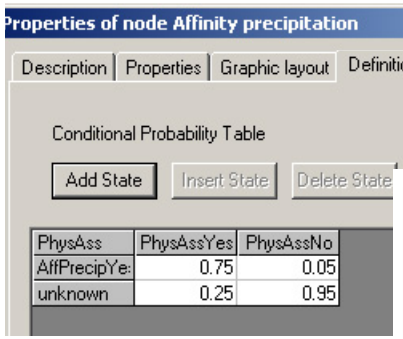
Pavlidis *et al.*
J Comp Biol 2002

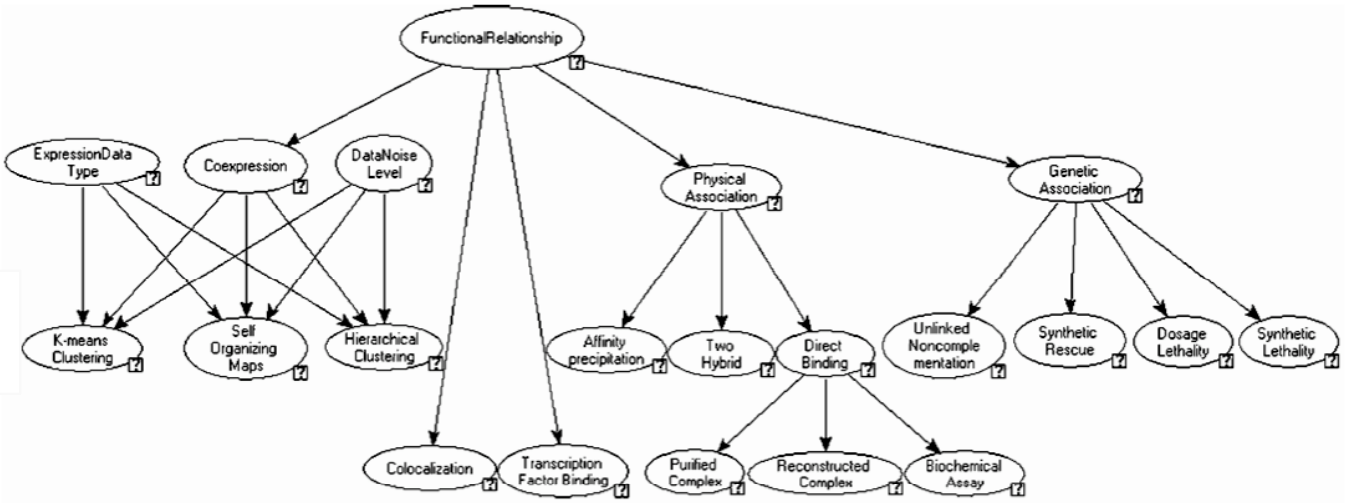
Integrating probabilities

- For example, through Bayesian networks...



