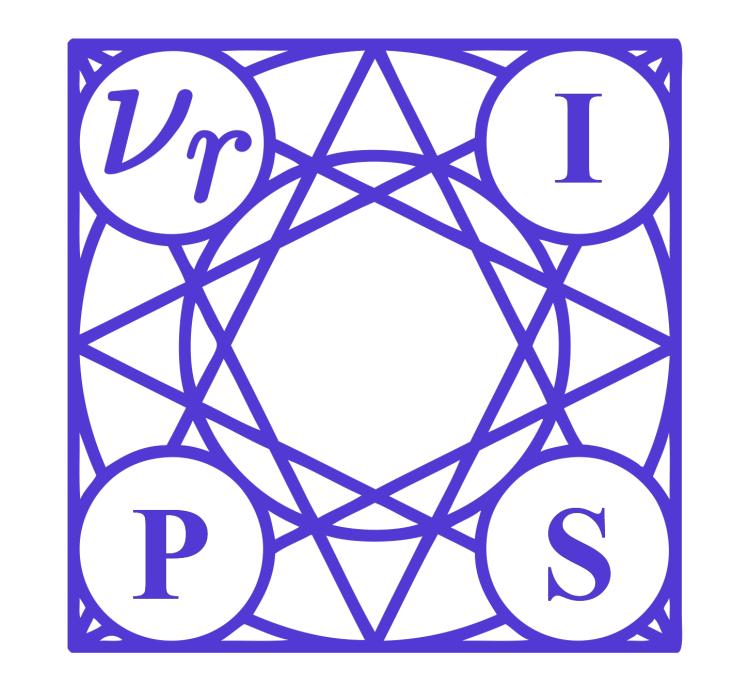


# KerGM: Kernelized Graph Matching

Zhen Zhang<sup>1</sup>, Yijian Xiang<sup>1</sup>, Lingfei Wu<sup>2</sup>, Bing Xue<sup>1</sup>, Arye Nehorai<sup>1</sup>

Washington University in St. Louis, <sup>2</sup>IBM Research



#### Abstract

Graph matching (GM), which aims at finding the optimal correspondence between nodes of two given graphs, is a longstanding problem due to its nonconvex objective function and binary constraints. It arises in many applications, ranging from recognizing actions to identifying functional orthologs of proteins. We provide a unifying view for two typical graph matching formulations: the Koopmans-Beckmann's QAP (quadratic assignment problem) and the Lawler's QAP. Furthermore, we develop the entropy-regularized Frank-Wolfe (EnFW) algorithm for optimizing QAPs, which is extremely fast in practice.

## A real-world example

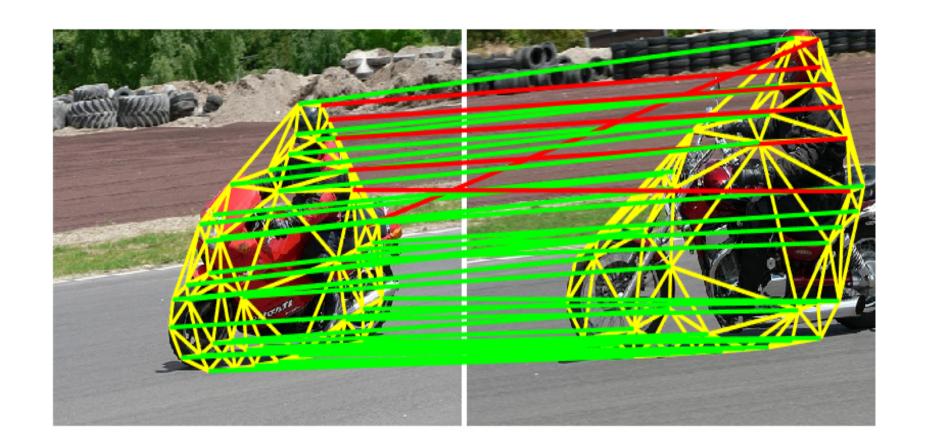


Figure 1: Landmarks matching in computer vision. The graphs are constructed by using Delaunay triangulation. Green lines: correctly matched; Red lines: incorrectly matched.

