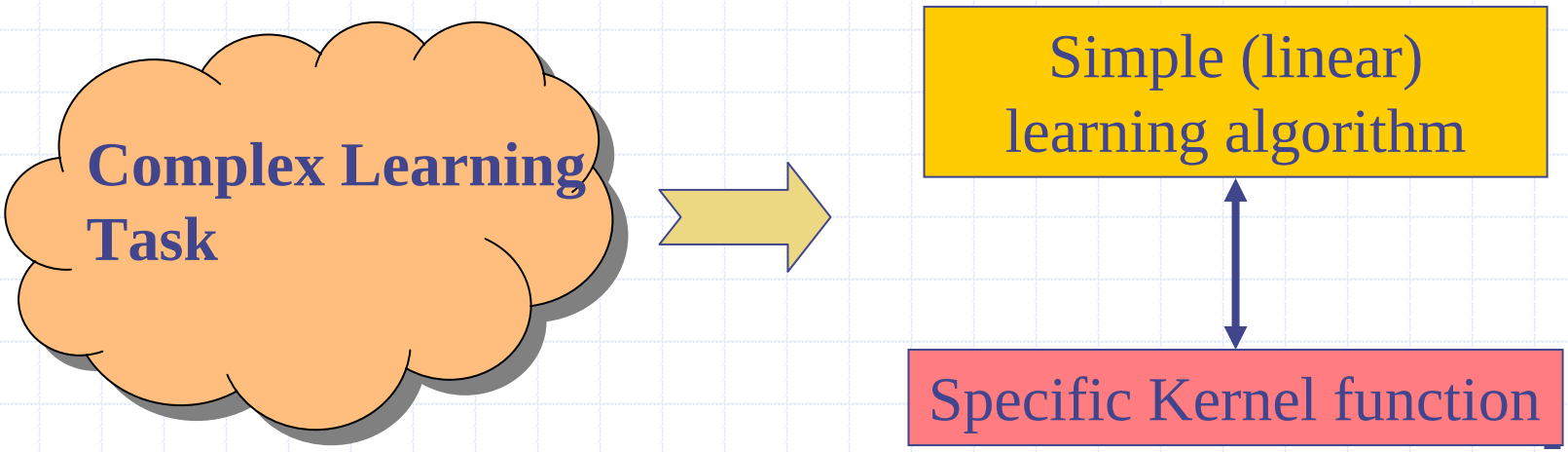
Marginalized Kernels & Diffusion Kernels

Max Planck Institute for Biological Cybernetics

Koji Tsuda

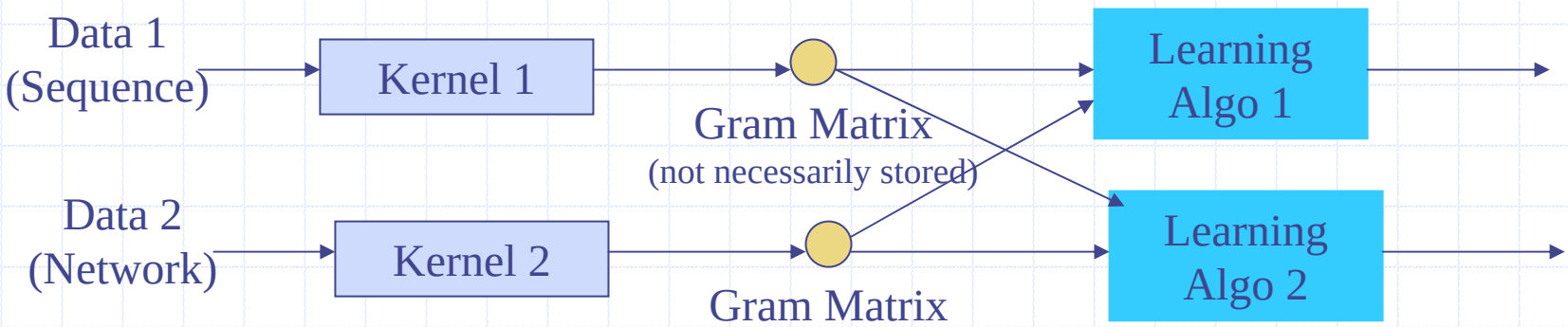
Kernels and Learning

- In Kernel-based learning algorithms, problem solving is now decoupled into:
  - A general purpose learning algorithm (e.g. SVM, PCA, …) – Often linear algorithm
  - A problem specific kernel



Current Synthesis

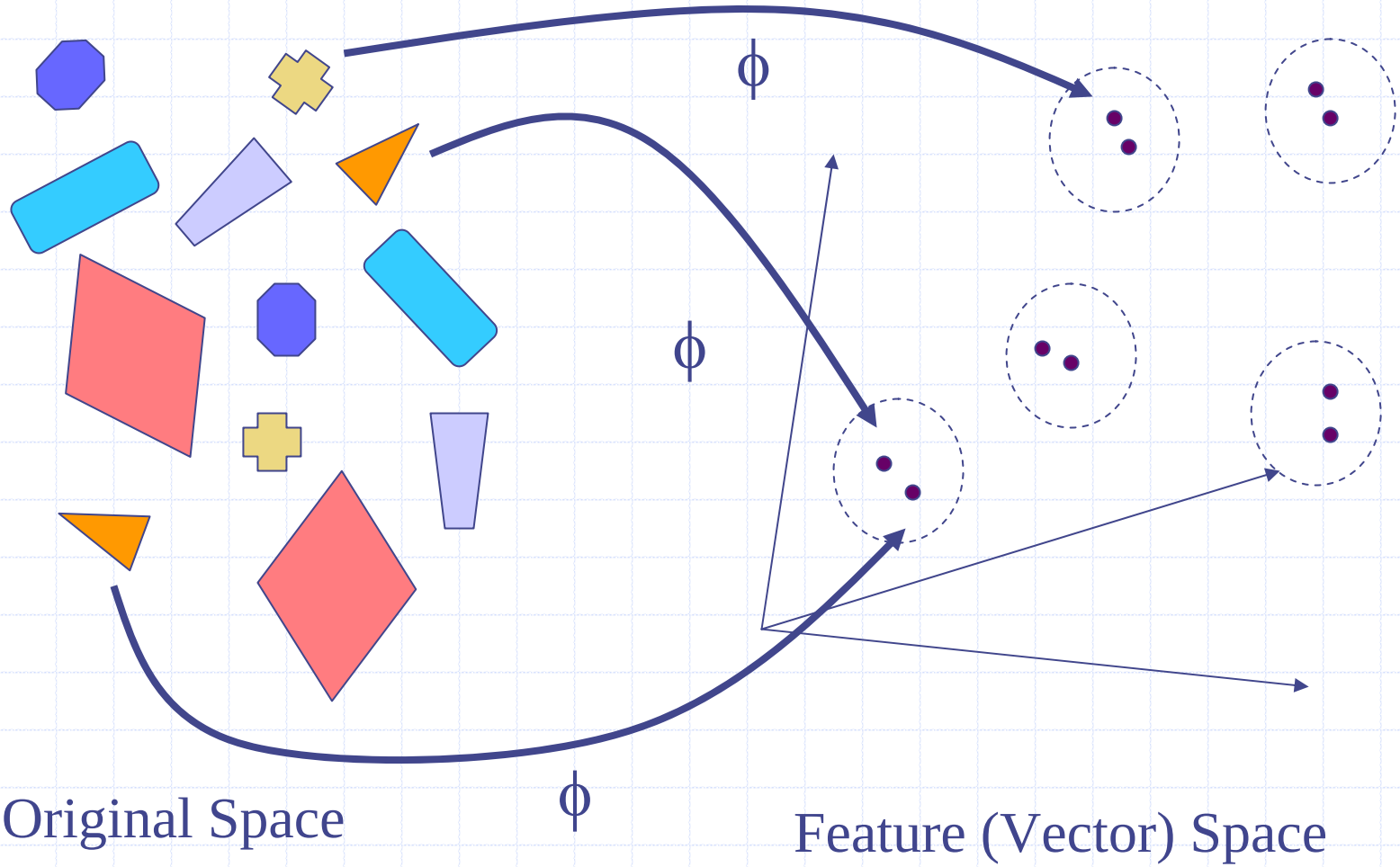
- Modularity and re-usability
  - Same kernel ,different learning algorithms
  - Different kernels, same learning algorithms



Lectures so far

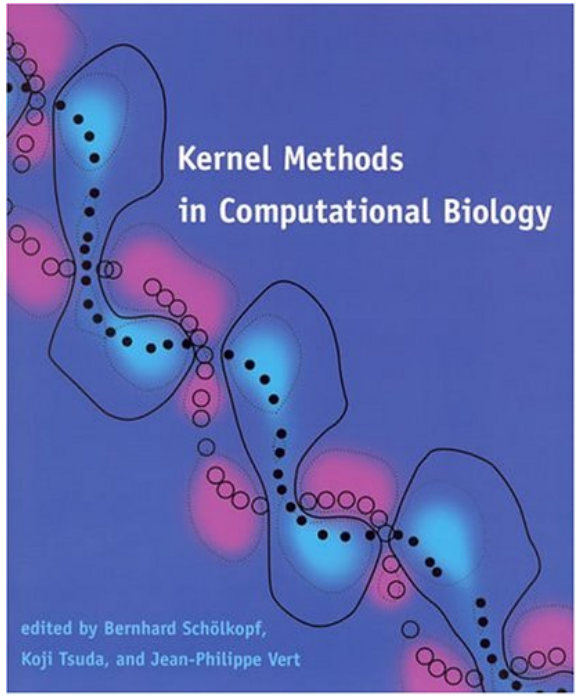
- Kernel represents the similarity between two objects, defined as the dot-product in the feature space
- Various String Kernels
- Importance of Positive Definiteness

Kernel Methods : the mapping



Family of kernels

- **Kernels for biological sequences**
  - Spectrum kernel (Leslie et al., 2002)
  - **Marginalized kernel** (Tsuda et al., 2002)
  - Profile kernel (Kuang et al., 2004)
  - Local alignment kernel (Saigo et al., 2004)



Kernel Methods in Computational Biology, MIT Press, 2004

- **Tree Kernels**
  - Kernel for phylogenetic profiles (Vert, 2002)
  - Kernel for natural language (Suzuki et al., 2003)
  - Kernel for RNA sequences (Kin et al., 2002)

Family of kernels

- **Kernels for nodes in a network**
  - **Diffusion kernel** (Kondor and Lafferty, 2002)
  - Locally constrained diffusion kernel (Tsuda and Noble, 2004)

- **Graph Kernels**
  - Marginalized Graph Kernels (Kashima et al., 2003)
  - MGK without tottering (Mahe et al., 2004)
  - Acyclic Pattern Kernels (Horvath et al., 2004)

Overview of this lecture

- **Marginalized kernels**
  - General idea about defining kernels using latent variables
  - An example in string kernel

- **Diffusion Kernels**
  - Closeness between nodes of a network
  - Used for function prediction of proteins based on biological networks (protein-protein

- **Supervised Network Inference Using Kernel Methods**
  - If time allows..

Marginalized kernels

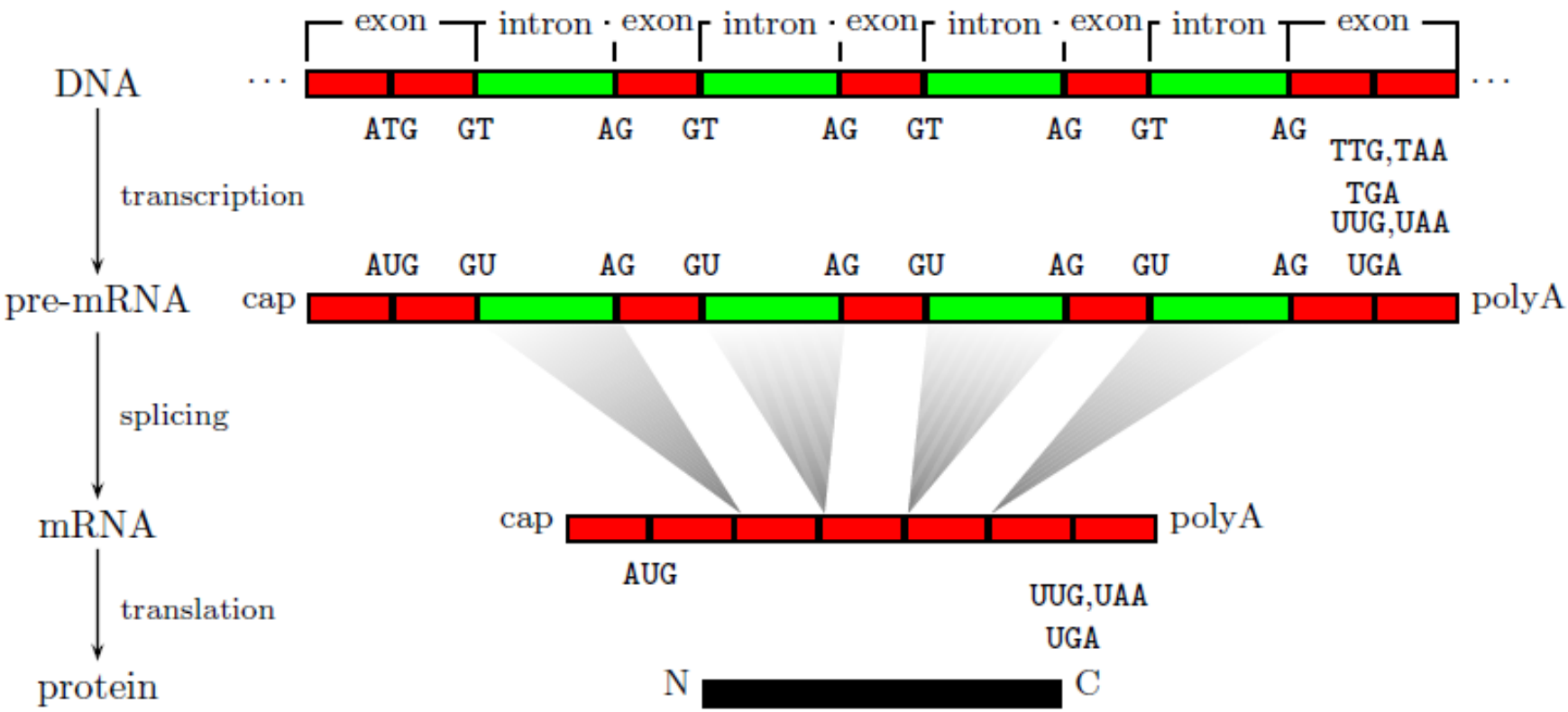
K. Tsuda, T. Kin, and K. Asai.
Marginalized kernels for biological sequences
Bioinformatics, 18(Suppl. 1):S268-S275, 2002.