Jansen *et al.* *Science* 2004





Troyanskaya *et al.* *PNAS* 2003

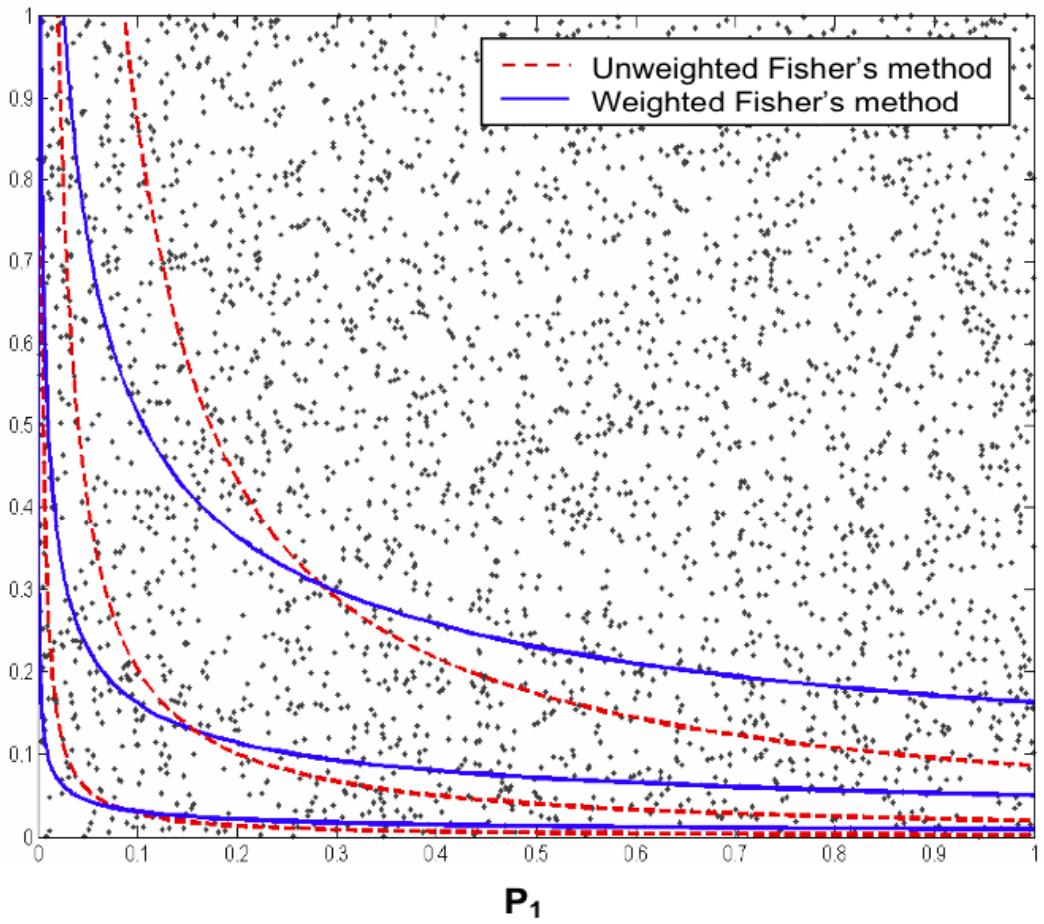
Integrating probabilities

- ...or by combining p-values or test scores

$$F_w = -2\sum_{i=1}^{n} w_i \ln(p_i)$$

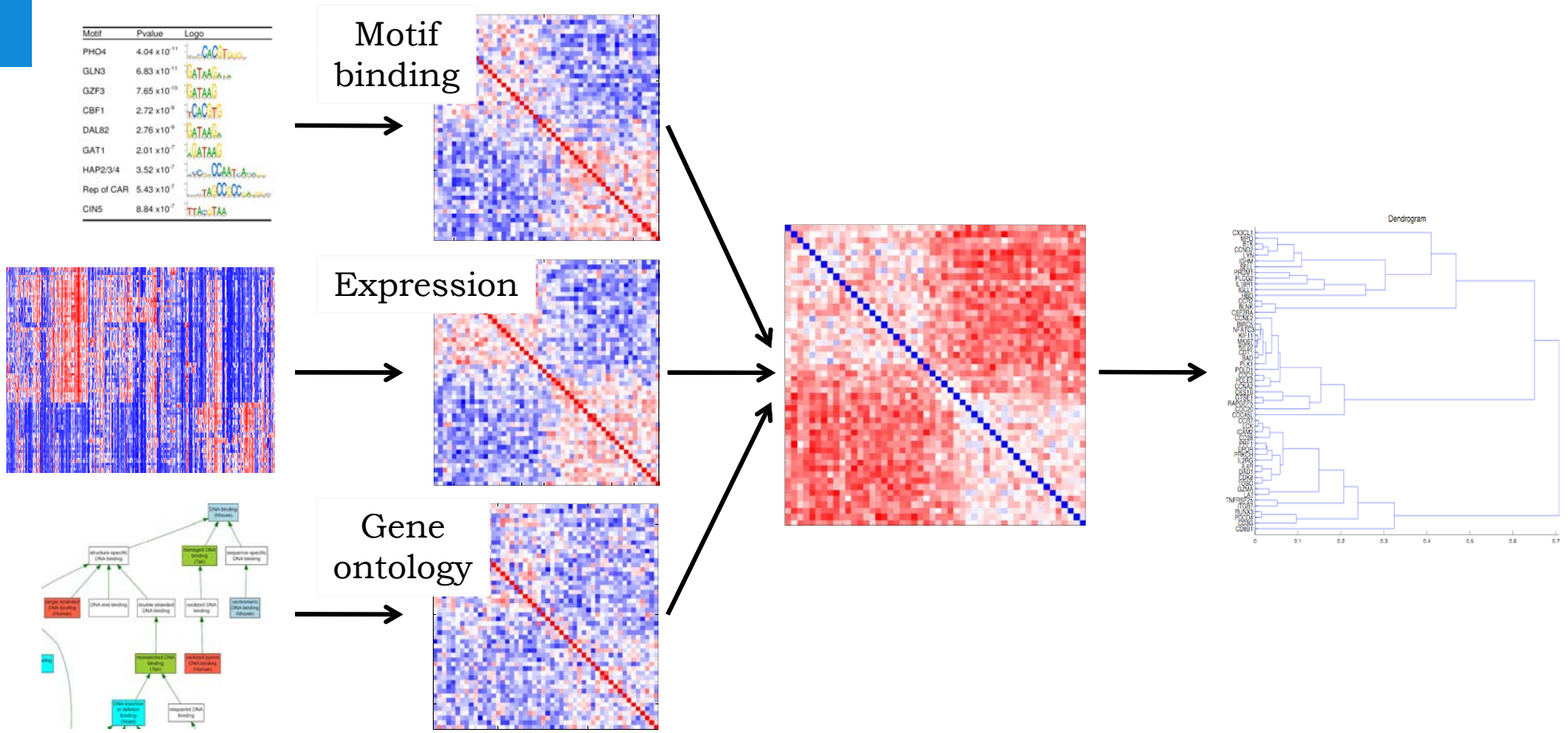
$$H_0 : F_w \sim \chi^2(2)$$

Hwang *et al.* *PNAS* 2005



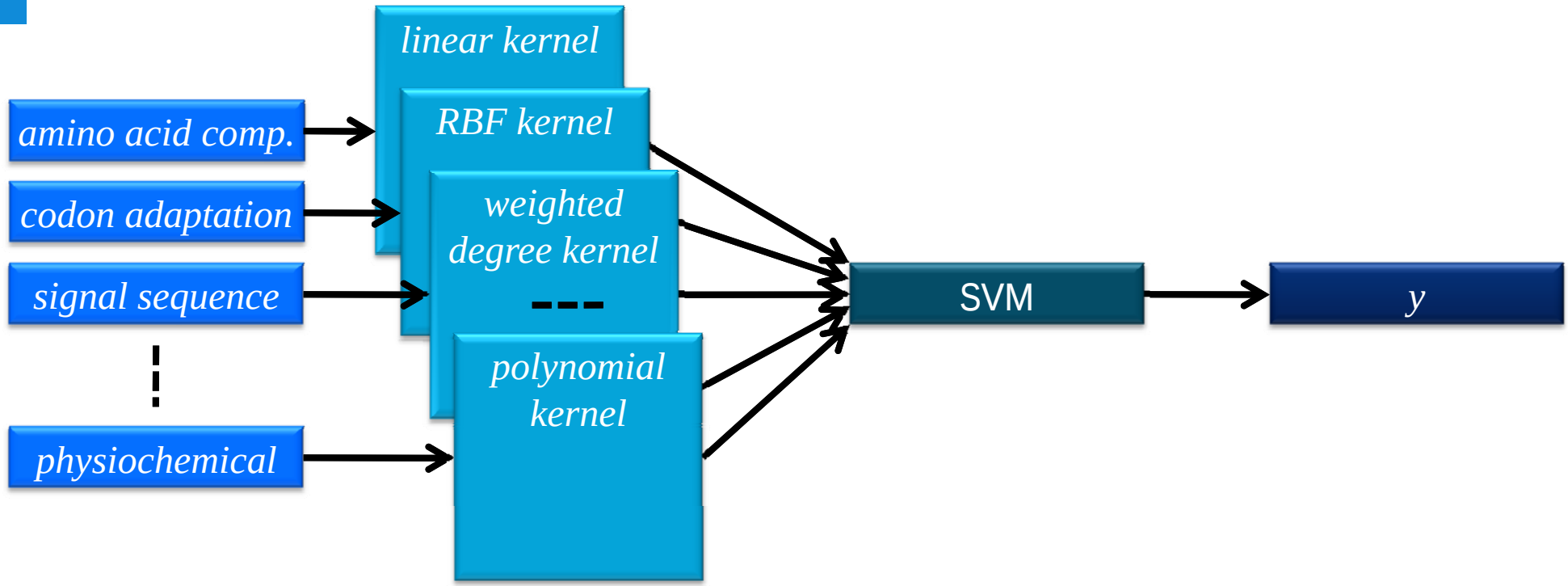
Integrating (dis)similarities

- For example, by adding distance matrices



Integrating kernels

- For example, by adding kernels – well-founded similarities



Integrative bioinformatics

Kernel-based algorithms

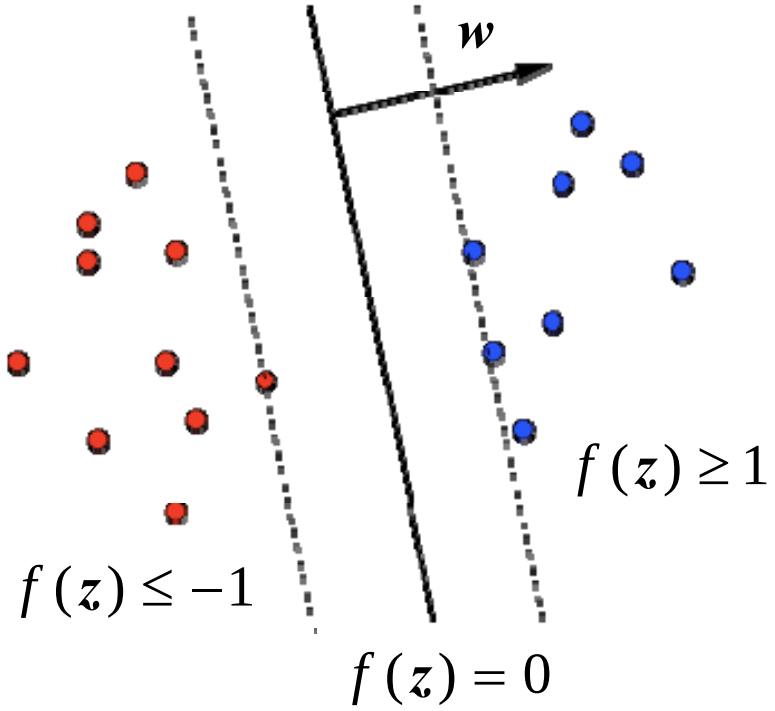
Kernels

Kernel combination

Example applications

Support vector machines

Boser, Guyon and Vapnik, *COLT* 1992;
Cortes and Vapnik, *Machine Learning* 1995



$$f(z) = w^T z + b$$

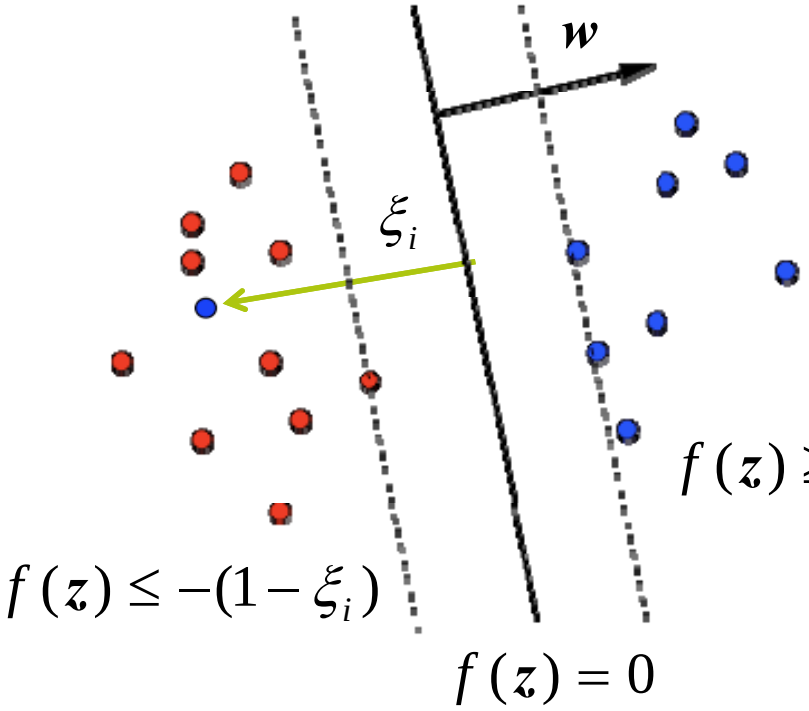
$$\min \|w\|^2$$

$$w^T x_i + b \geq +1, \quad y = +1$$

$$w^T x_i + b \leq -1, \quad y = -1$$

Support vector machines

- Slack variables: allow misclassifications on training set



$$f(z) = w^T z + b$$

$$\min \|w\|^2 + C\sum_{i=1}^{|X|} \xi_i$$

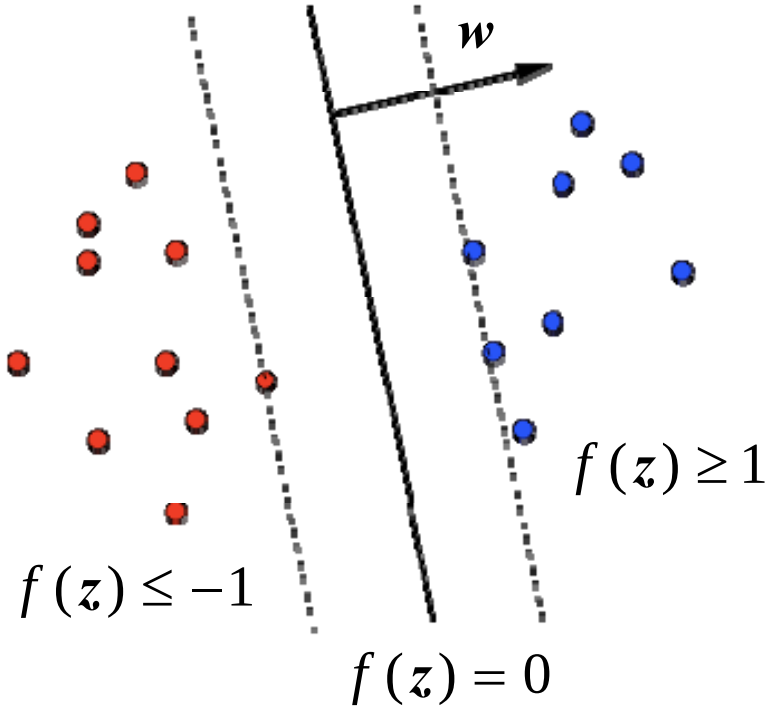
$$w^T x_i + b \geq +(1-\xi_i), \quad y = +1$$

$$w^T x_i + b \leq -(1-\xi_i), \quad y = -1$$

$$\xi_i \geq 0, \quad \forall i$$

Support vector machines (2)

- Rewrite (original) optimisation problem (dual) :



$$f(z) = w^T z + b$$

$$= \sum_{i=1}^{|X|} \alpha_i y_i x_i^T z + b$$

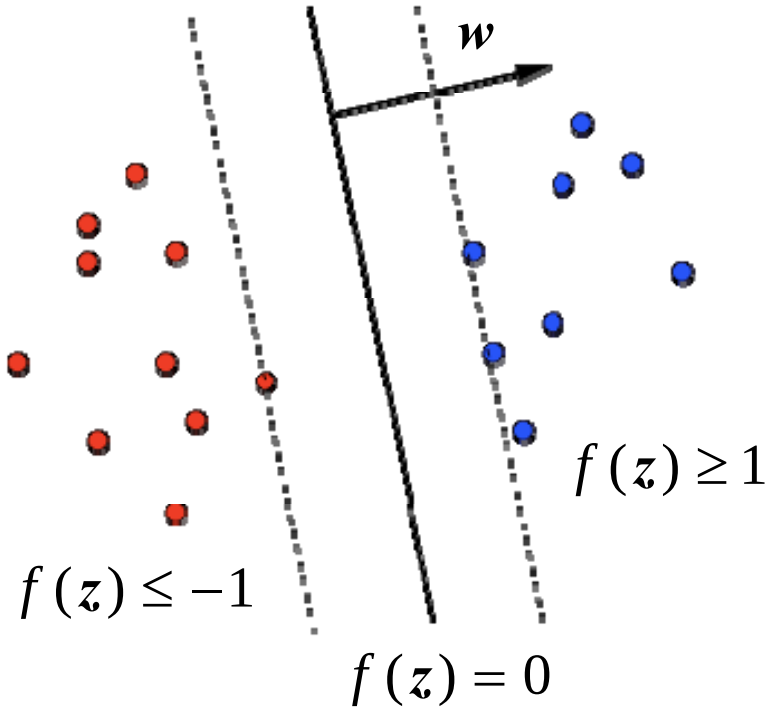
$$\max_{\alpha} \sum_{i=1}^{|X|} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{|X|} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$0 \leq \alpha_i \leq C, \quad \forall i$$

$$\sum_{i=1}^{|X|} \alpha_i y_i = 0$$

Support vector machines (3)

- Map input space into feature space using Φ:



$$f(z) = w^T z + b$$

$$= \sum_{i=1}^{|X|} \alpha_i y_i \Phi(x_i)^T \Phi(z) + b$$

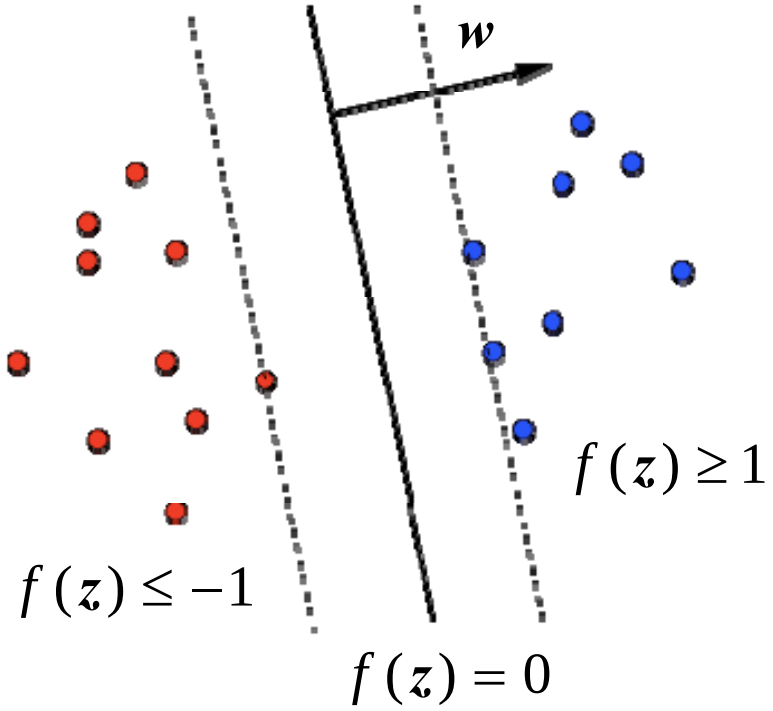
$$\max_{\alpha} \sum_{i=1}^{|X|} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{|X|} y_i y_j \alpha_i \alpha_j \Phi(x_i)^T \Phi(x_j)$$

$$0 \le \alpha_i \le C, \quad \forall i$$

$$\sum_{i=1}^{|X|} \alpha_i y_i = 0$$

Support vector machines (4)

- Replace inner product by kernel (similarity) function:



$$f(z) = w^T z + b$$

$$= \sum_{i=1}^{|X|} \alpha_i y_i K(x_i, z) + b$$

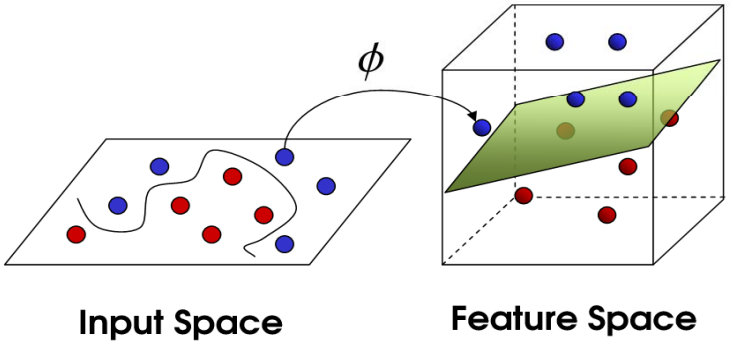
$$\max_{\alpha} \sum_{i=1}^{|X|} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{|X|} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$$

$$0 \leq \alpha_i \leq C, \quad \forall i$$

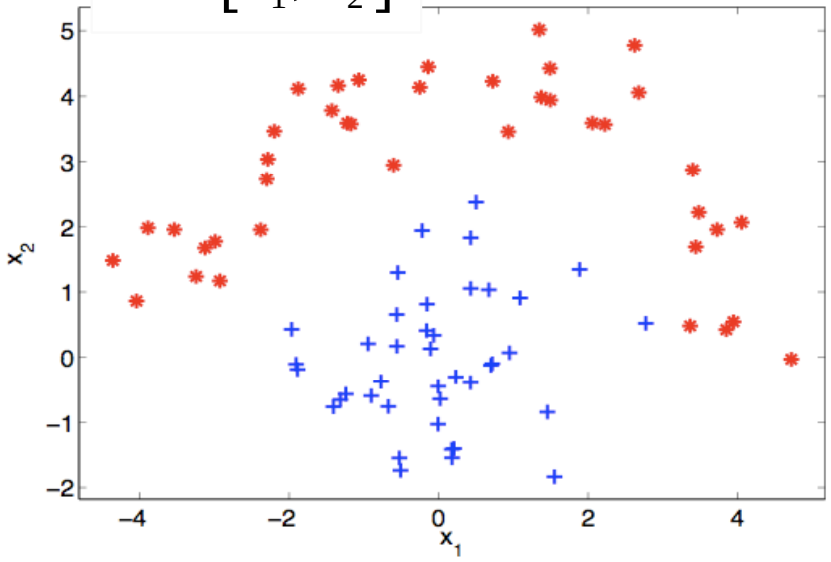
$$\sum_{i=1}^{|X|} \alpha_i y_i = 0$$

Feature space

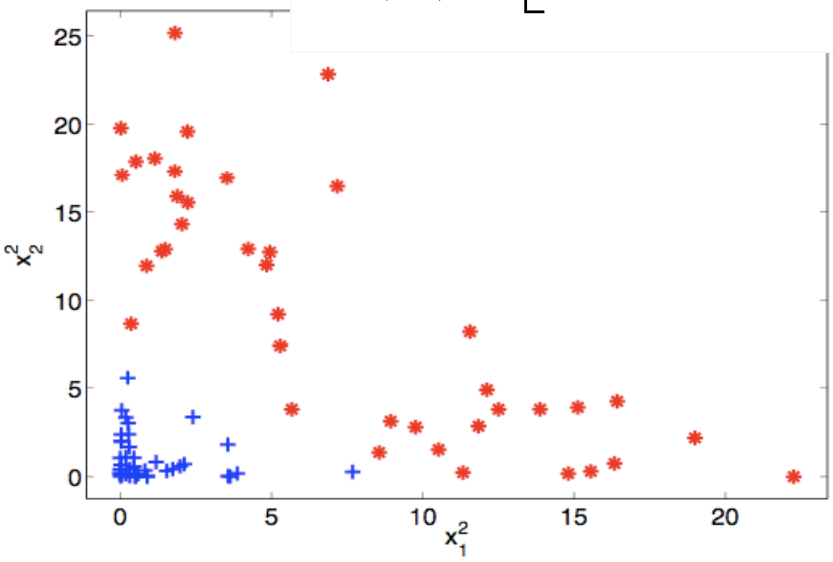
- Function Φ maps data into space in which classification may be easier



