IPSJ Transactions on Bioinformatics Vol.6 18–28 (June 2013)

[DOI: 10.2197/ipsjtbio.6.18]

Original Paper

Improved Protein-ligand Prediction Using Kernel Weighted Canonical Correlation Analysis

Raissa Relator[1,a)] Tsuyoshi Kato[1,b)] Richard Lemence[2,c)]

Received: January 24, 2013, Accepted: April 5, 2013, Released: June 24, 2013

Abstract: Protein-ligand interaction prediction plays an important role in drug design and discovery. However, wet lab procedures are inherently time consuming and expensive due to the vast number of candidate compounds and target genes. Hence, computational approaches became imperative and have become popular due to their promising results and practicality. Such methods require high accuracy and precision outputs for them to be useful, thus, the problem of devising such an algorithm remains very challenging. In this paper we propose an algorithm employing both support vector machines (SVM) and an extension of canonical correlation analysis (CCA). Following assumptions of recent chemogenomic approaches, we explore the effects of incorporating bias on similarity of compounds. We introduce kernel weighted CCA as a means of uncovering any underlying relationship between similarity of ligands and known ligands of target proteins. Experimental results indicate statistically significant improvement in the area under the ROC curve (AUC) and F-measure values obtained as opposed to those gathered when only SVM, or SVM with kernel CCA is employed, which translates to better quality of prediction.

Keywords: canonical correlation analysis, kernel methods, protein-ligand interaction, support vector machines

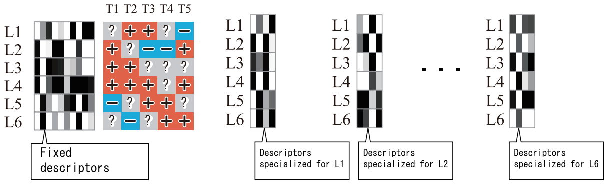
1. Introduction

Drug discovery is a multi-staged process which involves the determination of existing interactions between a compound and a protein. Many drugs are developed depending on the reaction they produce when coupled with the respective proteins acting during a biological process in the body. However, only a few existing interactions have actually been validated through experiments. Moreover, wet lab procedures are inherently time consuming and expensive due to the vast number of candidate compounds and target genes. Hence, computational approaches became imperative and have become popular due to their promising results and practicality.

The protein-ligand interaction prediction problem can be viewed as a task of filling up a protein-ligand matrix whose rows represent the candidate compounds and the columns represent the target proteins as shown in the example in **Fig. 1** (a). A matrix entry is +1 if there is interaction between the corresponding drug and target. Otherwise, −1. Only a few interactions have actually been verified and recorded which makes the protein-ligand matrix sparse. Termed as the ‘chemogenomic approach’ by Rognan [20], the ultimate goal of this task is to identify all the ligands of each target, thus, fully matching the ligand and target spaces [3].

Many in silico methods have already been developed to address this problem. We can classify these methods into two: the structure or docking approach and the ligand-based approach. Docking approaches make use of 3D structures of the chemical compounds or the proteins to find protein-ligand pairs which are more likely to bind [2], [4], [5]. On the other hand, ligand-based techniques usually employ machine learning algorithms in comparing known ligands and candidate ligands of a certain target even without any prior information regarding their structure [9], [14], [25]. In this study, we shall make use of the ligand-based approach.

There are two ways of approaching the task of interaction prediction: one is by using the global model [2], [17], and another one is via the local model [3], [14], [25]. The global model utilizes a large interaction matrix and imputation of missing values is done simultaneously. Each cell in the interaction matrix is considered as a sample to which statistical methods are applied. Descriptors of ligands in the form of a feature matrix and some information for target proteins are combined to generate a fused profile for each cell in the interaction matrix. An advantage is that interaction prediction for target proteins with few known interactions can still be formed. However, since the model aims



(a) Problem (b) Specialized descriptors

Fig. 1 Protein-ligand matrix and descriptors. In the example depicted in (a), the prediction task is to impute 11 missing entries in the 6×5 protein-ligand matrix using 10-dimensional raw descriptors of ligands. The problem can be divided into six sub-problems, each of which is to complete a row in the protein-ligand matrix. Our algorithm extracts compact descriptors specialized in each sub-problem.

[1] Department of Computer Science, Graduate School of Engineering, Gunma University, Kiryu, Gunma 376–8515, Japan

[2] Institute of Mathematics, College of Science, University of the Philippines, Diliman, Quezon City 1101, Philippines

[a)] relator-raissa@kato-lab.cs.gunma-u.ac.jp

[b)] katotsu@cs.gunma-u.ac.jp

[c)] rslemence@math.upd.edu.ph

to exploit information from similar columns, some useful information for learning the rule for prediction may be corrupted by information from irrelevant columns.

Meanwhile, in the local model approach, prediction is made for each column of the protein-ligand table independently — the approach finds unknown chemical compounds which are similar to known ligands interacting with the target protein of interest. The local model often suffers from a small-sample problem. Many columns in the protein-ligand interaction matrix include few positive interactions, causing machine learning algorithms to be trained with few positive samples despite very high dimensionality of ligand descriptors.

The goal of determining interactions between targets and compounds is established under twofold assumptions [3], [20]: First is that compounds with similar properties tend to share targets. And, targets with similar ligands share similarities in structures such as binding sites. These have been verified by recent studies by considering drug side effects [7] and similarities among ligands [16]. Moreover, integrated approaches exploring both protein and compound similarities have also been investigated [6], [14], [26]. Thus, recent methodologies have allowed us to make predictions on interactions based on similarity measures for ligands and targets.

Motivated by the assumption that similar ligands tend to have similar target proteins [15], [23], our goal is to uncover any underlying relationship between a set of ligands and exploit this relationship, together with some known ligand-target interactions, to predict new interactions. We search for ligands with strong associations by finding correlations between them using their features.

In this paper, we present a weighted extension of canonical correlation analysis (WCCA) in the reproducing kernel Hilbert space (RKHS) in an attempt to introduce advantageous properties of local models to the global model approach. To estimate the missing entries in each row of the interaction matrix, we use kernel WCCA (KWCCA) to extract essential features which are specialized in imputation of the corresponding row. The extracted features are compact enough for local models to be trained with a small training set composed from the column. Through the experiments with data of GPCRs and odorant receptors, the prediction performance is shown to be improved when our algorithm is applied compared to several existing methods.

1.1 Related Works

A popular and useful technique in investigating relationships between sets of data is the so-called canonical correlation analysis (CCA) [12]. First introduced by Hotelling [13], CCA generally aims to find linear transformations which maximize the correlation between a pair of data. However, the common information extracted from the data sources may not be as useful if nonlinear correlations exist. For this reason, kernel CCA (KCCA) was introduced to offer an alternative solution via the kernel trick, where CCA is performed in a reproducing kernel Hilbert space (RKHS), typically a higher dimensional feature space [1].

Several variants of CCA have been developed and applied to different problem settings. For instance, Yu et al. [31] introduced weights to CCA. Although we also introduce weighting in our proposed method, the authors' purpose and formulation are totally different from ours: they assumed more than two data sources and weight each source, whereas, in our formulation, we assume two data sources and each sample is weighted. On the other hand, in a biologically-related setting, Yamanishi et al. [29] employed multiple KCCA and integrated KCCA for gene cluster extraction. One is done by maximizing the sum of pairwise correlations and the other by maximizing correlation of combination of attributes.

For the problem of functional site prediction, Gonzalez et al. [11] incorporated KCCA to find amino acid pairs and protein functional classifications which are maximally correlated. This technique was motivated by the *Xdet* method [18] and CCA was employed as an alternative to computing Pearson correlation.

The indefinite kernel CCA (IKCCA) was developed by Samarov et al. [22] with a motivation similar to ours. They removed the similarity of samples outside the neighborhood to refine the analysis. The operation often yields an indefinite matrix. IKCCA finds a definite matrix close to the indefinite matrix to perform CCA on the definite matrix. However, their usage of employing CCA is different from ours: the inputs of their approach are positive pairs of ligands and proteins, whereas our approach applies CCA to two different types of ligand profiles. IKCCA is formulated with a saddle-point problem that is solved by minimizing a maximum, but the numerical algorithm to solve the problem has not been shown.

Another important variant is sparse CCA [27], [28] which uses *lasso* or *elastic net* techniques to encourage loading matrices to be sparse. This approach was also applied to a set of protein-ligand pairs with positive interactions in order to elucidate meaningful chemical descriptors in Ref. [28]. Another is the Supervised Regularized CCA [10] which allows integration of multimodal data. Such method can be very useful when involving non-image and image data samples.

2. Materials and Methods

2.1 Data

The data used for this study was originally from Ref. [21]. The given interaction matrix consists of 62 mammalian odorant receptors (ORs) as target proteins and 63 odorants as candidate ligands. It is binary in form and contains 340 positive interactions. The number of known positive interactions for each target protein is at least one and at most thirty-seven, while the median is three. Some randomly selected protein-ligand pairs are assumed to be unknown to test prediction methods, and the values of the cells are set to zero. Each row in the interaction matrix provides an *interaction profile* of the ligand.

From the chemical IDs supplied, we searched PubChem [*1] for the chemical structures of the odorants to obtain the descriptors of the ligands. Frequent substructures are employed as descriptors of ligands. The frequent substructures are mined with a software named *gSpan* [30]. The software is applied to the 63 chemical structures, and the 60,311 binary descriptors are obtained as

[*1] http://pubchem.ncbi.nlm.nih.gov/

chemical profiles.

2.2 Overview of the Algorithm

Our approach consists of two stages: First, we consider sub-problems, each of which involves imputation on a single row in the interaction matrix, and use weighted CCA to extract a compact vector representation for each sub-problem. Then, we apply SVM for prediction of each cell using the corresponding descriptor extracted in the previous stage. This technique is overviewed as follows.

Chemical profiles obtained from chemical structures contain numerous features that are not important for prediction. Extracting significant features from such chemical profiles is crucial for accurate prediction of protein-ligand interaction. To accomplish this, we have to find effective low-dimensional representations of the original chemical profiles lying in the extremely high-dimensional *chemical space*.

Interaction profiles describe the existence and the absence of interactions with several target proteins. More often than not, target proteins share similar properties. For this reason, interaction profiles approximately span a low-dimensional space, say $\mathbb{R}^m$, which we shall also extract from a high-dimensional *interaction space*, in a similar fashion as the chemical profiles.