Biological Sequences: Classification Tasks

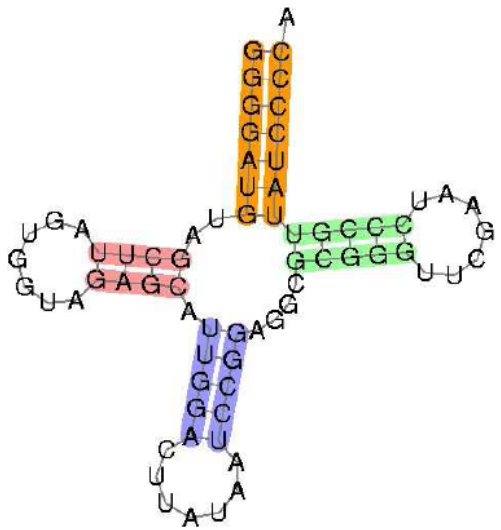
- DNA sequences (A,C,G,T)
  - Gene Finding, Splice Sites
- RNA sequences (A,C,G,U)
  - MicroRNA discovery, Classification into Rfam families
- Amino Acid Sequences (20 symbols)
  - Remote Homolog Detection, Fold recognition

Structures hidden in sequences (I)

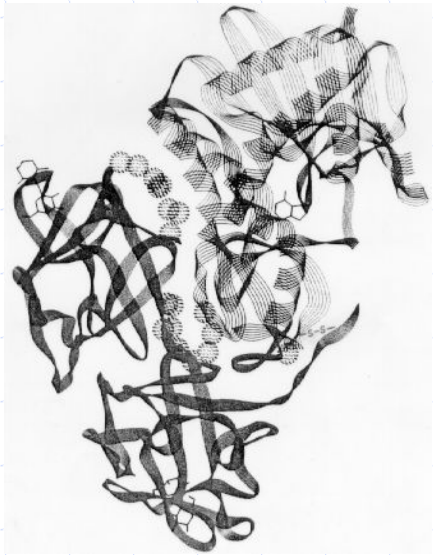
- Exon/intron of DNA (Gene)



Structures hidden in sequences (II)



RNA
Secondary
Structure



Protein
3D Structures

- It is crucial to infer hidden structures and exploit them for classification

Hidden Markov Models

- Visible Variable $\boldsymbol{x} = (x_1, \ldots, x_m)$: Symbol Sequence
- Hidden Variable $\boldsymbol{h} = (h_1, \ldots, h_m)$: Context

h: 1 2 2 1 2 2 1 2 2

x: A C G G T T C A A

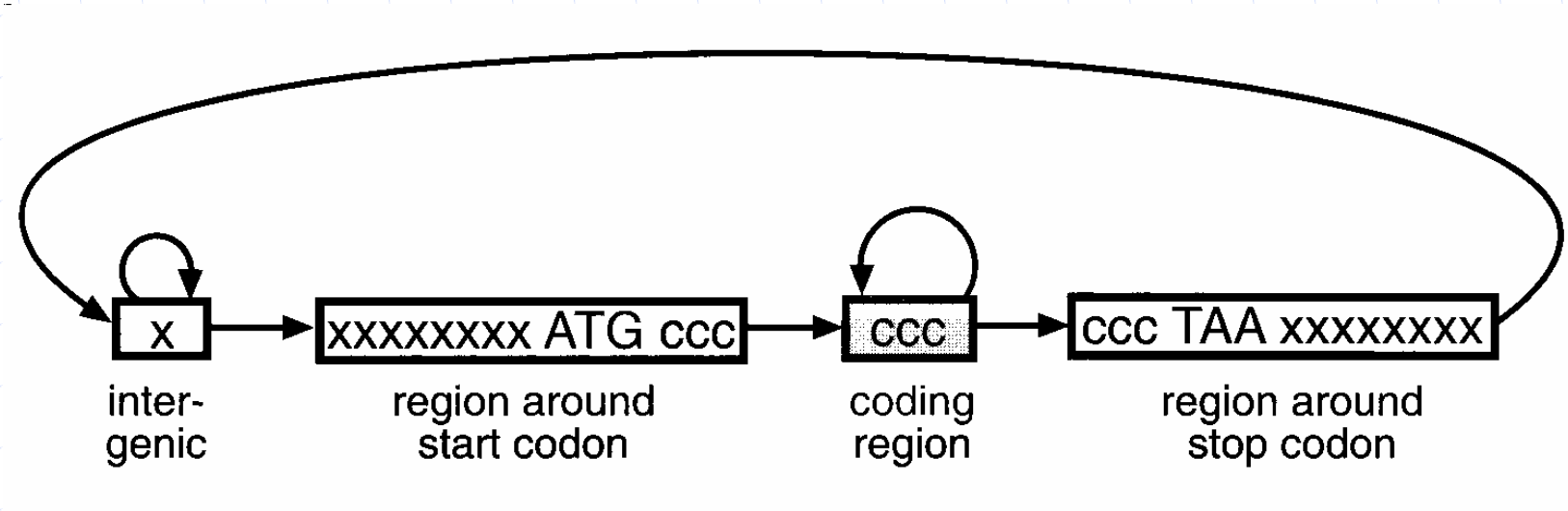
- HMM has parameters $\boldsymbol{\theta}$
  - Transition Probability
  - Emission Probability

- HMM models the joint probability $p(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{\theta})$

HMM for gene finding

Engineered HMM:
Some parameters are set to constants a priori

Reflect prior knowledge about the sequence



Training Hidden Markov Models

- Training examples consist of string-context pairs

$$(\boldsymbol{x}_1, \boldsymbol{h}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{h}_n)$$

  - E.g., Fragments of DNA sequences with known splice sites

- Parameters are estimated by the maximizing likelihood

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\boldsymbol{x}_i, \boldsymbol{h}_i | \boldsymbol{\theta})$$

Using trained hidden Markov models to estimate the context

- A trained HMM can compute the posterior probability
- Given the sequence x, what is the probability of the context h?

$$p(\boldsymbol{h}|\boldsymbol{x})$$

x: A C C T G T A A A

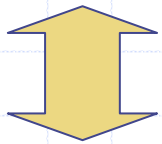
h: 1 2 1 2 2 2 2 1 1	0.0003
h: 2 2 1 1 1 1 2 1 1	0.0006

- You can never predict the context perfectly!

Kernels for Sequences

- Similarity between sequences of different lengths

ACGGTTCAA



ATATCGCGGGAA

- How do you use the trained HMM for computing the kernel?

Count Kernel

- Inner product between symbol counts

	A	C	G	T
ACGGTTCAA	3	2	2	2
ATATCGCGGGAA	4	2	4	2

- Extension: Spectrum kernels (Leslie et al., 2002)
  - Counts the number of k-mers (k-grams) efficiently

- Not good for sequences with frequent context change
  - E.g., coding/non-coding regions in DNA

Hidden Markov Models for Estimating Context

- Visible Variable $\boldsymbol{x} = (x_1, \ldots, x_m)$: Symbol Sequence
- Hidden Variable $\boldsymbol{h} = (h_1, \ldots, h_m)$: Context
- HMM can estimate the posterior probability of hidden variables $p(\boldsymbol{h}|\boldsymbol{x})$ from data

h: 1 2 2 1 2 2 1 2 2

x: A C G G T T C A A

Marginalized kernels

- Design a joint kernel $K_z(\boldsymbol{z}, \boldsymbol{z}')$ for combined $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{h})$
  - Hidden variable is not usually available
  - Take expectation with respect to the hidden variable
- The marginalized kernel for visible variables

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\boldsymbol{h} \in \mathcal{H}} \sum_{\boldsymbol{h}' \in \mathcal{H}} p(\boldsymbol{h}|\boldsymbol{x}) p(\boldsymbol{h}'|\boldsymbol{x}') K_z(\boldsymbol{z}, \boldsymbol{z}')$$

Designing a joint kernel for sequences

- Symbols are counted separately in each context

h: 1 2 2 1 2 2 1 2 2

x: A C G G T T C A A

(A,1) = 1 (C,1) = 1 (G,1) = 1 (T,1) = 0

(A,2) = 2 (C,2) = 1 (G,2) = 1 (T,2) = 2

- $c_{k\ell}(z)$:count of a combined symbol (k,l)

- Joint kernel: count kernel with context information

$$K_z(z, z') = \sum_{k=1}^{n_x} \sum_{\ell=1}^{n_h} c_{k\ell}(z) c_{k\ell}(z')$$

Marginalization of the joint kernel

- Joint kernel

$$K_z(\boldsymbol{z}, \boldsymbol{z}') = \sum_{k=1}^{n_x} \sum_{\ell=1}^{n_h} c_{k\ell}(\boldsymbol{z}) c_{k\ell}(\boldsymbol{z}')$$

- Marginalized count kernel

$$\begin{aligned} K(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} p(\boldsymbol{h}|\boldsymbol{x}) p(\boldsymbol{h}'|\boldsymbol{x}') K_z(\boldsymbol{z}, \boldsymbol{z}') \\ &= \sum_{k=1}^{n_x} \sum_{\ell=1}^{n_h} \gamma_{kl}(\boldsymbol{x}) \gamma_{kl}(\boldsymbol{x}') \end{aligned}$$

γ_{kl} is a marginalized count $\gamma_{kl}(\boldsymbol{x}) = \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{x}) c_{k\ell}(\boldsymbol{z})$

Computing Marginalized Counts from HMM

- Marginalized count is described as

$$\gamma_{kl}(\boldsymbol{x}) = \frac{1}{m} \sum_{i=1}^{m} \sum_{h_i=1}^{n_h} p(h_i|\boldsymbol{x}) I(x_i = k, h_i = \ell).$$

- Posterior probability of i-th hidden variable is efficiently computed by dynamic programming

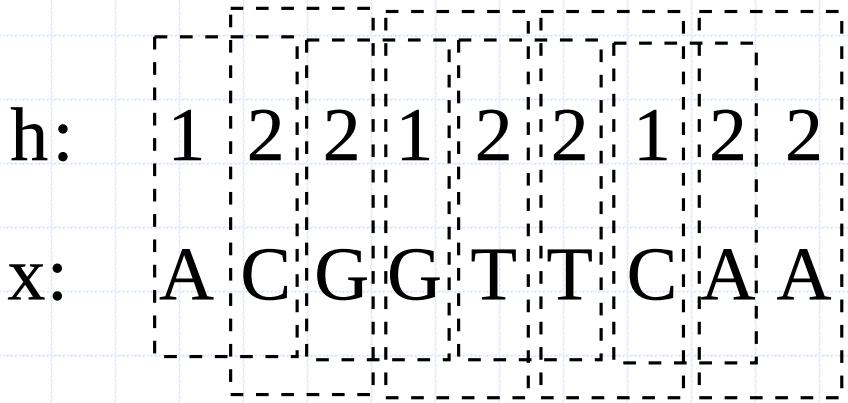
$$p(h_i = k|\boldsymbol{x}) = f_k(i) b_k(i) / p(\boldsymbol{x})$$

$f_k(i)$: forward variable, $b_k(i)$: backward variable

2nd order marginalized count kernel

- If adjacent relations between symbols have essential meanings, the count kernel is obviously not sufficient

- 2nd order marginalized count kernel
  - 4 neighboring symbols (i.e. 2 visible and 2 hidden) are combined and counted



Fisher Kernel

- Probabilistic mode $p(\boldsymbol{x}|\hat{\boldsymbol{\theta}}),\ \boldsymbol{x} \in \mathcal{X}$

  $\hat{\boldsymbol{\theta}} \in \Re^r$ is a parameter vector obtained from training samples

- Fisher Kernel
  - Mapping to a feature vector (Fisher score vector)

$$\boldsymbol{s}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) = \left(\frac{\partial \log p(\boldsymbol{x}|\hat{\boldsymbol{\theta}})}{\partial \theta_1}, \ldots, \frac{\partial \log p(\boldsymbol{x}|\hat{\boldsymbol{\theta}})}{\partial \theta_p}\right)$$