$$\boldsymbol{x} = [x_1, x_2]$$



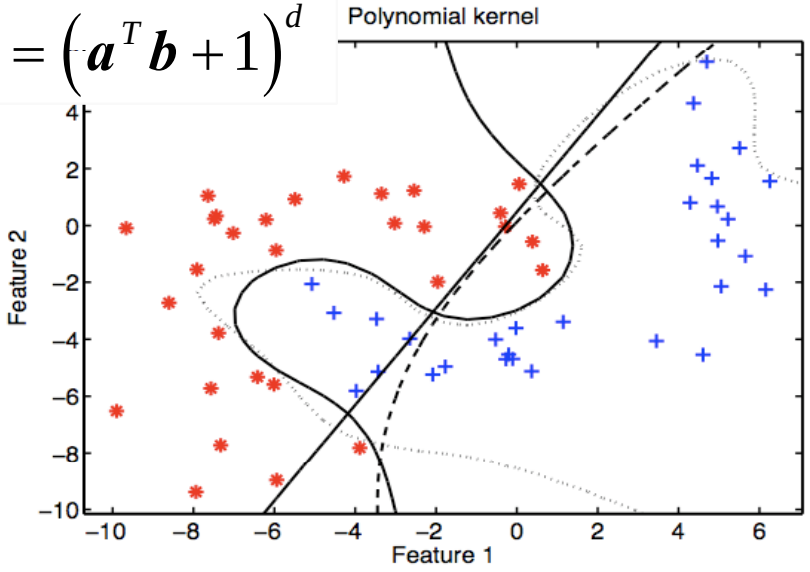
$$\Phi(\boldsymbol{x}) = \left[x_1^2, x_2^2, \sqrt{2}x_1x_2 \right]$$

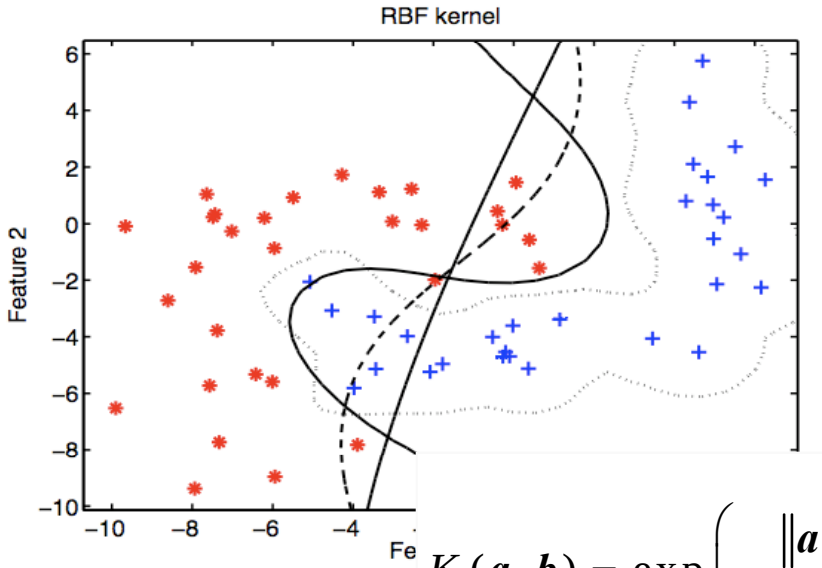


Kernels

- Kernels $K(\boldsymbol{a},\boldsymbol{b}) = \Phi(\boldsymbol{a})^T \Phi(\boldsymbol{b})$: using the same algorithm, obtain a nonlinear classifier in original space

$$K(\boldsymbol{a},\boldsymbol{b}) = \left(\boldsymbol{a}^T\boldsymbol{b}+1\right)^d$$





$$K(\boldsymbol{a},\boldsymbol{b}) = \exp\left(-\frac{\|\boldsymbol{a}-\boldsymbol{b}\|^2}{\sigma^2}\right)$$

Kernels (2)

- Not necessary to actually *know* $\Phi(\mathbf{.})$ to construct $K(\boldsymbol{x},\boldsymbol{y})$!
  Any kernel function is valid if it is *positive definite*,
  i.e. if for any input the resulting kernel matrix $\boldsymbol{K}$ is
  positive definite ($\boldsymbol{z}^T \boldsymbol{K}\boldsymbol{z} > 0, \quad \forall \boldsymbol{z} \in \mathbb{R}^n \neq \boldsymbol{0}$)

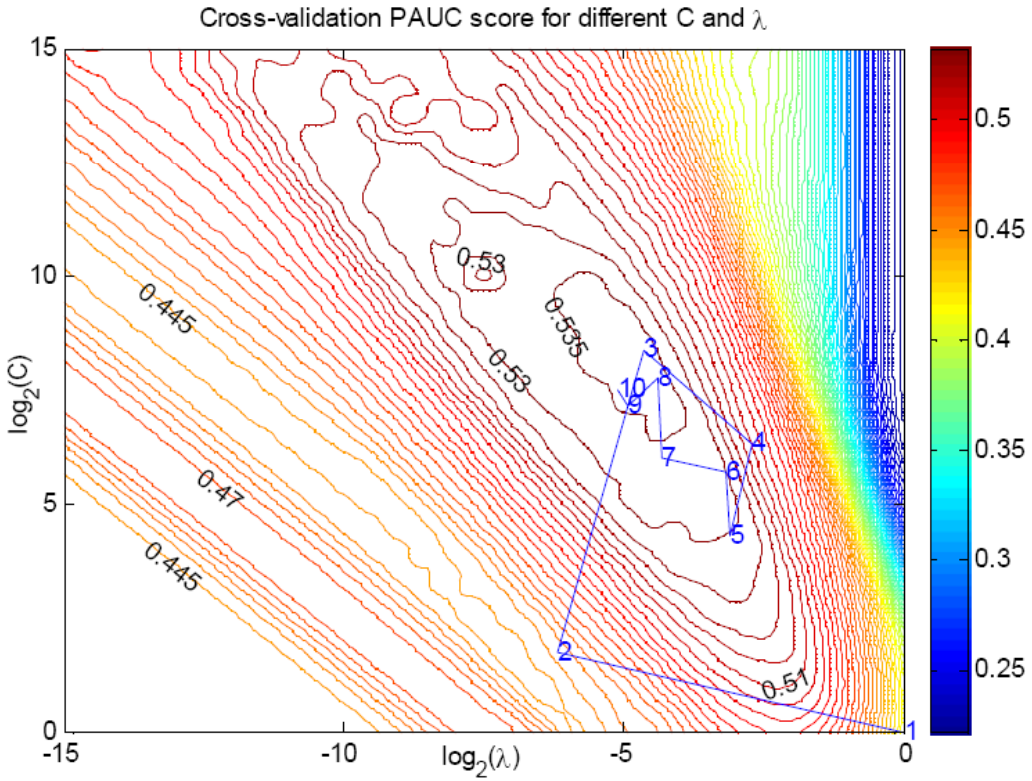
- If $\boldsymbol{K}$ is not positive definite: empirical kernel map (later)

- Other classifiers can be written in terms of inner products and similarly be “kernelised”: kernel nearest mean classifier, kernel k-nearest neighbour, kernel LDA (Fisher), ...

Training

- Optimisation of $\boldsymbol{w}$, b for SVM is a convex problem: can use standard solvers

- Find other parameters by grid search, using cross-validation error estimate
  - Trade-off parameter C (at least for SVC)
  - Kernel parameters: d, σ, ...



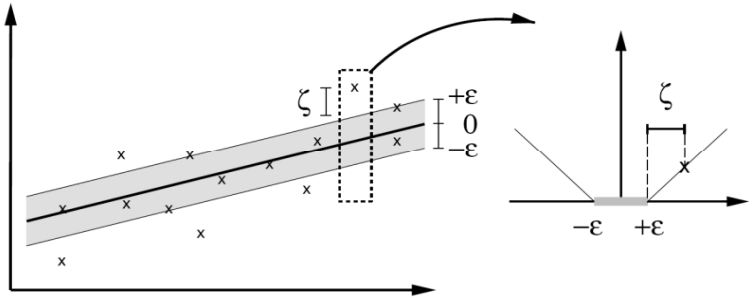
Support vector regression

- Regressor: $y = \boldsymbol{w}^T \boldsymbol{x} + b$

Smola & Schölkopf, 1998

- Loss function: ε-insensitive loss,

$$\xi = \begin{cases} 0 & |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$

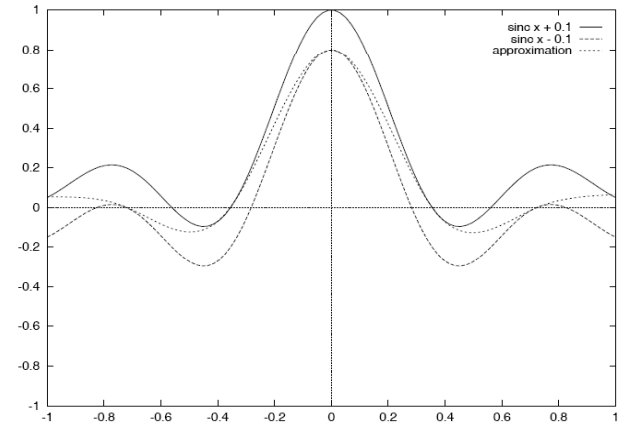


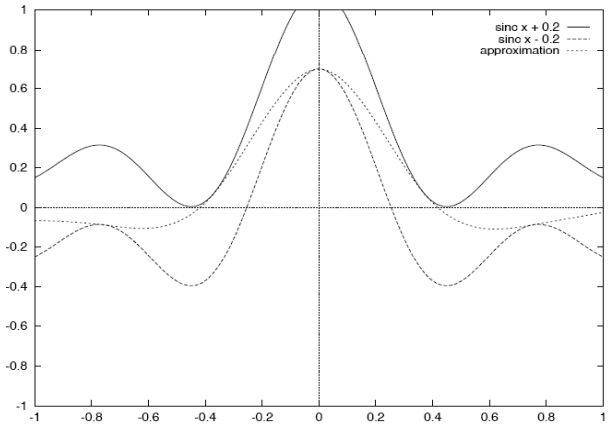
- Optimization problem: $\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right)$

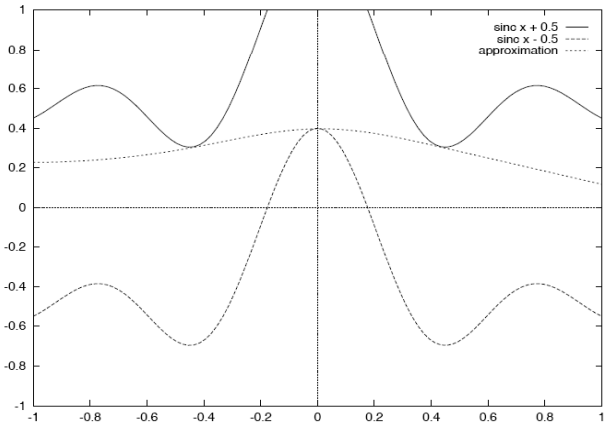
$$y_i - \boldsymbol{w}^T \boldsymbol{x}_i - b \le \varepsilon + \xi_i$$
$$\boldsymbol{w}^T \boldsymbol{x}_i + b - y_i \le \varepsilon + \xi_i^* \quad \forall i$$
$$\xi_i, \xi_i^* \ge 0$$

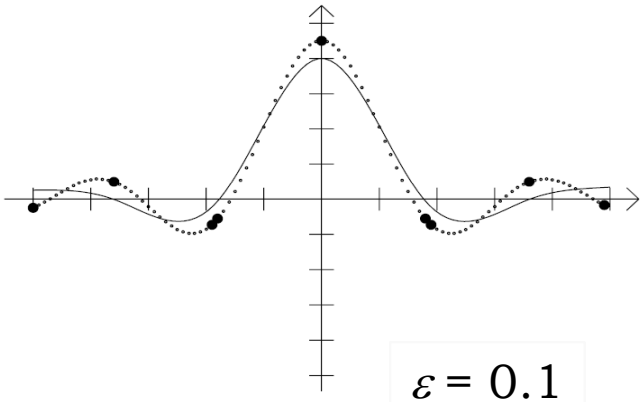
Support vector regression (2)

