BMC Bioinformatics

Proceedings

A new pairwise kernel for biological network inference with support vector machines

Jean-Philippe Vert*[1], Jian Qiu[2] and William S Noble[2,3]

Address: [1]Centre for Computational Biology, Ecole des Mines de Paris, 35 rue Saint-Honoré, 77300 Fontainebleau, France, [2]Department of Genome Sciences, University of Washington, 1705 NE Pacific Street, Seattle, WA 98195, USA and [3]Department of Computer Science and Engineering, University of Washington, 1705 NE Pacific Street, Seattle, WA 98195, USA

Email: Jean-Philippe Vert* - Jean-Philippe.Vert@ensmp.fr; Jian Qiu - jianq@u.washington.edu; William S Noble - noble@gs.washington.edu

* Corresponding author

This article is available from: http://www.biomedcentral.com/1471-2105/8/S10/S8

Abstract

Background: Much recent work in bioinformatics has focused on the inference of various types of biological networks, representing gene regulation, metabolic processes, protein-protein interactions, etc. A common setting involves inferring network edges in a supervised fashion from a set of high-confidence edges, possibly characterized by multiple, heterogeneous data sets (protein sequence, gene expression, etc.).

Results: Here, we distinguish between two modes of inference in this setting: direct inference based upon similarities between nodes joined by an edge, and indirect inference based upon similarities between one pair of nodes and another pair of nodes. We propose a supervised approach for the direct case by translating it into a distance metric learning problem. A relaxation of the resulting convex optimization problem leads to the support vector machine (SVM) algorithm with a particular kernel for pairs, which we call the *metric learning pairwise kernel*. This new kernel for pairs can easily be used by most SVM implementations to solve problems of supervised classification and inference of pairwise relationships from heterogeneous data. We demonstrate, using several real biological networks and genomic datasets, that this approach often improves upon the state-of-the-art SVM for indirect inference with another pairwise kernel, and that the combination of both kernels always improves upon each individual kernel.

Conclusion: The metric learning pairwise kernel is a new formulation to infer pairwise relationships with SVM, which provides state-of-the-art results for the inference of several biological networks from heterogeneous genomic data.

Background

Increasingly, molecular and systems biology is concerned with describing various types of subcellular networks. These include protein-protein interaction networks, metabolic networks, gene regulatory and signaling pathways, and genetic interaction networks. While some of these networks can be partly deciphered by high-throughput experimental methods, fully constructing any such network requires lengthy biochemical validation. Therefore, the automatic prediction of edges from other available data, such as protein sequences, global network topology or gene expression profiles, is of importance, either to speed up the elucidation of important pathways or to complement high-throughput methods that are subject to high levels of noise [1].

Edges in a network can be inferred from relevant data in at least two complementary ways. For concreteness, consider a network of protein-protein interactions derived from some noisy, high-throughput technology. Our confidence in the correctness of a particular edge A - B in this network increases if we observe, for example, that the two proteins A and B localize to the same cellular compartment or share similar evolutionary patterns [2-4]. Generally, in this type of *direct inference*, two genes or proteins are predicted to interact if they bear some direct similarity *to each other* in the available data.

An alternative mode of inference, which we call *indirect inference*, relies upon similarities between pairs of genes or proteins. In the example above, our confidence in A - B increases if we find some other, high-confidence edge C - D such that the pair $\{A, B\}$ resembles $\{C, D\}$ in some meaningful fashion. Note that in this model, the two connected proteins A and B might not be similar to one another. For example, if the goal is to detect edges in a regulatory network by using time series expression data, one would expect the time series of the regulated protein to be delayed in time compared to that of the regulatory protein. Therefore, in this case, the learning phase would involve learning this feature from other pairs of regulatory/regulated proteins. The most common application of the indirect inference approach in the case of protein-protein interaction involves comparing the amino acid sequences of A and B versus C and D (e.g., [5-8]).

Indirect inference amounts to a straightforward application of the machine learning paradigm to the problem of edge inference: each edge is an example, and the task is to learn to discriminate between "true" and "false" edges. Not surprisingly, therefore, several machine learning algorithms have been applied to predict network edges from properties of protein pairs. For example, in the context of machine learning with support vector machines (SVM) and kernel methods, Ben-Hur and Noble [8] describe how

to map an embedding of individual proteins onto an embedding of pairs of proteins. The mapping defines two pairs of proteins as similar to each other when each protein in a pair is similar to one corresponding protein in the other pair. In practice, the mapping is defined by deriving a kernel function on pairs of proteins from a kernel function on individual proteins, obtained by a tensorization of the initial feature space. We therefore call this pairwise kernel the *tensor product pairwise kernel* (TPPK, see Methods section).

Less attention has been paid to the use of machine learning approaches in the direct inference paradigm. Two exceptions are the works of Yamanishi *et al.* [9] and Vert *et al.* [10], who derive supervised machine learning algorithms to optimize the measure of similarity that underlies the direct approach by learning from examples of interacting and non-interacting pairs. Yamanishi *et al.* employ kernel canonical correlation analysis to embed the proteins into a feature space where distances are expected to correlate with the presence or absence of interactions between protein pairs. Vert *et al.* highlight the similarity of this approach with the problem of distance metric learning [11], while proposing an algorithm for that purpose.

