BIOINFORMATICS

Vol. 00 no. 00 2009
Pages 1–4

Supplementary Material for “Simultaneous Inference of Biological Networks of Multiple Species from Genome-wide Data and Evolutionary Information: A Semi-supervised Approach”

Hisashi Kashima [1,*], Yoshihiro Yamanishi [2], Tsuyoshi Kato [3], Masashi Sugiyama [4] and Koji Tsuda [5]

[1]IBM Research, Tokyo Research Laboratory, 1623-14 Shimo-tsuruma, Yamato, Kanagawa, 242-8502 Japan. [2][2]Mines ParisTech, Centre for Computational Biology, 35 rue Saint-Honore, F-77305 Fontainebleau Cedex, France, and Institut Curie, F-75248, Paris, France and INSERM, U900, F-75248, Paris, France. [3]Ochanomizu University, Center for Informational Biology, 2-1-1 Ohtsuka, Bunkyo-ku, Tokyo 112-8610 Japan. [4]Tokyo Institute of Technology, Department of Computer Science, 2-12-1, O-okayama, Meguro-ku, Tokyo 152-8552 Japan. [5]National Institute of Advanced Industrial Science and Technology, Computational Biology Research Center (AIST), 2-42 Aomi, Koto-ku, Tokyo 135-0064 Japan.

ABSTRACT

The purpose of this manuscript is to give the detailed derivation of the Link Propagation method proposed in the paper titled “*Simultaneous Inference of Biological Networks of Multiple Species from Genome-wide Data and Evolutionary Information*”. This manuscript also includes some experimental results that could not been included in the paper. Section 1 gives the review of the the problem description, and Section 2 gives the detailed formulation of the problem. Section 3 gives the derivation of the proposed algorithm. Section 4 gives the additional experimental results.

1 PROBLEM SETTING

We are considering the problem of inferring the biological networks of n species ($n = 3$ in our case). Let $m^{(k)}$ be the number of nodes (i.e. proteins) in the network of the k-th species, and m be the number of nodes in the largest network. Let $\mathsf{A}^{(k)}$ be the adjacency matrix for the k-th network ($k = 1, 2, \ldots, n$), where the (i, j)-th element $[\mathsf{A}^{(k)}]_{i,j} \equiv 1$ if a link exists between the i-th node and the j-th node in the k-th network, and $[\mathsf{A}^{(k)}]_{i,j} \equiv -1$ if no link exists. If the link status is unknown, we set $[\mathsf{A}^{(k)}]_{i,j} \equiv 0$. Let us denote by $A \equiv (\mathsf{A}^{(1)}, \mathsf{A}^{(2)}, \ldots, \mathsf{A}^{(n)})$ the ordered set of the adjacency matrices. Note that we assume $\mathsf{A}^{(k)}$ is symmetric in our experiments, but the discussion in this paper holds for directed networks if we consider $[\mathsf{A}^{(k)}]_{i,j}$ as a directed link from the i-th node to the j-th node.

Our goal is to infer whether or not a link exists for each of the node pairs whose corresponding element of A is 0. For this purpose, our algorithm provides an ordered set of n matrices $F = (\mathsf{F}^{(1)}, \mathsf{F}^{(2)}, \ldots, \mathsf{F}^{(n)})$, where $\mathsf{F}^{(k)}$ is an $m^{(k)} \times m^{(k)}$ matrix. The (i, j)-th element of $\mathsf{F}^{(k)}$, which we refer to as *link strength* between the i-th and the j-th nodes in the k-th network, represents how likely it is that a link exists between them. A large value of link strength indicates a high confidence that a link exists, and a small value indicates a high confidence that there is no link. One possible use of link strength is to prioritize the protein pairs whose link statuses should be confirmed in actual biological experiments.

Besides the known parts of the networks, we can also exploit biological information about the nodes (proteins) such as protein sequences and gene expressions. We assume that we are given n^2 non-negative similarity matrices $\{\mathsf{W}^{(k,\ell)}\}_{k,\ell=1}^{n}$, where $\mathsf{W}^{(k,\ell)}$ is the similarity matrix between the nodes in the k-th network and the nodes in the ℓ-th network. The (i, j)-th element $[\mathsf{W}^{(k,\ell)}]_{i,j}$ of $\mathsf{W}^{(k,\ell)}$ indicates the nonnegative similarity value between the i-th node in the k-th network and the j-th node in the ℓ-th network. Note that $\mathsf{W}^{(k,\ell)} = \mathsf{W}^{(\ell,k)\top}$. In our experiments, the intra-species similarity matrices ($\mathsf{W}^{(k,\ell)}$ for $k = \ell$) were defined by the gene expression, and the cross-species similarity matrices ($\mathsf{W}^{(k,\ell)}$ for $k \neq \ell$) were constructed from the protein sequence similarities.

Here is the summary of our task:

INPUT:

· An ordered set of n adjacency matrices $A = (\{\mathsf{A}^{(k)}\}_{k=1}^{n})$ representing the known parts of the n networks.