#### Two QAPs for graph matching

(i) **Koopmans-Beckmann's QAP**: Maximizing the Frobenius inner product maximization between two adjacency matrices  $A_1$  and  $A_2$  after permutation, i.e.,

$$\max \langle \mathbf{A}_1 \mathbf{X}, \mathbf{X} \mathbf{A}_2 \rangle_{\mathrm{F}}$$
s.t.  $\mathbf{X} \in \mathcal{P} = \{ \mathbf{X} \in \{0, 1\}^{n \times n} | \mathbf{X} \vec{\mathbf{1}} = \vec{\mathbf{1}}, \mathbf{X}^T \vec{\mathbf{1}} = \vec{\mathbf{1}} \}.$  (1)

(ii) **Lawler's QAP**: Maximizing the similarity between the node attributes of two graphs and the similarity between the edge attributes of two graphs,i.e.,

$$\max \langle \boldsymbol{K}^N, \boldsymbol{X} \rangle_{\mathrm{F}} + \mathrm{vec}(\boldsymbol{X})^T \boldsymbol{K} \mathrm{vec}(\boldsymbol{X}) \quad \text{s.t. } \boldsymbol{X} \in \mathcal{P}, \tag{2}$$

where  $\mathbf{K}^N \in \mathbb{R}^{n \times n}$  is the node affinity matrix,  $\mathbf{K}$  is an  $n^2 \times n^2$  matrix, defined such that  $\mathbf{K}_{ia,jb} = k^E(\mathbf{\vec{q}}_{ij}^1, \mathbf{\vec{q}}_{ab}^2)$  if  $i \neq j$ ,  $a \neq b$ ,  $e_{ij}^1 \in \mathcal{E}_1$ , and  $e_{ab}^2 \in \mathcal{E}_2$ , otherwise,  $\mathbf{K}_{ia,jb} = 0$ .

(iii) Note that the set of permutation matrix,  $\mathcal{P}$ , is usually relaxed as the set of double stochastic matrix,  $\mathcal{D} = \{ \mathbf{X} \in \mathbb{R}^{n \times n}_+ | \mathbf{X}\vec{\mathbf{1}} = \vec{\mathbf{1}}, \mathbf{X}^T\vec{\mathbf{1}} = \vec{\mathbf{1}} \}$ 

#### $\mathcal{H}$ -operations for Hilbert arrays

Let  $\mathcal{H}$  be any Hilbert space, coupled with the inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  taking values in  $\mathbb{R}$ .  $\Psi$  is said to be an  $n \times n$  Hilbert array if  $\Psi_{ij} \in \mathcal{H}$ ,  $\forall i, j = 1, 2, ..., n$ . we make the following addition, transpose, and multiplication rules ( $\mathcal{H}$ -operations). Let  $\Psi, \Xi \in \mathcal{H}^{n \times n}$ , and let  $X \in \mathbb{R}^{n \times n}$ , then we have

- $\Psi + \Xi$ ,  $\Psi^T \in \mathcal{H}^{n \times n}$ , where  $[\Psi + \Xi]_{ij} \triangleq \Psi_{ij} + \Xi_{ij} \in \mathcal{H}$ .
- $\Psi * \Xi \in \mathbb{R}^{n \times n}$ , where  $[\Psi * \Xi]_{ij} \triangleq \sum_{k=1}^{n} \langle \Psi_{ik}, \Xi_{kj} \rangle_{\mathcal{H}} \in \mathbb{R}$ .
- $\Psi \odot X$ ,  $X \odot \Psi \in \mathcal{H}^{n \times n}$ , where  $[\Psi \odot X]_{ij} \triangleq \sum_{k=1}^{n} X_{kj} \Psi_{ik} \in \mathcal{H}$  and  $[X \odot \Psi]_{ij} \triangleq \sum_{k=1}^{n} X_{ik} \Psi_{kj} \in \mathcal{H}$ .