Both of these direct inference approaches, however, suffer from two important drawbacks. First, they are based on the optimization of a proxy function that is slightly different from the objective of the embedding, namely, finding a distance metric such that interacting/non-interacting pairs fall above/below some threshold. Second, the methods of [9] and [10] are applicable only when the known part of the network used for training is the set of all edges among a subset of proteins in the network. In other words, in order to apply these methods, we must have a complete set of high-confidence edges for one set of proteins, from which we can infer edges in the rest of the network by assuming that edges not observed among the proteins in the training set are really absent. This setting is often unrealistic. In practice, our training data will generally consist of known positive and negative edges distributed throughout the target network. For example, in the case of protein-protein interactions, one typically derives positive examples of interactions from experimental assays, while negative examples can be sampled randomly among non-interacting pairs or generated from pairs of proteins known to be present in different cellular localization or expressed under different conditions; the methods of [9] and [10] can not be used in this setting.

In this paper we propose a convex formulation for supervised learning in the direct inference paradigm that overcomes both of the limitations mentioned above. This formulation stems from a particular formulation of the

distance metric learning problem [10,11]. We show that a slight relaxation of this formulation bears surprising similarities with the supervised approach of [8], in the sense that it amounts to defining a kernel between pairs of proteins from a kernel between individual proteins. We therefore call our method the *metric learning pairwise kernel* (MLPK). An important property of this formulation as an SVM is the possibility to learn from several data types simultaneously by combining kernels, which is of particular importance in various bioinformatics applications [12,13].

Several authors have proposed algorithms for distance metric learning with kernels related to our method. Tsang and Kwok [14] propose a quadratic program (QP) formulation of the problem, while Weinberger *et al.* [15] propose a semidefinite programming formulation in the context of distance metric learning for *k*-nearest-neighbour classifiers. In both cases, however, a specific algorithm must be implemented. To the contrary, the formulation we propose builds upon the well-known SVM algorithm. Any practitioner of SVM can therefore easily use it with most public SVM implementations, at the price of using a specific kernel. A second advantage of our SVM formulation is that it can be easily combined with other SVM formulation, such as the TPPK approach, by forming linear combinations of different kernels.

We validate the MLPK approach on the task of reconstructing two yeast networks: the network of metabolic pathways and the co-complex network. In each case, the network is inferred from a variety of genomic and proteomic data, including protein amino acid sequences, gene expression levels over a large set of experiments, and protein subcellular localization. We show that the MLPK approach nearly always provides better prediction performance than the state-of-the-art TPPK approach, and that the combination of the MLPK and TPPK together almost always leads to the best results.

Results and discussion

In this section we present a comparison of the previously described TPPK kernel and the new MLPK kernel for the reconstruction of two biological networks: the metabolic network and the co-complex protein network. For each network, we cast the problem of network reconstruction as a binary classification problem, where the presence or absence of edges must be inferred from various types of data relevant to the problem. Because the network contains relatively few edges compared to the total number of possible pairs, we created a balanced dataset by keeping all known edges as positive examples and randomly sampling an equal number of absent edges as negative examples. We compare the utilities of the TPPK and MLPK kernels in this context by assessing the performance of an

SVM for edge prediction in a five-fold cross-validation experiment repeated three times (3 × 5 cv) with different random folds. At each fold, the regularization parameter C of the SVM is chosen among 18 values evenly log-spaced on the interval [10^{-4}, 50] by minimizing the classification error estimated by five-fold cross-validation within the training set only. We also assess the performance of the pairwise kernel obtained by summing the TPPK and MLPK kernels, which we call *MLPK + TTPK* below. The MLPK + TPPK kernel is a simple way to combine the information contained in the MLPK and TTPK kernels. We also test two approaches to integrate the various genomic and proteomic data for edge prediction. First we construct an integrated kernel over genes, obtained by adding together all kernels defined by the various data, and deduce a TPPK, MLPK or MLPK + TTPK pairwise kernel from this integrated kernel. This is a simple approach to data integration that has proved useful in previous work [12,16]. Alternatively, we consider the pairwise kernels deduced from each individual genomic data, and add them together to form an integrated pairwise kernel.

As a baseline method for direct inference, for each kernel between genes we also assess the performance of a direct method that ranks the candidate edges by increasing distance between the two gene involved, where the distance between two genes is derived from the kernel value by the equation:

$$d_K(x, y) = \sqrt{K(x, x) + K(y, y) - 2K(x, y)}.$$

Metabolic network

Most biochemical reactions in living organisms are catalyzed by particular proteins called enzymes, and occur sequentially to form metabolic pathways. For example, the degradation of glucose into pyruvate (called glycolysis) involves a sequence of ten chemical reactions catalyzed by ten enzymes. The metabolic gene network is defined as an undirected graph with enzymes as vertices and with edges connecting pairs of enzymes that can catalyze successive chemical reactions. The reconstruction of metabolic pathways for various organisms is of critical importance, e.g., to find new ways to synthesize chemical compounds of interest. This problem motivated earlier work on supervised graph inference [9,10]. Focusing on the budding yeast *S. cerevisiae*, we collected the metabolic network and genomic data used in [9]. The network was extracted from the KEGG database and contains 769 vertices and 3702 undirected edges.

In order to infer the network, various independent data about the proteins can be used. In this experiment, we use four relevant sources of data provided by [9]: (1) a set of 157 gene expression measurements obtained from DNA microarrays; (2) the phylogenetic profiles of the genes,

represented as 145-bit vectors indicating the presence or absence of each gene in 145 fully sequenced genomes; (3) the protein's localization in the cell determined experimentally [17], represented as 23-bit vectors corresponding to 23 cellular compartments, and (4) yeast two-hybrid protein-protein interaction data [1], represented as a network. For the first three data sets, a Gaussian RBF kernel was used to represent the data as a kernel matrix. For the yeast two-hybrid network, we use a diffusion kernel [18]. All data were downloaded from http://web.kuicr.kyoto-u.ac.jp/~yoshi/ismb04

Table 1 shows the performance of each pairwise kernel, as well as that of the baseline direct approach, for the different data sets. The MLPK is never worse than the TPPK kernel, and both methods are always much better than the baseline direct method for edge inference. The two kernels have similar performance on the sum kernel; MLPK is slightly better than TPPK on the expression, localization and phylogenetic profile kernels, and much better on the yeast two-hybrid dataset (76.6% vs. 59.2% in accuracy). Finally we observe that the integrated kernel MLPK + TPPK is always at least as good as the best of MLPK or TPPK alone, confirming that MLPK and TTPK are complementary to one another.