· The n^2 similarity matrices $\{\mathsf{W}^{(k,\ell)}\}_{k,\ell=1}^{n}$ representing the similarities among the nodes.

OUTPUT: The n matrices $F = (\{\mathsf{F}^{(k)}\}_{k=1}^{n})$ representing the link strengths for all pairs of nodes.

2 FORMULATION

Our goal is to estimate the link strength by leveraging the known parts of the networks and the node similarities.

[*] Corresponding author. He is currently with the University of Tokyo, Department of Mathematical Informatics.

Our approach to inferring the unknown parts of the networks is based on the *link propagation principle*, which is, "If two pairs of nodes are similar to each other, then the two pairs have similar link strengths" (Fig. 1). This can be regarded as a pairwise extension of the hypothesis used in the label propagation method [Zhou et al., 2004, Zhu et al., 2003], which is a well-known semi-supervised learning method. For example, let us consider two node pairs, $(i^{(k)}, j^{(k)})$ in the k-th network and $(i^{(\ell)}, j^{(\ell)})$ in the ℓ-th network. Note that this includes the case of $k = \ell$. The hypothesis says that, if the two pairs are similar to each other, their link strength $[F^{(k)}]_{i^{(k)},j^{(k)}}$ and $[F^{(\ell)}]_{i^{(\ell)},j^{(\ell)}}$ should be close to each other.

To formulate this as the objective function of a minimization problem, we need to define a similarity metric between two node pairs. We define the similarity matrix between the node pairs in the k-th network and the node pairs in the ℓ-th network as an $m^{(k)^2} \times m^{(\ell)^2}$ matrix $\tilde{\mathsf{W}}^{(k,\ell)}$, where the $(i^{(k)} + m^{(k)}j^{(k)}, i^{(\ell)} + m^{(\ell)}j^{(\ell)})$-th element $[\tilde{\mathsf{W}}^{(k,\ell)}]_{i^{(k)}+m^{(k)}j^{(k)},i^{(\ell)}+m^{(\ell)}j^{(\ell)}}$ indicates the similarity between the $(i^{(k)}, j^{(k)})$-pair and the $(i^{(\ell)}, j^{(\ell)})$-pair. It is natural to regard two pairs of nodes as similar to each other if the two nodes from different pairs are similar to each other (See also Fig. 1.) In this paper, we define the similarity metric between two node pairs as the product of the node-wise similarities.

$$[\tilde{\mathsf{W}}^{(k,\ell)}]_{i^{(k)}+m^{(k)}j^{(k)},i^{(\ell)}+m^{(\ell)}j^{(\ell)}} \equiv [\mathsf{W}^{(k,\ell)}]_{i^{(k)},i^{(\ell)}} [\mathsf{W}^{(k,\ell)}]_{j^{(k)},j^{(\ell)}}$$

This similarity is equivalently expressed by using matrices as

$$\tilde{\mathsf{W}}^{(k,\ell)} = \mathsf{W}^{(k,\ell)} \otimes \mathsf{W}^{(k,\ell)}, \tag{1}$$

where $\otimes$ denotes the Kronecker product. This similarity metric is the same as the one used in the kernel methods [Basilico and Hofmann, 2004, Ben-Hur and Noble, 2005, Oyama and Manning, 2004].

To represent the link propagation principle, we define the regularization function

$$J_1(F) \equiv \sum_{k,\ell=1}^{n} \sum_{i^{(k)},j^{(k)}=1}^{m^{(k)}} \sum_{i^{(\ell)},j^{(\ell)}=1}^{m^{(\ell)}} [\tilde{\mathsf{W}}^{(k,\ell)}]_{i^{(k)}+m^{(k)}j^{(k)},i^{(\ell)}+m^{(\ell)}j^{(\ell)}} \left([\mathsf{F}^{(k)}]_{i^{(k)},j^{(k)}} - [\mathsf{F}^{(\ell)}]_{i^{(\ell)},j^{(\ell)}}\right)^2,$$

which will appear in our objective function (2).

In addition to making the link strength consistent with the similarity measure, they should also be consistent with the known parts of the networks. In other words, the link strength $[\mathsf{F}^{(k)}]_{i^{(k)},j^{(k)}}$ should be a large value if there is a link between $(i^{(k)}, j^{(k)})$-pair, and should be a small value if no link exists. Therefore, we consider the following loss function J_2 defined as

$$J_2(F) \equiv \sum_{k=1}^{n} \sum_{i^{(k)},j^{(k)}=1}^{m^{(k)}} \left([\mathsf{F}^{(k)}]_{i^{(k)},j^{(k)}} - [\mathsf{A}^{(k)^*}]_{i^{(k)},j^{(k)}}\right)^2,$$

where $\mathsf{A}^{(k)^*}$ is an $m^{(k)} \times m^{(k)}$ matrix which represents the target values defined as