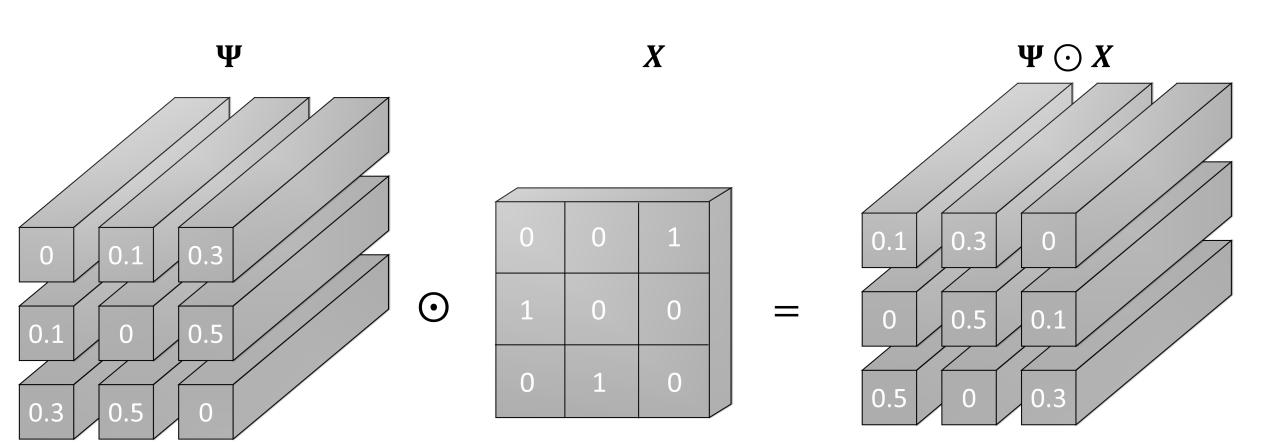


Figure 2: Visualization of the operation  $\Psi \odot X$ .

**Facts:** Define the function  $\langle \cdot, \cdot \rangle_{\mathcal{F}_{\mathcal{H}}} : \mathcal{H}^{n \times n} \times \mathcal{H}^{n \times n} \to \mathbb{R}$  such that  $\langle \Psi, \Xi \rangle_{\mathcal{F}_{\mathcal{H}}} \triangleq \operatorname{tr}(\Psi^T * \Xi) = \sum_{i,j=1}^n \langle \Psi_{ij}, \Xi_{ij} \rangle_{\mathcal{H}}, \ \forall \Psi, \ \Xi \in \mathcal{H}^{n \times n}$ . Then  $\langle \cdot, \cdot \rangle_{\mathcal{F}_{\mathcal{H}}}$  is an inner product on  $\mathcal{H}^{n \times n}$ .

## Kernelized graph matching

**Assumption:** We assume that the edge affinity function  $k^E : \mathbb{R}^{d_E} \times \mathbb{R}^{d_E} \to \mathbb{R}$  is a kernel. That is, there exist both an RKHS,  $\mathcal{H}$ , and an (implicit) feature map,  $\psi : \mathbb{R}^{d_E} \to \mathcal{H}$ , such that  $k^E(\vec{q}^1, \vec{q}^2) = \langle \psi(\vec{q}^1), \psi(\vec{q}^2) \rangle_{\mathcal{H}}, \forall \vec{q}^1, \vec{q}^2 \in \mathbb{R}^{d_E}$ . For any graph  $\mathcal{G} = \{A, \mathcal{V}, P, \mathcal{E}, Q\}$ , we can construct an array,  $\Psi \in \mathcal{H}^{n \times n}$ :

$$\Psi_{ij} = \begin{cases} \psi(\vec{q}_{ij}) \in \mathcal{H}, & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0_{\mathcal{H}} \in \mathcal{H}, & \text{otherwise} \end{cases}, \text{ where } 0_{\mathcal{H}} \text{ is the zero vector in } \mathcal{H}.$$