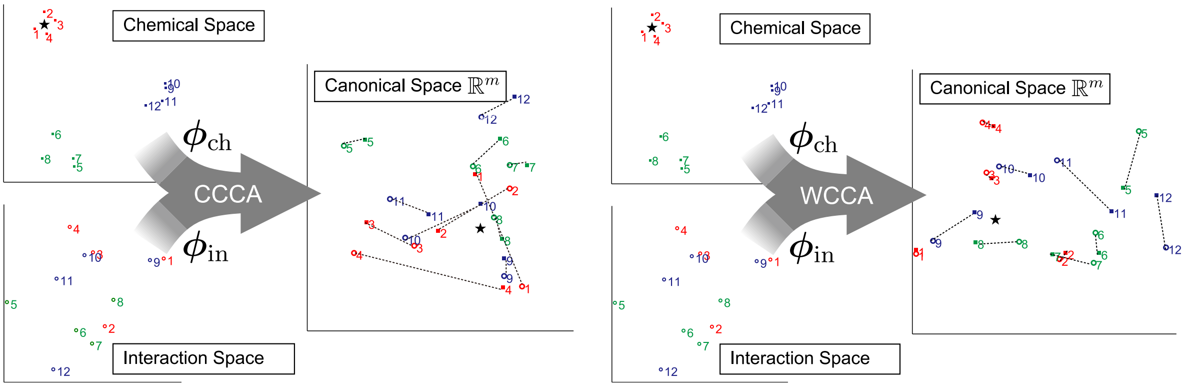
Canonical correlation analysis uses a set of chemical profiles and interaction profiles to find two projection functions, $\phi_{\rm ch}$ and $\phi_{\rm in}$, simultaneously: The projection $\phi_{\rm ch}$ is from the chemical space to the low-dimensional canonical space $\mathbb{R}^m$, and $\phi_{\rm in}$ is from the interaction space to $\mathbb{R}^m$. The images of $\phi_{\rm ch}$ are used to approximate the images of $\phi_{\rm in}$. The projections obtained by CCA are shown mathematically to be the minimizer of the expected deviation of the image of $\phi_{\rm ch}$ from the image of $\phi_{\rm in}$.

Figure 2 (a) is an illustration of how CCA works with chemical profiles and interaction profiles. In this figure, the shaded squares are data representations of the feature vector of each ligand in the chemical space. While the open circles are the data representation of the interaction vector of each ligand in the interaction space. The images under $\phi_{\rm ch}$ and $\phi_{\rm in}$ of these data points are plotted in the canonical space, and their corresponding images are linked with a dashed line. CCA finds the projections $\phi_{\rm ch}$ and $\phi_{\rm in}$ so that the average squared length of the dashed lines is minimized.

In application to protein-ligand interaction prediction, estimating the images for all ligands is not necessary; it is only for the ligand whose interactions we wish to predict that the image of the chemical compound is desired to be well approximated. To obtain a good approximation for a ligand of interest, it is sufficient to estimate projections so that only the images of similar ligands are approximated well. The precisions of the approximations for ligands dissimilar to the ligand of interest barely affect the accuracy of the solution. This consideration motivated us to assign weights to ligands according to their similarity to the ligand of interest, and to extend the classical CCA so that the weighted average deviation is minimized. The weighted CCA almost disregards ligands with small weights to find projections, achieving more accurate approximations for the ligand of interest. We refer to the extension of CCA as weighted CCA.

Figure 2 (b) illustrates the effects of weighted CCA when weights are added to similar ligands. In this context, we define similarity as the measure of affinity between features of compounds. This can be represented by the distance between the data representation of the ligands in the chemical space. In the given figure, the chemical profile for a ligand of interest is marked with a star, and profiles of similar ligands are colored red. In a similar manner, we interpret points of the same color as ligands sharing similarities in their chemical properties, hence their grouping in the chemical space. The two figures, (a) and (b), allow us to compare classical CCA with weighted CCA: the deviations for red points in (b) are smaller than those in (a). The deviations for other ligands are larger, which hardly worsen the performance of predicting the interaction of the protein of interest.

The final prediction result is obtained in the post-processing stage using SVM. The images of the projections are used for SVM learning. SVM is trained well if a good training set is given. Hence, ligands with poor approximations by CCA, which are noisy for SVM learning, are preferably excluded. The im-



(a) Classical CCA

(b) Weighted CCA

Fig. 2 Classical CCA and weighted CCA. Our approach projects chemical and interaction profiles into a low-dimensional canonical space so that the images are close to each other. The star point represents the ligand of interest, and red points are ligands sharing similarities with the ligand of interest. Although the classical CCA minimizes the average deviation over all the ligands, to achieve accurate prediction, it is sufficient that the deviations between the images of the target ligand and the ligands similar to it are small. The weighted CCA works with arbitrarily specified weights, which ensures small deviations for red points by giving them larger weights.

ages are already in a low-dimensional space in which SVM learning works well even with a small training set, encouraging us to assign smaller weights to ligands with poor approximations for SVM learning.

2.3 Weighted CCA

In this subsection we present the details of weighted CCA. We denote the chemical profile and the interaction profile, respectively, by a p_{ch}-dimensional vector $\boldsymbol{x}^{\text{ch}}$ and a p_{in}-dimensional vector $\boldsymbol{x}^{\text{in}}$. Assuming that the functions $\phi_{\text{ch}} : \mathbb{R}^{p_{\text{ch}}} \to \mathbb{R}^m$ and $\phi_{\text{in}} : \mathbb{R}^{p_{\text{in}}} \to \mathbb{R}^m$ are affine transformations allows us to express them as $\phi_{\text{ch}}(\boldsymbol{x}^{\text{ch}}) = \boldsymbol{W}_{\text{ch}}^{\top}(\boldsymbol{x}^{\text{ch}} - \boldsymbol{\mu}_{\text{ch}})$, $\phi_{\text{in}}(\boldsymbol{x}^{\text{in}}) = \boldsymbol{W}_{\text{in}}^{\top}(\boldsymbol{x}^{\text{in}} - \boldsymbol{\mu}_{\text{in}})$, where $\boldsymbol{W}_{\text{ch}} \in \mathbb{R}^{p_{\text{ch}} \times m}$, $\boldsymbol{\mu}_{\text{ch}} \in \mathbb{R}^{p_{\text{ch}}}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{p_{\text{in}} \times m}$, and $\boldsymbol{\mu}_{\text{in}} \in \mathbb{R}^{p_{\text{in}}}$ are their respective parameters. We wish to find the pair of projection functions minimizing the expected deviation between the images given by $J(\phi_{\text{ch}}, \phi_{\text{in}}) \equiv \mathbb{E}[\|\phi_{\text{ch}}(\boldsymbol{x}^{\text{ch}}) - \phi_{\text{in}}(\boldsymbol{x}^{\text{in}})\|^2]$, where $\mathbb{E}$ is the expectation operator.

The expected deviation can be reduced arbitrarily by setting the projections so that the images are scaled down. A trivial solution is $\boldsymbol{W}_{\text{ch}} = \boldsymbol{0}$ and $\boldsymbol{W}_{\text{in}} = \boldsymbol{0}$ at which the expected deviation vanishes for any dataset. To avoid trivial solutions, the size of the images is adjusted by fixing the second moment matrices, $\mathbb{E}[\phi_{\text{ch}}(\boldsymbol{x}^{\text{ch}})\phi_{\text{ch}}(\boldsymbol{x}^{\text{ch}})^{\top}]$ and $\mathbb{E}[\phi_{\text{in}}(\boldsymbol{x}^{\text{in}})\phi_{\text{in}}(\boldsymbol{x}^{\text{in}})^{\top}]$, to identity matrices.

The expectation appearing in the derivation and the second moment matrices operates according to an empirical probabilistic distribution. Supposing n ligands are given, the chemical profiles are denoted by $\boldsymbol{x}_1^{\text{ch}}, \ldots, \boldsymbol{x}_n^{\text{ch}}$, and the interaction profiles by $\boldsymbol{x}_1^{\text{in}}, \ldots, \boldsymbol{x}_n^{\text{in}}$. If we define an empirical distribution as $q(\boldsymbol{x}^{\text{ch}}, \boldsymbol{x}^{\text{in}}) = \sum_{j=1}^{n} v_j \delta(\boldsymbol{x}^{\text{ch}} - \boldsymbol{x}_j^{\text{ch}})\delta(\boldsymbol{x}^{\text{in}} - \boldsymbol{x}_j^{\text{in}})$, with weights $v_1, \ldots, v_n$ whose sum is one and $\delta(\cdot)$ is the Dirac delta function, then the expected deviation is reduced to the weighted average of deviation and can be expressed as

$$J(\phi_{\text{ch}}, \phi_{\text{in}}) = \sum_{j=1}^{n} v_j \left\|\phi_{\text{ch}}(\boldsymbol{x}_j^{\text{ch}}) - \phi_{\text{in}}(\boldsymbol{x}_j^{\text{in}})\right\|^2. \quad (1)$$

This implies that approximations are refined locally by setting the weights so that ligands dissimilar from the target ligand are given smaller weights.

The optimal projections can be computed via the generalized eigen-decomposition, as given in Algorithm 1 in the Appendix. When setting $v_j = 1/n$, the algorithm is shown to be equivalent to the classical CCA. Hence, we can say that weighted CCA is an extension of the classical CCA.

Kernelization of weighted CCA is formulated with a similarity function of chemical profiles $K_{\text{ch}}(\boldsymbol{x}_i^{\text{ch}}, \boldsymbol{x}_j^{\text{ch}})$ and a similarity function of interaction profiles $K_{\text{in}}(\boldsymbol{x}_i^{\text{in}}, \boldsymbol{x}_j^{\text{in}})$ without using the vectors themselves explicitly. These similarity functions are said to be valid kernels guaranteeing the theory of the algorithms, which map the profiles non-linearly into other (typically high-dimensional) spaces $\mathcal{H}_{\text{ch}}$ and $\mathcal{H}_{\text{in}}$, respectively, called an RKHS. Kernelized weighted CCA finds affine-transforms from RKHS to a canonical space $\mathbb{R}^m$, so that the expected deviation between images in $\mathbb{R}^m$ is minimized. If we denote the composite mapping functions by ψ_{ch} and ψ_{in}, respectively, the optimal solution is given by $\psi_{\text{ch}}(\boldsymbol{x}^{\text{ch}}) = \boldsymbol{A}_{\text{ch}}^{\top}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{k}}_{\text{ch}}(\boldsymbol{x}^{\text{ch}})$, $\psi_{\text{in}}(\boldsymbol{x}^{\text{in}}) = \boldsymbol{A}_{\text{in}}^{\top}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{k}}_{\text{in}}(\boldsymbol{x}^{\text{in}})$. The algorithm for computing the two matrices, $\boldsymbol{A}_{\text{ch}} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{A}_{\text{in}} \in \mathbb{R}^{m \times n}$, is presented in Algorithm 2 in the Appendix. The functions $\bar{\boldsymbol{k}}_{\text{ch}}(\cdot)$ and $\bar{\boldsymbol{k}}_{\text{in}}(\cdot)$ are called the empirical kernel mapping, and their definition is as given in Eq. (A.7) in the Appendix.

2.4 Weighted SVM

Prediction of the interaction between ligand i and target t is performed with the SVM score given by $f(\boldsymbol{x}_i^{\text{ch}}; \boldsymbol{w}_{(i,t)}, b_{(i,t)}) = \boldsymbol{w}_{(i,t)}^{\top}\psi_{\text{ch}}(\boldsymbol{x}_i^{\text{ch}}) + b_{(i,t)}$, where $\boldsymbol{x}_i^{\text{ch}}$ is the chemical profile of ligand i. The SVM parameters, $\boldsymbol{w}_{(i,t)}$ and $b_{(i,t)}$, are obtained beforehand by the SVM learning algorithm. This is performed only with ligands whose interaction with the target t is known. This study employs the similarity of ligands as weights in the learning process, as in the algorithm presented in the Appendix.