$$[\mathsf{A}^{(k)^*}]_{i^{(k)},j^{(k)}} \equiv \begin{cases} \frac{|A^{(k)^+}|+|A^{(k)^-}|}{|A^{(k)^+}|} & \text{if } [\mathsf{A}^{(k)}]_{i^{(k)},j^{(k)}} = 1, \\ -\frac{|A^{(k)^+}|+|A^{(k)^-}|}{|A^{(k)^-}|} & \text{if } [\mathsf{A}^{(k)}]_{i^{(k)},j^{(k)}} = -1, \\ 0 & \text{otherwise.} \end{cases}$$

where $|A^{(k)^+}|$ is the number of entries in $\mathsf{A}^{(k)}$ satisfying $[\mathsf{A}^{(k)}]_{i^{(k)},j^{(k)}} = 1$, and $|A^{(k)^-}|$ is the number of entries in $\mathsf{A}^{(k)}$ satisfying $[\mathsf{A}^{(k)}]_{i^{(k)},j^{(k)}} = -1$. The magnitude of target value differs depending on whether or not a link exist. This way of setting target values is the one which the Fisher discriminant uses [Bishop, 2006], so the proposed method can be interpreted as applying a semi-supervised version of the Fisher discriminant to pairs of nodes. For pairs whose link existence/absence are unknown, the target values are set to 0 for the predicted link strength not being far from 0, and for making the coefficient matrix of the resultant linear equation (7) regular for numerical stability.

Using the regularization function J_1 and the the loss function J_2, we devise the following objective function J as

$$J(F) \equiv \frac{\sigma}{2} J_1(F) + \frac{1}{2} J_2(F), \tag{2}$$

where σ is a positive constant which balances the two terms. Our method obtains the link strength F by minimizing the objective function J with respect to F.

Now, we rewrite the objective function (2) using matrices as

$$J(F) = \frac{\sigma}{2} \mathbf{vec}(F)^\top \mathsf{L} \mathbf{vec}(F) + \frac{1}{2} \parallel \mathbf{vec}(F) - \mathbf{vec}(A^*) \parallel_2^2, \tag{3}$$

where $A^* \equiv (\mathsf{A}^{(1)^*}, \mathsf{A}^{(2)^*}, \ldots, \mathsf{A}^{(n)^*})$. Here, the **vec** operation for a matrix refers to a vector constructed by stacking all the columns of the matrix. When the **vec** operation is applied to the ordered set of matrices F, it indicates that $\mathbf{vec}(F) \equiv \mathbf{vec}\left(\left[\mathbf{vec}\left(\mathsf{F}^{(1)}\right), \mathbf{vec}\left(\mathsf{F}^{(2)}\right), \ldots, \mathbf{vec}\left(\mathsf{F}^{(n)}\right)\right]\right)$. Also, L in Eq. (3) is the Laplacian matrix defined as

$$\mathsf{L} \equiv \begin{bmatrix} \mathsf{L}^{(1,1)} & \cdots & \mathsf{L}^{(1,n)} \\ \vdots & \ddots & \vdots \\ \mathsf{L}^{(n,1)} & \cdots & \mathsf{L}^{(n,n)} \end{bmatrix}$$
$$\equiv \begin{bmatrix} \tilde{\mathsf{D}}^{(1)} & & 0 \\ & \ddots & \\ 0 & & \tilde{\mathsf{D}}^{(n)} \end{bmatrix} - \begin{bmatrix} \tilde{\mathsf{W}}^{(1,1)} & \cdots & \tilde{\mathsf{W}}^{(1,n)} \\ \vdots & \ddots & \vdots \\ \tilde{\mathsf{W}}^{(n,1)} & \cdots & \tilde{\mathsf{W}}^{(n,n)} \end{bmatrix}, \tag{4}$$

where $\tilde{\mathsf{D}}^{(k)}$ is an $m^{(k)^2} \times m^{(k)^2}$ diagonal matrix whose diagonal elements are defined as

$$[\tilde{\mathsf{D}}^{(k)}]_{i,i} \equiv \sum_{\ell=1}^{n} \sum_{j=1}^{m^{(\ell)^2}} [\tilde{\mathsf{W}}^{(k,\ell)}]_{i,j}. \tag{5}$$

The diagonals of $\tilde{\mathsf{D}}^{(1)}, \tilde{\mathsf{D}}^{(2)}, \ldots, \tilde{\mathsf{D}}^{(n)}$ correspond to the row (or column) sum of $\tilde{\mathsf{W}}$. Remembering that $\tilde{\mathsf{W}}^{(k,\ell)}$ is the pairwise similarity matrix between the k-th species and the ℓ-th species defined by a matrix Kronecker product as in Eq. (1), $\tilde{\mathsf{D}}^{(k)}$ can also be represented by using matrix Kronecker product as

$$\tilde{\mathsf{D}}^{(k)} = \sum_{p=1}^{n} \mathsf{D}^{(k,p)} \otimes \mathsf{D}^{(k,p)}, \tag{6}$$

where $\mathsf{D}^{(k,p)}$ is an $m^{(k)} \times m^{(k)}$ matrix whose diagonal elements are $[\mathsf{D}^{(k,p)}]_{i,i} = \sum_{j=1}^{m^{(p)}} [\mathsf{W}^{(k,p)}]_{i,j}$.