Smola & Schölkopf, 1998

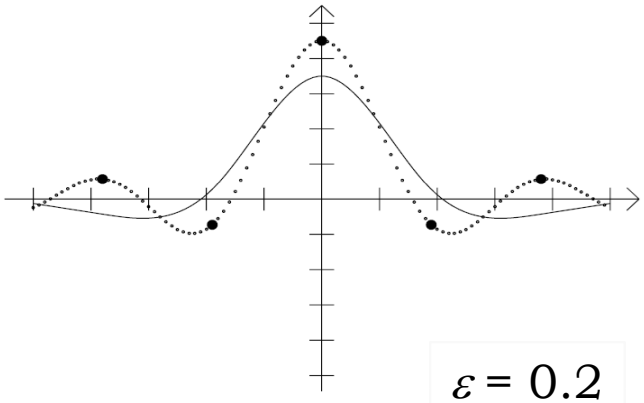




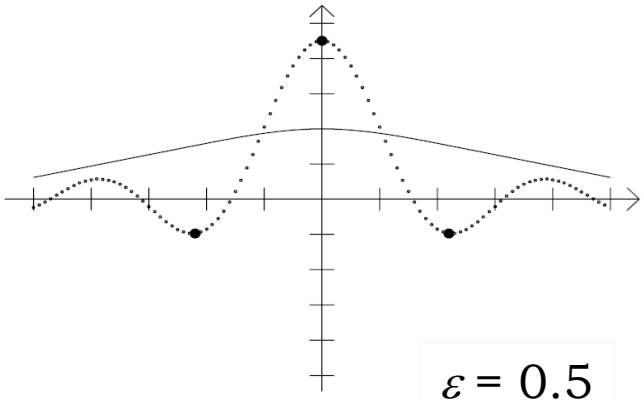




$\varepsilon = 0.1$



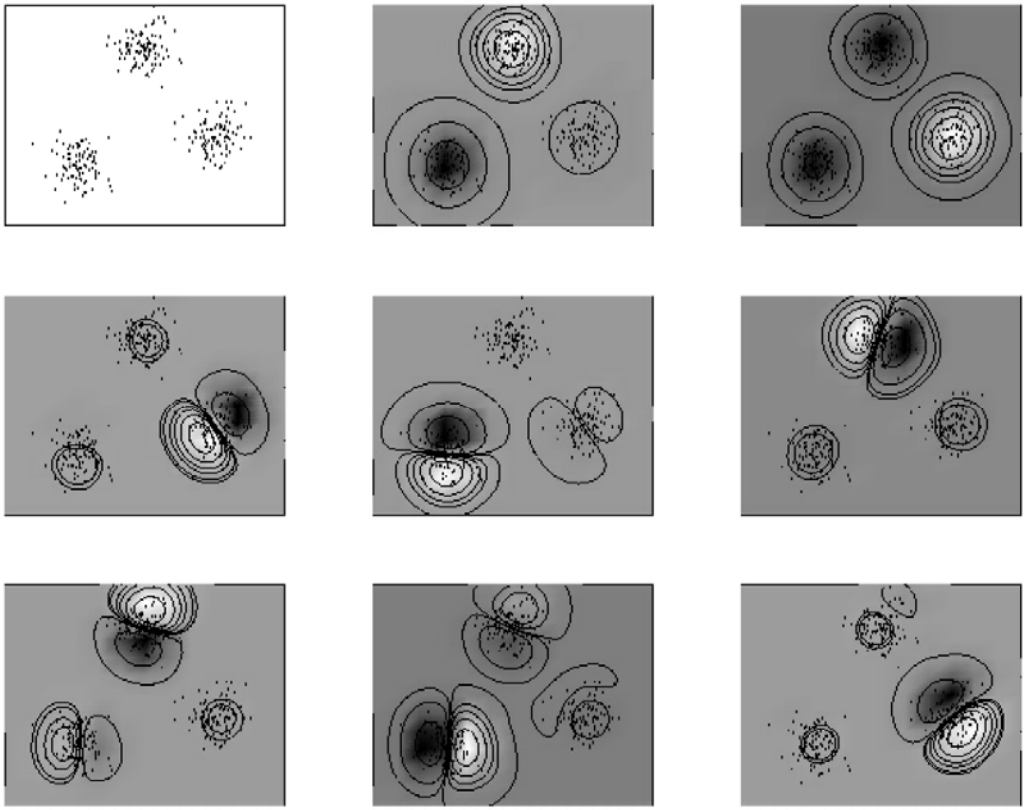
$\varepsilon = 0.2$



$\varepsilon = 0.5$

Kernel clustering

- Hierarchical clustering using kernel matrices
- Kernel k-means
- Kernel MDS
  (= kernel PCA)
- …

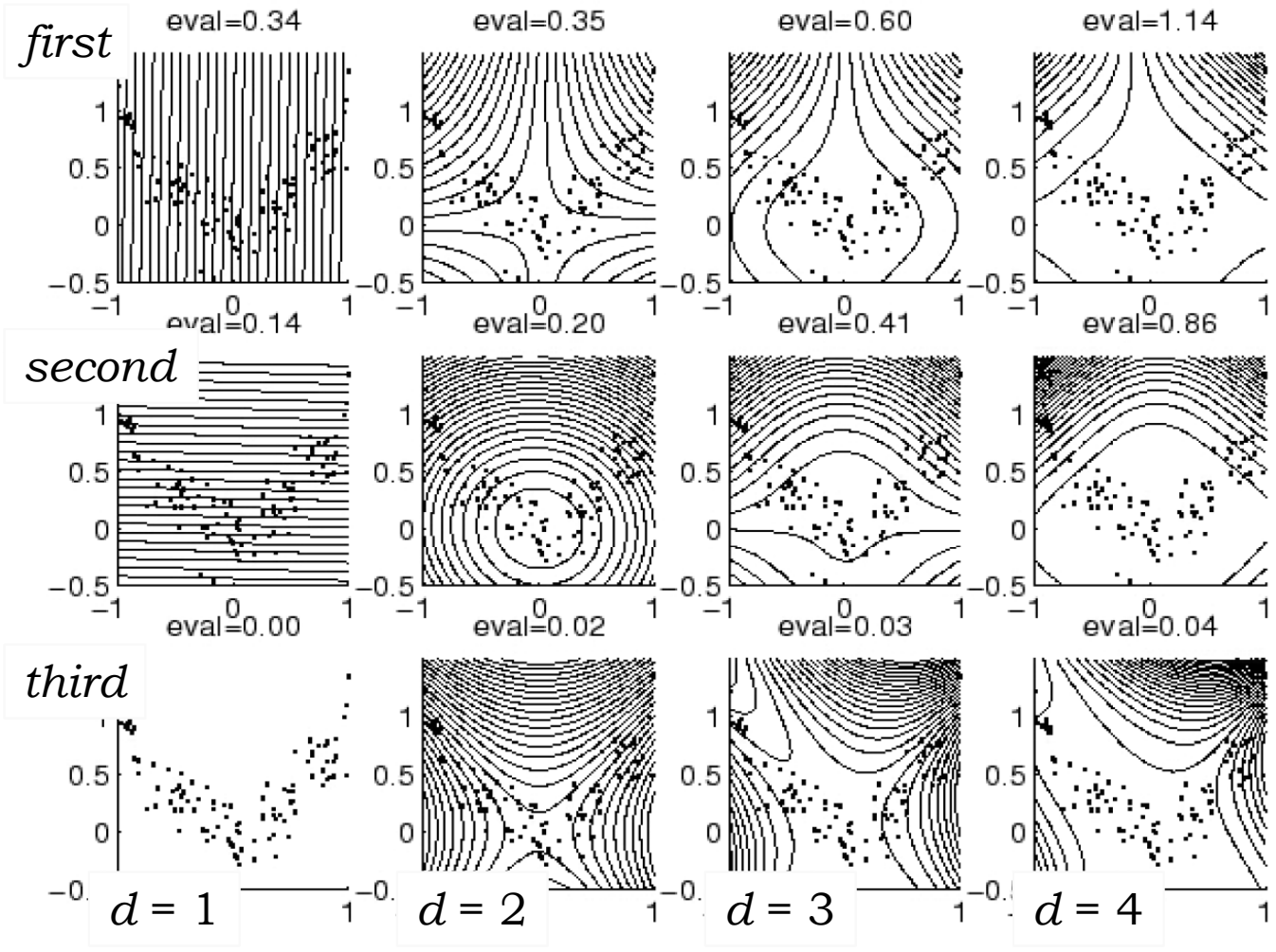


Schölkopf *et al.*, 1996

Kernel dimensionality reduction

Schölkopf *et al.*, 1996

- Example: principal component analysis, polynomial kernel
- Similarly:
  - kernel LDA
  - kernel CCA
  - ...



Integrative bioinformatics

Kernel-based algorithms

Kernels

Kernel combination

Example applications

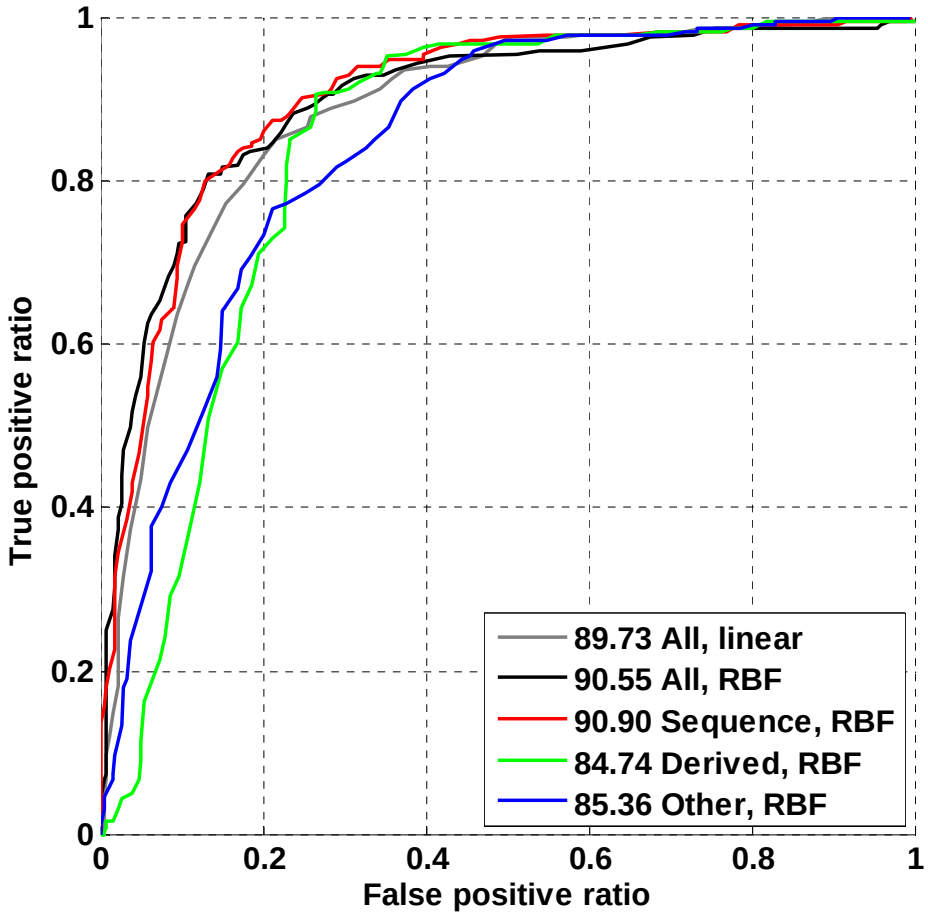
Vector kernels

- Linear $K(\boldsymbol{a},\boldsymbol{b}) = \langle \boldsymbol{a},\boldsymbol{b} \rangle^d$
- Polynomial $K(\boldsymbol{a},\boldsymbol{b}) = \left(\langle \boldsymbol{a},\boldsymbol{b} \rangle + 1 \right)^d$
- Radial basis function $K(\boldsymbol{a},\boldsymbol{b}) = \exp\left(-\frac{\|\boldsymbol{a}-\boldsymbol{b}\|^2}{\sigma^2} \right)$

Protein secretion prediction

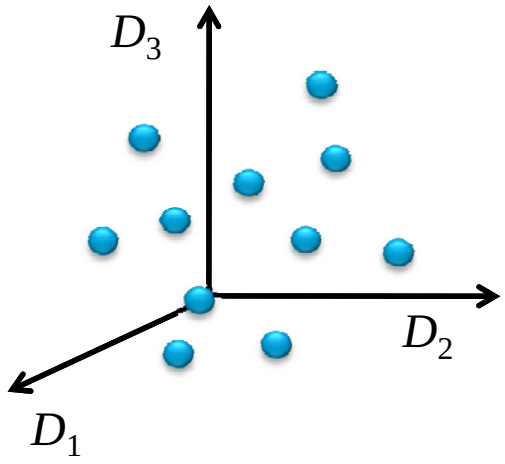
- Support vector classifier, 10-fold cross-validation error estimate

- Feature subsets
  - Sequence: amino-acid composition
  - Derived: hydrophobic & hydrophilic peaks
  - Other: length, CAI, pI, ...