2.5 Weighting Schemes

Ligands are given weights in both stages of the weighted CCA and the weighted SVM. These weights are dependent on the ligand to be predicted. Larger weights are given for ligands that are more similar to the ligand of interest. In predicting the interaction of the ith ligand, the weight of jth ligand is given by the normalization of $v'_j = \frac{1}{\left\|\bar{\boldsymbol{k}}_{\text{ch}}(\boldsymbol{x}_j^{\text{ch}}) - \bar{\boldsymbol{k}}_{\text{ch}}(\boldsymbol{x}_i^{\text{ch}})\right\| + \left\|\bar{\boldsymbol{k}}_{\text{in}}(\boldsymbol{x}_j^{\text{in}}) - \bar{\boldsymbol{k}}_{\text{in}}(\boldsymbol{x}_i^{\text{in}})\right\| + \epsilon}$, where ϵ is a positive constant and set to 10 in our analysis. Normalization is done by setting $v_j = \frac{v'_j}{\sum_{k=1}^{n} v'_k}$ so that the sum of the weights is one.

3. Results

3.1 Experimental Setting

To illustrate the effectiveness of the kernel weighted CCA (KWCCA), we carried out experiments on an interaction dataset of GPCRs and odorant receptors described in the previous section. For evaluation of prediction performance, we applied a 10-fold Monte-Carlo cross validation, where data is randomly divided into 2 disjoint sets of training and test data for 10 repetitions. Data was partitioned such that for each target protein, 50% of the positive and negative interactions are used for training, and the other half for testing. KCCA, KWCCA, and the weighted SVM were implemented in Matlab, and LIBSVM [8] was used for the classical SVM.

We also performed prediction using SVM in the global model setting for comparison. The kernel function for the global model here is defined as the product of the inner product among chemical profiles and the inner product among columns of the interaction matrix.

Parameters of the local models are determined by finding respective values where the test data perform best using SVM and KCCA. Namely, the regularization parameter C and the kernel function for SVM are chosen so that SVM achieves the highest prediction performance, while the regularization parameters for CCA γ_{ch} and γ_{in}, and the number of dimensions of the canonical space m, are determined via the performance of KCCA. As a result, the values of the parameters are set as $C = 1000$, $\gamma_{\text{ch}} = \gamma_{\text{in}} = 1$, and $m = 4$. The RBF kernel is applied and the kernel width is determined as the mean of the distance within sets. These mentioned parameters are then fed into the algorithm em-

Table 1 Abbreviation of methods.

Abbreviation	Description
WW	KWCCA + Weighted SVM
WU	KWCCA + Classical SVM
KW	Classical KCCA + Weighted SVM
KU	Classical KCCA + Classical SVM
S	SVM of local model
SGL	Linear SVM of global model
SGR	RBF SVM of global model

ploying KWCCA. The parameters are not tuned specifically for KWCCA. Thus, it is believed that there is a chance of improvement in the perfomance of this algorithm if careful and suitable parameter selection is done.

For the global model, the kernel which achieves the best performance is the linear kernel. The regularization parameter is chosen as $C = 10$, achieving the best performance among other values. Results for the case of the RBF kernel with the best C value obtained are also reported for comparison.

The methods based on KWCCA involve two stages upon implementation. First, we exploit KWCCA to extract a set of features for each compound. Second, we use them for training a machine learning algorithm employing SVMs before testing them to make predictions. In total, seven methods are implemented in the experiments: two using SVM in the global model setting, and the other five following the local model. One of the two global model methods uses RBF kernel for SVM, and the other uses the linear kernel. On the other hand, the methods used for the local models are as follows: SVM, KCCA with classical SVM, KCCA with weighted SVM, KWCCA with classical SVM, and KWCCA with weighted SVM. For simplicity of notation, we shall refer to each of the seven methods using the abbreviations in **Table 1**.

3.2 Performance Evaluation Criteria

The following criteria were used to compare the seven prediction methods:

(1) Area under the ROC curve (AUC) – The receiver operating characteristic (ROC) curve is a plot of the true positive rate (TPR) versus the false positive rate (FPR) where $\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, $\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$, and TP, FN, FP, and TN are the number of true positives, false negatives, false positives, and true negatives, respectively. For performance comparison using the ROC, the AUC value is further computed.

(2) F-measure – A value which is given by the harmonic mean of precision and recall: $F = \frac{2\mathrm{Prec}\times\mathrm{Recall}}{\mathrm{Prec+Recall}}$ where the precision Prec and the recall Recall are defined by $\mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, $\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, respectively. Since the problem is presented as a binary classification problem, only the maximum value of the F-measure values for each target is considered. The scores obtained via SVM are used as confidence levels, thus, changing the threshold yields different predictions.

These values are calculated for each target protein and averaged over the ten data divisions. However, there are instances when the test set does not contain a true positive interaction, hence AUC and F-measures cannot be computed. Therefore, these values were disregarded and, out of 62 target proteins, AUC and F-measures were computed for 49 of them. The Wilcoxon signed test was used for the statistical significance of the difference among the values of the evaluation measures.

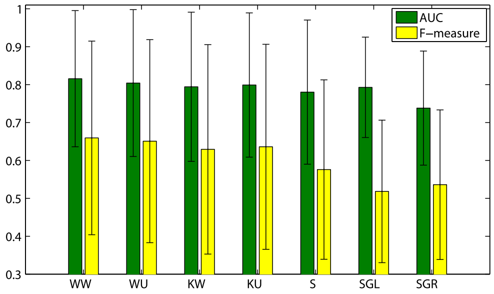


Fig. 3 Average performance of the methods. Data was randomly split into training and test sets, and 10 training-testing data divisions were used for each method. Following the local model, AUC and F-measure were computed for each of the 62 targets. The bar plots represent the average AUC (green) and average F-measure (yellow) over the 10 cross validation sets and the 49 targets containing true positives. The two KWCCA-based methods, WW and WU, and the other methods were implemented for comparison. The difference of the performances of WW and WU from the other five methods showed to be statistically significant in terms of the P-values (by Wilcoxon signed rank test).

3.3 Effects of the Use of CCA

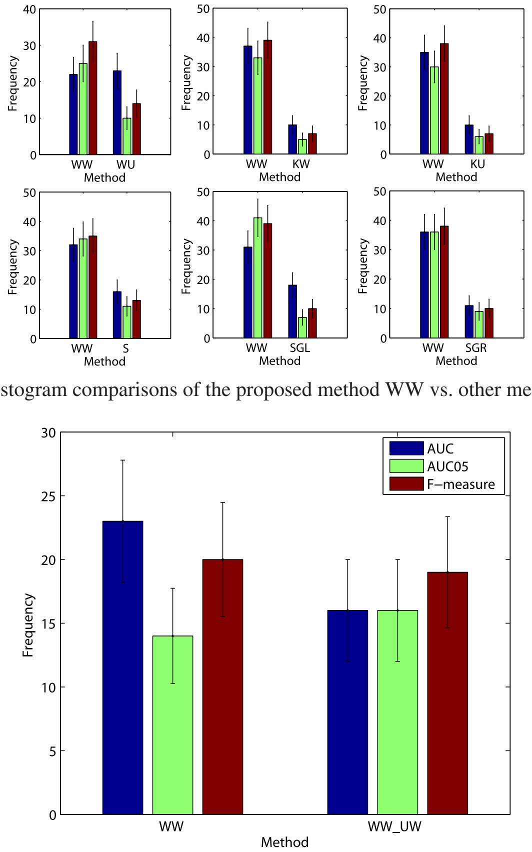
The average AUCs and F-measures are reported in **Fig. 3**. Error bars are also included to present standard deviations. In comparison with the local models, four CCA-based methods, WW, WU, KW, and KU, achieve remarkably better AUCs and F-measures compared to those of S: the differences between the AUCs and F-measures of KW, the worst among the four CCA-based methods, and S are 0.014 and 0.053, respectively, (P-values: 5.81×10^{-7} and 9.49×10^{-9} respectively). The AUC of the global model SGL is comparable to some of the local models, whereas the F-measure is not worse than that of S. A closer inspection on the results of SGL indicate that it has the lowest average number of true positives over all cross-validations among all models, around 161, which may be the reason behind a very small F-measure value.

3.4 Improvement by Weighting

The effects of the weighted extension of CCA are manifested via comparison among four CCA-based methods. WW achieves significantly higher AUC and F-measures in average compared to KW and KU, where the P-values for the difference in the AUCs are 4.85×10^{-11} and 6.91×10^{-10}, and the P-values for the F-measures are 3.59×10^{-7} and 3.96×10^{-6}, respectively.

3.5 Histogram Comparison

The frequencies of WW besting the AUC or F-measure values of the other methods in predicting interactions for a certain target protein are shown in the histograms in **Fig. 4** (a). These values represent the number of target proteins such that the evaluated AUC and F-measure values for the method WW is better than the AUC and F-measure values of the other method in comparison. Instances when there are ties between the methods were unaccounted. For the evaluated AUC and F-measure values, WW outputs are more desirable than most of the others which indicates higher quality of prediction performance.



(a) Histogram comparisons of the proposed method WW vs. other methods.

(b) Histogram comparisons of Weighted SVM using weighting scheme (2) vs. Classical SVM.

Fig. 4 Histogram comparisons between the proposed method WW and other methods. Frequencies when the AUC (blue), AUC between 0 and 0.05 (green), and F-measure (red) values of WW outperform the other methods, and vice-versa, are illustrated. It can be observed that AUC and F-measure values histograms for WW are more desirable than the rest.

3.6 Weighted and Classical SVM

WU yields interesting results in the histogram (Fig. 4 (a)): The frequency of WW yielding better AUCs are comparable to that of WU's, although frequency of better F-measures are relatively higher for WW than WU. To further investigate the comparison between WW and WU, we compute the area under the curve of the region of FPR between 0 and 0.05. This area, which we shall refer to as AUC05, allows us to evaluate the true positive rate with higher confidence. The histogram on AUC05 shows WW bests WU more frequently than WU does, which implies the use of weights in the SVM stage can find more true positives confidently than the classical SVM.

The motivation to endow the weights with training data in SVM learning is that the projections in the canonical space from chemical profiles with larger weights are expected to be better approximations of the projections from interaction profiles. It is possible to directly evaluate how good the approximations are by computing the distances among the projections. This motivation leads to another weighting scheme using the normalization of

$$v'_j = \frac{1}{\left\|\phi_{\text{ch}}(\boldsymbol{x}_j^{\text{ch}}) - \phi_{\text{in}}(\boldsymbol{x}_j^{\text{in}})\right\| + \epsilon} \qquad (2)$$

instead of Eq. (1) in the SVM learning stage. We investigate the performance when the weighting scheme is changed to Eq. (2) in the SVM learning stage. We refer to this approach as WW_{UW} hereinafter. The average AUC and F-measure of WW_{UW} are 0.802 and 0.649, respectively, which are slightly worse than those of WW. The number of target proteins, for which the prediction performance of WW_{UW} is better than that of WW is not larger than the number of WW besting WW_{UW}, as depicted in Fig. 4 (b). These facts imply that the changing weighting scheme in SVM learning does not achieve significant improvements.

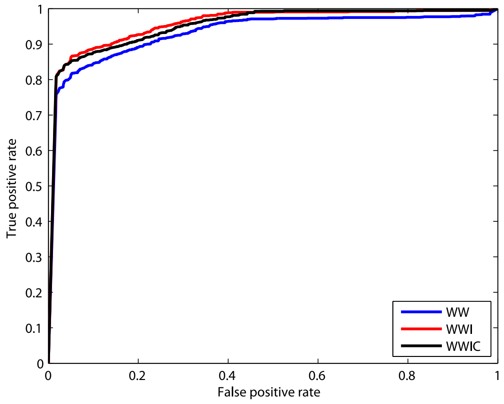


Fig. 5 ROC curves. WWI uses the projections from interaction profiles in the SVM stage, and WWIC uses the projections from both chemical and interaction profiles for SVM.