  - Inner product of Fisher scores

$$K_f(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{s}(\boldsymbol{x}, \hat{\boldsymbol{\theta}})^\top Z^{-1}(\hat{\boldsymbol{\theta}}) \boldsymbol{s}(\boldsymbol{x}', \hat{\boldsymbol{\theta}})$$

Z: Fisher information matrix

Fisher Kernel from HMM

- Derive FK from HMM (Jaakkola et al. 2000)
  - Derivative for emission probabilities only
  - No Fisher information matrix
- FK from HMM is a special case of marginalized kernels
  - Counts are centralized and weighted

$$K_f(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} p(\boldsymbol{h}|\boldsymbol{x}) p(\boldsymbol{h}'|\boldsymbol{x}') K_{fz}(\boldsymbol{z}, \boldsymbol{z}')$$

$$K_{fz}(\boldsymbol{z}, \boldsymbol{z}') = \sum_{k=1}^{n_x} \sum_{\ell=1}^{n_h} \frac{1}{e_{\ell k}^2} (c_{k\ell}(\boldsymbol{z}) - \bar{c}_{k\ell}(\boldsymbol{z}))(c_{k\ell}(\boldsymbol{z}') - \bar{c}_{k\ell}(\boldsymbol{z}'))$$

Difference between FK and Marginalized Kernels

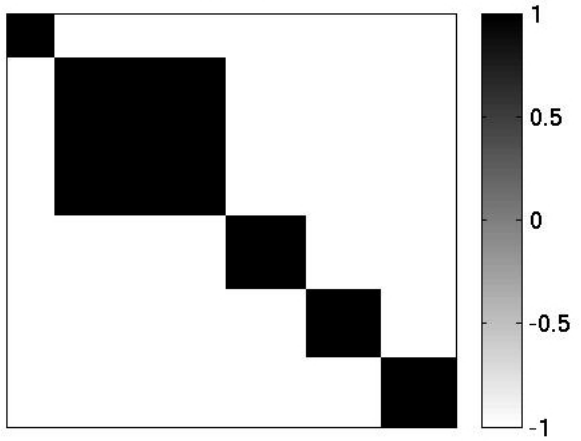
- FK: Probabilistic model determines the joint kernel and the posterior probabilities
- MK: You can determine them separately
  - More flexible design!

Protein clustering experiment

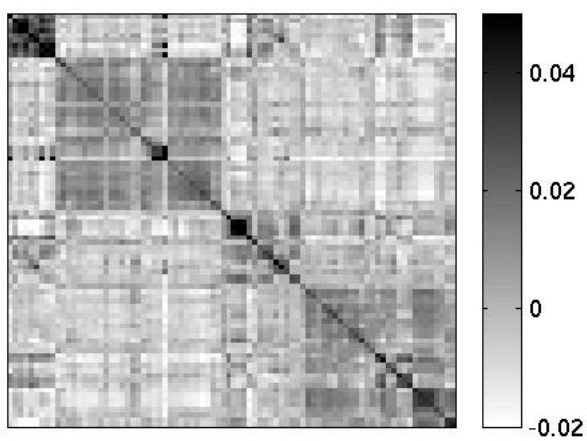
- 84 proteins containing five classes
  - gyrB proteins from five bacteria species
- Clustering methods
  - HMM + {FK,MCK1,MCK2}+K-Means
- Evaluation
  - Adjusted Rand Index (ARI)

Kernel Matrices

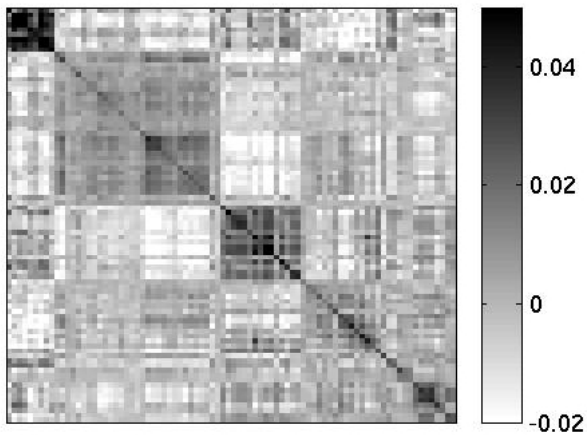
Ideal



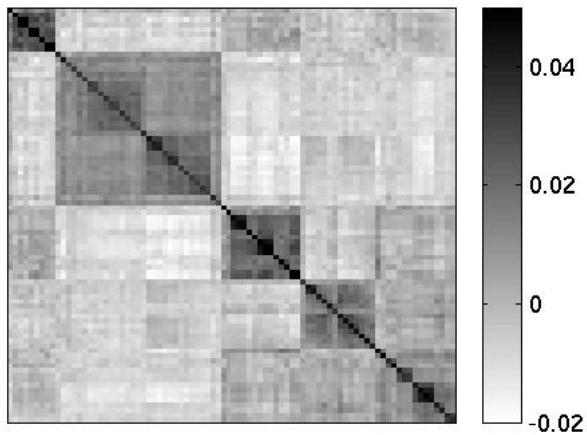
FK



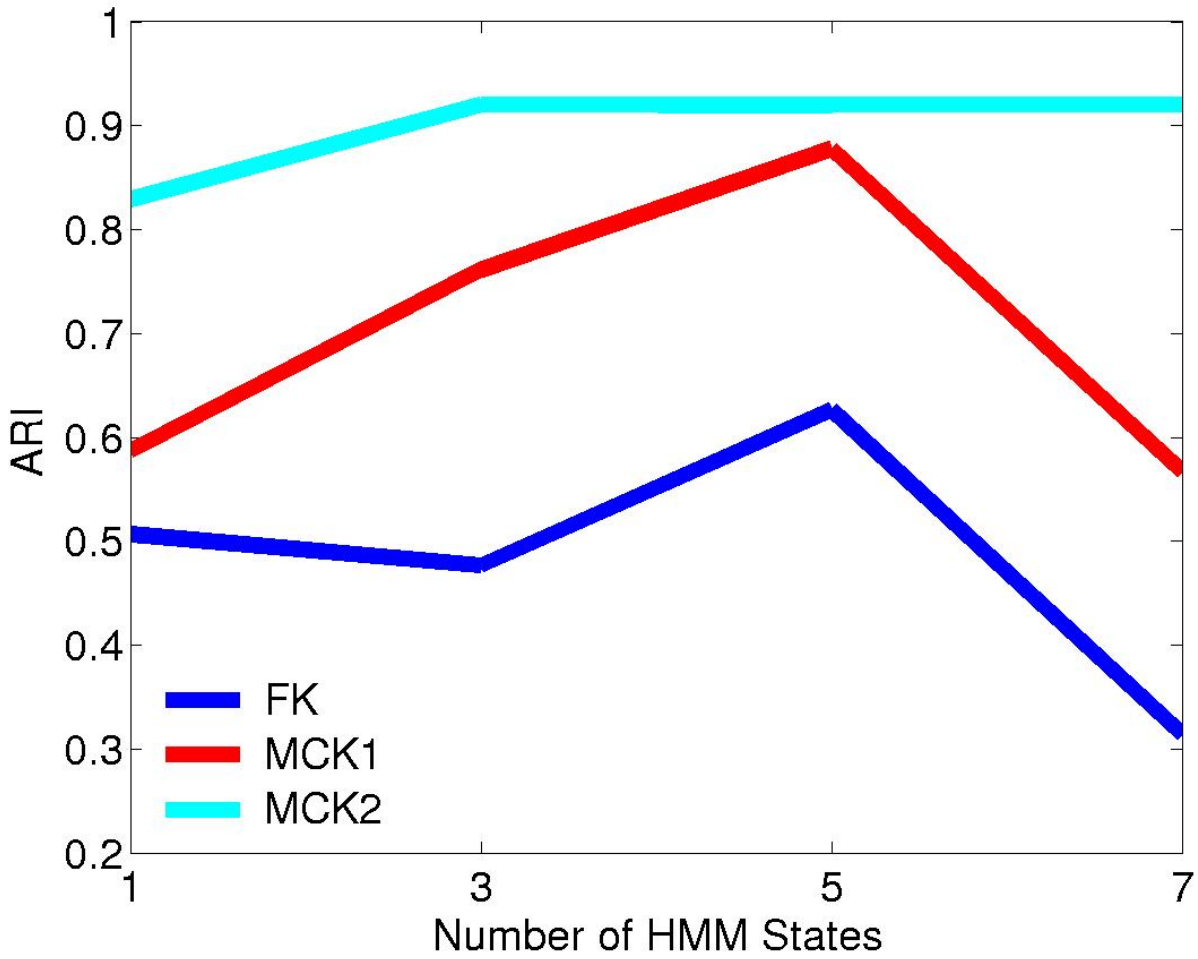
MCK1



MCK2



Clustering Evaluation



Applications since then..

- Marginalized Graph Kernels (Kashima et al., ICML 2003)
- Sensor networks (Nyugen et al., ICML 2004)
- Labeling of structured data (Kashima et al., ICML 2004)
- Robotics (Shimosaka et al., ICRA 2005)
- Kernels for Promoter Regions (Vert et al., NIPS 2005)
- Web data (Zhao et al., WWW 2006)

Summary (Marginalized Kernels)

- General Framework for using generative model for defining kernels
- Fisher kernel as a special case
- Broad applications
- Combination with CRFs and other advanced stuff?

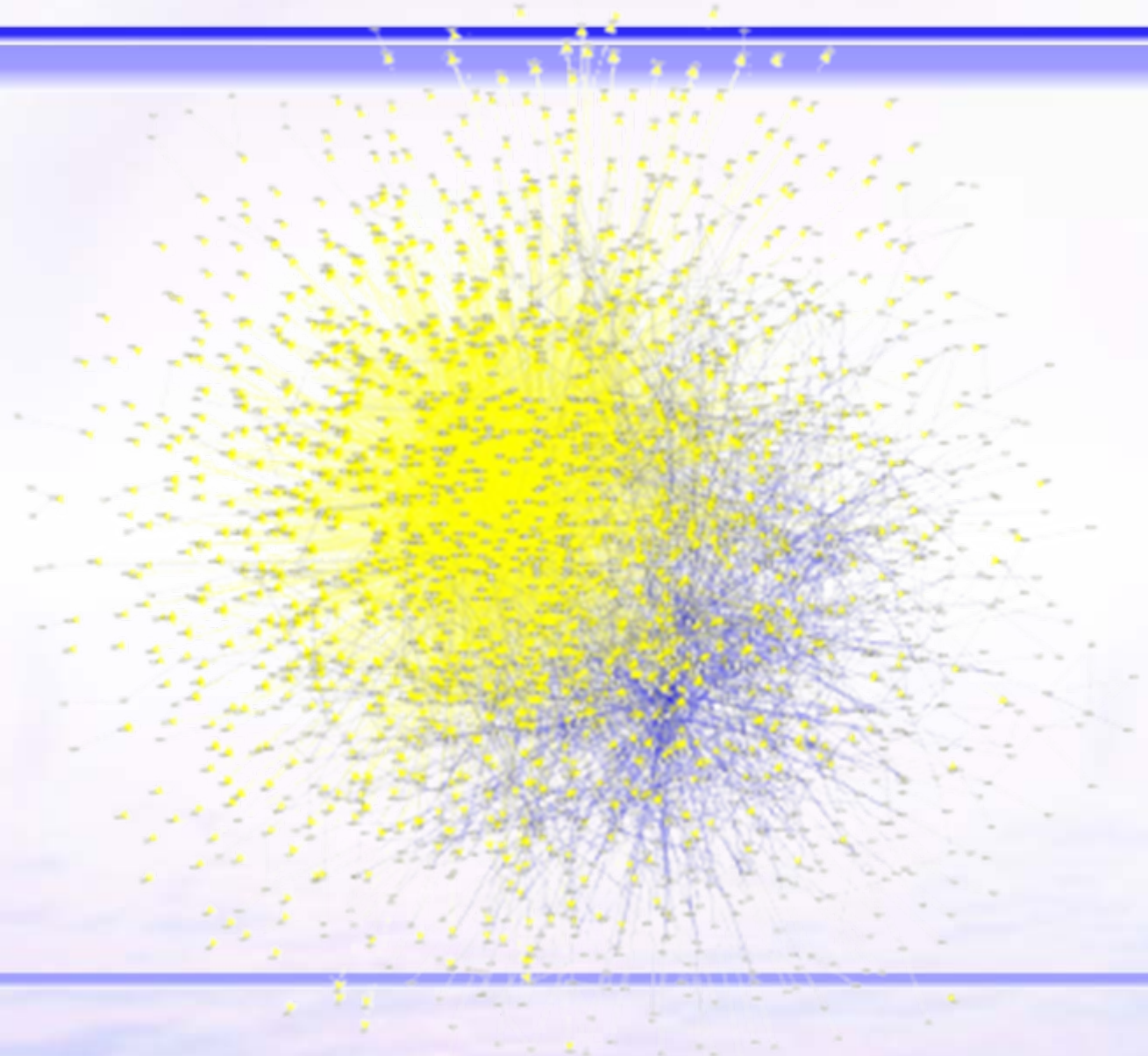
Diffusion Kernels: Biological Network Analysis

Biological Networks

- **Protein-protein physical interaction**
- **Metabolic networks**
- **Gene regulatory networks**
- **Network induced from sequence similarity**