Empirical kernel map

- For almost *any* other data type: empirical kernel map
  - Any distance measure (not necessarily positive definite) can be used to construct a vector with distances to a number of other objects (the "template set"), e.g. BLAST -log(E)-values to all proteins



  - This vector can then be used in a vector kernel:

Template set

$$\begin{array}{cc} & \boldsymbol{x}_1 \;\; \boldsymbol{x}_2 \;\; \boldsymbol{x}_3 \;\; \boldsymbol{x}_4 \;\; \ldots \boldsymbol{x}_n \\ \boldsymbol{a} & (\, D_{a1} \;\; D_{a2} \;\; D_{a3} \;\; D_{a4} \;\; \ldots D_{an}\,) = \boldsymbol{a'} \\ \boldsymbol{b} & (\, D_{b1} \;\; D_{b2} \;\; D_{b3} \;\; D_{b4} \;\; \ldots D_{bn}\,) = \boldsymbol{b'} \end{array} \quad \Big\} \; K(\boldsymbol{a},\boldsymbol{b}) = \langle \boldsymbol{a}', \boldsymbol{b}' \rangle$$

Kernel kernels

- Kernel addition $K(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{k} w_i K_i(\boldsymbol{a},\boldsymbol{b}), \quad w_i > 0 \quad \forall i$
- Kernel pointwise multiplication $K(\boldsymbol{a},\boldsymbol{b}) = \prod_{i=1}^{k} K_i(\boldsymbol{a},\boldsymbol{b})$
- Generalized RBF kernel

$$K(\boldsymbol{a},\boldsymbol{b}) = 1 + \exp\left(-\frac{\overbrace{K'(\boldsymbol{a},\boldsymbol{a}) - 2K'(\boldsymbol{a},\boldsymbol{b}) + K'(\boldsymbol{b},\boldsymbol{b})}^{D(\boldsymbol{a},\boldsymbol{b})}}{2\sigma^2} \right)$$

- Kernel normalization $K(\boldsymbol{a},\boldsymbol{b}) = \frac{K'(\boldsymbol{a},\boldsymbol{b})}{\sqrt{K'(\boldsymbol{a},\boldsymbol{a})K'(\boldsymbol{b},\boldsymbol{b})}}$

Kernel kernels (2)

- Convolution kernel:
  - When subkernels operate on subparts, but it is not clear which subparts
  - Try all possible decompositions into subparts:

$$K_1 \otimes K_2 \otimes \ldots \otimes K_n(\boldsymbol{a}, \boldsymbol{b}) = \sum_{\substack{\boldsymbol{a}=\boldsymbol{a}_1\boldsymbol{a}_2\ldots\boldsymbol{a}_n \\ \boldsymbol{b}=\boldsymbol{b}_1\boldsymbol{b}_2\ldots\boldsymbol{b}_n}} K_1(\boldsymbol{a}_1, \boldsymbol{b}_1) K_2(\boldsymbol{a}_2, \boldsymbol{b}_2) \ldots K_n(\boldsymbol{a}_n, \boldsymbol{b}_n)$$

Local alignment kernel

$$K_{la}(\boldsymbol{a},\boldsymbol{b}) = \sum_{n=0}^{\infty} K_{la(n)}(\boldsymbol{a},\boldsymbol{b})$$

$$K_{la(n)}(\boldsymbol{a},\boldsymbol{b}) = K_t \otimes \left(K_a \otimes K_g\right)^{(n-1)} \otimes K_a \otimes K_t(\boldsymbol{a},\boldsymbol{b})$$

Trivial kernel:

$$K_t(\boldsymbol{a},\boldsymbol{b}) = 1$$

Gap kernel

$$K_g(\boldsymbol{a},\boldsymbol{b}) = \exp\left(\beta\left(|\boldsymbol{a}| + |\boldsymbol{b}|\right)\right)$$

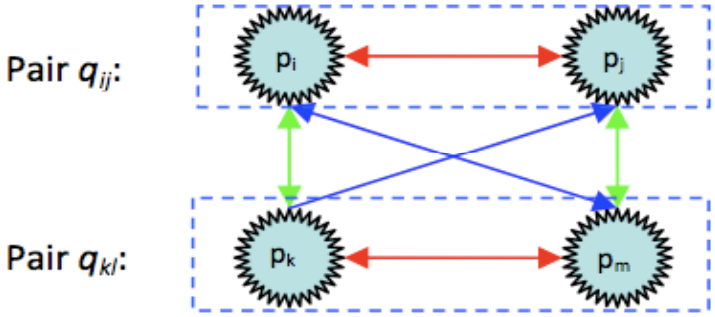
Letter alignment kernel:

$$K_a(\boldsymbol{a},\boldsymbol{b}) = \begin{cases} 0 & |\boldsymbol{a}| > 1 \vee |\boldsymbol{b}| > 1 \\ \exp\left(\beta S(\boldsymbol{a},\boldsymbol{b})\right) & \text{otherwise} \end{cases}$$

with S the substitution cost, e.g. BLOSUM

Kernel kernels (3)

- Pairwise kernel:
  - Kernel between pairs of objects rather than individual ones
  - Alternative to linear vector kernel on pair kernels



Protein similarity kernel:

$$k_s(q_{ij}, q_{km}) = \langle k_{ps}(p_i, p_j), k_{ps}(p_k, p_m) \rangle$$

Pairwise kernel (similarity between pairs):

$$k_{pw}(q_{ij}, q_{km}) = k_{ps}(p_i, p_k) k_{ps}(p_j, p_m) + k_{ps}(p_i, p_m) k_{ps}(p_j, p_k)$$

Set kernels

- Let $\mu(A)$ be a probability distribution over sets A on a domain D (for example $\mu(A) = I_A$, the indicator function)

- Intersection kernel: $$K_{\cap}(A,B) = \int_D I_A(a) I_B(a) d\,\mu(a)$$

- Union complement kernel: $$\tilde{K}(A,B) = \int_D I_{D\setminus A}(a) I_{D\setminus B}(a) d\,\mu(a)$$

- Agreement kernel: $$K(A,B) = \tilde{K}(A,B) + K_{\cap}(A,B)$$

- Example: documents represented as sets of words; $\mu(A)$ can be measure of “uniqueness” of word

String kernels

- Spectrum kernel:
  - create a dictionary of all k-mers
  - construct vector with #occurences of each k-mer
  - use this in a linear kernel
- Similar:
  - versions with gaps, mismatches
  - mixed spectrum kernel, sum over all $k = 1,\ldots,d$
  - motif kernel, look for specific set(s) of k-mers
- Example, $k = 4$:

a = **aabbbababa**

b = **abbaabbbab**

aabb	abba	bbab	baba	abab	bbaa	baab
1	1	1	2	1	0	0
1	2	1	0	0	1	1

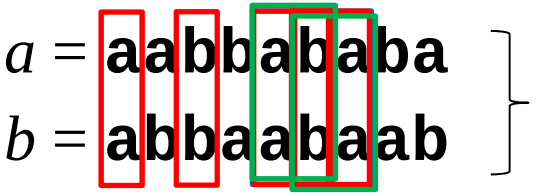
$$K(a,b) = 4$$

String kernels (2)

- Weighted degree kernel: take position into account
  - count number of matching k-mers at identical position, for $k = 1,\ldots,d$
  - discount by length of match, i.e. $w = d - k+1$

- Example, $d = 4$:

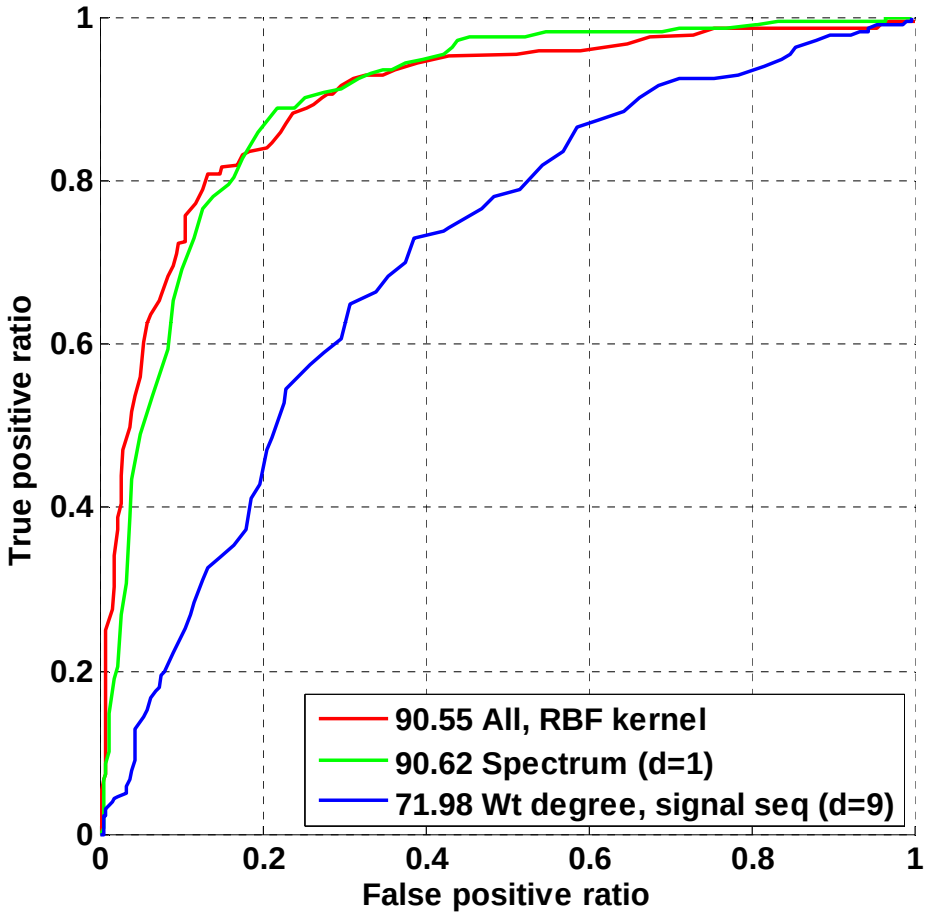
Christian Widmer
TB6, Thu 12.30



$$K(a,b) = (4-1+1)\cdot 5 + (4-2+1)\cdot 2 + (4-3+1)\cdot 1 = 20+4+1 = 25$$

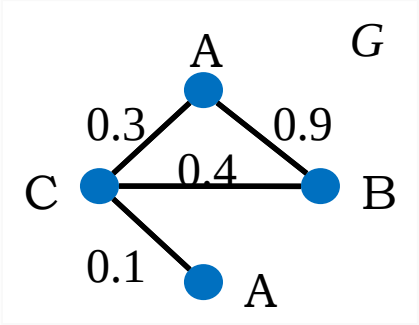
Protein secretion prediction

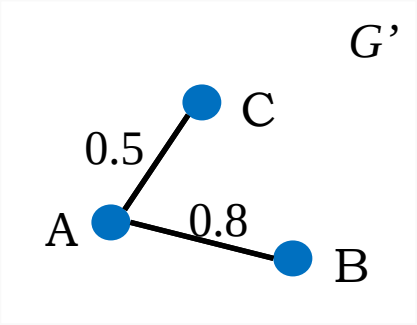
- String kernel on protein sequences slightly better than kernels on original feature vectors



Advanced kernels

- Graph kernels
  - encode graph as string
  - compare random walks



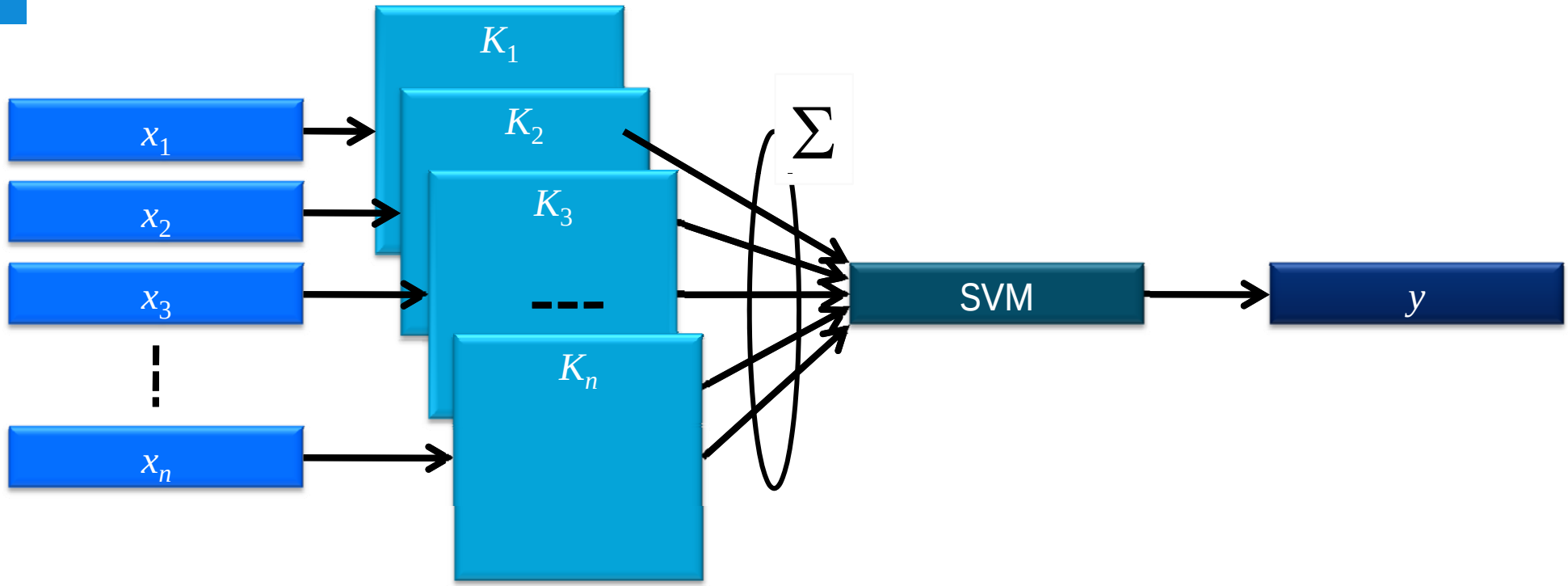


- Generative model kernels:
  - $K(\boldsymbol{a},\boldsymbol{b}) = P(\boldsymbol{a},\boldsymbol{b} \mid M)$: joint probability of $\boldsymbol{a}$ and $\boldsymbol{b}$ given a model M (for example, a hidden Markov model)
  - Fisher kernel

- Etc. etc.

Kernel combination

- Zoo of kernels, applicable for different data sources...