3.7 Using Interaction Profiles

When a sufficient number of known positive and negative interactions are given for a certain ligand, the image of the interaction profile in the canonical image can provide good descriptors for predicting the remaining interactions. We further implemented two methods, herein referred to as WWI and WWIC, to investigate the performance of the interaction profile. WWI replaces the image of a chemical profile with the image of the interaction profile in the SVM stage, while WWIC concatenates the two images to feed them to the weighted SVM. The two methods achieved significant improvement. WWI achieved an average AUC of 0.857 and average F-measure of 0.699, while WWIC obtained a 0.835 average AUC and a 0.692 average F-measure. The P-values of the differences on AUC from WW are 5.27×10^{-9} and 0.021, respectively, and P-values on F-measures are 1.05×10^{-5} and 9.17×10^{-7}, respectively. **Figure 5** compares the average ROC curves of WW, WWI, and WWIC. The curves of WWI and WWIC are higher than that of WW, which supports the claim that introducing the interaction profiles improves the prediction performance.

4. Conclusions

A kernel version of weighted canonical correlation analysis is proposed, which is implemented using a derived form of the generalized eigenvalue problem. Similar to the linear CCA and its kernelized version, this can be applied to machine learning problems for dimension reduction and feature extraction. The paper presents an application to improving the prediction quality obtained in the protein-ligand interaction problem setting. By adding bias to more similar samples, better prediction can be made which is evident on the higher AUC and F-measure values obtained. Weighting scheme on SVM based on CCA outputs were also explored and are judged to be better than classical SVM.

Even in the field of computational biology, CCA for more than two data sources has been widely used [19], [24], [29] and their usual objectives involve maximizing the sum of correlations for every pair of data sources. For future work, it could be worth exploring the extension of weighted CCA for analysis of multiple data sets in a biological setting. It could also be interesting to investigate the effectiveness of applying the proposed method to other biological problems aside from protein-ligand interaction prediction.

Acknowledgments RR acknowledges the support of the Ministry of Education, Culture, Sports, Science and Technology of Japan. TK was supported by Grant-in-Aid for Scientific Research (C) 23500373.

References

[1] Akaho, S: Kernel method for canonical correlation analysis, *Proc. International Meeting of Psychometric Society*, Osaka, Japan (2001).

[2] Antes, I., Siu, S.W. and Lengauer, T.: DynaPred: A structure and sequence based method for the prediction of MHC class I binding peptide sequences and conformations, *Bioinformatics*, Vol.22, No.14, pp.e16–24 (July 2006).

[3] Bajorath, J.: Computational analysis of ligand relationships within target families, *Curr. Opin. Chem. Biol.*, Vol.12, No.3, pp.352–358 (June 2008).

[4] Ballester, P.J. and Mitchell, J.B.: A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking, *Bioinformatics*, Vol.26, No.9, pp.1169–1175 (May 2010).

[5] Biniashvili, T., Schreiber, E. and Kliger, Y.: Improving classical substructure-based virtual screening to handle extrapolation challenges, *J. Chem. Inf. Model*, Vol.52, No.3, pp.678–685 (Mar. 2012).

[6] Bleakley, K. and Yamanishi, Y.: Supervised prediction of drug-target interactions using bipartite local models, *Bioinformatics*, Vol.25, No.18, pp.2397–2403 (Sep. 2009).

[7] Campillos, M., Kuhn, M., Gavin, A.-C., Jensen, L.J. and Bork, P.: Drug target identification using side-effect similarity, *Science*, Vol.321, pp.263–266 (2008).

[8] Chang, C.-C. and Lin, C.-J.: LIBSVM: A library for support vector machines, *ACM Trans. Intelligent Systems and Technology*, Vol.2, No.27, pp.1–27 (2011).

[9] Chen, B., Harrison, R.F., Papadatos, G., Willett, P., Wood, D.J., Lewell, X.Q., Greenidge, P. and Stiefl, N.: Evaluation of machine-learning methods for ligand-based virtual screening, *J. Comput. Aided Mol. Des.*, Vol.21, No.1-3, pp.53–62 (Jan.-Mar. 2007).

[10] Golugula, A., Lee, G., Master, S.R.. Feldman, M.D., Tomaszewski, J.E. and Madabhushi, A.: Supervised regularized canonical correlation analysis: Integrating histologic and proteomic data for predicting biochemical failures, *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, pp.6434–6437 (2011).

[11] Gonzalez, A.J., Liao, L. and Wu, C.H.: Predicting ligand binding residues and functional sites using multi-positional correlations with graph theoretic clustering and kernel CCA, *IEEE/ACM Trans. Comput. Biol. Bioinform.* (Oct. 2011).

[12] Hardoon, D.R., Szedmak, S. and Shawe-Taylor, J.: Canonical correlation analysis: An overview with application to learning methods, *Neural Comput.*, Vol.16, No.12, pp.2639–2664 (Dec. 2004).

[13] Hotelling, H.: Relation between two sets of variates, *Biometrika*, Vol.28, No.3 (1936).

[14] Jacob, L. and Vert, J.P.: Protein-ligand interaction prediction: An improved chemogenomics approach, *Bioinformatics*, Vol.24, No.19, pp.2149–2156 (Oct. 2008).

[15] Klabunde, T.: Chemogenomic approaches to drug discovery: Similar receptors bind similar ligands, *Br. J. Pharmacol.*, Vol.152, pp.5–7 (2007).

[16] Martin, Y.C., Kofron, J.L. and Traphagen, L.M.: Do structurally similar molecules have similar biological activity, *J. Med. Chem.*, Vol.45, pp.4350–4358 (2002).

[17] Paolini, G.V., Shapland, R.H.B., van Hoorn, W.P., Mason, J.S. and Hopkins, A.L.: Global mapping of the pharmacological space, *Nat. Biotechnol.*, Vol.24, pp.805–815 (2006).

[18] Pazos, F., Rausell, A. and Valencia, A.: Phylogeny-independent detection of functional residues, *Bioinformatics*, Vol.22, No.12, pp.1440–1448 (June 2006).

[19] Peng, Y., Zhang, D. and Zhang, J.: A new canonical correlation analysis algorithm with local discrimination, *Neural Process Lett.*, Vol.31, pp.1–15 (2010).

[20] Rognan, D.: Chemogenomic approaches to rational drug design, *Br. J. Pharmacol.*, Vol.152, No.1, pp.38–52 (Sep. 2007).

[21] Saito, H., Kubota, M., Roberts, R.W., Chi, Q. and Matsunami, H.: RTP family members induce functional expression of mammalian odorant receptors, *Cell*, Vol.119, No.5, pp.679–691 (Nov. 2004).

[22] Samarov, D., Marron, J.S., Liu, Y., Grulke, C. and Tropsha, A.: Local kernel canonical correlation analysis with application to virtual drug screening, *Ann. Appl. Stat.*, Vol.5, No.3, pp.2169–2196 (Sep. 2011).

[23] Schuffenhauer, A., Floersheim, P., Acklin, P. and Jacoby, E.: Similarity metrics for ligands reflecting the similarity of the target proteins, *J. Chem. Inf. Comput. Sci.*, Vol.43, pp.391–405 (2003).

[24] Tang, C.S. and Ferreira, M.A.: A gene-based test of association using canonical correlation analysis, *Bioinformatics*, Vol.28, No.6. pp.845–850 (Mar. 2012).

[25] van Laarhoven, T., Nabuurs, S.B. and Marchiori, E.: Gaussian interaction profile kernels for predicting drug-target interaction, *Bioinformatics*, Vol.27, No.21, pp.3036–3043 (Nov. 2011).

[26] Wassermann, A.M., Geppert, H. and Bajorath, J.: Ligand prediction for orphan targets using support vector machines and various target-ligand kernels is dominated by nearest neighbor effects, *J. Chem. Inf. Model.*, Vol.49, pp.2155–2167 (2009).

[27] Witten, D.M., Tibshirani, R. and Hastie, T.: A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, *Biostatistics*, Vol.10, No.3, pp.515–534 (July 2009).

[28] Yamanishi, Y., Pauwels, E., Saigo, H. and Stoven, V.: Extracting sets of chemical substructures and protein domains governing drug-target interactions, *J. Chem. Inf. Model.* (May 2011).

[29] Yamanishi, Y., Vert, J.P., Nakaya, A. and Kanehisa, M.: Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis, *Bioinformatics*, Vol.19, Suppl.1, pp.i323–330 (2003).

[30] Yan, X. and Han, J.: gSpan: Graph-based substructure pattern mining, *Proc. 2002 Int'l Conf. Data Mining* (*ICDM '02*), pp.721–724 (2002).

[31] Yu, S., De Moor, B. and Moreau, Y.: Learning with heterogenous data sets by weighted multiple kernel canonical correlation analysis, *Proc. IEEE Machine Learning for Signal Processing* (*IEEE MLSP 2008*), pp.81–86, Thessaloniki, Greece (Aug. 2007).

Appendix

Here we present the details of the supporting theories applied in this study. Throughout the discussion, we shall use the following notations. We will denote vectors by boldface lower-case letters and matrices by boldface upper-case letters. The inverse of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^{-1}$ and its transpose is $\boldsymbol{A}^{\top}$. The $n \times n$ identity matrix is given by $\boldsymbol{I}_n$, while $\boldsymbol{D}_{\boldsymbol{v}}$ is defined as diag($\boldsymbol{v}$), the matrix whose diagonal entries are from vector $\boldsymbol{v}$. We let $\mathbb{S}^n$ be the set of all $n \times n$ symmetric matrices, $\mathbb{S}^n_+$ as the set of all $n \times n$ symmetric positive semi-definite matrices, $\mathbb{S}^n_{++}$, the set of all $n \times n$ symmetric positive definite matrices, and $\mathbb{O}^{m\times n}$ as the set of all $m \times n$ orthonormal matrices. We define $\mathbb{N}_n = \{i \in \mathbb{N} | i \leq n\}$, and $\langle \cdot, \cdot \rangle$ is used to denote the inner product between vectors.

A.1 Generalized Eigendecomposition

The generalized eigendecomposition is defined as follows:

Theorem 1. (Generalized Eigendecomposition) Let $n \in \mathbb{N}$. For any $\boldsymbol{A} \in \mathbb{S}^n$ and any $\boldsymbol{B} \in \mathbb{S}^n_{++}$,

$$\exists \boldsymbol{U} \in \mathbb{R}^{n\times n}, \exists \boldsymbol{\lambda} \in \mathbb{R}^n, \text{ s.t. } \quad \begin{aligned} \boldsymbol{AU} &= \boldsymbol{BUD}_{\boldsymbol{\lambda}}, \\ \boldsymbol{U}^{\top}\boldsymbol{BU} &= \boldsymbol{I}_n. \end{aligned}$$

The entries of $\boldsymbol{\lambda}$ are called *generalized eigenvalues*, and the columns of $\boldsymbol{U}$ are called *generalized eigenvectors*.