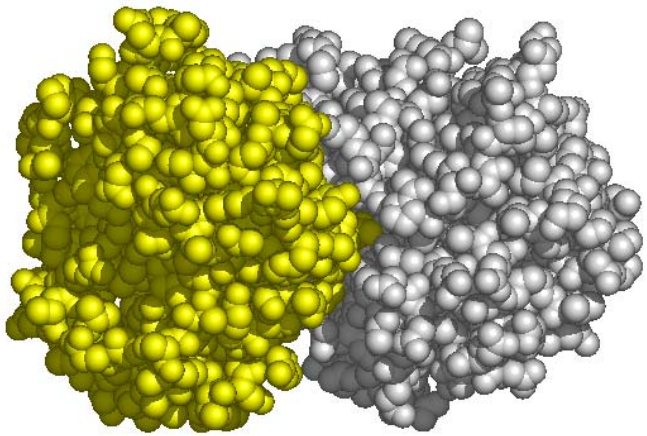
- **Thousands of nodes (genes/proteins)**
- **100000s of edges (interactions)**

Physical Interaction Network

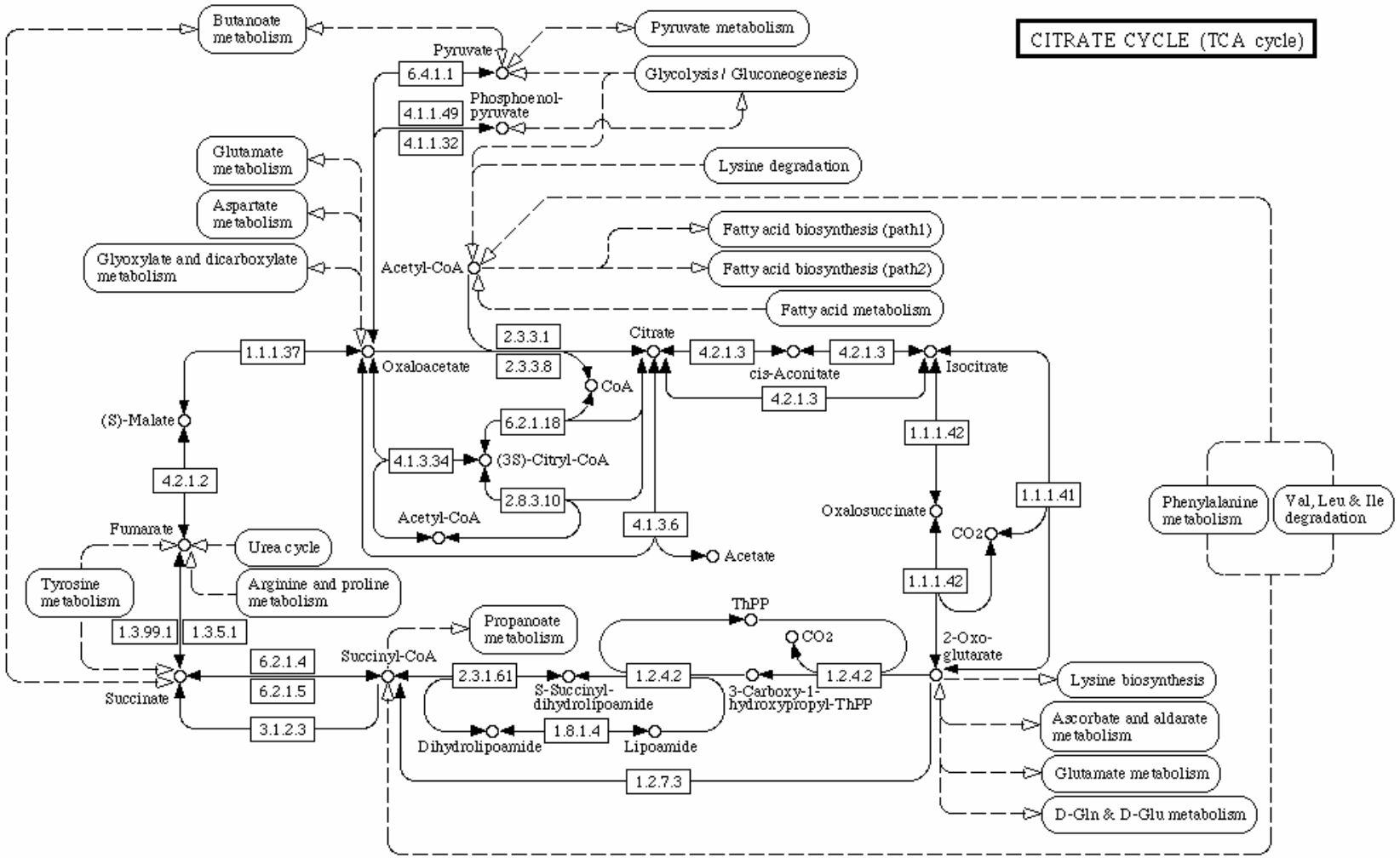


Physical Interaction Network

- **Undirected graphs of proteins**
- **Edge exists if two proteins physically interact**
  - **Docking (Key – Keyhole)**
- **Interacting proteins tend to have the same biological function**



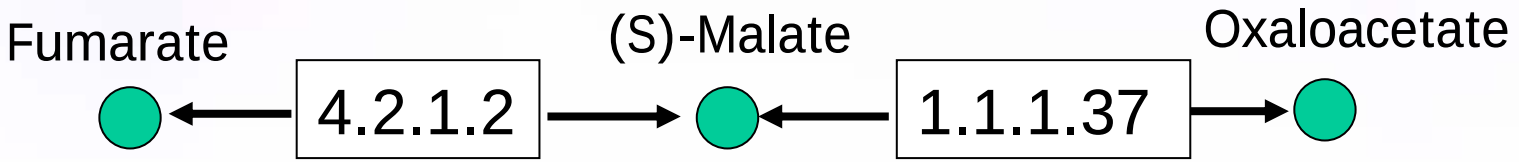
Metabolic Network



00020 3/19/04

Metabolic Network

- **Node: Chemical compounds**
- **Edge: Enzyme catalyzing the reaction (EC Number)**



- **KEGG Database (Kyoto University)**
- **Collection of pathways (subnetworks)**
- **Can be converted as a network of enzymes (proteins)**

Protein Function Prediction

- **For some proteins, their functions are known**
- **But still functions of many proteins are unknown**

	GO ID	*No. of protein*	*Function description*
1	7165	143	Signal transduction
2	902	102	Cellular morphogenesis
3	7010	254	Cytoskeleton organization and biogenesis
4	7005	100	Mitochondrion organization and biogenesis
5	7033	125	Vacuole organization and biogenesis
6	6364	116	rRNA processing
7	7047	127	Cell wall organization and biogenesis
8	16568	126	Chromatin modification
9	6260	100	DNA replication
10	7067	120	Mitosis
11	7126	111	Meiosis
12	74	116	Regulation of cell cycle
13	6605	209	Protein targeting

Function Prediction Using a Network

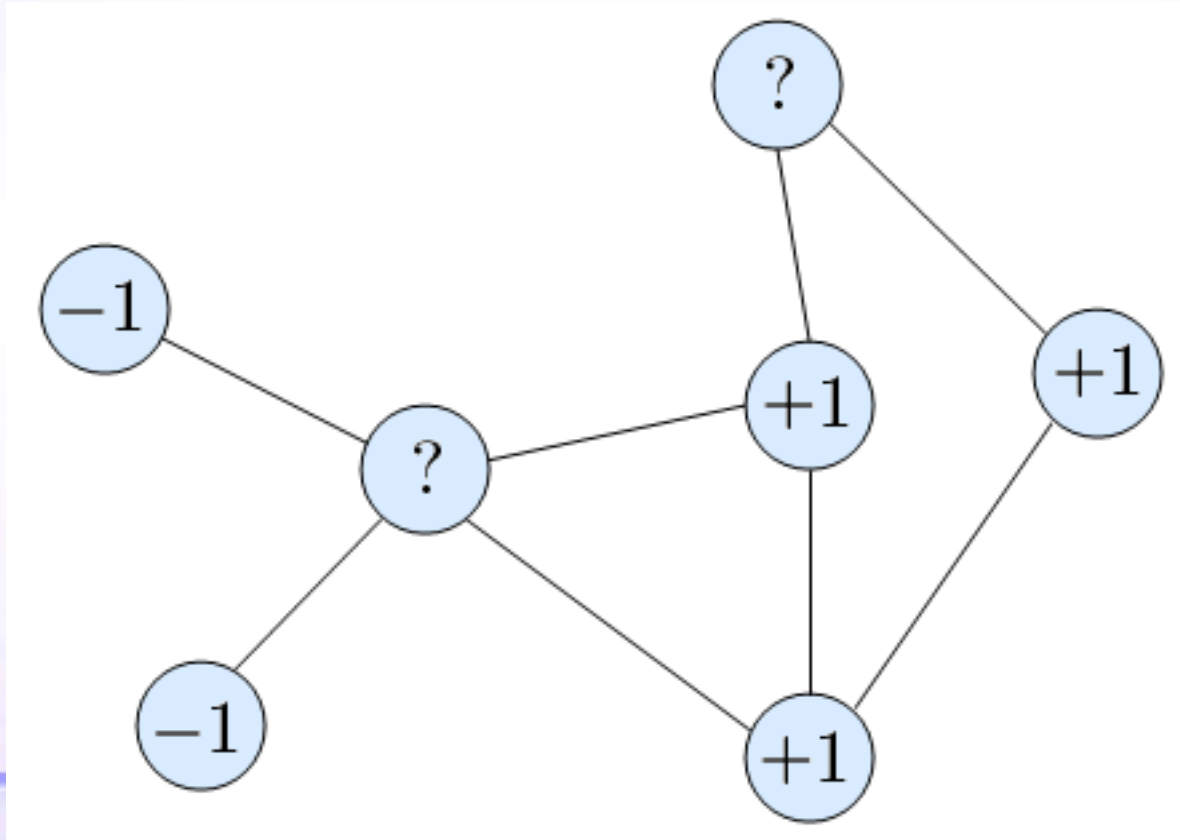
- **Determination of protein's function is a central goal of molecular biology**
- **It has to be determined by biological experiments, but accurate computational prediction helps**

- **Proteins close to each other in the networks tend to share the same functional category**
- **Use the network for function prediction!**

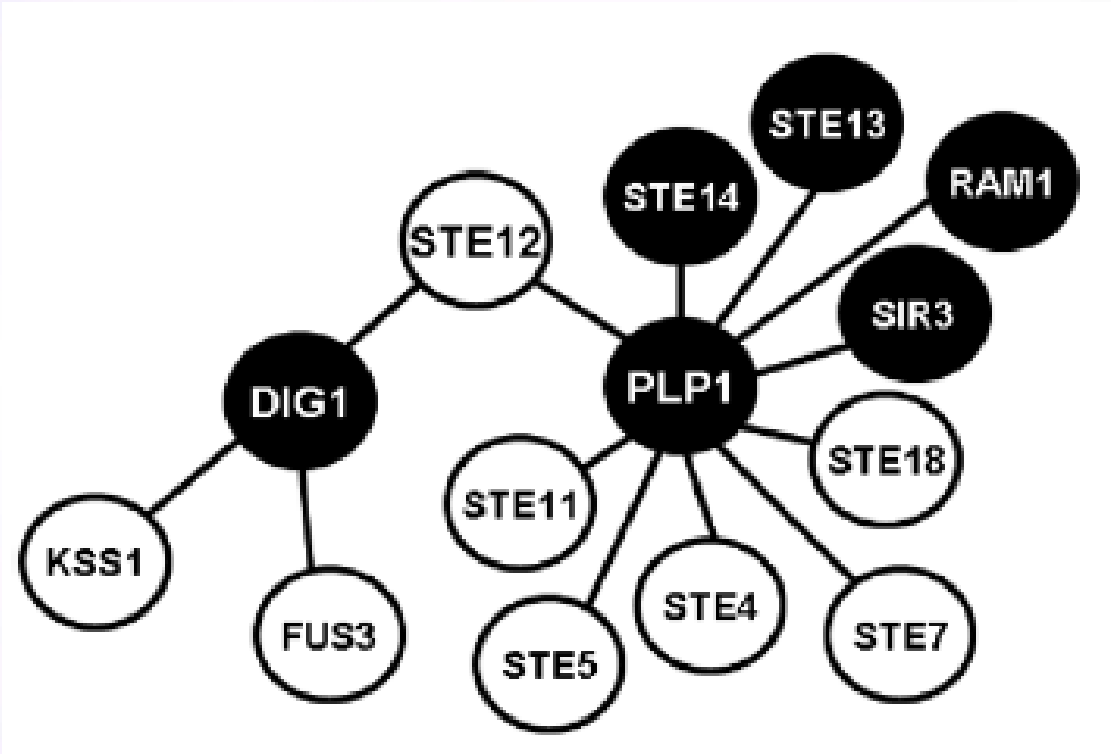
- **(Combination with other information sources)**

Prediction of one functional category

- **+1/-1： Labeled proteins with/without a specific function**
- **?: Unlabeled proteins**