- Combine, for example by simply adding kernel matrices

Integrative bioinformatics

Kernel-based algorithms

Kernels

Kernel combination

Example applications

Weighted kernel combination

- Combination: weighted sum of (normalised) kernel matrices

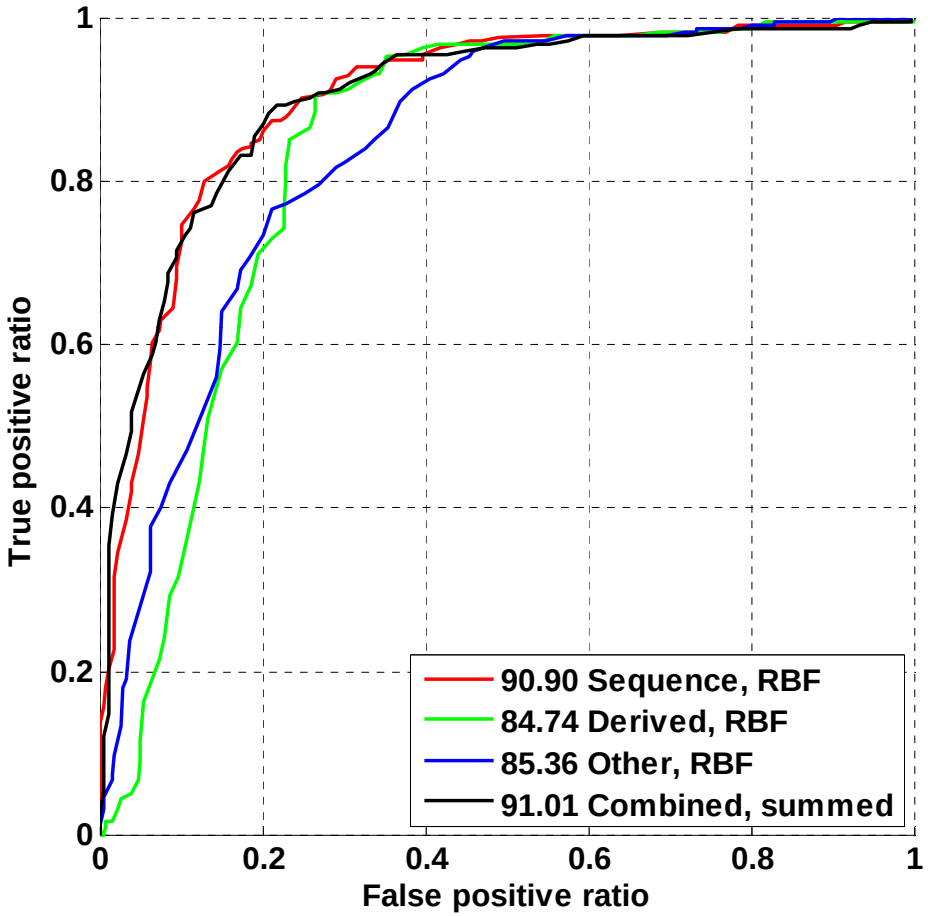
$$K_k'(\boldsymbol{a},\boldsymbol{b}) = \frac{K_k(\boldsymbol{a},\boldsymbol{b})}{\sqrt{K_k(\boldsymbol{a},\boldsymbol{a})K_k(\boldsymbol{b},\boldsymbol{b})}} \qquad K_{combined}(\boldsymbol{a},\boldsymbol{b}) = \sum_{k=1}^{n} \mu_k K_k'(\boldsymbol{a},\boldsymbol{b})$$

- Goals:
  - Improve performance
  - Determine important features
  - Sparser model

- Simplest approach: $\mu_k = 1 \ \forall k$
  - Note: sum of linear kernels is equal to feature space concatenation

Protein secretion prediction

- Kernels normalised and summed: slightly better than best individual kernel
- Can we optimize the weights μ_k ?



Weight optimisation

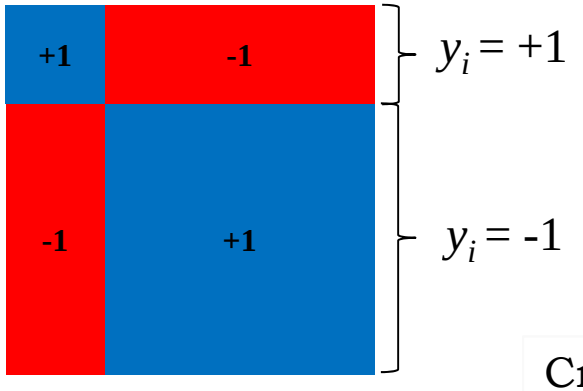
Szafranski *et al.*, *Machine Learning* 2010

1. Filter approach: optimize a derived criterion

- Example: maximise the kernel alignment w.r.t. μ

$$A\left(K_{combined}\right) = \frac{\left\langle K_{combined}, K_{ideal} \right\rangle_F}{\sqrt{\left\langle K_{combined}, K_{combined} \right\rangle_F \left\langle K_{ideal}, K_{ideal} \right\rangle_F}}$$

where $K_{ideal} = \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$ is the ideal kernel matrix, and $\left\langle K_1, K_2 \right\rangle_F = \sum_i \sum_j K_{1_{ij}} K_{2_{ij}}$ is the Frobenius norm and $K_{combined}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{k=1}^{n} \mu_k K_k^{'}(\boldsymbol{a}, \boldsymbol{b})$



Cristianini *et al.*, *NIPS* 2001

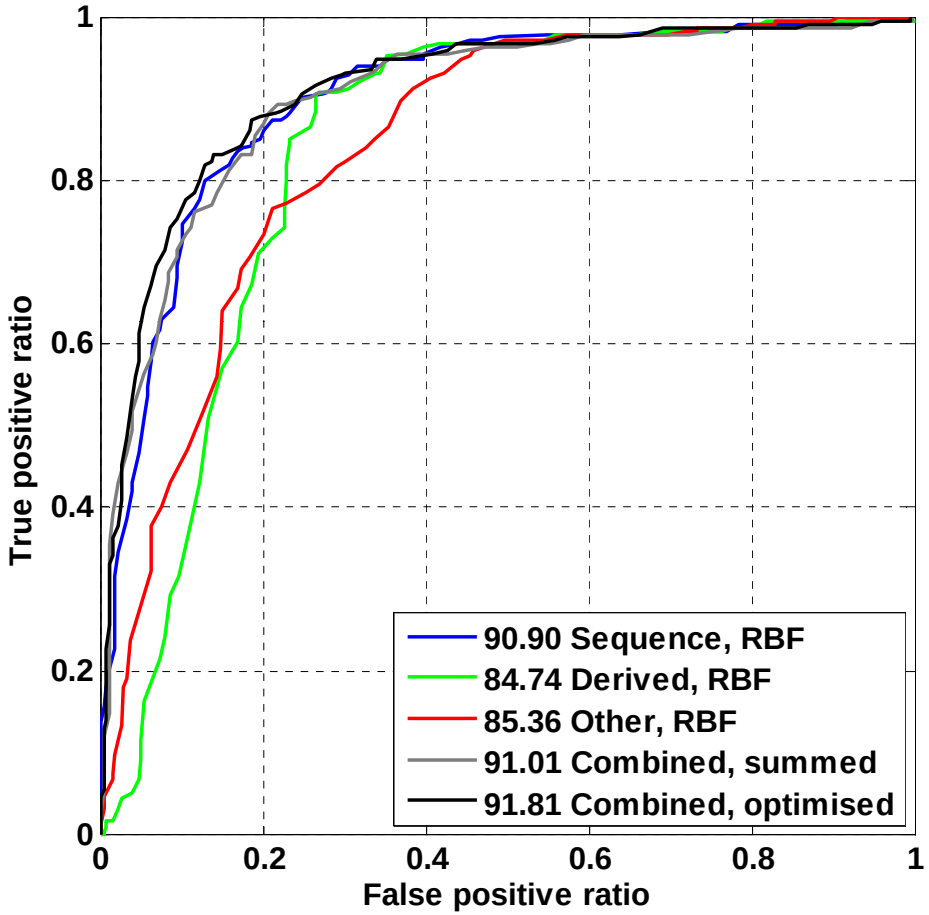
Weight optimisation (2)

2. Wrapper approach: optimise SVM performance
   - grid search
   - evolutionary algorithms
   - gradient descent, using estimated derivative of the generalisation error, E
     1. set initial guess for μ, use to combine kernels
     2. train SVM on combined kernel
     3. update μ to minimise E, recombine kernels
     4. go to 2

Chapelle *et al.*, *Machine Learning* 2002

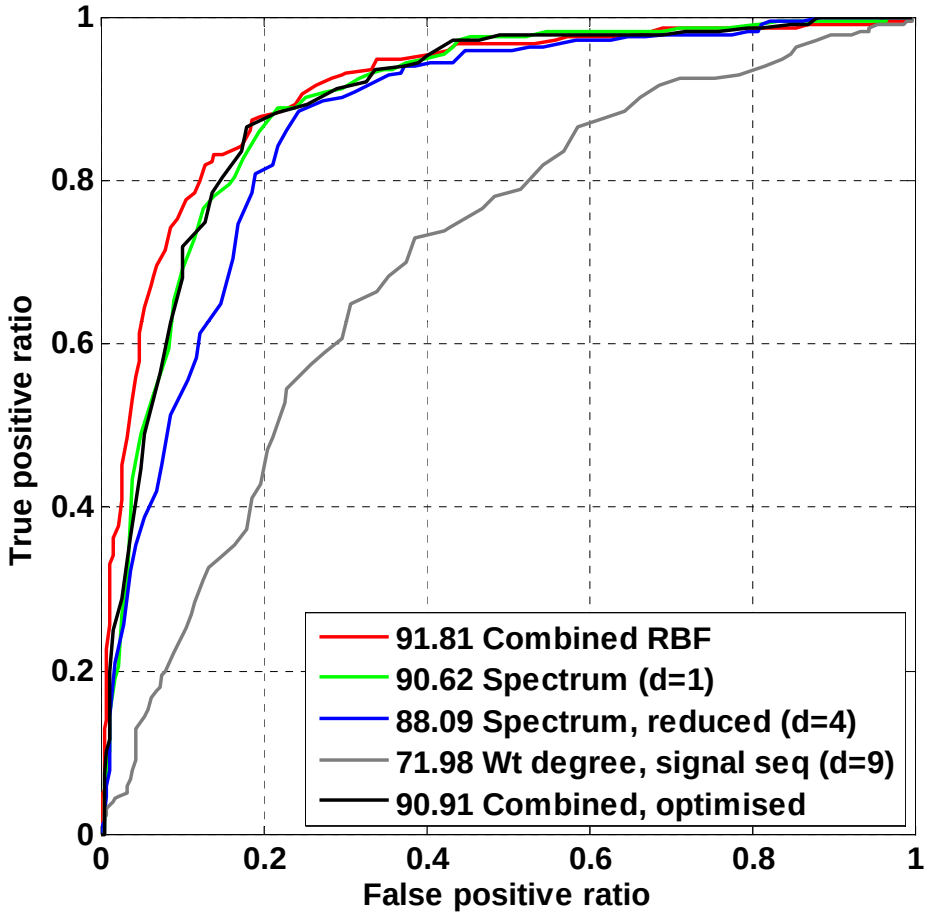
Protein secretion prediction

- Three RBF kernels combined, weights set by grid search:
  - Sequence: $\mu_1 = 1.0$
  - Derived: $\mu_2 = 0.0$
  - Other: $\mu_3 = 0.5$



Protein secretion prediction (2)

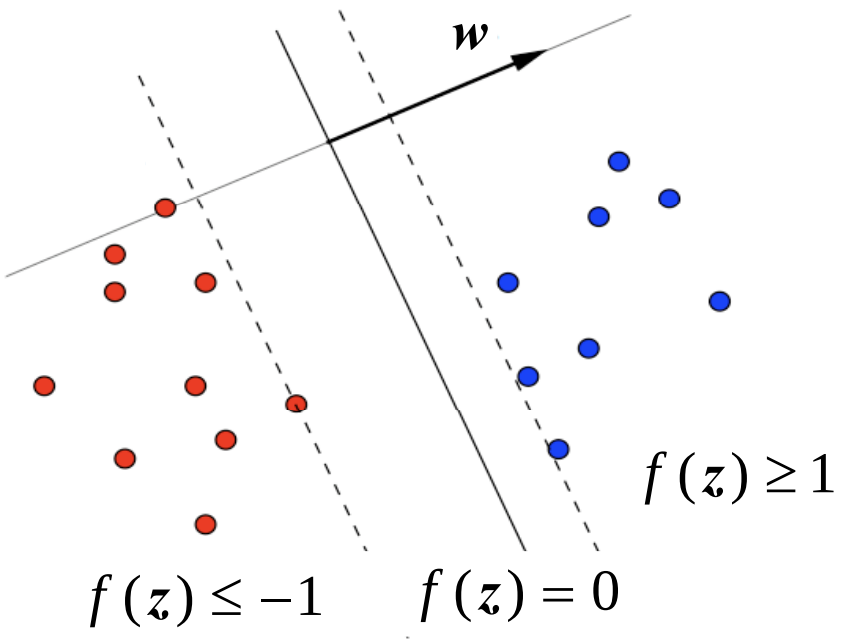
- Adding specific kernels does not help here



Weight optimisation (3)