Interestingly, we note that although connected pairs, i.e., pairs of enzymes acting successively in a pathway, are expected to have similar expression, phylogenetic profiles and localization (explaining the good performance of the MLPK on these datasets), the indirect approach implemented by the TPPK also gives good results for these data. This result implies that for these data, interacting pairs in the training set are often similar not only to each other but also to other interacting pairs in the training set. This observation is not surprising because, for example, if two proteins in the test set are co-localized in a particular organelle, then it is likely that interacting pairs of proteins

co-localized in the same organelle are also present in the training set.

In the case of yeast two-hybrid data, on the other hand, the kernel between single proteins is defined as a diffusion kernel over the yeast two-hybrid graph. One can speculate that, in that case, similarity between pairs can be easily assessed and used by the MLPK to predict edges, but similarity between pairs as defined by the TPPK kernel is less likely to be observed. In a sense, the dimensionality of the feature space of the diffusion kernels is much larger than that defined by the other kernels, and a protein is only close to its neighbors in the yeast two-hybrid graph.

Regarding the integration of heterogeneous data sets, the pairwise kernels deduced from the sum of the individual kernels performs slightly better than the sum of the pairwise kernels deduced from individual kernels, which performs itself always better than the best of the pairwise kernels deduced from individual kernels. This confirms that the simple addition of kernels is a simple and powerful means to learn from heterogeneous data, and shows that in the case of pairwise kernels it seems better to first integrate heterogeneous data at the level of individual genes, before converting this integrated kernel into a pairwise kernel.

Protein complex network
Many proteins carry out their biological functions by acting together in multi-protein structures known as complexes. Understanding protein function therefore requires identification of these complexes. In the co-complex network, nodes are proteins, and an edge between proteins *A* and *B* exists if *A* and *B* are members of the same protein complex. Some high-throughput experimental methods, such as tandem affinity purification followed by mass spectrometry, explicitly identify these co-complex relationships, albeit in a noisy fashion. Also, computational

Table 1: Performance on reconstruction of the yeast metabolic networks.

Data	MLPK		TPPK		MLPK + TPPK		Direct
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	AUC
Expression	77.9 ± 1.2	84.8 ± 1.2	77.4 ± 0.9	84.1 ± 0.4	78.2 ± 0.9	84.9 ± 1.3	51.9 ± 1.6
Localization	63.8 ± 2.2	67.5 ± 3.0	62.4 ± 1.0	65.6 ± 0.8	64.4 ± 0.9	66.3 ± 1.0	55.1 ± 1.4
Phylogenetic profile	79.5 ± 0.9	84.3 ± 0.9	77.7 ± 1.6	83.6 ± 1.7	80.7 ± 0.8	85.4 ± 1.1	60.7 ± 1.4
Yeast two-hybrid	75.9 ± 1.2	82.5 ± 1.4	59.4 ± 1.0	65.4 ± 1.7	76.7 ± 0.8	83.0 ± 0.4	51.6 ± 1.4
Sum	83.9 ± 0.7	91.6 ± 0.5	84.0 ± 0.7	91.2 ± 0.4	83.9 ± 0.9	91.5 ± 0.6	60.6 ± 1.3
Pairwise sum	81.4 ± 0.5	89.0 ± 0.4	80.7 ± 1.1	88.6 ± 0.6	81.6 ± 0.7	89.2 ± 0.8	-

The table lists, for each type of data, the accuracy and area under the ROC curve obtained by each pairwise kernel. Values in the tables are means and standard errors in a 3 × 5 cv experiment. TPPK is the tensor product pairwise kernel, and MLPK is the metric learning pairwise kernel. The column MLPK + TTPK shows the results when an SVM is trained with the sum of the MLPK and TPPK pairwise kernels. The row *Sum* shows the results when the kernel between the genes is the sum of the expression, localization, phylogenetic profile and yeast two-hybrid kernels. The line *Pairwise sum* shows the results obtained with the SVM when the pairwise kernel used is the sum of pairwise kernels derived from the expression, localization, phylogenetic profile and yeast two-hybrid kernels, respectively. The *Direct* column shows the result of the direct method, where gene pairs are ranked according to their distance as defined by each kernel to predict edges.

methods exist for inferring the co-complex network from individual data types or from multiple data types simultaneously [19,20]. We derived the co-complex data set based on an intersection of the manually curated MIPS complex catalogue [21] and the BIND complex data set [22]. The co-complex network contains 3280 edges connecting 797 proteins. In addition, our data set contains 3081 proteins with no co-complex relationships.

For this evaluation, we again use four different data sets that we consider relevant to the co-complex network. The first data set is the same localization data that we used above [17]. The second is derived from a chip-based version of the chromatin immunoprecipitation assay (so-called "ChIP-chip" data) [23]. This assay provides evidence that a transcription factor binds to the upstream region of a given gene and is likely to regulate the expression of the given gene. Our data set contains data for 113 transcription factors, and so yields a vector of length 113 for each protein. The final two data sets are derived from the amino acid sequences of the yeast proteins. For the first, we compared each yeast protein to every model in the Pfam database of protein domain HMMs (pfam.wustl.edu) and recorded the E-value of the match. This comparison yields a vector of length 8183 for each protein. Finally, in a similar fashion, we compared each yeast protein to each protein in the Swiss-Prot database version 40 (ca.expasy.org/sprot) using PSI-BLAST [24], yielding vectors of length 101,602. Each of the four data sets is represented using a scalar product kernel.