Algorithm 1 Conjugate Gradient $(\mathsf{A}, \mathbf{f}^*, \epsilon)$.

```
1: $\mathbf{f}(0) := \mathbf{f}^*$
2: $\mathbf{r}(0) := \mathbf{f}^* - \mathsf{A}\mathbf{f}(0)$, and $\mathbf{p}(0) := \mathbf{r}(0)$
3: for $t = 0, 1, 2, \ldots$ do
4:     $\mathbf{q}(t) := \mathsf{A}\mathbf{p}(t)$
5:     $\alpha(t) := \langle \mathbf{r}(t), \mathbf{p}(t) \rangle / \langle \mathbf{p}(t), \mathbf{q}(t) \rangle$
6:     $\mathbf{f}(t+1) := \mathbf{f}(t) + \alpha(t)\mathbf{p}(t)$
7:     $\mathbf{r}(t+1) := \mathbf{r}(t) - \alpha(t)\mathbf{q}(t)$
8:     $\beta(t) := \| \mathbf{r}(t+1) \|_2^2 / \| \mathbf{r}(t) \|_2^2$
9:     if $\| \mathbf{r}(t+1) \|_2^2 / \| \mathbf{r}(0) \|_2^2 < \epsilon^2$, return $\mathbf{f}(t+1)$
10:    $\mathbf{p}(t+1) := \mathbf{r}(t+1) + \beta(t)\mathbf{p}(t)$
11: end for
```

Algorithm 2 *Link Propagation Algorithm* $(\{\mathsf{W}^{(k,\ell)}\}_{k,\ell=1}^{n}, A^*, \sigma, \epsilon)$.

```
1: $F(0) := A^*$
2: $R(0) := -\sigma \left[\mathsf{G}^{(1)}(F(0)), \ldots, \mathsf{G}^{(n)}(F(0))\right]$ and $P(0) = R(0)$.
3: for $t = 0, 1, 2, \ldots$ do,
4:     $Q(t) := \sigma \left[\mathsf{G}^{(1)}(P(t)), \ldots, \mathsf{G}^{(n)}(P(t))\right] + P(t)$
5:     $\alpha(t) := \sum_{k=1}^n \langle \mathsf{R}^{(k)}(t), \mathsf{P}^{(k)}(t) \rangle / \sum_{k=1}^n \langle \mathsf{P}^{(k)}(t), \mathsf{Q}^{(k)}(t) \rangle$
6:     $F(t+1) := F(t) + \alpha(t)P(t)$
7:     $R(t+1) := R(t) - \alpha(t)Q(t)$
8:     $\beta(t) := \sum_{k=1}^n \| \mathsf{R}^{(k)}(t+1) \|_2^2 / \sum_{k=1}^n \| \mathsf{R}^{(k)}(t) \|_2^2$
9:     if $\frac{\sum_{k=1}^n \|\mathsf{R}^{(k)}(t+1)\|_2^2}{\sum_{k=1}^n \|\mathsf{R}^{(k)}(0)\|_2^2} < \epsilon^2$, return $F(t+1)$
10:    $P(t+1) := R(t+1) + \beta(t)P(t)$
11: end for
```

To obtain the F that minimizes Eq. (3), we differentiate Eq. (3) with respect to $\mathbf{vec}(\mathsf{F})$, which results in

$$\frac{\partial J(F)}{\partial \mathbf{vec}(F)} = \sigma \mathsf{L} \mathbf{vec}(F) + \mathbf{vec}(F) - \mathbf{vec}(A^*).$$

Setting $\frac{\partial J}{\partial \mathbf{vec}(F)} = \mathbf{0}$ to find the stationary point, we obtain the linear equation

$$(\sigma \mathsf{L} + \mathsf{I})\, \mathbf{vec}(F) = \mathbf{vec}(A^*), \tag{7}$$

which has a unique solution.

3 ALGORITHM

3.1 Conjugate gradient method

To find a solution for the linear equation (7), we use the conjugate gradient method [Golub and Loan, 1996], which is a standard approach to solving linear equations. The basic conjugate gradient algorithm for a linear equation $\mathsf{A}\mathbf{f} = \mathbf{f}^*$ is given as Algorithm 1. In our case, setting

$$\mathsf{A} \equiv \sigma \mathsf{L} + \mathsf{I},\ \mathbf{f}^* \equiv \mathbf{vec}(A^*),\ \text{and}\ \mathbf{f} \equiv \mathbf{vec}(F), \tag{8}$$