3. Embedded approach: directly optimize SVM margin

- Multiple kernel learning (MKL):



$$\max_{\alpha} \sum_{i=1}^{|X|} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{|X|} y_i y_j \alpha_i \alpha_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$$

$$\alpha_i \geq 0, \quad \forall i$$

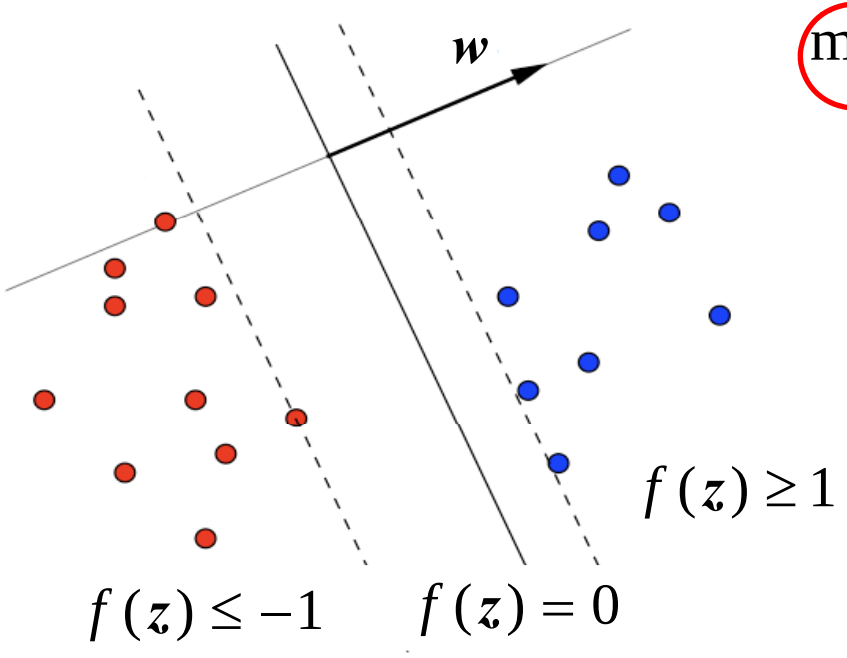
$$\sum_{i=1}^{|X|} \alpha_i y_i = 0$$

Lanckriet *et al.*, *ICML* 2002

Weight optimisation (3)

3. Embedded approach: directly optimize SVM margin

- Multiple kernel learning (MKL):



$$\min_{\mu} \max_{\alpha} \sum_{i=1}^{|X|} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{|X|} y_i y_j \alpha_i \alpha_j \sum_{k=1}^{n} \mu_k K_k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\alpha_i \geq 0, \quad \forall i$$

$$\sum_{i=1}^{|X|} \alpha_i y_i = 0$$

$$\mu_k \geq 0, \quad \forall k$$

$$\sum_{k=1}^{n} \mu_k \operatorname{tr}(K_k) = c$$

Lanckriet *et al.*, *ICML* 2002

Multiple kernel learning

- Original L_2-SVM primal:

$$\min \|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{|X|}\xi_i$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \geq +(1-\xi_i), \quad y=+1$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \leq -(1-\xi_i), \quad y=-1$$

$$\xi_i \geq 0, \quad \forall i$$

- Corresponding MKL primal:

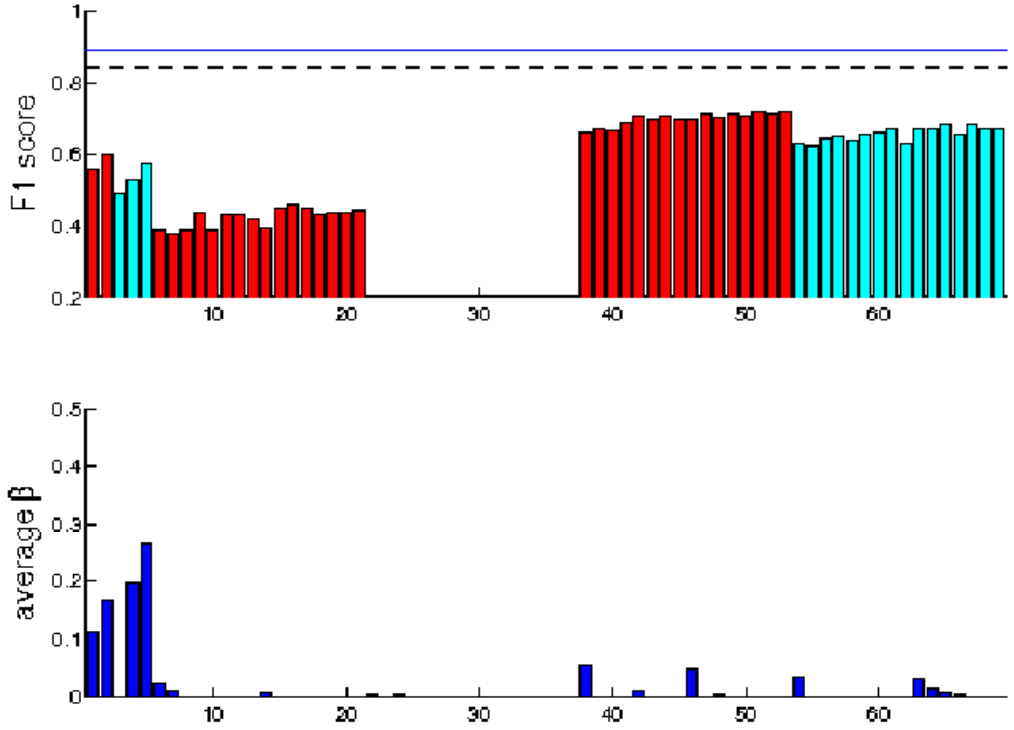
$$\min \left(\sum_{k=1}^{n}\|\boldsymbol{w}_k\|_2\right)^2 + C\sum_{i=1}^{|X|}\xi_i$$

  - L_1-norm over per-kernel L_2 norm
  - Promotes sparsity at kernel level, similar to group-LASSO regression; kernels with non-zero weight are "support kernels"

Bach *et al.*, *ICML* 2004

Multiple kernel learning (2)

- Example: protein localisation prediction, with 69 kernels in six groups

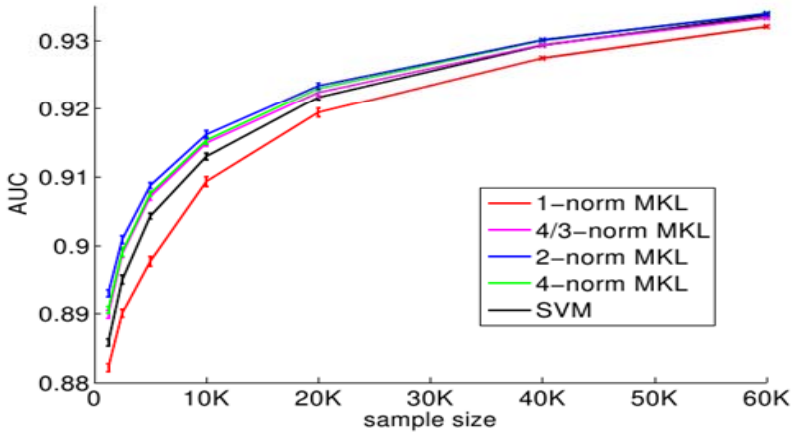


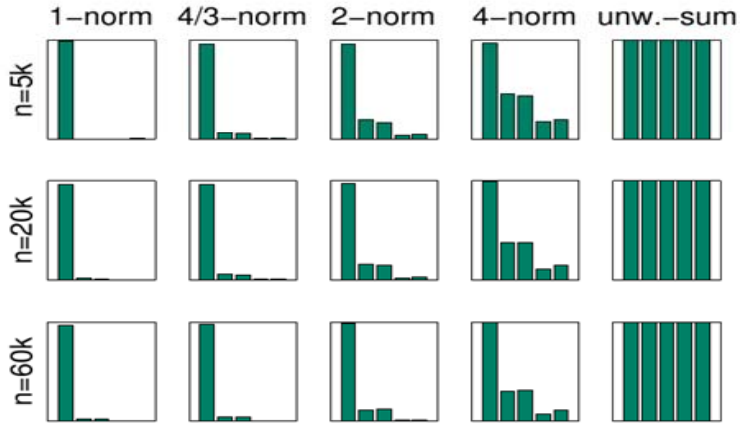
Ong & Zien, *WABI* 2008

Multiple kernel learning (3)

- Sparsity does not always bring better performance
  - Move from L_1 norm to L_p norm

Kloft *et al.*, *NIPS* 2008





- Other extensions:
  - Localized multiple kernel learning (Gönen *et al.*, *ICML* 2008)
  - Nonlinear kernel combinations (Cortes *et al.*, *NIPS* 2009)
  - More complex sparsity structures (Szafranski *et al.*, *M. Learning* 2010)

Integrative bioinformatics

Kernel-based algorithms

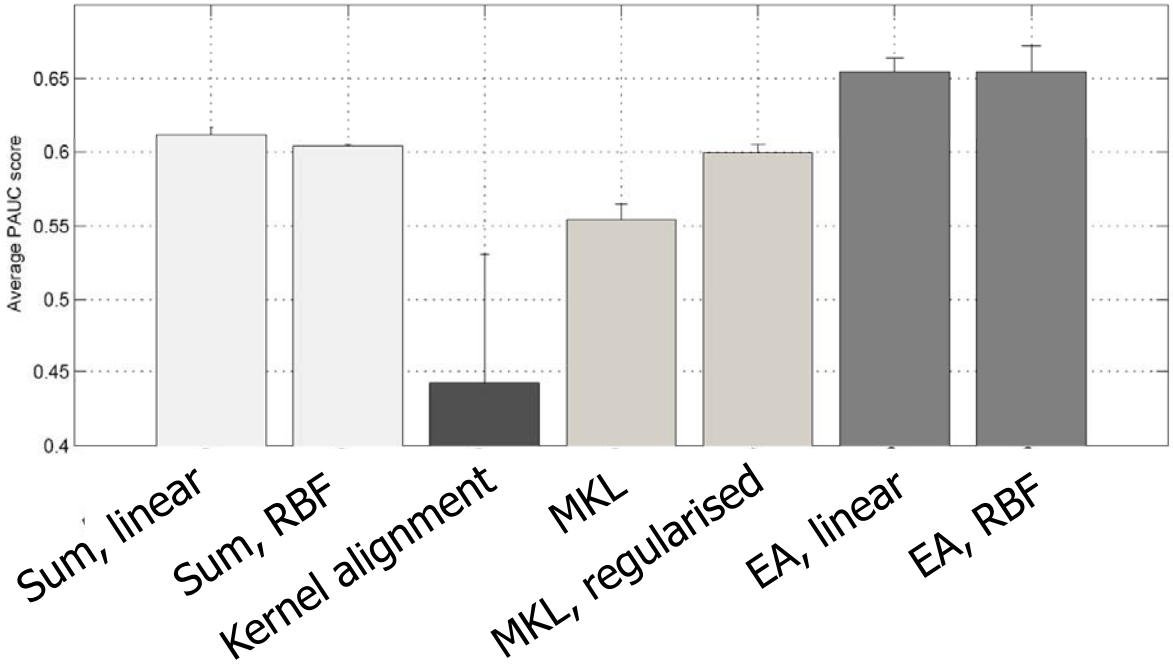
Kernels

Kernel combination

Example applications

A. Protein-protein interaction

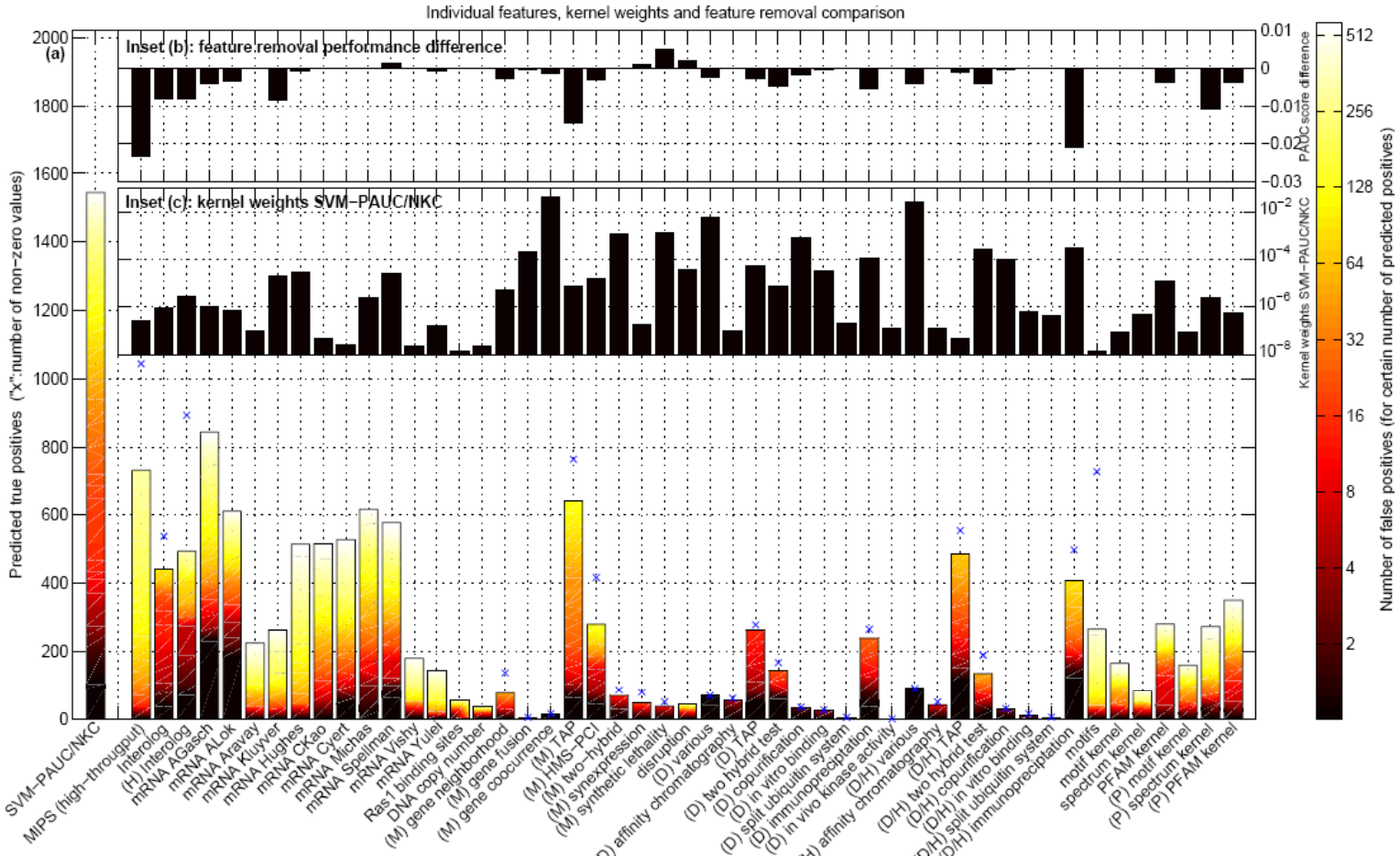
- Input x : homology, co-expression, co-localization, etc. (49)
- Output y : protein interaction (0/1)