We used the same experimental procedure to compare the quality of edge predictors for the co-complex network using MLPK, TPPK and their combination MLPK + TPPK. The results, shown in Table 2, again show the value of the MLPK approach. Using either performance metric (accuracy or ROC area), the MLPK approach performs better than the TPPK approach on three out of four data sets. Both methods strongly outperform the direct approach on all datasets.

Most striking is the improvement for the ChIP-chip data set (accuracy from 63.8% to 82.2%). This result is expected, because we know that proteins in the same complex must act in concert. As such, they are typically regulated by a common set of transcription factors.

In contrast, the MLPK approach does not perform better than TPPK on the localization data set. This is, at first, suprising because two proteins must co-localize in order to participate in a common complex. This problem is thus an example of the direct inference case for which the MLPK is designed. However, the localization data is somewhat complex because (1) only approximately 70% of yeast proteins are assigned any localization at all, and (2)

many proteins are assigned to multiple locations. As a result, among 3280 positive edges in the training set, only 1852 (56%) of those protein pairs share exactly the same localization. Furthermore, 550 (16.8%) of the 3280 negative edges used in training connect proteins with the same localization, primarily "Unknown." These factors make direct inference using this data set difficult. The indirect method, by contrast, is apparently able to identify useful relationships, corresponding to specific localizations, that are enriched among the positive pairs relative to the negative pairs.

The fact that the MLPK and TPPK capture complementary information is further demonstrated by the good performance of the combined MLPK + TPPK approach, which is always better than both TPPK and MLPK alone on all datasets. Finally, the relevance of heterogeneous data integration by kernel summation is again demonstrated by the excellent results obtained in this case, with a slight advantage to the construction of a pairwise kernel over the integrated kernel for genes. The combination of MLPK + TPPK over the integrated kernel results in the best performance.

Conclusion

We showed that a particular formulation of metric distance learning for graph inference can be formulated as a convex optimization problem and can be applied to any data set endowed with a positive definite kernel. A relaxation of this problem leads to the SVM algorithm with the new MLPK kernel (5) between pairs. Experiments on two biological networks confirm the value of this approach for the reconstruction of biological network from heterogeneous genomic and proteomic data.

The MLPK kernel is derived from a new formulation for distance metric learning. Contrary to other formulations [14,15] the resulting algorithm is a classical SVM with a particular kernel. This formulation can therefore benefit from the popularity of SVM in the computational biology community coupled with the availability of numerous public implementations of SVM, to solve various problems of gene or protein network inference, or more generally pairwise relationships inference.

This formulation, however, is obtained at the price of relaxing a positive definiteness constraint for the sake of computational efficiency. While the experimental results validate the approach for practical gene network inference, the relaxed formulation can not be considered as a distance metric learning algorithm anymore, because the final metric matrix may have negative eigenvalues. This discrepancy between the motivation of our approach (formulating graph inference as distance metric learning) and the final algorithm might complicate the interpretation of

the results obtained, and will be subject to further investigations in the future.

Beyond the direct and indirect approaches to graph inference mentioned in the introduction, there exist many alternative ways to infer networks, such as estimating conditional independence between vertices with Bayesian networks [25]. An interesting property of methods based on supervised learning, such as the SVM with the TPPK and MLPK kernels, is the limited hypothesis made on the nature of the edges; the only hypothesis made is that there is information related to the presence or absence of edges in the data, and we let the learning algorithm model this information. The good accuracy obtained on two completely different networks (metabolic and co-complex) supports the general utility of this approach.

An interesting and important avenue for future research is the extension of these approaches to inference of directed graphs, e.g., regulatory networks. Although the TPPK and MLPK approaches are not adapted as such to this problem, variants involving for example kernels between ordered pairs could be studied.

Methods
In this section we first explain how SVM can be used for graph inference, present the TPPK and MLPK kernels and provide some intuitive analysis of their differences. We then provide a detailed derivation of the MLPK kernel in the context of distance metric learning. After explaining the link between graph inference and distance metric learning, we first propose a new algorithm for distance metric learning when the genomic data are represented by vectors. We then generalize this algorithm to the case where the data are not necessarily finite-dimensional vectors, but more generally when a positive definite kernel is defined over the vertices. Finally, we introduce a relaxation of the resulting optimization problem, and we show that the problem is then equivalent to an SVM for a particular pairwise kernel, which we explicitly identify as the MLPK.

SVM and positive definite kernels
Our method for graph inference is based on the SVM algorithm, a widely-used algorithm for supervised binary classification [26,27]. Given a set of points $x_1,...,x_n$ with binary labels $y_1,...,y_n \in \{-1, 1\}$, SVM estimate a function:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) = b, \qquad (1)$$

to predict the label of any new point x by the sign of $f(x)$. The function K in (1) is the so-called *kernel*, which must be a symmetric and positive definite function (i.e., for any integer p and any set of points $u_1,...,u_p$ the square $p \times p$ matrix $K_{i,j} = K(u_i, u_j)$ must be symmetric and positive semidefinite). The weights α_i ($i = 1,...,n$) and offset b in (1) are obtained by solving the following quadratic program:

$$\min_{\alpha,b,\zeta} \sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) + C \sum_{i=1}^{n} \zeta_i, \qquad (2)$$

under the constraints

$$\zeta_i \geq 0, \quad i = 1,...,n,$$

$$\zeta_i \geq 1 - y_i \left(\sum_{j=1}^{n} \alpha_j K(x_j, x_i) + b \right), \quad i = 1,...,n. \qquad (3)$$

An interesting property of SVM is the complete modularity between the choice of the kernel K, on the one hand, and the algorithm. In other words the same SVM implementation can be used to process different data and solve different problems by simply modifying the data and the kernel used.