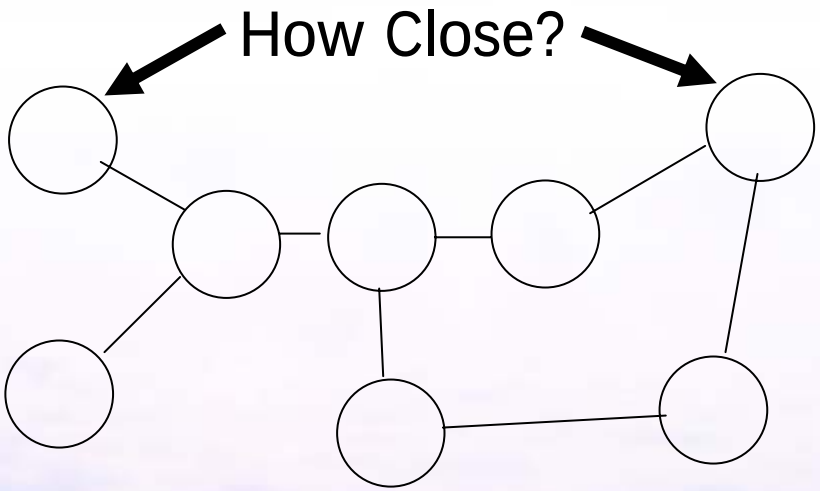


Indirect connection



Diffusion kernels (Kondor and Lafferty, 2002)

- **Function prediction by SVM using a network**
  - **Kernels are needed !**
- **Define closeness of two nodes**
  - **Has to be positive definite**



Definition of Diffusion Kernel

- **A: Adjacency matrix,**
- **D: Diagonal matrix of Degrees**
- **L = D-A: Graph Laplacian Matrix**
- ***Diffusion kernel matrix***

$$K = \exp(-\beta L)$$

  - β : **Diffusion paramater**

- **Matrix exponential, not elementwise exponential**

Computation of Matrix Exponential

- **Definition**

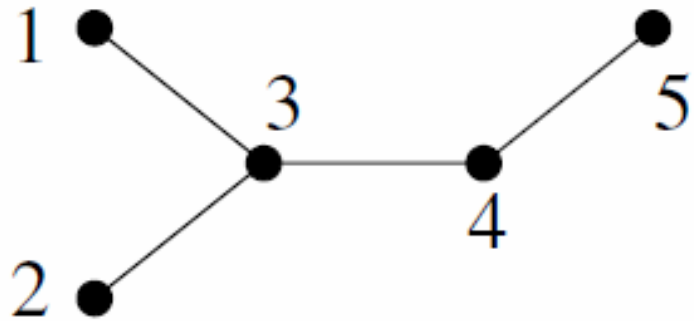
$$\exp(A) = \lim_{s \to \infty} (I + \frac{A}{s})^s$$

- **Eigen-decomposition**

$$A = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

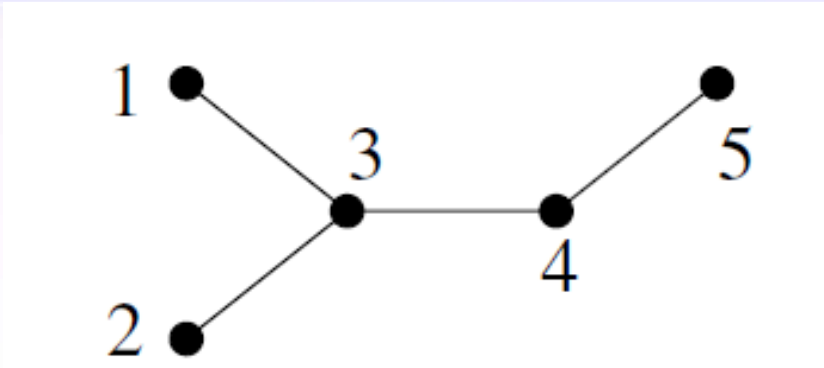
$$\exp(A) = \sum_{i=1}^{n} \exp(\lambda_i) \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

Adjacency Matrix and Degree Matrix



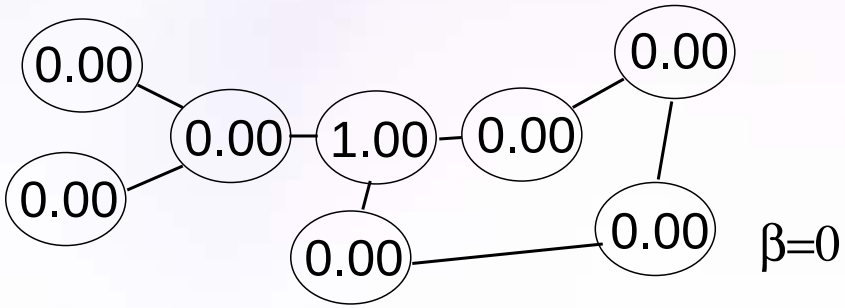
$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Graph Laplacian Matrix L

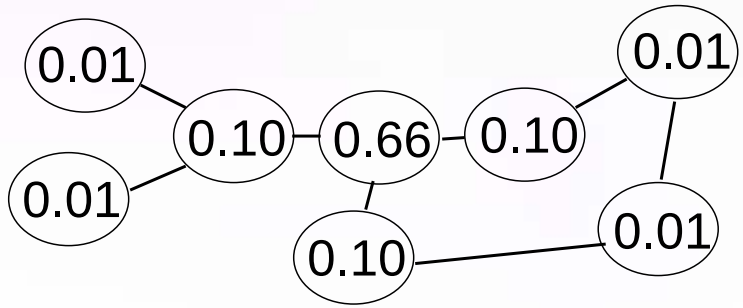


$$L = D - A = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

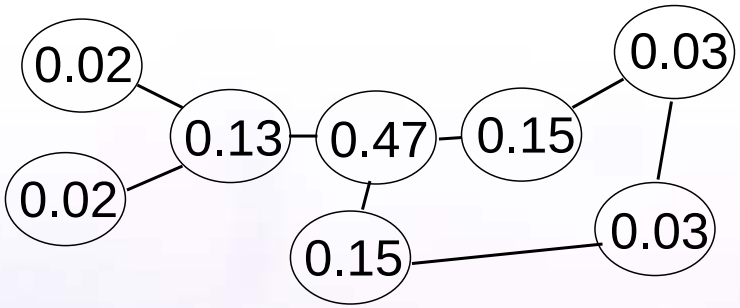
Actual Values of Diffusion Kernels



β=0



β=0.15



β=0.3

Closeness from the "central node"

Interpretation: Stochastic Process

- **For each node i ,consider random variable $Z_i(t)$**
- **Initial condition $Z_i(0)$**
  - **Zero mean, Variance σ^2**
  - **Independent to each other (covariance zero).**
- **Each variable sends a fraction to the neighbors**

$$Z_i(t+1) = Z_i(t) + \beta \sum_{j \sim i} (Z_j(t) - Z_i(t))$$

$$Z(t+1) = (I - \beta L) Z(t)$$

Stochastic Process (2)

- **Time Evolution Operator** $T(t) = (I - \beta L)^t$

- **Covariance**

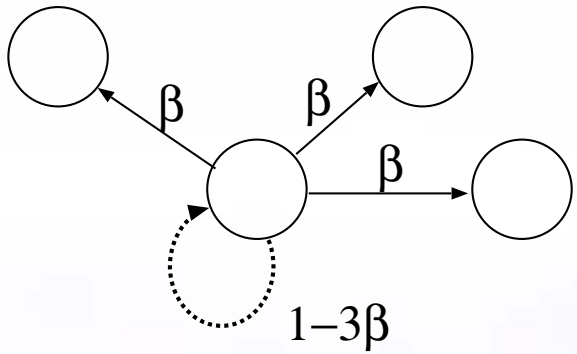
$$\mathrm{Cov}_{ij}(t) = \sigma^2 T_{ij}(2t)$$

- **Reduce the time step 1 to** Δt
- **Diffusion parameter** $\beta \Delta t$
- **Taking the limit** $\Delta t \to 0$

$$\mathrm{Cov}(t) = \sigma^2 \exp(-2\beta t L)$$

Interpretation via random walking

- **Random walking according to transition probability**
- **Transition probability is constant** β
- **Remaining probability = Self loop**

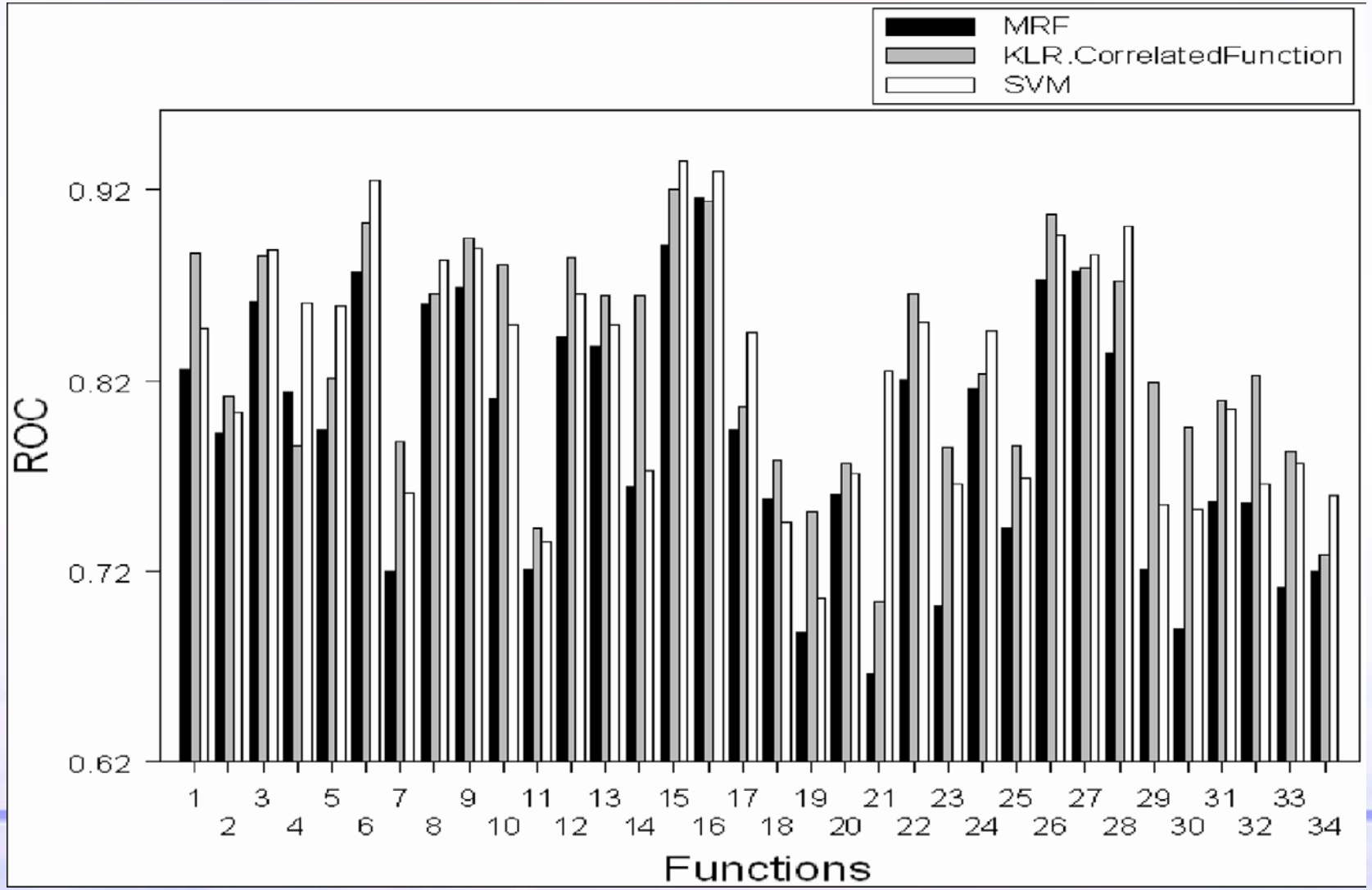


- K_{ij} **is equal to the probability of the walk that started at i being at j after infinite time steps**

Experimental Results by Lee et al. (2006)

- **Yeast Proteins**
- **34 functional categories**
  - **Decomposed into binary classification problems**
- **Physical Interaction Network only**
- **Methods**
  - **Markov Random Field**
  - **Kernel Logistic Regression**
    - **Use additional knowledge of correlated functions**
  - **Support Vector Machine**
- **ROC score**
  - **Higher is better**

Experimental results by Lee et al. (2006)



Kernel-based inference of networks

T. Kato, K. Tsuda, and K. Asai.
Selective integration of multiple biological data for supervised network inference.
Bioinformatics, 21(10):2488-2495, 2005.