# Equivalent formulation of Lawler's QAP

Let  $\Psi^{(1)}$  and  $\Psi^{(2)}$  be the corresponding Hilbert arrays of  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , respectively. Then the Lawler's QAP can be written as

min 
$$J_{\rm gm}(\boldsymbol{X}) = -\langle \boldsymbol{K}^N, \boldsymbol{X} \rangle_{\rm F} - \langle \boldsymbol{\Psi}^{(1)} \odot \boldsymbol{X}, \boldsymbol{X} \odot \boldsymbol{\Psi}^{(2)} \rangle_{\rm F_{\mathcal{H}}}$$
 s.t.  $\boldsymbol{X} \in \mathcal{P}$ . convex and concave relaxations:

$$J_{ ext{vex}}(\boldsymbol{X}) = -\langle \boldsymbol{K}^N, \boldsymbol{X} \rangle_{ ext{F}} + \frac{1}{2} \| \boldsymbol{\Psi}^{(1)} \odot \boldsymbol{X} - \boldsymbol{X} \odot \boldsymbol{\Psi}^{(2)} \|_{ ext{F}_{\mathcal{H}}}^2$$
 $J_{ ext{cav}}(\boldsymbol{X}) = -\langle \boldsymbol{K}^N, \boldsymbol{X} \rangle_{ ext{F}} - \frac{1}{2} \| \boldsymbol{\Psi}^{(1)} \odot \boldsymbol{X} + \boldsymbol{X} \odot \boldsymbol{\Psi}^{(2)} \|_{ ext{F}_{\mathcal{H}}}^2.$ 

Path-following strategy:

$$\min J_{\alpha}(\boldsymbol{X}) = (1 - \alpha)J_{\text{vex}}(\boldsymbol{X}) + \alpha J_{\text{cav}}(\boldsymbol{X}), \alpha = 0 : \Delta\alpha : 1.$$

## Kernelized Graph Matching (Cont.)

**Approximate explicit feature maps:** For the kernel  $k^E : \mathbb{R}^{d_E} \times \mathbb{R}^{d_E} \to \mathbb{R}$ , we may find an explicit feature map  $\hat{\psi} : \mathbb{R}^{d_E} \to \mathbb{R}^D$ , such that

$$\forall \vec{q}^1, \vec{q}^2 \in \mathbb{R}^{d_E}, \ \langle \hat{\psi}(\vec{q}^1), \hat{\psi}(\vec{q}^2) \rangle = \hat{k}^E(\vec{q}^1, \vec{q}^2) \approx k^E(\vec{q}^1, \vec{q}^2). \tag{3}$$
For example, if  $k^E(\vec{q}^1, \vec{q}^2) = \exp(-\gamma ||\vec{q}^1 - \vec{q}^2||_2^2)$ , then  $\hat{\psi}$  is the Fourier random feature map  $\hat{\psi}(\vec{q}) = \sqrt{\frac{2}{D}} \left[ \cos(\omega_1^T \vec{q} + b_1), ..., \cos(\omega_D^T \vec{q} + b_D) \right]^T$ , where  $\omega_i \sim N(\vec{0}, \gamma^2 I)$  and  $b_i \sim U[0, 1]$ . We obtain a new graph representation  $\hat{\Psi} \in (\mathbb{R}^D)^{n \times n}$ :

$$\hat{\boldsymbol{\Psi}}_{ij} = \begin{cases} \hat{\psi}(\vec{\boldsymbol{q}}_{ij}) \in \mathbb{R}^D, & \text{if } (v_i, v_j) \in \mathcal{E} \\ \vec{\boldsymbol{0}} & \in \mathbb{R}^D, & \text{otherwise} \end{cases}, \text{ where } \vec{\boldsymbol{0}} \text{ is the zero vector in } \mathbb{R}^D.$$

Similarly, the **Lawler's QAP** can be written as

$$\min \ J_{\text{gm}}(\boldsymbol{X}) = -\langle \boldsymbol{K}^N, \boldsymbol{X} \rangle_{\text{F}} - \langle \hat{\boldsymbol{\Psi}}^{(1)} \odot \boldsymbol{X}, \boldsymbol{X} \odot \hat{\boldsymbol{\Psi}}^{(2)} \rangle_{\text{F}_{\mathcal{H}}} \quad \text{s.t. } \boldsymbol{X} \in \mathcal{P}.$$