This section deals with the case when the two matrices $\boldsymbol{A} \in \mathbb{S}^{n+m}$ and $\boldsymbol{B} \in \mathbb{S}^{n+m}_{++}$ are of the form

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{m\times m} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0}_{n\times n} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_x & \boldsymbol{0}_{m\times n} \\ \boldsymbol{0}_{n\times m} & \boldsymbol{B}_y \end{bmatrix}$$

where $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ with $r \equiv \mathrm{rank}(\boldsymbol{C})$. Let us denote the generalized eigendecomposition of $(\boldsymbol{A}, \boldsymbol{B})$ by

$$\boldsymbol{A}\boldsymbol{U}_{\mathrm{all}} = \boldsymbol{B}\boldsymbol{U}_{\mathrm{all}}\boldsymbol{\Lambda}_{\mathrm{all}},$$

where $\boldsymbol{U}_{\mathrm{all}}^\top\boldsymbol{B}\boldsymbol{U}_{\mathrm{all}} = \boldsymbol{I}_{m+n}$, and $\boldsymbol{\Lambda}_{\mathrm{all}} = \mathrm{diag}(\boldsymbol{\lambda}_{\mathrm{all}})$ and $\boldsymbol{\lambda}_{\mathrm{all}} = [\lambda_1, \dots, \lambda_{m+n}]^\top$, such that $\lambda_1 \geq \cdots \geq \lambda_{m+n}$. We denote the columns in $\boldsymbol{U}_{\mathrm{all}}$ by

$$\boldsymbol{U}_{\mathrm{all}} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{m+n}] = \begin{bmatrix} \boldsymbol{u}_{x,1}, \cdots, \boldsymbol{u}_{x,m+n} \\ \boldsymbol{u}_{y,1}, \cdots, \boldsymbol{u}_{y,m+n} \end{bmatrix}$$

where $(\forall i \in \mathbb{N}_{m+n})\, \boldsymbol{u}_{x,i} \in \mathbb{R}^m$, $\boldsymbol{u}_{y,i} \in \mathbb{R}^n$, and define

$$\boldsymbol{U}_x \equiv [\boldsymbol{u}_{x,1}, \dots \boldsymbol{u}_{x,r}], \quad \boldsymbol{U}_y \equiv [\boldsymbol{u}_{y,1}, \dots, \boldsymbol{u}_{y,r}].$$

The following theorem is the main result of this section.

Theorem 2. Consider the following optimization problem

$$\begin{aligned} \max \quad & \mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{C}\boldsymbol{Y}) \\ \mathrm{wrt} \quad & \boldsymbol{X} \in \mathbb{R}^{m\times k}, \boldsymbol{Y} \in \mathbb{R}^{n\times k}, \\ \mathrm{subj\ to} \quad & \boldsymbol{X}^\top\boldsymbol{B}_x\boldsymbol{X} = \boldsymbol{Y}^\top\boldsymbol{B}_y\boldsymbol{Y} = \boldsymbol{I}_k. \end{aligned}$$

An optimal solution is given by

$$\boldsymbol{X} = \sqrt{2}[\boldsymbol{u}_{x,1}, \dots, \boldsymbol{u}_{x,k}], \quad \boldsymbol{Y} = \sqrt{2}[\boldsymbol{u}_{y,1}, \dots, \boldsymbol{u}_{y,k}].$$

To prove Theorem 2, we will use the following lemma.

Lemma 1.

$$\boldsymbol{U}_x^\top\boldsymbol{B}_x\boldsymbol{U}_x = \boldsymbol{U}_y^\top\boldsymbol{B}_y\boldsymbol{U}_y = \frac{1}{2}\boldsymbol{I}_r.$$

Proof. (of Lemma 1) Let

$$\boldsymbol{\Lambda} \equiv \mathrm{diag}\{\lambda_1, \dots, \lambda_r\}.$$

We assume $\lambda_r > 0$ here. From this assumption,

$$\boldsymbol{C}\boldsymbol{U}_y = \boldsymbol{B}_x\boldsymbol{U}_x\boldsymbol{\Lambda}, \quad \boldsymbol{C}^\top\boldsymbol{U}_x = \boldsymbol{B}_y\boldsymbol{U}_y\boldsymbol{\Lambda}.$$

Pre-multiplying the former equation by $\boldsymbol{U}_x^\top$ and post-multiplying the transpose of the latter equation by $\boldsymbol{U}_y$ yield

$$\boldsymbol{U}_x^\top\boldsymbol{B}_x\boldsymbol{U}_x\boldsymbol{\Lambda} = \boldsymbol{U}_x^\top\boldsymbol{C}\boldsymbol{U}_y = \boldsymbol{\Lambda}\boldsymbol{U}_y^\top\boldsymbol{B}_y\boldsymbol{U}_y.$$

For the diagonal entries of the above equality,

$$\forall i \in \mathbb{N}_r \, : \, \lambda_i\boldsymbol{u}_{x,i}^\top\boldsymbol{B}_x\boldsymbol{u}_{x,i} = \lambda_i\boldsymbol{u}_{y,i}^\top\boldsymbol{B}_y\boldsymbol{u}_{y,i},$$

resulting in

$$\boldsymbol{u}_{x,i}^\top\boldsymbol{B}_x\boldsymbol{u}_{x,i} = \boldsymbol{u}_{y,i}^\top\boldsymbol{B}_y\boldsymbol{u}_{y,i} = \frac{1}{2} \tag{A.1}$$

since $\lambda_i > 0$ and from the assumption that $\boldsymbol{U}_{\mathrm{all}}^\top\boldsymbol{B}\boldsymbol{U}_{\mathrm{all}} = \boldsymbol{I}_{m+n}$. For the off-diagonal entries,

$$\boldsymbol{u}_{x,i}^\top\boldsymbol{B}_x\boldsymbol{u}_{x,j} - \frac{\lambda_i}{\lambda_j}\boldsymbol{u}_{y,i}^\top\boldsymbol{B}_y\boldsymbol{u}_{y,j} = 0. \tag{A.2}$$

From the assumption $\boldsymbol{U}_{\mathrm{all}}^\top\boldsymbol{B}\boldsymbol{U}_{\mathrm{all}} = \boldsymbol{I}_{m+n}$,

$$\begin{bmatrix} \boldsymbol{u}_{x,i} \\ \boldsymbol{u}_{y,i} \end{bmatrix}^\top \boldsymbol{B} \begin{bmatrix} \boldsymbol{u}_{x,j} \\ \boldsymbol{u}_{y,j} \end{bmatrix} = \boldsymbol{u}_{x,i}^\top\boldsymbol{B}_x\boldsymbol{u}_{x,j} + \boldsymbol{u}_{y,i}^\top\boldsymbol{B}_y\boldsymbol{u}_{y,j} = 0. \tag{A.3}$$

From the two equations (A.2) and (A.3), we have

$$\boldsymbol{u}_{x,i}^\top\boldsymbol{B}_x\boldsymbol{u}_{x,j} = \boldsymbol{u}_{y,i}^\top\boldsymbol{B}_y\boldsymbol{u}_{y,j} = 0. \tag{A.4}$$

Thus, Eqs. (A.1) and (A.4) establish Lemma 1. □

Proof. (of Theorem 2) Let

$$\boldsymbol{Z} \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}.$$

There exists $\boldsymbol{R} \in \mathbb{R}^{(m+n)\times k}$ such that $\boldsymbol{Z} = \boldsymbol{U}_{\mathrm{all}}\boldsymbol{R}$, and matrix $\boldsymbol{R}$ is orthonormal, i.e., $\boldsymbol{R} \in \mathbb{O}^{(m+n)\times k}$, since

$$\begin{aligned} \boldsymbol{R}^\top\boldsymbol{R} &= \boldsymbol{R}^\top\boldsymbol{I}_{(m+n)}\boldsymbol{R} = \boldsymbol{R}^\top\boldsymbol{U}_{\mathrm{all}}^\top\boldsymbol{B}\boldsymbol{U}_{\mathrm{all}}\boldsymbol{R} = \boldsymbol{Z}^\top\boldsymbol{B}\boldsymbol{Z} \\ &= \frac{1}{2}\boldsymbol{X}^\top\boldsymbol{B}_x\boldsymbol{X} + \frac{1}{2}\boldsymbol{Y}^\top\boldsymbol{B}_y\boldsymbol{Y} = \boldsymbol{I}_k. \end{aligned}$$

Let $\boldsymbol{r}_i \in \mathbb{R}^k$ denote the ith row vector, i.e.,

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}_1^\top \\ \vdots \\ \boldsymbol{r}_{m+n}^\top \end{bmatrix}.$$

Then we have

$$\begin{aligned} \mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{C}\boldsymbol{Y}) &= \frac{1}{2}\mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{C}\boldsymbol{Y}) + \frac{1}{2}\mathrm{tr}(\boldsymbol{Y}^\top\boldsymbol{C}^\top\boldsymbol{X}) \\ &= \frac{1}{2}\mathrm{tr}\left(\left[\boldsymbol{X}^\top, \boldsymbol{Y}^\top\right]\begin{bmatrix} \boldsymbol{0}_{m\times m} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0}_{n\times n} \end{bmatrix}\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}\right) \\ &= \mathrm{tr}(\boldsymbol{Z}^\top\boldsymbol{A}\boldsymbol{Z}) = \mathrm{tr}(\boldsymbol{R}^\top\boldsymbol{U}_{\mathrm{all}}^\top\boldsymbol{A}\boldsymbol{U}_{\mathrm{all}}\boldsymbol{R}) \\ &= \mathrm{tr}(\boldsymbol{R}^\top\boldsymbol{U}_{\mathrm{all}}^\top\boldsymbol{B}\boldsymbol{U}_{\mathrm{all}}\boldsymbol{D}_{\boldsymbol{\lambda}_{\mathrm{all}}}\boldsymbol{R}) = \mathrm{tr}(\boldsymbol{R}^\top\boldsymbol{D}_{\boldsymbol{\lambda}_{\mathrm{all}}}\boldsymbol{R}) \\ &= \sum_{i=1}^{m+n}\lambda_i\|\boldsymbol{r}_i\|^2 = \langle\boldsymbol{w}, \boldsymbol{\lambda}_{\mathrm{all}}\rangle, \end{aligned} \tag{A.5}$$

where $\boldsymbol{w} \in \mathbb{R}_+^{m+n}$ is a nonnegative vector in which the ith entry is defined by $w_i = \|\boldsymbol{r}_i\|^2$. To conclude the proof, we shall make use of the following two properties:

First, note that there exists $\boldsymbol{S} \in \mathbb{O}^{(m+n)\times(m+n-k)}$ such that $\boldsymbol{R}^\top\boldsymbol{S} = \boldsymbol{0}_{k\times(m+n-k)}$. Then $[\boldsymbol{R}, \boldsymbol{S}] \in \mathbb{O}^{(m+n)\times(m+n)}$. Let $\boldsymbol{s}_i \in \mathbb{R}^{(m+n-k)}$ denote the ith row vector, i.e.,

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_1^\top \\ \vdots \\ \boldsymbol{s}_{m+n}^\top \end{bmatrix}.$$