Statistical Inference of Networks

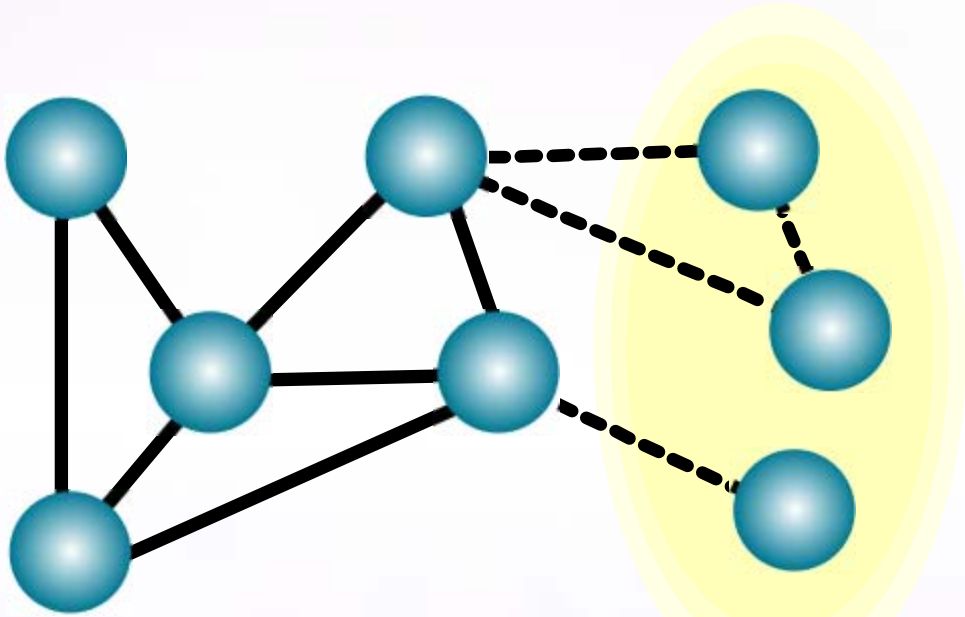
- **Infer the network by data about proteins**
  - **Gene expressions, Phylogenetic profiles etc**
- **Propose a Kernel-based inference method**
  - **1. Supervised Inference**
    - **Learning from data and training network**
  - **2. Weighted combination of multiple data**
    - **Identify unnecessary data that do not contribute for network inference**

Unsupervised ⇨ Supervised Inference

- **Unsupervised network inference**
  - **Bayesian network (Friedman et al., 2000)**
  - **Infer every edge from scratch (no known edges)**

- **Supervised network inference**
  - **A part of the network is known (training network)**
  - **Infer the rest of the network from data and training net**
  - **Kernel CCA (Yamanishi et al., ISMB, 2004)**

Supervised Network Inference



Training network

Extra nodes

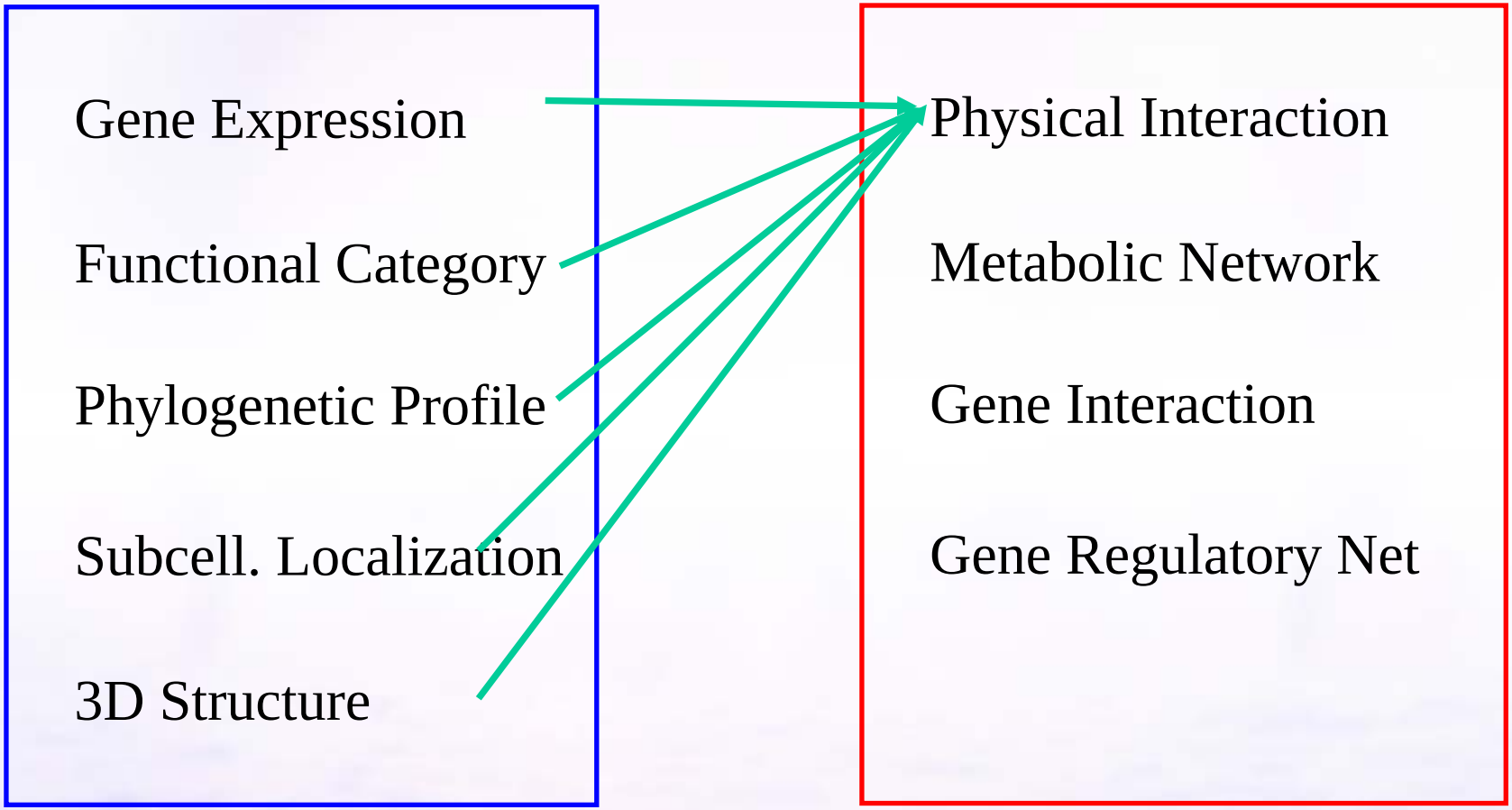
Single Data ⇨ Multiple Data

- **Multiple data for inferring networks**
  - **Gene expression profiles**
  - **Subcellular locations**
  - **Phylogenetic profiles**

- **Identify relevant data for inference**
- **Weighted integration of multiple data !**
  - **Feature selection to data selection**

- **Kernel CCA: No mechanism for data selection**

Inferring a Network from Multiple Data



Outline

- **Network Inference from a kernel matrix**
  - **Unsupervised, Single Data**
  - **Thresholding: Nearest neighbor connection**
- **Incorporating the training network**
  - **Supervised, Single Data**
  - **Kernel Matrix Completion (Tsuda et al., 2003)**
- **Weighted integration of multiple data**
  - **Supervised, Multiple Data**
  - **Weights determined by the EM algorithm**

Unsupervised, Single Data

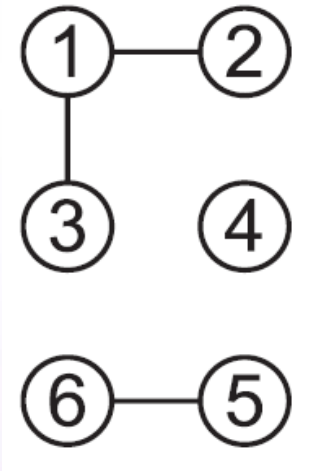
- **Convert the data to a kernel matrix**
  - **Similarity among proteins**
  - **Gene expression: Pearson correlation**
  - **Phylogenetic profile: Tree kernel (Vert 2002)**
  - **3D structure: Graph kernel (Borgwardt et al., 2005)**

$$\begin{bmatrix} 1.000 & 0.460 & 0.460 & 0.175 & 0.023 & 0.004 \\ 0.460 & 1.000 & 0.183 & 0.385 & 0.073 & 0.017 \\ 0.460 & 0.183 & 1.000 & 0.385 & 0.073 & 0.017 \\ 0.175 & 0.385 & 0.385 & 1.000 & 0.366 & 0.124 \\ 0.023 & 0.073 & 0.073 & 0.366 & 1.000 & 0.603 \\ 0.004 & 0.017 & 0.017 & 0.124 & 0.603 & 1.000 \end{bmatrix}$$

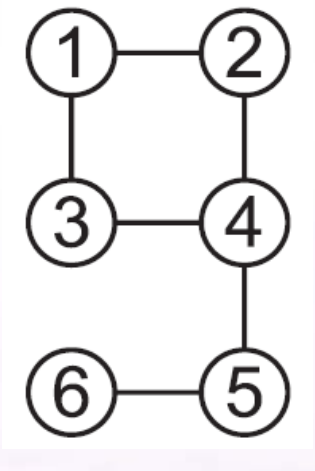
Construct the network by thresholding

- **Establish an edge where the kernel value is more than threshold**

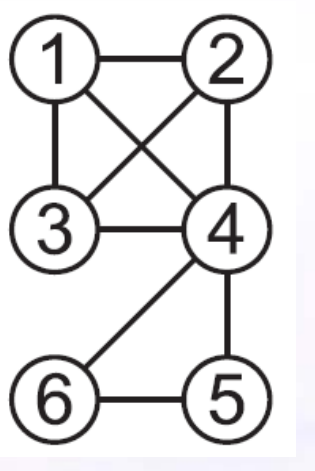
t=0.1



t=0.2

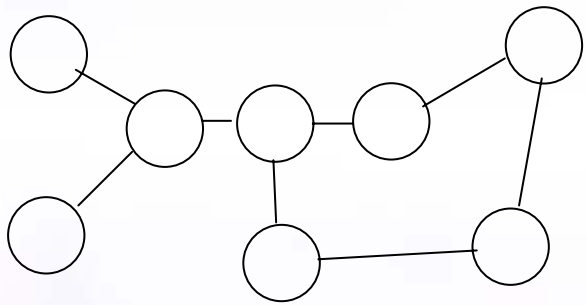


t=0.4



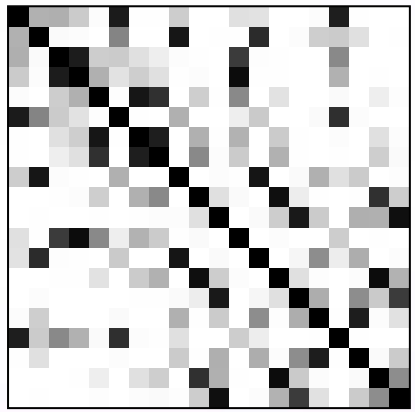
Supervised, Single Data

- **Known Training Network (Only for first n nodes)**



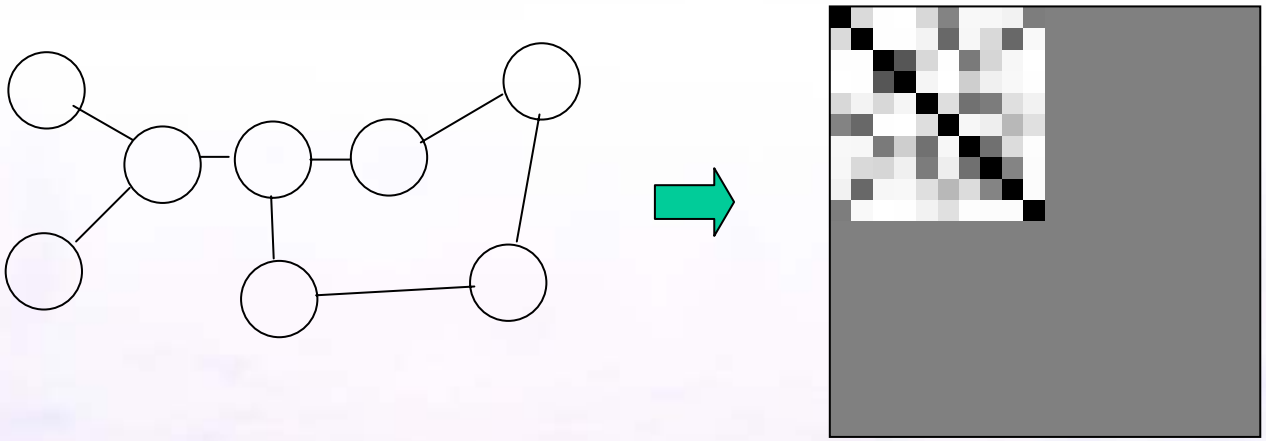
- **Data about all proteins**

Kernel Matrix



Incomplete kernel matrix from training network

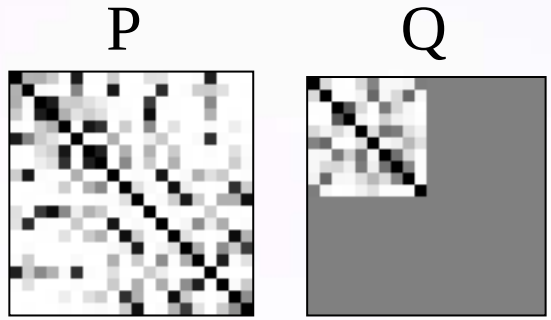
- **Convert the training graph to a kernel matrix**
- **Synchronizing the representation**
- **Diffusion kernel** **(Kondor and Lafferty, 2002)**
  - **Measure closeness of nodes by random walking**



*Thresholding approximately recover the original network

Kernel Matrix Completion

- **P: Kernel matrix of the data**
- **Q: Incomplete kernel matrix**



$$Q = \begin{bmatrix} K_I & Q_{vh} \\ Q_{vh}^{\mathrm{T}} & Q_{hh} \end{bmatrix}$$