# Gradient Computation

$$\nabla J_{\alpha}(\boldsymbol{X}) = (1 - 2\alpha) \left[ (\hat{\boldsymbol{\Psi}}^{(1)} * \hat{\boldsymbol{\Psi}}^{(1)}) \boldsymbol{X} + \boldsymbol{X} (\hat{\boldsymbol{\Psi}}^{(2)} * \hat{\boldsymbol{\Psi}}^{(2)}) \right]$$
$$- 2(\hat{\boldsymbol{\Psi}}^{(1)} \odot \boldsymbol{X}) * \hat{\boldsymbol{\Psi}}^{(2)} - \boldsymbol{K}^{N},$$

where

$$(\hat{\mathbf{\Psi}}^{(1)} * \hat{\mathbf{\Psi}}^{(1)}) \mathbf{X} = \sum_{i=1}^{D} \hat{\mathbf{\Psi}}^{(1)}(:,:,i) \hat{\mathbf{\Psi}}^{(1)}(:,:,i) \mathbf{X},$$

$$\mathbf{X}(\hat{\mathbf{\Psi}}^{(2)} * \hat{\mathbf{\Psi}}^{(2)}) = \sum_{i=1}^{D} \mathbf{X} \hat{\mathbf{\Psi}}^{(2)}(:,:,i) \hat{\mathbf{\Psi}}^{(2)}(:,:,i),$$
and 
$$(\hat{\mathbf{\Psi}}^{(1)} \odot \mathbf{X}) * \hat{\mathbf{\Psi}}^{(2)} = \sum_{i=1}^{D} \hat{\mathbf{\Psi}}^{(1)}(:,:,i) \mathbf{X} \hat{\mathbf{\Psi}}^{(2)}(:,:,i).$$

# Entropy-regularized Frank-Wolfe algorithm.

#### Motivation: Hungarian vs Sinkhorn

Hungarian algorithm for  $\min_{\boldsymbol{Y}} \langle \nabla J_{\alpha}(\boldsymbol{X}), \boldsymbol{Y} \rangle_{F}$  s.t.  $\boldsymbol{Y} \in \mathcal{D}$ , Sinkhorn algorithm for  $\min_{\boldsymbol{Y}} \langle \nabla J_{\alpha}(\boldsymbol{X}), \boldsymbol{Y} \rangle_{F} + \lambda H(\boldsymbol{Y})$  s.t.  $\boldsymbol{Y} \in \mathcal{D}$ .