Pairwise kernels for graph inference
We formulate the problem of supervised graph inference as follows: given a set of known interacting and non-interacting pairs of genes, build a classification function to predict for all pairs not used in the training phase whether they interact or not. In order to formalize this problem let us assume that a gene is represented by a point x, and that

Table 2: Performance on reconstruction of the yeast co-complex networks.

Data	MLPK Accuracy	MLPK AUC	TPPK Accuracy	TPPK AUC	MLPK + TPPK Accuracy	MLPK + TPPK AUC	Direct AUC
Localization	76.2 ± 1.0	76.9 ± 2.0	79.5 ± 1.8	82.9 ± 1.7	80.6 ± 0.7	83.0 ± 1.2	73.9 ± 1.4
Chip-chip	82.2 ± 1.1	89.7 ± 0.8	63.8 ± 1.2	68.0 ± 1.1	84.4 ± 1.2	90.8 ± 1.2	58.4 ± 1.5
Pfam	92.1 ± 0.9	98.0 ± 0.5	86.1 ± 1.0	91.8 ± 0.9	93.8 ± 0.3	98.5 ± 0.1	67.3 ± 1.2
PSI-BLAST	89.0 ± 0.9	97.0 ± 0.1	88.3 ± 1.0	93.5 ± 0.9	93.1 ± 0.6	97.9 ± 0.2	67.8 ± 1.2
Sum	93.6 ± 0.3	98.7 ± 0.2	94.1 ± 0.6	98.0 ± 0.3	95.8 ± 0.3	99.1 ± 0.3	79.9 ± 0.8
Pairwise sum	93.3 ± 0.8	98.2 ± 0.4	90.5 ± 0.9	96.3 ± 0.7	95.2 ± 0.3	98.9 ± 0.2	-

The table lists, with the notation conventions explained in Figure 1, the results of the different methods on the reconstruction of the yeast co-complex networks.

a kernel K between genes has been chosen. This kernel can for example be derived from genomic data, such as a microarray expression profile. We consider a set of n genes $x_1,...,x_n$, and a training set $\mathcal{T} = \mathcal{I} \cup \mathcal{N}$ of interacting ($\mathcal{I}$) and non-interacting ($\mathcal{N}$) pairs; our objective is to learn a function to predict which pairs outside the training set interact or not.

By labeling +1 interacting pairs and -1 non-interacting pairs, this problem is a classical binary supervised classification problem, which can be solved with a SVM as soon as a kernel is defined. The difficulty is that the patterns to be classified are *pairs* of genes, while we assume that only a kernel between *individual* genes is available.

Ben-Hur and Noble proposed in [8] a general formula to create a kernel between pairs or patterns from a kernel between individual patterns:

$$K_{TPPK}((x_1, x_2), (x_3, x_4)) = K(x_1, x_3)K(x_2, x_4) + K(x_1, x_4)K(x_2, x_3). \qquad (4)$$

The rationale behind this tensor product pairwise kernel (TPPK) is that the comparison between a pair (x_1, x_2) and another pair (x_3, x_4) is done through the comparison of x_1 with x_3 and x_2 with x_4 (using the kernel between individual genes), on the one hand, and the comparions of x_1 with x_4 and x_2 with x_3, on the other hand.

In this paper we propose another pairwise kernel as follows:

$$K_{MLPK}((x_1, x_2), (x_3, x_4)) = (K(x_1, x_3) - K(x_1, x_4) - K(x_2, x_3) + K(x_2, x_4))^2. \qquad (5)$$

This metric learning pairwise kernel (MLPK) is justified in detail in the following subsections and its link with the problem of distance metric learning highlighted. Although the formula of the MLPK (5) might seem less intuitive than the TPPK (4), some simple algebra can help highlight their difference. Indeed, any positive definite kernel can be written as an inner product after embedding the points to some Hilbert space [28]:

$$K(x, x') = \Phi(x)\top\Phi(x'), \qquad (6)$$

where Φ is the mapping from the space of pattern to the feature Hilbert space. Consequently the MLPK can be rewritten as follows by plugging (6) into (5):

$$K_{MLPK}((x_1, x_2), (x_3, x_4)) = [(\Phi(x_1) - \Phi(x_2))\top(\Phi(x_3) - \Phi(x_4))]^2. \qquad (7)$$

This equation suggests that, up to the square exponent, the MLPK is an inner product between pairs after mapping a pair (x_1, x_2) to the vector $\Phi(x_1) - \Phi(x_2)$. Hence a major difference between the TPPK and MLPK is that the former involves comparison between individual genes of the first pair and individual genes of the second pair, while the later compares pairs through the differences between their elements (in the feature space). In particular two pairs might be very similar with respect to the MLPK kernel even if the patterns of the first pair are very different from the patterns of the second pair, resulting in a large dissimilarity with respect to the TPPK kernel.

The rest of this section is devoted to a more rigorous derivation of the MLPK kernel, in particular to show its relationship to distance metric learning

Distance metric learning

Following [10], we note that a possible approach to solve the problem of graph inference is to learn a distance metric d between genes with the property that pairs of nearby genes with respect to d are connected by an edge, while pairs of genes far from each other are not. If such a metric is available, then the prediction of an edge between a candidate pair of genes simply amounts to computing their distance to each other and predicting an edge if the distance is below a threshold.