we thus obtain our version of the conjugate gradient algorithm detailed in Algorithm 2. Note that the algorithm is described using matrix notation in contrast to the standard conjugate gradient algorithm (Algorithm 1) being described in terms of vectors. Instead of the vectors $\mathbf{f}(t)$, $\mathbf{r}(t)$, $\mathbf{p}(t)$, and $\mathbf{q}(t)$ in the original conjugate gradient algorithm (Algorithm 1), we define the corresponding ordered sets of matrices, $F(t) \equiv \left(\mathsf{F}^{(1)}(t), \ldots, \mathsf{F}^{(n)}(t)\right)$, $R(t) \equiv \left(\mathsf{R}^{(1)}(t), \ldots, \mathsf{R}^{(n)}(t)\right)$, $P(t) \equiv \left(\mathsf{P}^{(1)}(t), \ldots, \mathsf{P}^{(n)}(t)\right)$, $Q(t) \equiv \left(\mathsf{Q}^{(1)}(t), \ldots, \mathsf{Q}^{(n)}(t)\right)$, where the k-th element of each set is an $m^{(k)} \times m^{(k)}$ matrix. The $\langle \cdot, \cdot \rangle$ indicates the inner product of two matrices.

Although most of the steps in Algorithm 2 are obtained naturally by replacing the vectors by the ordered sets of matrices, Line 2 and Line 4 need some derivation since it involves multiplication of A and a vectorized ordered set of matrices.

Let us derive Line 2 of the algorithm. First, we define an operation $\mathsf{Res}(\mathbf{v})$ which reshapes a vector $\mathbf{v}$ to a $\sqrt{|\mathbf{v}|} \times \sqrt{|\mathbf{v}|}$ matrix. In other words, for any square matrix S, it holds that $\mathsf{S} = \mathsf{Res}(\mathbf{vec}(\mathsf{S}))$. Substituting Eqs. (8) into $\mathbf{r}(0) \equiv \mathbf{f}^* - \mathsf{A}\mathbf{f}(0)$, we obtain

$$\begin{aligned} \mathbf{vec}(R(0)) &= \mathbf{vec}(F(0)) - (\sigma \mathsf{L} + \mathsf{I})\mathbf{vec}(F(0)) \\ &= -\sigma \begin{bmatrix} \sum_{\ell=1}^n \mathsf{L}^{(1,\ell)} \mathbf{vec}\left(\mathsf{F}^{(\ell)}(0)\right) \\ \vdots \\ \sum_{\ell=1}^n \mathsf{L}^{(n,\ell)} \mathbf{vec}\left(\mathsf{F}^{(\ell)}(0)\right) \end{bmatrix} \end{aligned}$$

Removing the $\mathbf{vec}$ operation (i.e. applying $\mathbf{Res}$ operation), we obtain

$$\begin{aligned} R(0) = -\sigma \Bigg[&\sum_{\ell=1}^n \mathsf{Res}\left(\mathsf{L}^{(1,\ell)} \mathbf{vec}\left(\mathsf{F}^{(\ell)}(0)\right)\right), \ldots, \\ &\mathsf{Res}\left(\sum_{\ell=1}^n \mathsf{L}^{(n,\ell)} \mathbf{vec}\left(\mathsf{F}^{(\ell)}(0)\right)\right)\Bigg]. \end{aligned} \tag{9}$$

Now, let us define $\mathsf{G}^{(k)}(B)$ as

$$\mathsf{G}^{(k)}(B) \equiv \sum_{\ell=1}^n \mathsf{Res}\left(\mathsf{L}^{(k,\ell)} \mathbf{vec}\left(\mathsf{B}^{(\ell)}\right)\right), \tag{10}$$

where $B \equiv (\mathsf{B}^{(1)}, \mathsf{B}^{(2)}, \ldots, \mathsf{B}^{(n)})$ is an ordered set of n matrices where $\mathsf{B}^{(\ell)}$ is an $m^{(\ell)} \times m^{(\ell)}$ matrix. Substituting Eq. (10) into Eq. (9), we obtain Line 2. Line 4 is also obtained similarly.

3.2 Efficient computation

The computational bottleneck is the evaluation of $\mathsf{G}^{(k)}(B)$ in Eq. (10). Since $\mathsf{L}^{(k,\ell)}$ is an ${m^{(\ell)}}^2 \times {m^{(\ell)}}^2$ matrix, it is quite time- and space-consuming to explicitly build large $\mathsf{L}^{(k,\ell)}$ in the memory ($O(m^4)$ space) and compute their multiplications ($O(m^4)$ time).

Since $\mathsf{L}^{(k,\ell)} \equiv \delta(k = \ell)\tilde{\mathsf{D}}^{(k)} - \tilde{\mathsf{W}}^{(k,\ell)}$ from Eq. (4), Eq. (10) is rewritten as

$$\mathsf{G}^{(k)}(B) = \sum_{\ell=1}^n \mathsf{Res}\left(\delta(k = \ell)\tilde{\mathsf{D}}^{(k)} \mathbf{vec}\left(\mathsf{B}^{(\ell)}\right) - \tilde{\mathsf{W}}^{(k,\ell)} \mathbf{vec}\left(\mathsf{B}^{(\ell)}\right)\right).$$