- **Missing values estimated by minimizing the KL divergence**

$$\textbf{Minimize } \mathrm{KL}(Q,P) \textbf{ w.r.t. } Q_{vh}, Q_{hh}$$

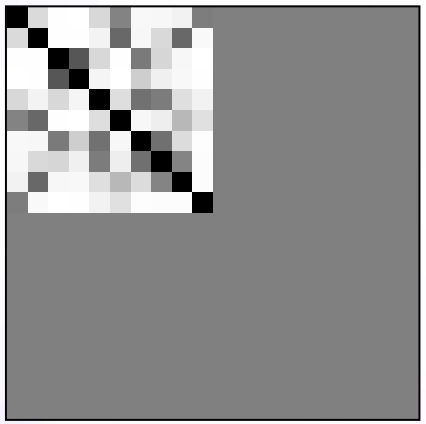
$$\mathrm{KL}(Q,P) = \tfrac{1}{2}\mathrm{tr}(P^{-1}Q) - \tfrac{1}{2}\mathrm{logdet}(P^{-1}Q) - \tfrac{1}{2}\ell$$

- **Closed from solution Q***
- **Threshold Q* to obtain the network**

Supervised, Multiple Data

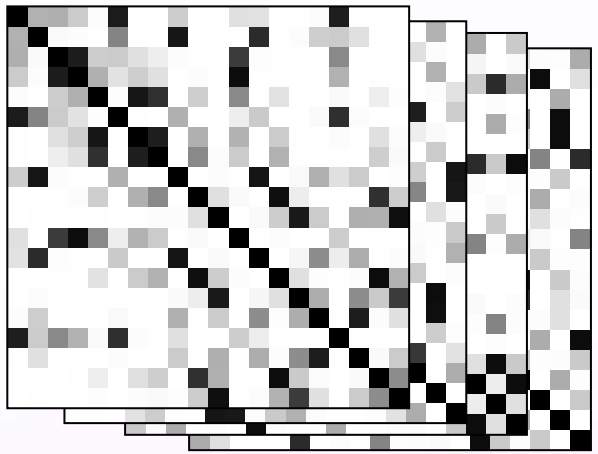
- **Known Training Network**

Diffusion Kernel Matrix

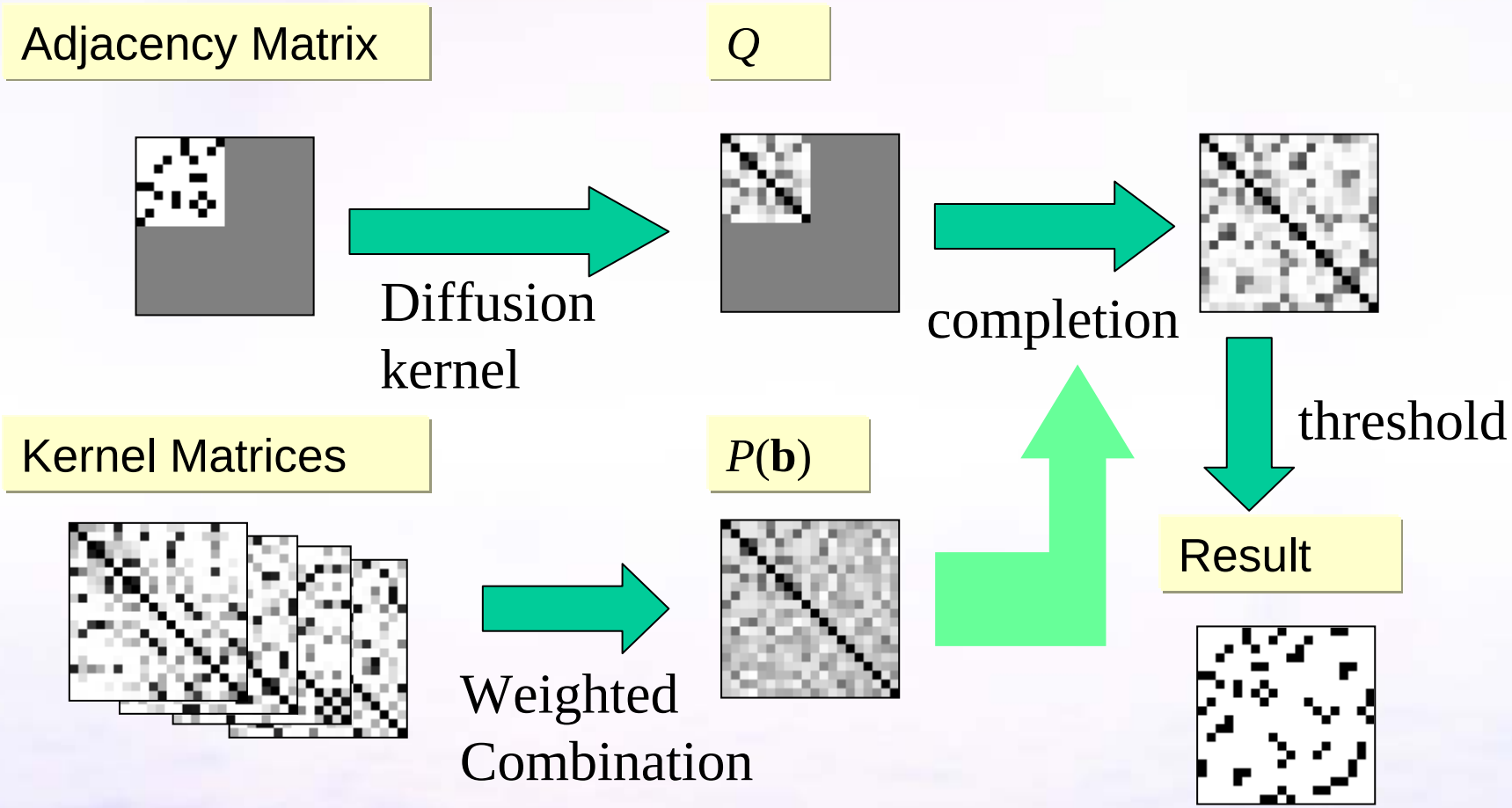


- **Multiple data about all proteins**

Kernel Matrices



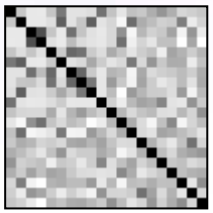
Overview of Our Approach



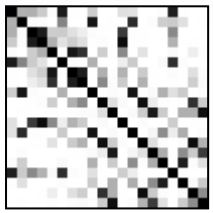
Notations



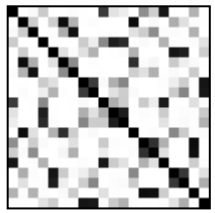
$$Q = \begin{bmatrix} K_I & Q_{vh} \\ Q_{vh}^{\mathrm{T}} & Q_{hh} \end{bmatrix}$$



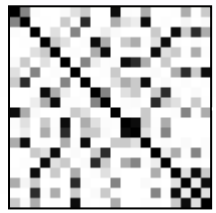
$P(\mathbf{b})$



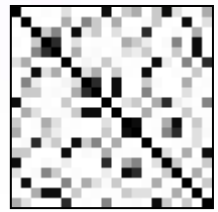
K_1



K_2



K_3



K_4

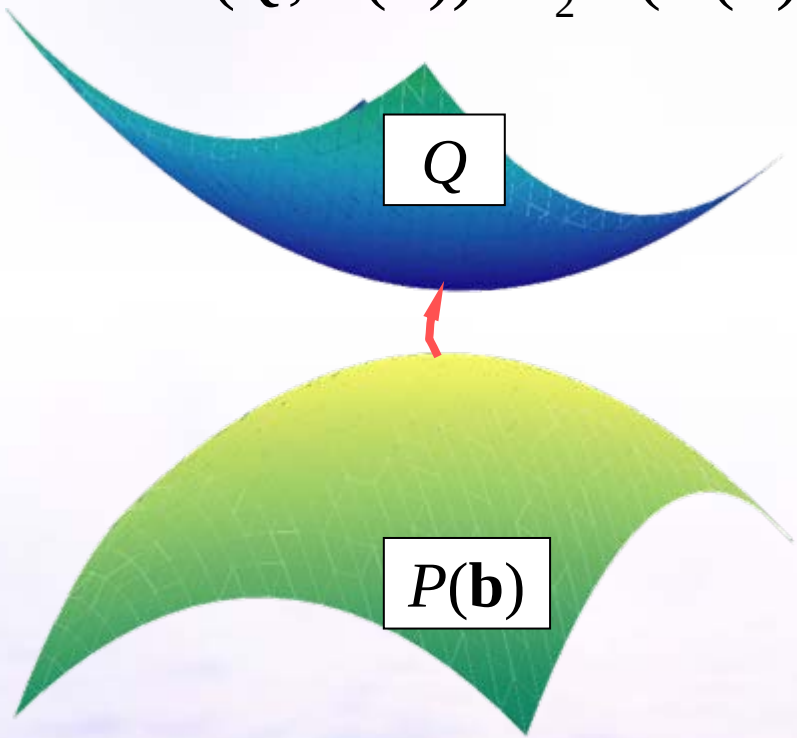
$$P(\mathbf{b}) = \sum_{i=1}^{n_k} b_i K_i + \sigma^2 I$$

Unknowns： Submatrices Q_{vh},Q_{hh}, Weights $\mathbf{b}$

Objective Function

- **KL divergence**

$$\mathrm{KL}(Q, P(\mathbf{b})) = \tfrac{1}{2}\,\mathrm{tr}(P(\mathbf{b})^{-1}Q) - \tfrac{1}{2}\,\mathrm{logdet}(P(\mathbf{b})^{-1}Q) - \tfrac{1}{2}\,\ell$$



Minimize w.r.t.

Submatrices Q_{vh},Q_{hh},
Weights **b**

Solved by the EM algorithm

EM Algorithm

- **Repeat the following two steps**
  1. **E-step: minimize** $\mathrm{KL}(Q, P(\mathbf{b}))$ **w.r.t.** Q_{vh}, Q_{hh}
  2. **M-step: minimize** $\mathrm{KL}(Q, P(\mathbf{b}))$ **w.r.t.** $\mathbf{b}$

- **E-step:**
  - **Same as the single kernel case**

- **M-step:**
  - **Solved in closed form**

Edge prediction experiments

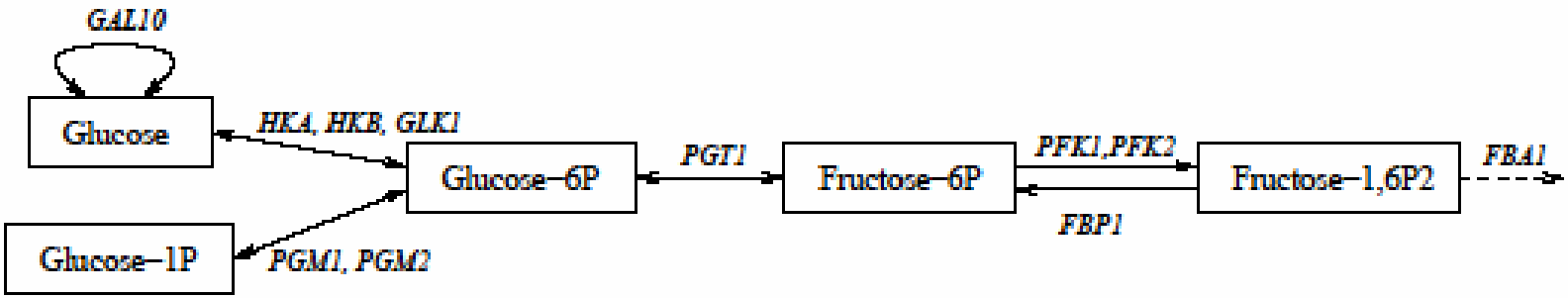
Network	・ **Metabolic Network（KEGG）** ・ **Protein interaction network（von Mering, 2002）**
Data	・ **exp: gene expression** ・ **y2h: Interaction net by yeast2hybrid** ・ **loc: subcellular location** ・ **phy: phylogenetic profile** ・ **rnd1,…,rnd4: random noise**
Methods	・ **Q: Proposed method** ・ **P: Simple combination of kernel matrices** ・ **cca: kernel CCA (without noises)**
Evaluation	**ROC score of edge prediction accuracy** **(10-fold cross validation)**

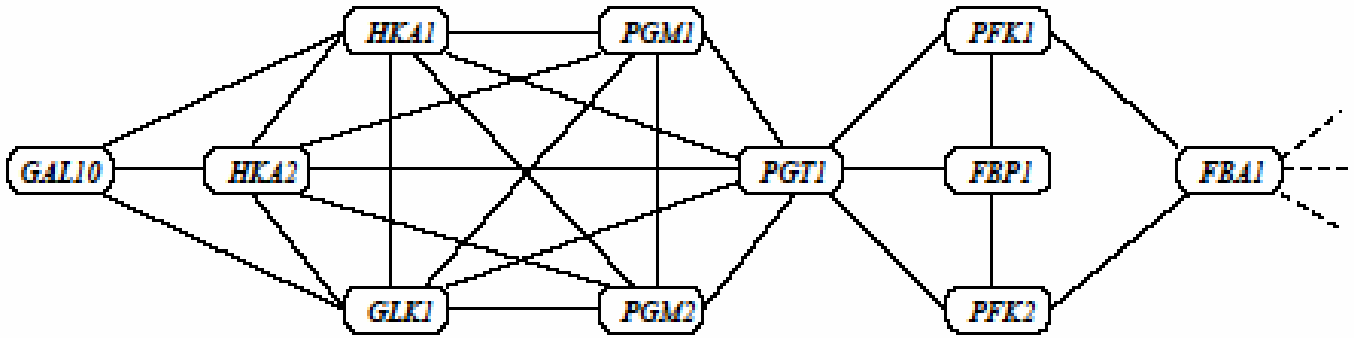
Metabolic network

- Made from LIGAND Database (KEGG)