More formally, let us first assume that genes are represented by finite-dimensional vectors and investigate distance metrics obtained by linear transformations of the input space. Such metrics are indexed by symmetric positive semidefinite matrices M as follows:

$$d_M(x, x') = (x - x')\top M(x - x').$$

Our goal is to learn a distance metric which separates interacting from non-interacting pairs, while controlling over-fitting to the training set. Following the spirit of the SVM algorithm, we enforce an arbitrary margin of 2 between the distances of interacting and non-interacting variables up to slack variables, and control the Frobenius norm of M by considering the following problem:

$$\min_{\gamma, M, \zeta} \|M\|^2_{Fro} + C \sum_{(i,j)\in\mathcal{T}} \zeta_{ij}, \qquad (8)$$

under the constraints:

$$\begin{aligned} \zeta_{ij} &\geq 0, \quad (i, j) \in \mathcal{T}, \\ d_M(x_i, x_j) &\leq \gamma - 1 + \zeta_{ij}, \quad (i, j) \in \mathcal{I}, \\ d_M(x_i, x_j) &\geq \gamma + 1 - \zeta_{ij}, \quad (i, j) \in \mathcal{N}, \\ M &\succeq 0. \end{aligned} \qquad (9)$$

In order to solve this problem we first prove the following extension to the representer theorem [29]:

Theorem 1
The solution of (8–9) can be expanded as:

$$M = \sum_{(i,j) \in \mathcal{T}} \alpha_{ij}(x_i - x_j)(x_i - x_j)^{\top},$$

with $\alpha_{ij} \in \mathbb{R}$ for $(i, j) \in \mathcal{T}$.

Proof
For any pair (i, j), let us denote $u_{ij} = x_i - x_j$, and let D_{ij} be the $p \times p$ matrix $D_{ij} = (x_i - x_j)(x_i - x_j)^{\top} = u_{ij} u_{ij}^{\top}$. Then we can rewrite

$$d_M(x_i, x_j) = \langle M, D_{ij} \rangle_{Fro},$$

where $\langle A, B \rangle_{Fro} = Trace(A^{\top}B)$ is the Frobenius inner product. Introducing the hinge loss function $L(\gamma, \gamma') = \max(1 - \gamma\gamma', 0)$ for $\gamma, \gamma' \in \mathbb{R}$, and the indicator variables:

$$\gamma_{ij} = \begin{cases} 1 & \text{if } (i,j) \in N, \\ -1 & \text{if } (i,j) \in I, \end{cases}$$

we can eliminate the slack variables and rewrite the problem (8–9) as:

$$\min_{M \succeq 0, \gamma \in \mathbb{R}} \|M\|_{Fro}^2 + C \sum_{(i,j) \in \mathcal{T}} L(\langle M, D_{ij} \rangle_{Fro} - \gamma, \gamma_{ij}).$$

$$(10)$$

This shows that the optimization problem is in fact equivalent, up to the positive semidefinitiveness constraint, to an SVM in the linear space of symmetric matrices endowed with the Frobenius inner product. Each edge example is then mapped to the matrix D_{ij}. In particular, if the constraint on M was not present, then Theorem 1 would be exactly the representer theorem. Here we need to show that it still holds with the constraint $M \succeq 0$. For this purpose let $M \succeq 0$ and $\gamma \in \mathbb{R}$ be the solution of (8–9). M can be uniquely decomposed as $M = M_S + M_{\perp}$, where M_S is in the linear span of $(D_{ij}, (i, j) \in \mathcal{T})$ and $\langle M_{\perp}, D_{ij} \rangle_{Fro} = 0$ for $(i, j) \in \mathcal{T}$. By the Pythagorean theorem we have $\|M\|_{Fro}^2 = \|M_S\|_{Fro}^2 + \|M_{\perp}\|_{Fro}^2$, so if $M_{\perp} \neq 0$ the functional minimized in (10) is strictly smaller at (M_S, γ) than at (M, γ); this would be a contradiction if $M_S \succeq 0$. Therefore, to prove the theorem it suffices to show $M_S \succeq 0$. Let $v \in \mathbb{R}^p$ be any vector. We can decompose that vector uniquely as $v =$

$v_S + v_{\perp}$, where v_S is in the linear span of the u_{ij}, $(i, j) \in \mathcal{T}$, and $v_{\perp}^{\top} u_{ij} = 0$ for $(i, j) \in \mathcal{T}$. We then have $M_S v_{\perp} = 0$ and $M_{\perp} v_S = 0$, and therefore

$$v^{\top} M_S v = v_S^{\top} M_S v_S = v_S^{\top} M_S v_S + v_S^{\top} M_{\perp} v_S = v_S^{\top} M v_S \geq 0,$$

where we used the fact that $M \succeq 0$ in the last inequality. This is true for any $v \in \mathbb{R}^p$, which shows that $M_S \succeq 0$, concluding the proof. ■

By plugging the result of Theorem 1 into (8–9) we see that this problem is equivalent to that of finding α_{ij}, $(i, j) \in \mathcal{T}$ and γ. In order to write out the problem explicitly, let us introduce the following kernel between two pairs (x_1, x_2) and (x_3, x_4):

$$\begin{aligned} K_{MLPK}((x_1, x_2), (x_3, x_4)) &= \langle D_{x_1 x_2}, D_{x_3 x_4} \rangle_{Fro} \\ &= Trace\left((x_1 - x_2)(x_1 - x_2)^{\top}(x_3 - x_4)(x_3 - x_4)^{\top}\right) \\ &= \left((x_1 - x_2)^{\top}(x_3 - x_4)\right)^2 \\ &= \left(x_1^{\top} x_3 - x_1^{\top} x_4 - x_2^{\top} x_3 + x_2^{\top} x_4\right)^2. \end{aligned}$$

$$(11)$$

This kernel is positive definite because it is the Frobenius inner product between the matrices D_{ab} representing the pairs. Moreover, although K_{MLPK} is formally defined for ordered pairs only, we observe that it is invariant by permutation of the elements of each pair (e.g., when x_1 and x_2 are flipped). It can therefore be considered as a positive definite kernel over the set of *unordered pairs*, seen as the quotient space of the set of ordered proteins with respect to the equivalence relation of permutation among each pair. We call this kernel for unordered pairs the *metric learning pairwise kernel* (MLPK), hence the notation K_{MLPK}.