- EA = evolutionary algorithm

Hulsman *et al.*, *IEEE TCBB* 2009

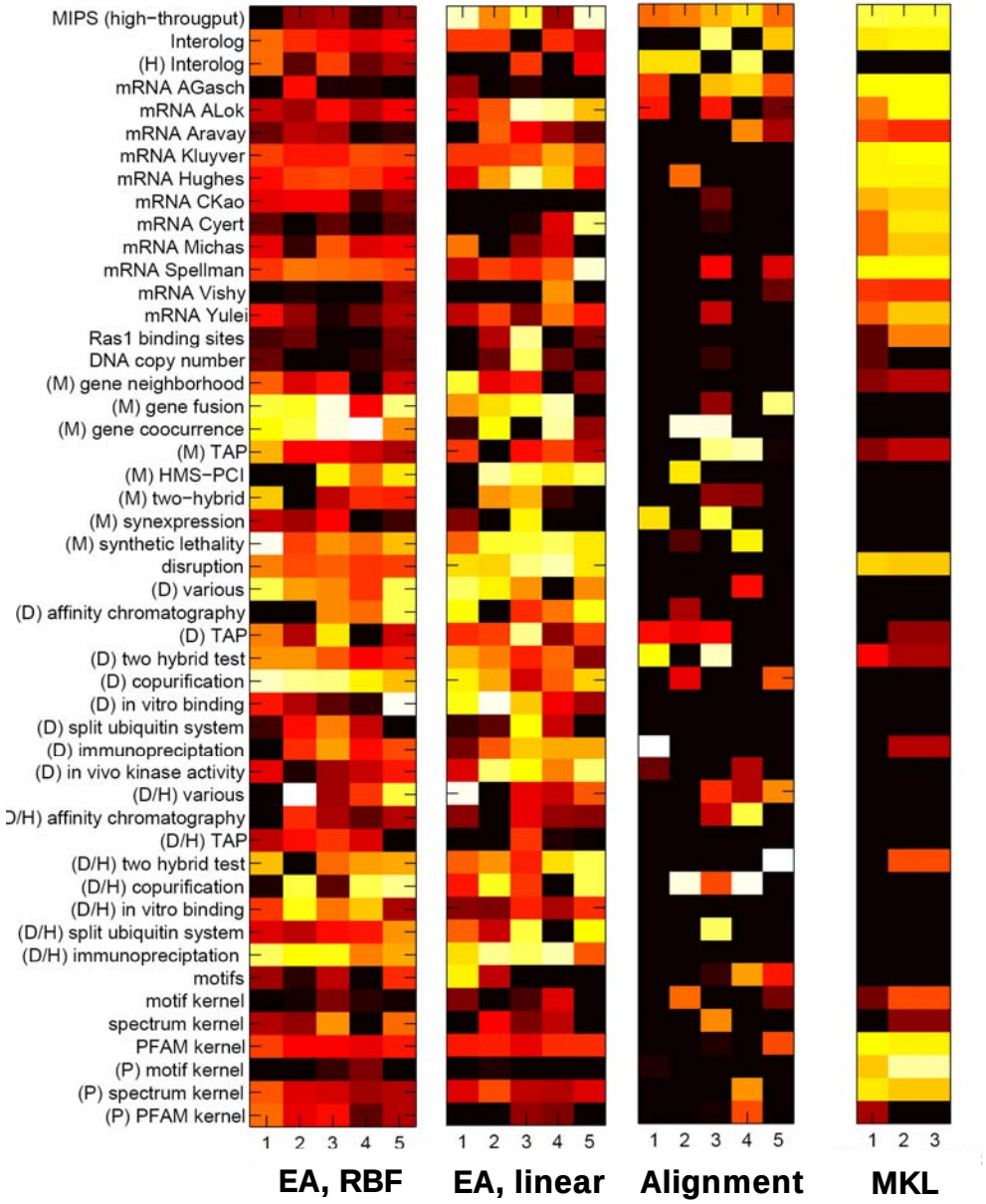
A. Protein-protein interaction (2)



A. Protein-protein interaction (3)

- Alignment and MKL give sparse solutions

  (columns are runs)

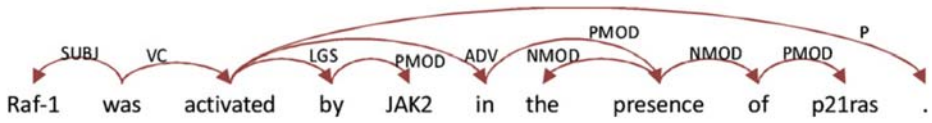


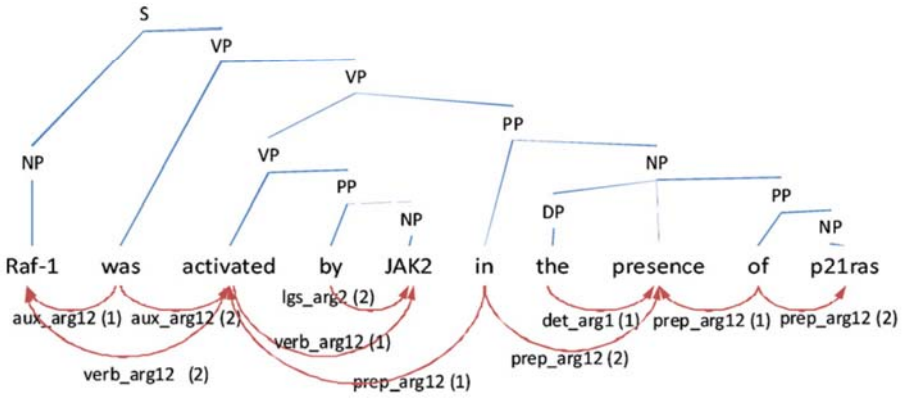
B. PPI from literature

- Input x : sentence with two identified proteins, e.g. *"**Raf-1** was activated by **JAK2** in the presence of p21ras"*
- Output y : protein interaction (0/1)

Miwa *et al.*, *Int J Med Inf* 2009

- Two parsers result in trees:



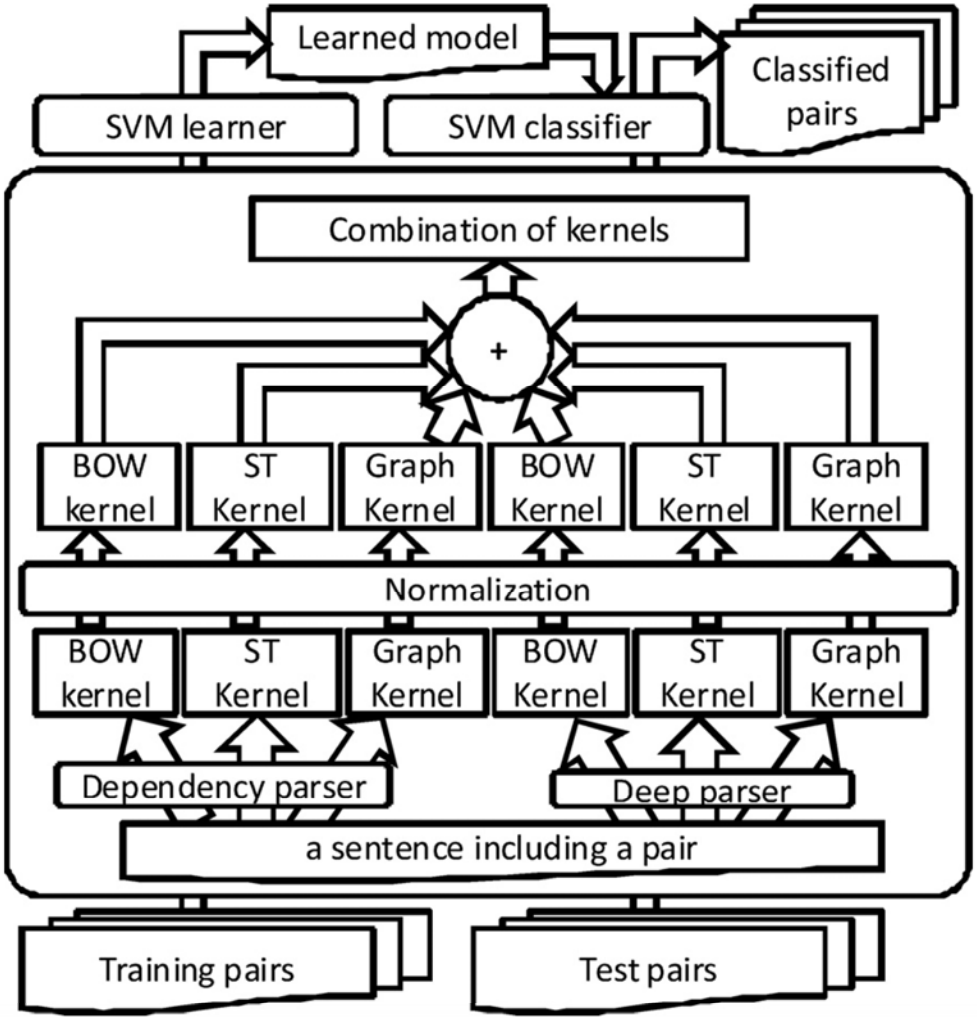


B. PPI from literature (2)

Miwa *et al.*,
Int J Med Inf 2009

- Three kernels applied to each parse:
  - Bag-of-words (set kernel)
  - Subset tree kernel (#common subtrees)
  - Graph kernel (random walks)

	L	
	F	AUC
C	30.2	-
B	52.7	0.822
T	55.1	0.799
G	59.1	0.854
T+B	58.5	0.849
G+B	57.0	0.847
T+G	62.0	0.873
T+G+B	59.9	0.863



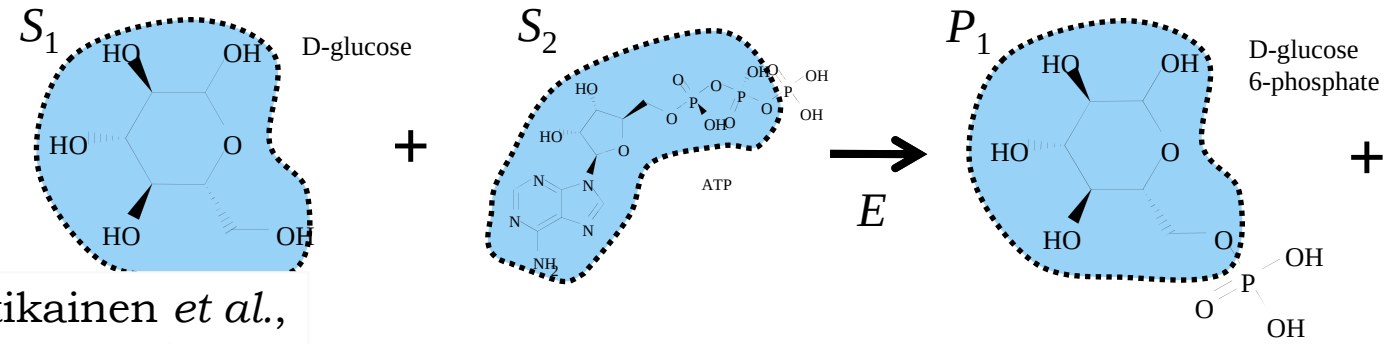
C. Enzyme function prediction

- Input $\boldsymbol{x}$: sequence representation of enzyme E
- Output $\boldsymbol{y}$: representation of metabolic reaction R
  (structured output prediction: density over R's)

$$K(R^a, R^b) = K(S^a, S^b)K(P^a, P^b)$$

$$K(S^a, S^b) = \sum_{i,j} K_{molecule}(S_i^a, S_j^b)$$

$K_{molecule}$ counts number of common small subgraphs



Astikainen *et al.*, *ICB* 2010

D-glucose + ATP → D-glucose 6-phosphate + ADP

Juho Rousu, TA2, Thu 10.15

CONCLUSIONS

Conclusions

- Integrative bioinformatics:
  - combining prior knowledge & measurements to infer and annotate molecular interaction networks
  - heterogeneous data calls for intermediate integration
- Kernels are ideal vehicles for this
  - many standard algorithms have been “kernelised”
  - a wide variety of applicable kernels is available
  - theory of kernel algorithms and combination is well-developed, and still ongoing research
- Try it yourself!
  Good start: **`http://www.shogun-toolbox.org/`**

But…

- Kernel combination is still no free lunch:
  - kernel normalisation is essential
  - kernel combinations can overtrain as well
  - kernel and classifier parameters (C, σ) have large impact, but require intensive procedures to set
  - computationally intensive, particularly for genome-wide datasets

- In practice:
  - measurement coverage & bias are problematic
  - choosing the right kernel(s) to combine is still an art
  - as often in bionformatics, the KISS principle applies: simple summation often already works quite well

Thank you!

http://bioinformatics.tudelft.nl/