Using Eq. (1) and (6), this is further rewritten as

$$\begin{aligned} \mathsf{G}^{(k)}(B) = \sum_{\ell=1}^n \Bigg(&\mathsf{Res}\left(\left(\mathsf{D}^{(k,\ell)} \otimes \mathsf{D}^{(k,\ell)}\right) \mathbf{vec}\left(\mathsf{B}^{(k)}\right)\right) \\ &- \mathsf{Res}\left(-\left(\mathsf{W}^{(k,\ell)} \otimes \mathsf{W}^{(k,\ell)}\right) \mathbf{vec}\left(\mathsf{B}^{(\ell)}\right)\right)\Bigg). \end{aligned} \tag{11}$$

The key for efficient computation is based on the following equation called "vec-trick" [Laub, 2005, Vishwanathan et al., 2007] (See also Fig. 2),

$$\left(\mathsf{W}^{(k,\ell)} \otimes \mathsf{W}^{(k,\ell)}\right) \mathbf{vec}\left(\mathsf{B}^{(\ell)}\right) = \mathbf{vec}\left(\mathsf{W}^{(k,\ell)}\mathsf{B}^{(\ell)}\mathsf{W}^{(k,\ell)\top}\right). \quad (12)$$

The right-hand side of Eq. (12) needs only $O(m^3)$ time and $O(m^2)$ space. Using Eq. (12), we can rewrite Eq. (11) as

$$\mathsf{G}^{(k)}(B) = \sum_{\ell=1}^{n} \left(\mathsf{D}^{(k,\ell)}\mathsf{B}^{(k)}\mathsf{D}^{(k,\ell)} - \mathsf{W}^{(k,\ell)}\mathsf{B}^{(\ell)}\mathsf{W}^{(\ell,k)}\right). \quad (13)$$

Note that $\mathsf{D}^{(k,\ell)}$ is a diagonal matrix, so is symmetric. The right-hand side of Eq. (13) reduces the time- and space-complexity of Line 2 and Line 4 in Algorithm 2, compared with Eq. (10).

3.3 Computational Cost

To deal with numerous proteins of many species, scalability is the key factor. However, the memory requirement of the P-SVM is $O(m^4)$, which severely limits its scalability: even for several thousands of proteins, the P-SVM requires tens of gigabytes of the memory. Since the quadratic programming problem needs cubic time complexity with respect to the number of training examples, the time complexity of the P-SVM is theoretically $O(m^6)$. As many fast optimization methods have been developed for the SVM, its practical speed is not too slow in general. Nevertheless, as it is difficult to keep the whole pairwise kernel matrix in memory, we cannot always use the fastest software packages in our problems.

A great advantage of our Link Propagation algorithm is in its memory efficiency. It requires only $O(m^2)$ memory thanks to the "vec-trick". In terms of the computational complexity, it requires $O(m^5)$ time theoretically, because each iteration of our algorithm can be conducted in $O(m^3)$ time, and $O(m^2)$ iterations are required to solve the linear equations completely. However, the number of iterations in practice is much smaller than $O(m^2)$, and in our comparison, the Link Propagation algorithm was roughly 100 times faster than the P-SVM.

4 ADDITIONAL RESULTS

In this section, we give the experimental results that we could not include in the paper. Tables 1, 2, and 3 show the predictive performances measured in AUC of individual inferences and simultaneous inference using various methods including the kernel regression (KR), the P-SVM, and the proposed method, respectively.

Tables 4, 5, and 6 show the predictive performances measured in sensitivity-specificity equilibria where sensitivity is equal to specificity.

REFERENCES

J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

A. Ben-Hur and W.S. Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(Suppl. 1):i38–i46, 2005.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

G.H. Golub and C.F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996.

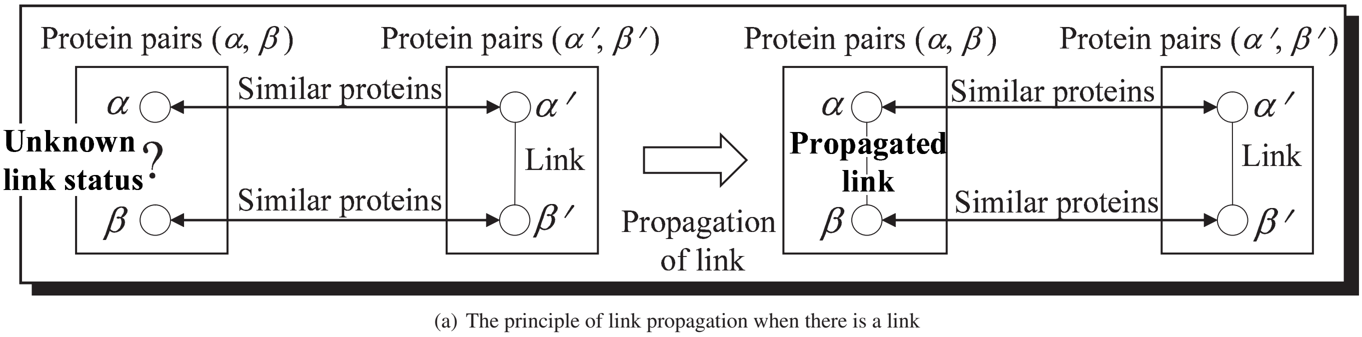
A.J. Laub. *Matrix Analysis for Scientists and Engineers*. Society for Industrial and Applied Mathematics, 2005.