(Vert and Kanehisa, NIPS, 2003)

- Connect enzymes of two successive reactions
- 769 nodes, 3702 edges





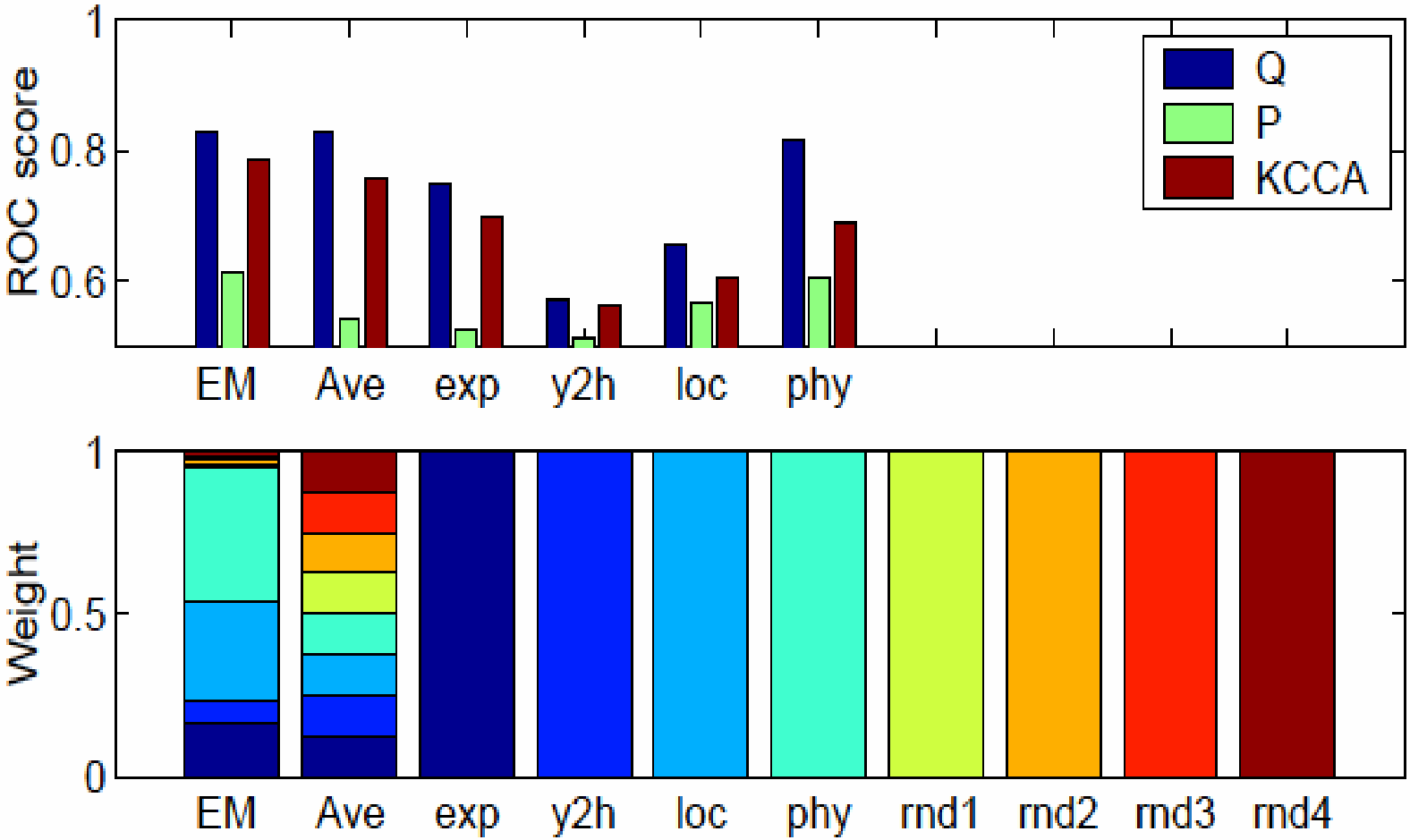
Interaction network (Von Mering et al., Nature, 2002)

- **Middle Confidence**
- **Interactions validated by multiple experiments**
  - **High-throughput yeast two hybrid**
  - **Correlated mRNA expression**
  - **Genetic interaction**
  - **Tandem affinity purification,**
  - **High-throughput mass-spectrometric protein complex identification**
- **984nodes, 2438 edges**

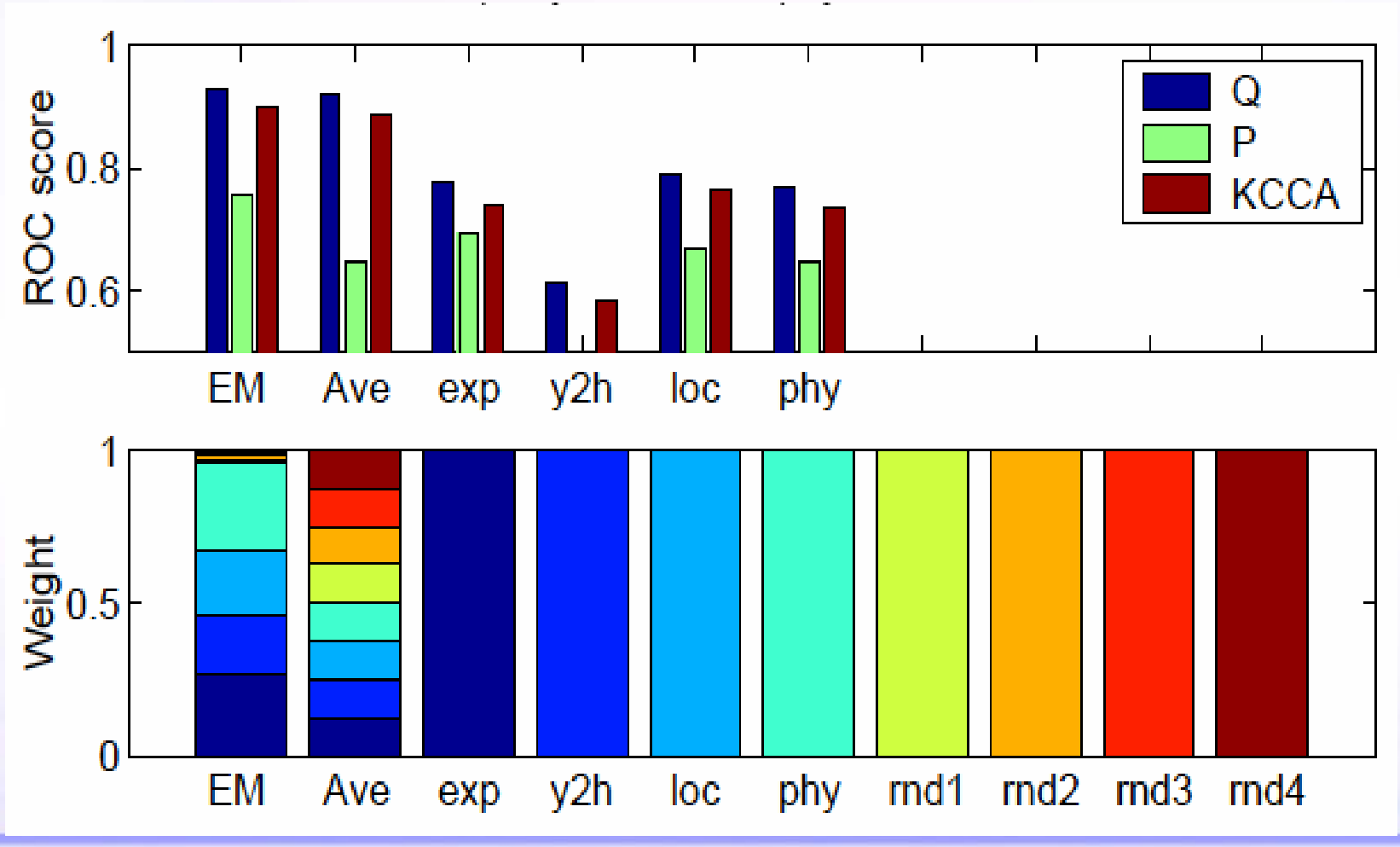
Dataset Details

Metabolic Net	**http://www.genome.jp/kegg/**
Interaction	**Von Mering et al., Nature, 417 399--403 , 2002**
Expression	**Spellman et al., MBC, 9, 3273—3297, 1998** **Eisen et al., PNAS, 95, 14863—8, 1998**
Y2H	**Ito et al., PNAS, 98, 4569—74, 2001** **Uetz et al., Nature, 10, 601—3, 2000**
Subcellular location	**Huh et al. Nature, 425, 686-91, 2003**
Phylogenetic profile	**http://www.genome.jp/kegg/**

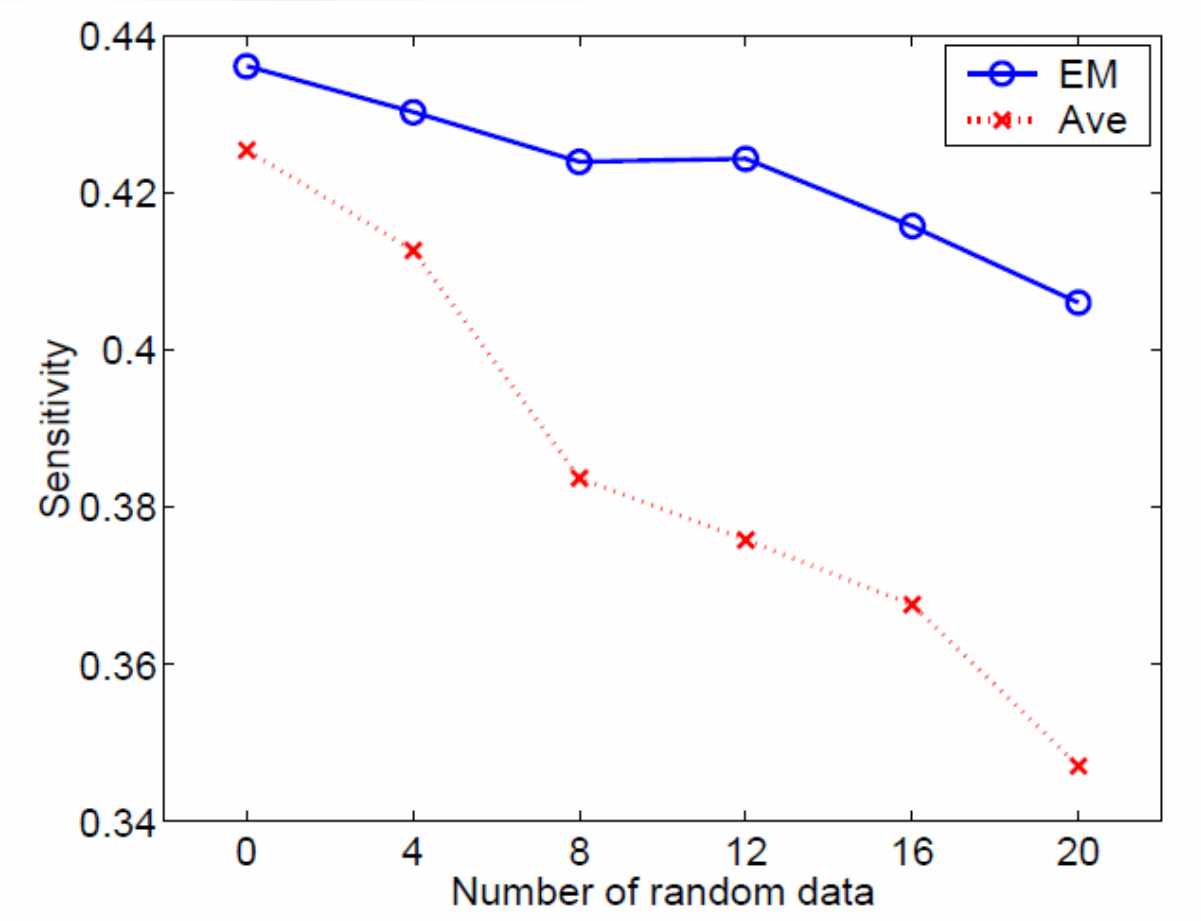
Metabolic Network



Physical Interaction Network



Introduce More Random Matrices (Metabolic network)



Sensitivity at 95% specificity

Summary of Experiments

- **Simple combination (P) < Completed matrix (Q)**
  - **Training network is essential**
- **Selection did not improve accuracy**
- **Accuracy comparable to kernel CCA**
- **Automatic selection of datasets**
  - **4 noise kernel matrices removed**

Summary (Network Estimation)

- **Supervised Inference of Network**
  - **Part of network known**
  - **Selection from multiple data**
  - **Formulation as kernel completion problem**
  - **Validation experiments on metabolic and interaction networks**
- **Future work**
  - **Biological interpretation of selection results**
  - **Applications to non-bio data**

Concluding Remarks

- Kernel methods have been applied to many different objects
  - Marginalized Kernels: Latent variables
  - Diffusion Kernels: Networks
- Graph Kernels: Next Lecture
- THANK YOU VERY MUCH!!