Then, $\forall i \in \mathbb{N}_n$ we have

$$1 = \left\|\begin{bmatrix} \boldsymbol{r}_i \\ \boldsymbol{s}_i \end{bmatrix}\right\|^2 = \|\boldsymbol{r}_i\|^2 + \|\boldsymbol{s}_i\|^2 \geq \|\boldsymbol{r}_i\|^2 = w_i,$$

where the first equality follows from the property of the square orthonormal matrix: $[\boldsymbol{R}, \boldsymbol{S}][\boldsymbol{R}, \boldsymbol{S}]^\top = \boldsymbol{I}_n$. Second, observe that

$$\|\boldsymbol{w}\|_1 = \sum_{i=1}^{n} w_i = \sum_{i=1}^{n}\|\boldsymbol{r}_k\|^2 = \mathrm{tr}(\boldsymbol{R}\boldsymbol{R}^\top) = \mathrm{tr}(\boldsymbol{R}^\top\boldsymbol{R}) = \mathrm{tr}\boldsymbol{I}_k = k.$$

Now, the objective function $\mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{C}\boldsymbol{Y})$ is maximized when

$$w_i = \begin{cases} 1, & \text{if } 1 \le i \le k, \\ 0, & \text{if } k < i \le n, \end{cases}$$

and so the value of Eq. (A.5) is bounded above by

$$\langle \boldsymbol{w}, \boldsymbol{\lambda} \rangle \le \sum_{i=1}^{k} \lambda_i.$$

Finally, we show that equality holds when

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{I}_k \\ \boldsymbol{0}_{(m+n-k)\times k} \end{bmatrix}.$$

Since $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{R} = \boldsymbol{U}_{\mathrm{ma}}$, where $\boldsymbol{U}_{\mathrm{ma}} \equiv [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$, we have

$$\mathrm{tr}(\boldsymbol{Z}^\top \boldsymbol{A}\boldsymbol{Z}) = \mathrm{tr}(\boldsymbol{U}_{\mathrm{ma}}^\top \boldsymbol{A}\boldsymbol{U}_{\mathrm{ma}}) = \sum_{i=1}^{k} \lambda_i.$$

Thus, $\boldsymbol{Z} = \boldsymbol{U}_{\mathrm{ma}}$ is an optimum, which implies

$$\boldsymbol{X} = \sqrt{2}[\boldsymbol{u}_{x,1}, \ldots, \boldsymbol{u}_{x,k}], \quad \boldsymbol{Y} = \sqrt{2}[\boldsymbol{u}_{y,1}, \ldots, \boldsymbol{u}_{y,k}].$$

□

A.2 Linear Weighted CCA

Algorithm 1 (Linear Weighted CCA) Define two matrices,

$$\boldsymbol{X}_{\mathrm{ch}} \equiv [\boldsymbol{x}_1^{\mathrm{ch}}, \ldots, \boldsymbol{x}_n^{\mathrm{ch}}], \qquad \boldsymbol{X}_{\mathrm{in}} \equiv [\boldsymbol{x}_1^{\mathrm{in}}, \ldots, \boldsymbol{x}_n^{\mathrm{in}}],$$

and let $\boldsymbol{v} = [v_1, \ldots, v_n]^\top$. Then the optimal offsets, $\boldsymbol{\mu}_{\mathrm{ch}}$, $\boldsymbol{\mu}_{\mathrm{in}}$, are computed as

$$\boldsymbol{\mu}_{\mathrm{ch}} = \boldsymbol{X}_{\mathrm{ch}}\boldsymbol{v}, \qquad \boldsymbol{\mu}_{\mathrm{in}} = \boldsymbol{X}_{\mathrm{in}}\boldsymbol{v}.$$

We use the optimal offsets to define $\boldsymbol{C}_{\mathrm{ch,ch}}, \boldsymbol{C}_{\mathrm{ch,in}}, \boldsymbol{C}_{\mathrm{in,ch}}, \boldsymbol{C}_{\mathrm{in,in}}$ as

$$\boldsymbol{C}_{\mathrm{ch,ch}} \equiv \boldsymbol{X}_{\mathrm{ch}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{ch}}^\top - \boldsymbol{\mu}_{\mathrm{ch}}\boldsymbol{\mu}_{\mathrm{ch}}^\top, \quad \boldsymbol{C}_{\mathrm{ch,in}} \equiv \boldsymbol{X}_{\mathrm{ch}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{in}}^\top - \boldsymbol{\mu}_{\mathrm{ch}}\boldsymbol{\mu}_{\mathrm{in}}^\top,$$
$$\boldsymbol{C}_{\mathrm{in,ch}} \equiv \boldsymbol{X}_{\mathrm{in}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{ch}}^\top - \boldsymbol{\mu}_{\mathrm{in}}\boldsymbol{\mu}_{\mathrm{ch}}^\top, \quad \boldsymbol{C}_{\mathrm{in,in}} \equiv \boldsymbol{X}_{\mathrm{in}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{in}}^\top - \boldsymbol{\mu}_{\mathrm{in}}\boldsymbol{\mu}_{\mathrm{in}}^\top,$$

and consider the following generalized eigen-decomposition problem:

$$\begin{bmatrix} \boldsymbol{0}_{p_{\mathrm{ch}}\times p_{\mathrm{ch}}} & \boldsymbol{C}_{\mathrm{ch,in}} \\ \boldsymbol{C}_{\mathrm{in,ch}} & \boldsymbol{0}_{p_{\mathrm{in}}\times p_{\mathrm{in}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^{\mathrm{ch}} \\ \boldsymbol{w}^{\mathrm{in}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{\mathrm{in,in}} & \boldsymbol{0}_{p_{\mathrm{in}}\times p_{\mathrm{ch}}} \\ \boldsymbol{0}_{p_{\mathrm{ch}}\times p_{\mathrm{in}}} & \boldsymbol{C}_{\mathrm{ch,ch}} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^{\mathrm{ch}} \\ \boldsymbol{w}^{\mathrm{in}} \end{bmatrix}.$$

Denote the hth major eigen-vector by $\begin{bmatrix} \boldsymbol{w}_h^{\mathrm{ch}} \\ \boldsymbol{w}_h^{\mathrm{in}} \end{bmatrix}$. The optimal loading matrices $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$ are computed by setting the hth columns of $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$ to $\boldsymbol{w}_h^{\mathrm{ch}}$ and $\boldsymbol{w}_h^{\mathrm{in}}$, respectively.

Theorem 3. Algorithm 1 yields the parameters of the mapping functions $(\phi_{\mathrm{ch}}, \phi_{\mathrm{in}})$ which minimize the expected deviation subject to the scaling constraints that the second moment matrices are the identity matrix.

To prove this theorem, we employ the following result.

Let two affine mapping functions $\phi_{\mathrm{ch}} : \mathbb{R}^{p_{\mathrm{ch}}} \to \mathbb{R}^m$ and $\phi_{\mathrm{in}} : \mathbb{R}^{p_{\mathrm{in}}} \to \mathbb{R}^m$ be parametrized by

$$\phi_{\mathrm{ch}}(\boldsymbol{x}_{\mathrm{ch}}) = \boldsymbol{W}_{\mathrm{ch}}^\top(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}}), \quad \text{and} \quad \phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}}) = \boldsymbol{W}_{\mathrm{in}}^\top(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}}),$$

respectively. Define the expected deviation by

$$J = \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{in}})}[\|\phi_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}}) - \phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\|^2],$$

where the expectation is according to a probabilistic distribution $p(\boldsymbol{x}^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{in}})$. If we consider the optimization problem for minimizing the expected deviation with respect to the parameters $(\boldsymbol{W}_{\mathrm{ch}}, \boldsymbol{\mu}_{\mathrm{ch}}, \boldsymbol{W}_{\mathrm{in}}, \boldsymbol{\mu}_{\mathrm{in}})$ subject to

$$\mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}})}[\phi_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})\phi_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})^\top] = \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{in}})}[\phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})^\top] = \boldsymbol{I}_m,$$

a minimizer of the optimization problem is found by the following optimization problem.

Set the offset vectors as $\boldsymbol{\mu}_{\mathrm{ch}} = \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}})}[\boldsymbol{x}^{\mathrm{ch}}]$ and $\boldsymbol{\mu}_{\mathrm{in}} = \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{in}})}[\boldsymbol{x}^{\mathrm{in}}]$. The optimal loading matrices $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$ are computed by setting the hth columns of $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$ to $\boldsymbol{w}_h^{\mathrm{ch}}$ and $\boldsymbol{w}_h^{\mathrm{in}}$, respectively, and denoting by $\begin{bmatrix} \boldsymbol{w}_h^{\mathrm{ch}} \\ \boldsymbol{w}_h^{\mathrm{in}} \end{bmatrix}$ the hth major eigenvector of the generalized eigendecomposition:

$$\begin{bmatrix} \boldsymbol{0}_{p_{\mathrm{ch}}\times p_{\mathrm{ch}}} & \check{\boldsymbol{C}}_{\mathrm{ch,in}} \\ \check{\boldsymbol{C}}_{\mathrm{in,ch}} & \boldsymbol{0}_{p_{\mathrm{in}}\times p_{\mathrm{in}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^{\mathrm{ch}} \\ \boldsymbol{w}^{\mathrm{in}} \end{bmatrix} = \begin{bmatrix} \check{\boldsymbol{C}}_{\mathrm{in,in}} & \boldsymbol{0}_{p_{\mathrm{in}}\times p_{\mathrm{ch}}} \\ \boldsymbol{0}_{p_{\mathrm{ch}}\times p_{\mathrm{in}}} & \check{\boldsymbol{C}}_{\mathrm{ch,ch}} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^{\mathrm{ch}} \\ \boldsymbol{w}^{\mathrm{in}} \end{bmatrix}, \tag{A.6}$$

where the covariance matrices are given by

$$\check{\boldsymbol{C}}_{\mathrm{ch,ch}} \equiv \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}})}[(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})^\top],$$
$$\check{\boldsymbol{C}}_{\mathrm{ch,in}} \equiv \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{in}})}[(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})^\top],$$
$$\check{\boldsymbol{C}}_{\mathrm{in,ch}} \equiv \check{\boldsymbol{C}}_{\mathrm{ch,in}}^\top,$$
$$\check{\boldsymbol{C}}_{\mathrm{in,in}} \equiv \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{in}})}[(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})^\top].$$

To verify this, we let

$$\boldsymbol{x} \equiv \begin{bmatrix} \boldsymbol{x}^{\mathrm{ch}} \\ \boldsymbol{x}^{\mathrm{in}} \end{bmatrix}, \qquad \boldsymbol{\mu}_{\mathrm{tot}} \equiv \begin{bmatrix} \boldsymbol{\mu}_{\mathrm{ch}} \\ \boldsymbol{\mu}_{\mathrm{in}} \end{bmatrix}.$$