S. Oyama and C.D. Manning. Using feature conjunctions across examples for learning pairwise classifiers. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, pages 322–333, 2004.

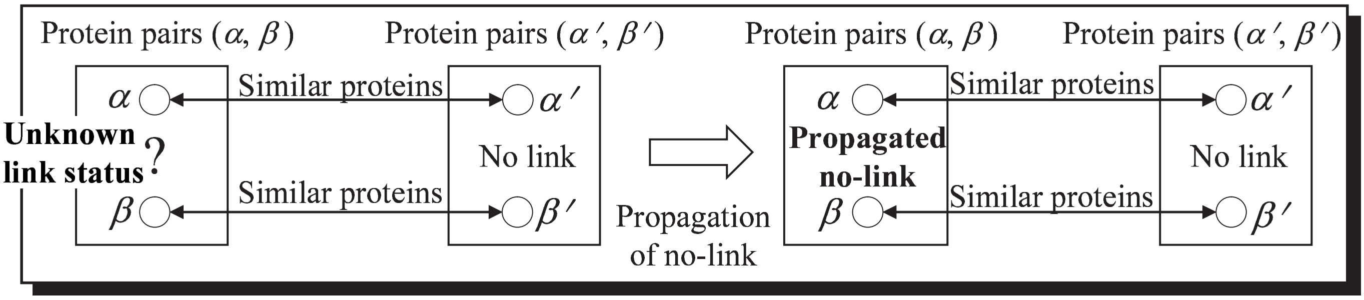
S. V. N. Vishwanathan, K. Borgwardt, and N.N. Schraudolph. Fast computation of graph kernels. In *Advances in Neural Information Processing Systems 19*, 2007.

D. Zhou, O. Bousquet, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328, 2004.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.



(a) The principle of link propagation when there is a link



(b) The principle of link propagation when no link exists

Fig. 1. The principle of link propagation for protein pair (α, β) and protein pair (α', β'). Figure (a) depicts that if two protein pairs are similar to each other, the link for one protein pair is propagated to the other pair whose link status was unknown. Figure (b) shows the case of the no-link status being propagated. The method applies this principle to all protein pairs.

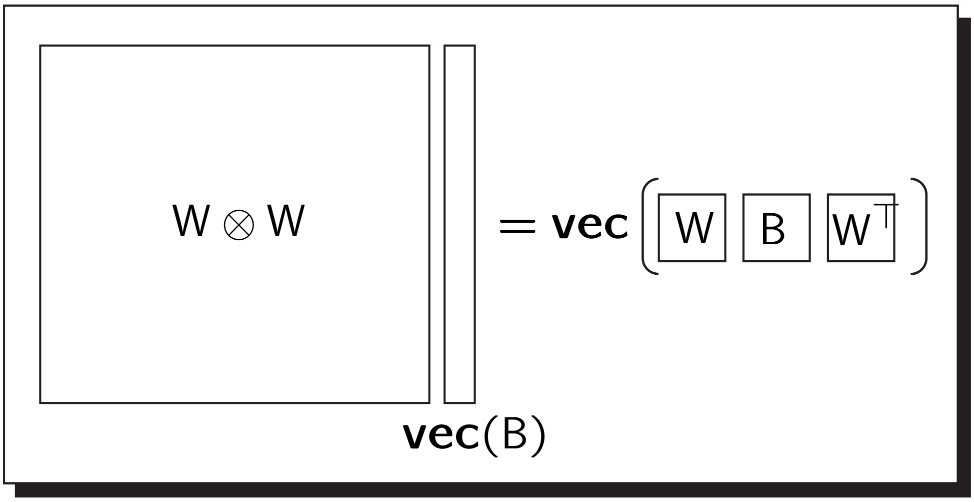

$W \otimes W$
$= \textbf{vec}\left(\boxed{W}\ \boxed{B}\ \boxed{W^\top}\right)$
$\textbf{vec}(B)$

Fig. 2. The "vec-trick" [Laub, 2005, Vishwanathan et al., 2007] accelerates a multiplication of a Kronecker product of two matrices and a vectorized matrix (the l.h.s) by replacing it by matrix multiplications (the r.h.s), which reduces the computational complexity by one order, and the space requirement of the r.h.s becomes the square root of the l.h.s. This equation plays a crucial role in the proposed algorithm (Algorithm 2).

Table 1. Comparison of AUCs by individual inferences and simultaneous inference using the kernel regression (KR).