In order to express the problem (8–9) in terms of the α variables provided by Theorem 1, we need to express the constraint $M \succeq 0$ in terms of α. Denoting pairs of indices $t = (i, j)$, Theorem 1 ensures that M can be written as $M = \sum_{t \in \mathcal{T}} \alpha_t u_t u_t^{\top}$. As we showed in the proof of Theorem 1, this implies that M is null on the space orthogonal to the linear span of $(u_t, t \in \mathcal{T})$. Therefore, $M \succeq 0$ if and only if $v^{\top} M v \geq 0$ for any v in the linear span of $(u_t, t \in \mathcal{T})$. This is equivalent to the fact that the $|\mathcal{T}| \times |\mathcal{T}|$ matrix F defined by $F_{t,t'} = u_t^{\top} M u_{t'}$ is positive semidefinite. Finally, if we denote by F_t the $|\mathcal{T}| \times |\mathcal{T}|$ matrix whose (t_1, t_2)

entry is $u_{t_1}^> D_t u_{t_2} = u_{t_1}^> u_t u_t^> u_{t_2}$, this is equivalent to $\sum_{t \in \mathcal{T}} \alpha_t F_t '' 0$.

Plugging the representation of Theorem 1 into (8–9), and replacing the Frobenius inner product by the MLPK kernel, we show that the problem is equivalent to

$$\min_{\alpha, \gamma, \zeta} \sum_{(i,j) \in \mathcal{T}} \sum_{(k,l) \in \mathcal{T}} \alpha_{ij} \alpha_{kl} K_{MLPK}((x_i, x_j), (x_k, x_l)) + C \sum_{(i,j) \in \mathcal{T}} \zeta_{ij},$$

(12)

under the constraints:

$$\zeta_{ij} \geq 0, \quad (i,j) \in \mathcal{T},$$

$$\sum_{(k,l) \in \mathcal{T}} \alpha_{kl} K_{MLPK}((x_i, x_j), (x_k, x_l)) \leq \gamma - 1 + \zeta_{ij}, \quad (i,j) \in \mathcal{I},$$

$$\sum_{(k,l) \in \mathcal{T}} \alpha_{kl} K_{MLPK}((x_i, x_j), (x_k, x_l)) \geq \gamma + 1 - \zeta_{ij}, \quad (i,j) \in \mathcal{N},$$

$$\sum_{(k,l) \in \mathcal{T}} \alpha_{kl} F_{kl} '' 0$$

(13)

Kernelization
An important property of the problem (13) is that the data only appear through the kernel K_{MLPK} and the matrices F_{ij}. Furthermore, the MLPK kernel itself (5) computed between two pairs of vectors only involves inner products between the vectors; similarly the (t_1, t_2)-th entry of the matrix F_t is a product of inner products, which can easily be computed from the inner products of the data themselves. As a result, we can apply the kernel trick to extend the problem (12–13) to any data space endowed with a positive definite kernel K_g. The resulting MLPK kernel between pairs becomes

$$K_{MLPK}((x_1, x_2), (x_3, x_4)) = (K_g(x_1, x_3) - K_g(x_1, x_4) - K_g(x_2, x_3) + K_g(x_2, x_4))^2,$$

and for any three pairs $t = (i, j)$, $t_1 = (i_1, j_1)$, $t_2 = (i_2, j_2)$ in $\mathcal{T}$ the entry (t_1, t_2) of F_t is

$[K_g(x_{i_1}, x_i) - K_g(x_{i_1}, x_j) - K_g(x_{j_1}, x_i) + K_g(x_{j_1}, x_j)] \times [K_g(x_{i_2}, x_i) - K_g(x_{i_2}, x_j) - K_g(x_{j_2}, x_i) + K_g(x_{j_2}, x_j)].$

Relaxation
The problem (12–13) is a convex problem over the cone of positive semidefinite matrices that can in theory be solved by algorithms such as interior-point methods [30]. The dimension of this problem, however, is $2|\mathcal{T}| + 1$. This is typically of the order of several thousands for small biological networks with a few hundreds or thousands vertices, which poses serious convergence issues for general-purpose optimization software.

If we relax the condition $M \geqq 0$ in the original problem, then it becomes the quadratic program of the SVM, for which dedicated optimization algorithms have been developed: current implementations of SVM easily handle several tens of thousands of dimensions [27]. The obvious drawback of this relaxation is that if the matrix M is not positive semidefinite, then it does not define a metric. Although this can be a serious problem for classical applications of distance metric learning such as clustering [11], we note that in our case the goal of metric learning is just to provide a decision function $f(x, x') = d_M(x, x')$ for predicting connected pairs, and negativity of this decision function is not a problem in itself. Therefore, we propose to relax the constraint $M \geqq 0$, or equivalently $\sum_{(kl) \in \mathcal{T}} \alpha_{kl} F_{k,l} '' 0$ in (13), and to solve the initial problem using an SVM over pairs with the MLPK kernel (5).

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
JPV proposed and implemented the method, carried out the experiments and drafted the manuscript. JQ and WSN helped prepare the data, participated in the design of the study and contributed to the redaction. All authors read and approved the final manuscript.