Then the second order moment matrices are rewritten as

$$\mathbb{E}[\phi_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})\phi_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})^\top] = \boldsymbol{W}_{\mathrm{ch}}^\top\mathbb{E}[(\boldsymbol{\mu}_{\mathrm{ch}} - \boldsymbol{x}^{\mathrm{ch}})(\boldsymbol{\mu}_{\mathrm{ch}} - \boldsymbol{x}^{\mathrm{ch}})^\top]\boldsymbol{W}_{\mathrm{ch}},$$
$$\mathbb{E}[\phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})^\top] = \boldsymbol{W}_{\mathrm{in}}^\top\mathbb{E}[(\boldsymbol{\mu}_{\mathrm{in}} - \boldsymbol{x}^{\mathrm{in}})(\boldsymbol{\mu}_{\mathrm{in}} - \boldsymbol{x}^{\mathrm{in}})^\top]\boldsymbol{W}_{\mathrm{in}},$$

respectively, and the expected deviation is arranged as

$$\begin{aligned} J &= \mathbb{E}[\|\phi_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}}) - \phi_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\|^2] \\ &= \mathbb{E}[\|\boldsymbol{W}_{\mathrm{ch}}^\top(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}}) - \boldsymbol{W}_{\mathrm{in}}^\top(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})\|^2] \\ &= \mathrm{tr}(\boldsymbol{W}_{\mathrm{ch}}^\top\mathbb{E}[(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})^\top]\boldsymbol{W}_{\mathrm{ch}}) \\ &\quad + \mathrm{tr}(\boldsymbol{W}_{\mathrm{in}}^\top\mathbb{E}[(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})^\top]\boldsymbol{W}_{\mathrm{in}}) \\ &\quad - 2\mathrm{tr}(\boldsymbol{W}_{\mathrm{ch}}^\top\mathbb{E}[(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})^\top]\boldsymbol{W}_{\mathrm{in}}) \\ &= 2m - 2\mathrm{tr}(\boldsymbol{W}_{\mathrm{ch}}^\top\mathbb{E}[(\boldsymbol{x}^{\mathrm{ch}} - \boldsymbol{\mu}_{\mathrm{ch}})(\boldsymbol{x}^{\mathrm{in}} - \boldsymbol{\mu}_{\mathrm{in}})^\top]\boldsymbol{W}_{\mathrm{in}}). \end{aligned}$$

From here, we will first derive the optimal value of $\boldsymbol{\mu}_{\mathrm{tot}}$, and then give the algorithm to find the optimal $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$. Introducing the Lagrangian multipliers $\boldsymbol{\Lambda}_{\mathrm{ch}} \in \mathbb{S}^{p_{\mathrm{ch}}}$ and $\boldsymbol{\Lambda}_{\mathrm{in}} \in \mathbb{S}^{p_{\mathrm{in}}}$, the Lagrangian function is written as

$$\begin{aligned} \mathcal{L}_{\mathrm{A}} &= 2m - 2\mathrm{tr}(\boldsymbol{W}_{\mathrm{ch}}^\top\mathbb{E}[(\boldsymbol{\mu}_{\mathrm{ch}} - \boldsymbol{x}^{\mathrm{ch}})(\boldsymbol{\mu}_{\mathrm{in}} - \boldsymbol{x}^{\mathrm{in}})^\top]\boldsymbol{W}_{\mathrm{in}}) \\ &\quad - \langle \boldsymbol{\Lambda}_{\mathrm{ch}}, \boldsymbol{I}_m - \boldsymbol{W}_{\mathrm{ch}}^\top\mathbb{E}[(\boldsymbol{\mu}_{\mathrm{ch}} - \boldsymbol{x}^{\mathrm{ch}})(\boldsymbol{\mu}_{\mathrm{ch}} - \boldsymbol{x}^{\mathrm{ch}})^\top]\boldsymbol{W}_{\mathrm{ch}} \rangle \\ &\quad - \langle \boldsymbol{\Lambda}_{\mathrm{in}}, \boldsymbol{I}_m - \boldsymbol{W}_{\mathrm{in}}^\top\mathbb{E}[(\boldsymbol{\mu}_{\mathrm{in}} - \boldsymbol{x}^{\mathrm{in}})(\boldsymbol{\mu}_{\mathrm{in}} - \boldsymbol{x}^{\mathrm{in}})^\top]\boldsymbol{W}_{\mathrm{in}} \rangle. \end{aligned}$$

To obtain the values of $\boldsymbol{\mu}_{\mathrm{ch}}$ and $\boldsymbol{\mu}_{\mathrm{in}}$ at the saddle point, we set the derivative of the Lagrangian to zero:

$$
\begin{aligned}
\frac{\partial \mathcal{L}_{\mathrm{A}}}{\partial \boldsymbol{\mu}_{\mathrm{ch}}} &= 2\boldsymbol{W}_{\mathrm{ch}}\boldsymbol{W}_{\mathrm{in}}^{\top}(\boldsymbol{\mu}_{\mathrm{in}} - \mathbb{E}[\boldsymbol{x}^{\mathrm{in}}]) \\
&\quad + 2\boldsymbol{W}_{\mathrm{ch}}\boldsymbol{\Lambda}_{\mathrm{ch}}\boldsymbol{W}_{\mathrm{ch}}^{\top}(\boldsymbol{\mu}_{\mathrm{ch}} - \mathbb{E}[\boldsymbol{x}^{\mathrm{ch}}]), \\
\frac{\partial \mathcal{L}_{\mathrm{A}}}{\partial \boldsymbol{\mu}_{\mathrm{in}}} &= 2\boldsymbol{W}_{\mathrm{in}}\boldsymbol{W}_{\mathrm{ch}}^{\top}(\boldsymbol{\mu}_{\mathrm{ch}} - \mathbb{E}[\boldsymbol{x}^{\mathrm{ch}}]) \\
&\quad + 2\boldsymbol{W}_{\mathrm{in}}\boldsymbol{\Lambda}_{\mathrm{in}}\boldsymbol{W}_{\mathrm{in}}^{\top}(\boldsymbol{\mu}_{\mathrm{in}} - \mathbb{E}[\boldsymbol{x}^{\mathrm{in}}]).
\end{aligned}
$$

The two derivatives vanish simultaneously when

$$\boldsymbol{\mu}_{\mathrm{ch}} = \mathbb{E}[\boldsymbol{x}^{\mathrm{ch}}], \qquad \text{and} \qquad \boldsymbol{\mu}_{\mathrm{in}} = \mathbb{E}[\boldsymbol{x}^{\mathrm{in}}].$$

Next, we derive the optimal $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$. Substituting the definitions of the covariance matrices into the expected deviation and the second moments of the affine transformations, we have

$$J = 2m - 2\mathrm{tr}(\boldsymbol{W}_{\mathrm{ch}}^{\top}\check{\boldsymbol{C}}_{\mathrm{ch,in}}\boldsymbol{W}_{\mathrm{in}})$$

and

$$
\begin{aligned}
\mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}})}[\boldsymbol{\phi}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})\boldsymbol{\phi}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})^{\top}] &= \boldsymbol{W}_{\mathrm{ch}}^{\top}\check{\boldsymbol{C}}_{\mathrm{ch,ch}}\boldsymbol{W}_{\mathrm{ch}}, \\
\mathbb{E}_{p(\boldsymbol{x}^{\mathrm{in}})}[\boldsymbol{\phi}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\boldsymbol{\phi}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})^{\top}] &= \boldsymbol{W}_{\mathrm{in}}^{\top}\check{\boldsymbol{C}}_{\mathrm{in,in}}\boldsymbol{W}_{\mathrm{in}}.
\end{aligned}
$$

Then, by omitting the constants, the problem for finding the optimal $\boldsymbol{W}_{\mathrm{ch}}$ and $\boldsymbol{W}_{\mathrm{in}}$ is reduced to the following optimization problem:

$$
\begin{aligned}
\text{max} \quad & \mathrm{tr}(\boldsymbol{W}_{\mathrm{ch}}^{\top}\check{\boldsymbol{C}}_{\mathrm{ch,in}}\boldsymbol{W}_{\mathrm{in}}) \\
\text{wrt} \quad & \boldsymbol{W}_{\mathrm{ch}} \in \mathbb{R}^{p_{\mathrm{ch}} \times m}, \boldsymbol{W}_{\mathrm{in}} \in \mathbb{R}^{p_{\mathrm{in}} \times m} \\
\text{subj to} \quad & \boldsymbol{W}_{\mathrm{ch}}^{\top}\check{\boldsymbol{C}}_{\mathrm{ch,ch}}\boldsymbol{W}_{\mathrm{ch}} = \boldsymbol{W}_{\mathrm{in}}^{\top}\check{\boldsymbol{C}}_{\mathrm{in,in}}\boldsymbol{W}_{\mathrm{in}} = \boldsymbol{I}_m.
\end{aligned}
$$

From Theorem 2, the optimization problem is solved by generalized eigendecomposition (A.6) previously described. Hence, the given algorithm finds a minimizer of the optimization problem.

Now we present the proof of Theorem 3.

Proof. Observe that if we substitute the probabilistic distribution $p(\boldsymbol{x}^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{in}})$ to the empirical distribution $q(\boldsymbol{x}^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{in}})$ defined in the main text, the first moments and the second moments are expressed as

$$
\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{x}^{\mathrm{ch}})}[\boldsymbol{x}^{\mathrm{ch}}] = \boldsymbol{X}_{\mathrm{ch}}\boldsymbol{v}, && \mathbb{E}_{p(\boldsymbol{x}^{\mathrm{in}})}[\boldsymbol{x}^{\mathrm{in}}] = \boldsymbol{X}_{\mathrm{in}}\boldsymbol{v}, \\
&\check{\boldsymbol{C}}_{\mathrm{ch,ch}} = \boldsymbol{X}_{\mathrm{ch}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{ch}}^{\top} - \boldsymbol{\mu}_{\mathrm{ch}}\boldsymbol{\mu}_{\mathrm{ch}}^{\top}, && \check{\boldsymbol{C}}_{\mathrm{ch,in}} = \boldsymbol{X}_{\mathrm{ch}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{in}}^{\top} - \boldsymbol{\mu}_{\mathrm{ch}}\boldsymbol{\mu}_{\mathrm{in}}^{\top}, \\
&\check{\boldsymbol{C}}_{\mathrm{in,ch}} = \check{\boldsymbol{C}}_{\mathrm{ch,in}}^{\top}, && \check{\boldsymbol{C}}_{\mathrm{in,in}} = \boldsymbol{X}_{\mathrm{in}}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}_{\mathrm{in}}^{\top} - \boldsymbol{\mu}_{\mathrm{in}}\boldsymbol{\mu}_{\mathrm{in}}^{\top},
\end{aligned}
$$

which implies that the optimization algorithm given in the result above is equivalent to Algorithm 1 in this case. Thus, Theorem 3 is established. □

A.3 Kernel Weighted CCA

Suppose we are given n drugs. The kernel matrices of the chemical and interaction kernels $\boldsymbol{K}_{\mathrm{ch}} \in \mathbb{S}^{n \times n}$ and $\boldsymbol{K}_{\mathrm{in}} \in \mathbb{S}^{n \times n}$ are defined so that their respective entries are the values of the kernel functions, as given by

$$\forall i, \forall j \in \mathbb{N}_n : \quad K_{i,j}^{\mathrm{ch}} = K_{\mathrm{ch}}(\boldsymbol{x}_i^{\mathrm{ch}}, \boldsymbol{x}_j^{\mathrm{ch}}), \quad K_{i,j}^{\mathrm{in}} = K_{\mathrm{in}}(\boldsymbol{x}_i^{\mathrm{in}}, \boldsymbol{x}_j^{\mathrm{in}}).$$

The kernel empirical mapping of the two information sources are defined as

$$
\begin{aligned}
\boldsymbol{k}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}}) &\equiv [K_{\mathrm{ch}}(\boldsymbol{x}_1^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{ch}}), \ldots, K_{\mathrm{ch}}(\boldsymbol{x}_n^{\mathrm{ch}}, \boldsymbol{x}^{\mathrm{ch}})]^{\top}, \\
\boldsymbol{k}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}}) &\equiv [K_{\mathrm{in}}(\boldsymbol{x}_1^{\mathrm{in}}, \boldsymbol{x}^{\mathrm{in}}), \ldots, K_{\mathrm{in}}(\boldsymbol{x}_n^{\mathrm{in}}, \boldsymbol{x}^{\mathrm{in}})]^{\top}.
\end{aligned}
$$