	C. elegans		H. pylori		S. cerevisiae		total	
ratio of training data	KR (individual)	KR (simultaneous)	KR (individual)	KR (simultaneous)	KR (individual)	KR (simultaneous)	KR (individual)	KR (simultaneous)
25 %	0.591±0.003	0.593±0.002	0.560±0.013	0.565±0.009	0.788±0.009	0.822±0.009	0.715±0.002	0.727±0.002
50 %	0.592±0.005	0.599±0.006	0.563±0.003	0.565±0.005	0.859±0.003	0.883±0.002	0.744±0.003	0.755±0.003
75 %	0.591±0.007	0.605±0.012	0.574±0.014	0.575±0.009	0.897±0.005	0.914±0.006	0.752±0.004	0.765±0.004

Table 2. Comparison of AUCs by individual inferences and simultaneous inference using the P-SVM.

	C. elegans		H. pylori		S. cerevisiae		total	
ratio of training data	P-SVM (individual)	P-SVM (simultaneous)	P-SVM (individual)	P-SVM (simultaneous)	P-SVM (individual)	P-SVM (simultaneous)	P-SVM (individual)	P-SVM (simultaneous)
25 %	0.695±0.005	0.722±0.007	0.594±0.004	0.604±0.002	0.810±0.010	0.832±0.007	0.731±0.005	0.746±0.005
50 %	0.717±0.004	0.752±0.008	0.612±0.009	0.628±0.012	0.866±0.009	0.884±0.005	0.771±0.006	0.789±0.006
75 %	0.776±0.009	0.774±0.013	0.621±0.018	0.648±0.018	0.891±0.010	0.914±0.004	0.821±0.005	0.813±0.004

Table 3. Comparison of AUCs by individual inferences and simultaneous inference using the proposed method.

	C. elegans		H. pylori		S. cerevisiae		total	
ratio of training data	proposed (individual)	proposed (simultaneous)	proposed (individual)	proposed (simultaneous)	proposed (individual)	proposed (simultaneous)	proposed (individual)	proposed (simultaneous)
25 %	0.702±0.004	0.747±0.005	0.600±0.007	0.616±0.007	0.851±0.005)	0.865±0.004	0.749±0.002	0.780±0.002
50 %	0.712±0.005	0.776±0.008	0.617±0.009	0.635±0.008	0.901±0.005	0.909±0.005	0.786±0.005	0.820±0.005
75 %	0.727±0.008	0.791±0.008	0.629±0.016	0.653±0.021	0.921±0.008	0.928±0.009	0.806±0.006	0.840±0.005

Table 4. Comparison of sensitivity-specificity equilibria by individual inferences and simultaneous inference using the kernel regression (KR).

	C. elegans		H. pylori		S. cerevisiae		total	
ratio of training data	KR (individual)	KR (simultaneous)	KR (individual)	KR (simultaneous)	KR (individual)	KR (simultaneous)	KR (individual)	KR (simultaneous)
25 %	0.563±0.004	0.582±0.002	0.544±0.013	0.548±0.010	0.687±0.009	0.724±0.007	0.663±0.002	0.677±0.001
50 %	0.563±0.004	0.587±0.005	0.545±0.006	0.548±0.009	0.751±0.005	0.779±0.006	0.686±0.002	0.698±0.001
75 %	0.559±0.005	0.589±0.010	0.552±0.020	0.555±0.019	0.790±0.005	0.815±0.007	0.694±0.003	0.704±0.003

Table 5. Comparison of sensitivity-specificity equilibria by individual inferences and simultaneous inference using the P-SVM.

	C. elegans		H. pylori		S. cerevisiae		total	
ratio of training data	P-SVM (individual)	P-SVM (simultaneous)	P-SVM (individual)	P-SVM (simultaneous)	P-SVM (individual)	P-SVM (simultaneous)	P-SVM (individual)	P-SVM (simultaneous)
25 %	0.659±0.005	0.667±0.009	0.570±0.003	0.576±0.002	0.726±0.013	0.745±0.011	0.691±0.006	0.689±0.007
50 %	0.680±0.003	0.689±0.008	0.587±0.010	0.590±0.010	0.783±0.010	0.804±0.010	0.723±0.006	0.727±0.007
75 %	0.692±0.005	0.709±0.011	0.599±0.010	0.606±0.014	0.816±0.016	0.842±0.018	0.744±0.008	0.753±0.001

Table 6. Comparison of sensitivity-specificity equilibria by individual inferences and simultaneous inference using the proposed method.

	C. elegans		H. pylori		S. cerevisiae		total	
ratio of training data	proposed (individual)	proposed (simultaneous)	proposed (individual)	proposed (simultaneous)	proposed (individual)	proposed (simultaneous)	proposed (individual)	proposed (simultaneous)
25 %	0.668±0.004	0.693±0.006	0.581±0.009	0.592±0.008	0.759±0.007	0.773±0.005	0.706±0.002	0.722±0.003
50 %	0.679±0.004	0.716±0.007	0.591±0.006	0.603±0.005	0.811±0.011	0.821±0.008	0.735±0.005	0.756±0.004
75 %	0.693±0.009	0.729±0.009	0.607±0.015	0.627±0.023	0.840±0.012	0.851±0.014	0.754±0.007	0.776±0.006