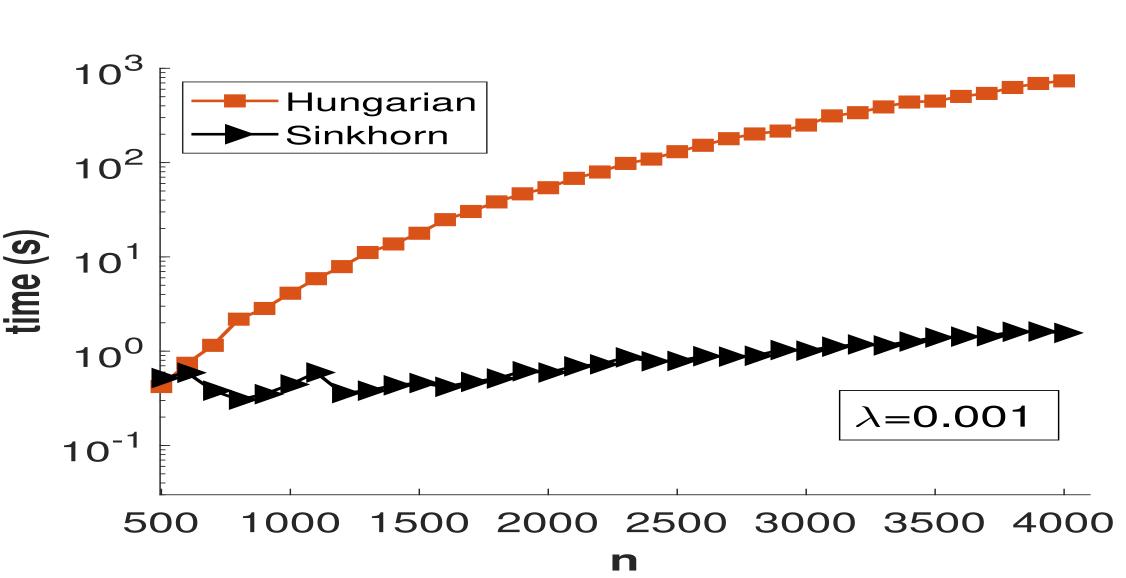


Figure 3: Comparison between the run-times of Hungarian and Sinkhorn.

# Entropy-regularized Frank-Wolfe algorithm (Cont.)

In this work, instead of minimizing  $J_{\alpha}(\mathbf{X})$ , we consider the following entropy-regularized QAP,

$$\min_{\mathbf{X}} F_{\alpha}(\mathbf{X}) = J_{\alpha}(\mathbf{X}) + \lambda H(\mathbf{X}) \quad \text{s.t. } \mathbf{X} \in \mathcal{D},$$

The observation is that if  $\lambda$  is set to be small enough, the solution of the above problem will approximate that of minimizing  $J_{\alpha}(\mathbf{X})$  as much as possible. With the entropy-regularized Frank-Wolfe algorithm, in each outer iteration, we need to solve the following nonlinear problem

$$\min \langle \nabla J_{\alpha}(\boldsymbol{X}), \boldsymbol{Y} \rangle_{\mathrm{F}} + \lambda H(\boldsymbol{Y}) \quad \text{s.t. } \boldsymbol{Y} \in \mathcal{D}_{n},$$
 (4)

which can be extremely efficiently solved by the Sinkhorn algorithm.