KWCCA needs the kernel values among shifted vectors in the kernel Hilbert spaces. The shifted kernels for chemical profiles can be computed as

$$
\begin{aligned}
\bar{K}_{\mathrm{ch}}(\boldsymbol{x}_i^{\mathrm{ch}}, \boldsymbol{x}_j^{\mathrm{ch}}) &\equiv K_{\mathrm{ch}}(\boldsymbol{x}_i^{\mathrm{ch}}, \boldsymbol{x}_j^{\mathrm{ch}}) - \boldsymbol{v}^{\top}(\boldsymbol{k}_{\mathrm{ch}}(\boldsymbol{x}_i^{\mathrm{ch}}) - \boldsymbol{k}_{\mathrm{ch}}(\boldsymbol{x}_j^{\mathrm{ch}})) \\
&\qquad + \boldsymbol{v}^{\top}\boldsymbol{K}_{\mathrm{ch}}\boldsymbol{v}, \\
\bar{\boldsymbol{k}}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}}) &\equiv (\boldsymbol{I}_n - \boldsymbol{1}_n\boldsymbol{v}^{\top})(\boldsymbol{k}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}}) - \boldsymbol{K}_{\mathrm{ch}}\boldsymbol{v}), \\
\bar{\boldsymbol{K}}_{\mathrm{ch}} &\equiv (\boldsymbol{I}_n - \boldsymbol{1}_n\boldsymbol{v}^{\top})\boldsymbol{K}_{\mathrm{ch}}(\boldsymbol{I}_n - \boldsymbol{v}\boldsymbol{1}_n^{\top}),
\end{aligned}
\tag{A.7}
$$

and $\bar{K}_{\mathrm{in}}(\boldsymbol{x}_i^{\mathrm{in}}, \boldsymbol{x}_j^{\mathrm{in}})$, $\bar{\boldsymbol{k}}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})$, and $\bar{\boldsymbol{K}}_{\mathrm{in}}$ are defined similarly.

If we define $\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}} \equiv \boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{K}}_{\mathrm{ch}}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}$, $\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}} \equiv \boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{K}}_{\mathrm{in}}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}$, the expected deviation between the images in $\mathbb{R}^m$ is expressed as

$$\mathbb{E}[\|\boldsymbol{\psi}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}}) - \boldsymbol{\psi}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\|^2] = 2m - 2\mathrm{tr}(\boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}\boldsymbol{A}_{\mathrm{in}}). \tag{A.8}$$

The second moment matrices can be written as

$$
\begin{aligned}
\mathbb{E}[\boldsymbol{\psi}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})\boldsymbol{\psi}_{\mathrm{ch}}(\boldsymbol{x}^{\mathrm{ch}})^{\top}] &= \boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}^2\boldsymbol{A}_{\mathrm{ch}}, \\
\mathbb{E}[\boldsymbol{\psi}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})\boldsymbol{\psi}_{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}})^{\top}] &= \boldsymbol{A}_{\mathrm{in}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}^2\boldsymbol{A}_{\mathrm{in}}.
\end{aligned}
$$

Hence, the algorithm can be reduced to the following maximization problem:

$$
\begin{aligned}
\text{max} \quad & \mathrm{tr}(\boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}\boldsymbol{A}_{\mathrm{in}}) \\
\text{wrt} \quad & \boldsymbol{A}_{\mathrm{ch}}, \boldsymbol{A}_{\mathrm{in}} \in \mathbb{R}^{n \times m}, \\
\text{subj to} \quad & \boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}^2\boldsymbol{A}_{\mathrm{ch}} = \boldsymbol{A}_{\mathrm{in}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}^2\boldsymbol{A}_{\mathrm{in}} = \boldsymbol{I}_m.
\end{aligned}
$$

Since the rank of $\boldsymbol{I}_n - \boldsymbol{v}\boldsymbol{1}_n^{\top}$ is $n-1$, the two shifted kernel matrices, $\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}$ and $\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}$, are always singular. To avoid overfitting, we introduce two regularization terms $\gamma_{\mathrm{ch}}\boldsymbol{A}_{\mathrm{ch}}^{\top}\boldsymbol{A}_{\mathrm{ch}}$ and $\gamma_{\mathrm{in}}\boldsymbol{A}_{\mathrm{in}}^{\top}\boldsymbol{A}_{\mathrm{in}}$ to scale constraints as

$$\boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}^2\boldsymbol{A}_{\mathrm{ch}} + \gamma_{\mathrm{ch}}\boldsymbol{A}_{\mathrm{ch}}^{\top}\boldsymbol{A}_{\mathrm{ch}} = \boldsymbol{A}_{\mathrm{in}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}^2\boldsymbol{A}_{\mathrm{in}} + \gamma_{\mathrm{in}}\boldsymbol{A}_{\mathrm{in}}^{\top}\boldsymbol{A}_{\mathrm{in}} = \boldsymbol{I}_m, \tag{A.9}$$

where γ_{ch} and γ_{in} are constants. The following algorithm finds $\boldsymbol{A}_{\mathrm{ch}}$ and $\boldsymbol{A}_{\mathrm{in}}$ minimizing the expected deviation subject to the regularized constraints.

Algorithm 2 (Kernel Weighted CCA) Solve the following generalized eigen-decomposition:

$$
\begin{bmatrix} \boldsymbol{0}_{n \times n} & \tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}} \\ \tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}} & \boldsymbol{0}_{n \times n} \end{bmatrix}
\begin{bmatrix} \boldsymbol{\alpha}^{\mathrm{ch}} \\ \boldsymbol{\alpha}^{\mathrm{in}} \end{bmatrix}
= \begin{bmatrix} \tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}^2 + \gamma_{\mathrm{ch}}\boldsymbol{I}_n & \boldsymbol{0}_{n \times n} \\ \boldsymbol{0}_{n \times n} & \tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}^2 + \gamma_{\mathrm{in}}\boldsymbol{I}_n \end{bmatrix}
\begin{bmatrix} \boldsymbol{\alpha}^{\mathrm{ch}} \\ \boldsymbol{\alpha}^{\mathrm{in}} \end{bmatrix}
$$

The matrix $\begin{bmatrix} \boldsymbol{A}_{\mathrm{ch}} \\ \boldsymbol{A}_{\mathrm{in}} \end{bmatrix}$ is set so that its hth column is set to the hth major generalized eigenvector.

Theorem 4. Algorithm 2 yields the parameters of the mapping functions $(\boldsymbol{\psi}_{\mathrm{ch}}, \boldsymbol{\psi}_{\mathrm{in}})$ which minimize the expected squared deviation (A.8) subject to the scaling constraints given in Eq. (A.9).

Proof. The problem of minimizing Eq. (A.8) subject to Eq. (A.9) is rewritten as

$$\begin{aligned} \text{max} \quad & \mathrm{tr}(\boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}\boldsymbol{A}_{\mathrm{in}}) \\ \text{wrt} \quad & \boldsymbol{A}_{\mathrm{ch}} \in \mathbb{R}^{n\times m}, \boldsymbol{A}_{\mathrm{in}} \in \mathbb{R}^{n\times m}, \\ \text{subj to} \quad & \quad \boldsymbol{A}_{\mathrm{ch}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{ch}}^{2}\boldsymbol{A}_{\mathrm{ch}} + \gamma_{\mathrm{ch}}\boldsymbol{A}_{\mathrm{ch}}^{\top}\boldsymbol{A}_{\mathrm{ch}} \\ & = \boldsymbol{A}_{\mathrm{in}}^{\top}\tilde{\bar{\boldsymbol{K}}}_{\mathrm{in}}^{2}\boldsymbol{A}_{\mathrm{in}} + \gamma_{\mathrm{in}}\boldsymbol{A}_{\mathrm{in}}^{\top}\boldsymbol{A}_{\mathrm{in}} = \boldsymbol{I}_{m}. \end{aligned}$$

From Theorem 2, the optimization problem is solved by generalized eigendecomposition in Algorithm 2. □

A.4 Weighted SVM

Let us denote the protein-ligand interaction table by $\boldsymbol{Y} \in \{\pm 1, 0\}^{n\times p_{\mathrm{in}}}$, where each row represents a ligand, and each column represents a target protein. Each cell in the table $\boldsymbol{Y}$ takes one of three values, ±1 and 0: +1 and −1 indicate the existence and the absence of the interaction, respectively, and unknowns are 0.

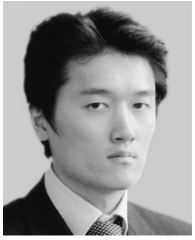
SVM learning algorithm finds a classification boundary minimizing the violations for the constraints that training points are kept out of the margin. The weighted SVM employed in this study counts the violations with weights v_j (See Sections 2.5 and 3.6) given to each training data.

When the interaction between a ligand i and a target t is predicted, ligands whose interactions with the target t are known are selected as training data. If the index set of the ligands for training is denoted by $\mathcal{I}_i$, then the weighted SVM minimizes

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{j\in\mathcal{I}_i} v_j \max\left(0, 1 - Y_{j,t} f(\boldsymbol{x}_j^{\mathrm{ch}}; \boldsymbol{w}, b)\right),$$

where C is constant. This algorithm is reduced to the classical SVM if all v_j's are set to be equal in value. The dual problem of the weighted SVM learning algorithm can be derived as a quadratic program with box constraints and a single equality linear constraint, enabling fast learning with kernels.

Raissa Relator was born in 1984. She received her B.S. and M.S. degrees in Mathematics from the University of the Philippines-Diliman in 2005 and 2008, respectively. She is currently pursuing a Ph.D. in Computer Science at the Graduate School of Engineering, Gunma University.



Tsuyoshi Kato was born in 1975. He received his B.E., M.E., and Ph.D. degrees in engineering from Tohoku University, Sendai, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2005, he was with the National Institute of Advanced Industrial Science Technology (AIST) as a Postdoctoral Fellow in the Computational Biology Research Center (CBRC), Tokyo. From 2005 to 2008, he was an Assistant Professor at the Graduate School of Frontier Sciences, the University of Tokyo, Japan. He became an Associate Professor at the Center for Informational Biology, Ochanomizu University, Tokyo, Japan, in 2008. Then, he stayed in the University of Tokyo for seven months, and now he is an associate professor at the Graduate School of Engineering, Gunma University. His current scientific interests include bioinformatics and statistical pattern recognition. Dr. Kato is a member of The Institute of Electronics, Information and Communication Engineers (IEICE) and Japanese Society for Bioinformatics (JSBi).

Richard Lemence was born in 1977. He received his Ph.D. degree in 2005 from Niigata University, Niigata City, Japan under his adviser Dr. Kouie Sekigawa. In 2009–2012, he was a JST post-doctoral fellow at Ochanomizu University under the supervision of Dr. Jesper Jansson. Currently, he is an Associate Professor at the Institute of Mathematics, College of Science, University of the Philippines-Diliman. He is a member of the Mathematical Society of the Philippines and the American Mathematical Society.

(Communicated by *Kengo Kinoshita*)