Acknowledgements

References
1. von Mering C, Krause R, Snel B, Cornell M, Olivier SG, Fields S, Bork P: **Comparative assessment of large-scale data sets of protein-protein interactions.** *Nature* 2002, **417:**399-403.
2. Ramani A, Marcotte E: **Exploiting the co-evolution of interacting proteins to discover interaction specificity.** *Journal of Molecular Biology* 2003, **327:**273-284.
3. Pazos F, Valencia A: **In silico two-hybrid system for the selection of physically interacting protein pairs.** *Proteins: Structure, Function and Genetics* 2002, **47(2):**219-227.
4. Marcotte EM, Pellegrini M, Ng HL, Rice DW, Yeates TO, Eisenberg D: **Detecting protein function and protein-protein interactions from genome sequences.** *Science* 1999, **285:**751-753.
5. Sprinzak E, Margalit H: **Correlated sequence-signatures as markers of protein-protein interaction.** *Journal of Molecular Biology* 2001, **311:**681-692.
6. Gomez SM, Noble WS, Rzhetsky A: **Learning to predict protein-protein interactions.** *Bioinformatics* 2003, **19:**1875-1881.
7. Martin S, Roe D, Faulon JL: **Predicting protein-protein interactions using signature products.** *Bioinformatics* 2005, **21(2):**218-226.
8. Ben-Hur A, Noble WS: **Kernel methods for predicting protein-protein interactions.** *Bioinformatics* 2005, **21(suppl 1):**i38-i46.

9. Yamanishi Y, Vert JP, Kanehisa M: **Protein network inference from multiple genomic data: a supervised approach.** *Bioinformatics* 2004, **20:**i363-i370.
10. Vert JP, Yamanishi Y: **Supervised Graph Inference.** In *Advances in Neural Information Processing Systems Volume 17*. Edited by: *Saul LK, Weiss Y, Bottou L. Cambridge, MA: MIT Press*; 2005:1433-1440.
11. Xing E, Ng A, Jordan M, Russell S: **Distance Metric Learning with Application to Clustering with Side-Information.** In *Adv Neural Inform Process Syst Volume 15*. Edited by: *S Becker ST, Obermayer K. Cambridge, MA: MIT Press*; 2003:505-512.
12. Pavlidis P, Weston J, Cai J, Grundy WN: **Gene functional classification from heterogeneous data.** *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology* 2001:242-248.
13. Lanckriet GRG, Bie TD, Cristianini N, Jordan MI, Noble WS: **A statistical framework for genomic data fusion.** *Bioinformatics* 2004, **20(16):**2626-2635.
14. Tsang IW, Kwok JT: **Distance metric learning with kernels.** *Proceedings of the International Conference on Artificial Neural Networks* 2003:126-129.
15. Weinberger KQ, Blitzer J, Saul LK: **Distance metric learning for large margin nearest neighbor classification.** In *Adv Neural Inform Process Syst Volume 18*. Edited by: *Weiss Y, Schoelkopf B, Platt J. Cambridge, MA: MIT Press*; 2006.
16. Yamanishi Y, Vert JP, Nakaya A, Kanehisa M: **Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis.** *Bioinformatics* 2003, **19(Suppl 1):**i323-i330.
17. Huh WK, Falvo JV, Gerke LC, Carroll AS, Howson RW, Weissman JS, O'Shea EK: **Global analysis of protein localization in budding yeast.** *Nature* 2003, **425:**686-691.
18. Kondor RI, Lafferty J: **Diffusion kernels on graphs and other discrete input spaces.** In *Proceedings of the International Conference on Machine Learning* Edited by: *Sammut C, Hoffmann A. Morgan Kaufmann*; 2002.
19. Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M: **A Bayesian networks approach for predicting protein-protein interactions from genomic data.** *Science* 2003, **302:**449-453.
20. Qi Y, Bar-Joseph Z, Klein-Seetharaman J: **Evaluation of different biological data and computational classification methods for use in protein interaction prediction.** *Proteins: Structure, Function, and Bioinformatics* 2006, **63:**490-500.
21. Mewes HW, Frishman D, Gruber C, Geier B, Haase D, Kaps A, Lemcke K, Mannhaupt G, Pfeiffer F, Schüller C, Stocker S, Weil B: **MIPS: a database for genomes and protein sequences.** *Nucleic Acids Research* 2000, **28:**37-40.
22. Bader GD, Donaldson I, Wolting C, Ouellette BF, Pawson T, Hogue CW: **BIND-The Biomolecular Interaction Network Database.** *Nucleic Acids Res* 2001, **29:**242-245.
23. Harbison C, Gordon D, Lee T, Rinaldi N, Macisaac K, Danford T, Hannett N, Tagne JB, Reynolds D, Yoo J, Jennings E, Zeitlinger J, Pokholok D, Kellis M, Rolfe P, Takusagawa K, Lander E, Gifford D, Fraenkel E, Young R: **Transcriptional Regulatory Code of a Eukaryotic Genome.** *Nature* 2004, **431:**99-104.
24. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: A new generation of protein database search programs.** *Nucleic Acids Research* 1997, **25:**3389-3402.
25. Friedman N, Linial M, Nachman I, Pe'er D: **Using Bayesian networks to analyze expression data.** *J Comput Biol* 2000, **7(3–4):**601-620.
26. Vapnik VN: *Statistical Learning Theory New-York: Wiley*; 1998.
27. Schölkopf B, Smola A: *Learning with Kernels Cambridge, MA: MIT Press*; 2002.
28. Aronszajn N: **Theory of reproducing kernels.** *Trans Am Math Soc* 1950, **68:**337-404.
29. Kimeldorf GS, Wahba G: **Some results on Tchebycheffian spline functions.** *J Math Anal Appl* 1971, **33:**82-95.
30. Boyd S, Vandenberghe L: *Convex Optimization New York, NY, USA: Cambridge University Press*; 2004.