#### Experimental results

(1). For the synthetic graph dataset

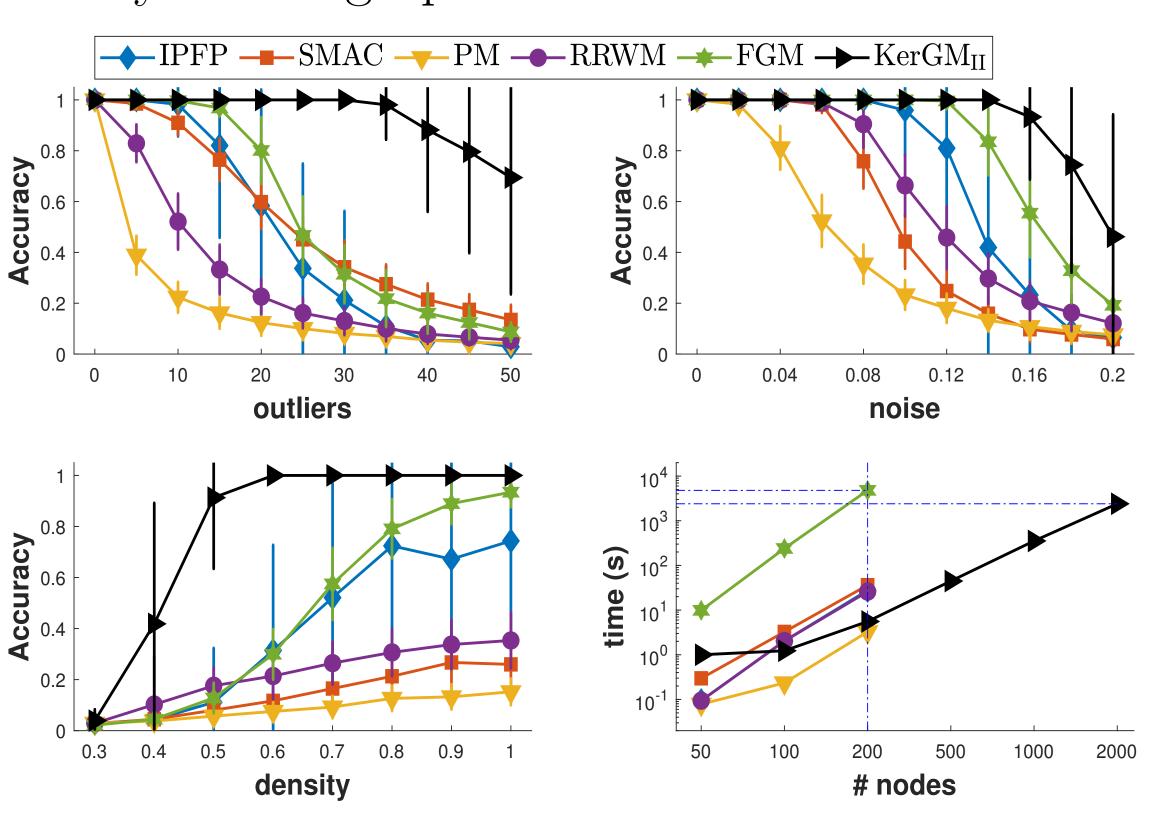


Figure 4: Matching results on synthetic graph dataset.

(2). For the protein-protein interaction (PPI) network dataset

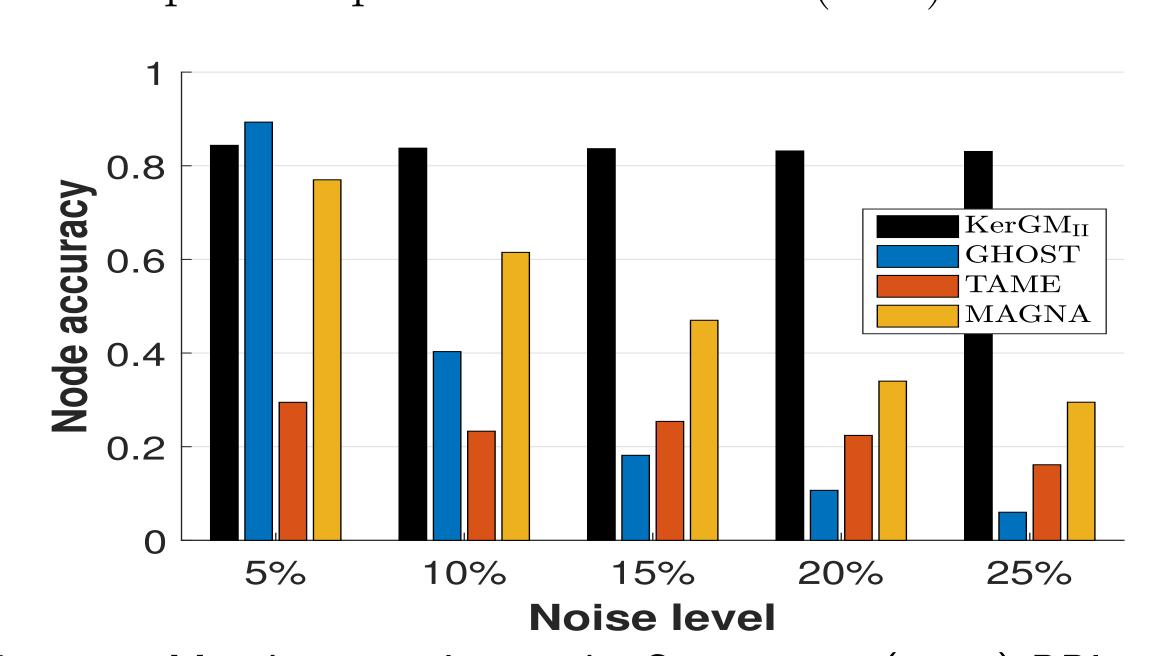


Figure 5: Matching results on the S.cerevisiae (yeast) PPI network.