Artificial Life and Robotics (2018) 23:205–212
https://doi.org/10.1007/s10015-017-0416-8

ORIGINAL ARTICLE

PKRank: a novel learning-to-rank method for ligand-based virtual screening using pairwise kernel and RankSVM

Shogo D. Suzuki[1,2] · **Masahito Ohue**[1,3,4] · **Yutaka Akiyama**[1,2,3,4,5]

Received: 30 March 2017 / Accepted: 27 November 2017 / Published online: 18 December 2017

Abstract
The development of a new drug takes over 10 years and costs approximately US $2.6 billion. Virtual compound screening (VS) is a part of efforts to reduce this cost. Learning-to-rank is a machine learning technique in information retrieval that was recently introduced to VS. It works well because the application of VS requires the ranking of compounds. Moreover, learning-to-rank can treat multiple heterogeneous experimental data because it is trained using only the order of activity of compounds. In this study, we propose PKRank, a novel learning-to-rank method for ligand-based VS that uses a pairwise kernel and RankSVM. PKRank is a general case of the method proposed by Zhang et al. with the advantage of extensibility in terms of kernel selection. In comparisons of predictive accuracy, PKRank yielded a more accurate model than the previous method.

Keywords Cheminformatics · Learning-to-rank · Machine learning · Virtual screening · PKRank

1 Introduction

The development of a safe and effective drug takes more than 10 years and costs approximately US $2.6 billion [1]. Virtual screening (VS), which predicts the activity of untested compounds at a target drug protein using computational methods, is widely used in drug discovery research to reduce the developmental cost of medication [2]. Ligand-based virtual screening (LBVS) is a VS method [3] where predictions are formulated as classification or regression problems, and the activity of untested compounds is predicted by machine learning methods using the activity of tested compounds.

Learning-to-rank is a machine learning framework in the field of information retrieval used to treat ranking models [4], and has lately been introduced to LBVS. In LBVS with learning-to-rank, VS is formulated as a problem of ranking prediction concerning the activity of compounds, such as half the maximum inhibitory concentration (IC_{50}), and ranking is predicted using the ranking prediction model of learning-to-rank. Agarwal et al. [5] introduced learning-to-rank to VS for the first time, and showed that their method using RankSVM outperformed the method that simply employs the support vector machine (SVM) and the support vector regression (SVR). Rathke et al. [6] proposed StructRank, which directly solves the ranking problem and focuses on the

This work was presented in part at the 22nd International Symposium on Artificial Life and Robotics, Beppu, Oita, January 19–21, 2017.

✉ Yutaka Akiyama
akiyama@c.titech.ac.jp

[1] Department of Computer Science, School of Computing, Tokyo Institute of Technology, 2-12-1 W8-76 Ookayama, Meguro-ku, Tokyo 152-8550, Japan

[2] Education Academy of Computational Life Sciences (ACLS), Tokyo Institute of Technology, 2-12-1 W8-93 Ookayama, Meguro-ku, Tokyo 152-8550, Japan

[3] Advanced Computational Drug Discovery Unit (ACDD), Institute of Innovative Research, Tokyo Institute of Technology, 4259 Nagatsutacho, Midori-ku, Yokohama, Kanagawa 226-8501, Japan

[4] AIST-Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory (RWBC-OIL), National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan

[5] Molecular Profiling Research Center for Drug Discovery (molprof), National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan

most promising compounds in terms of activity. Zhang et al. [7] compared several learning-to-rank prediction models, and concluded that RankSVM is the best. Furthermore, they noted that learning-to-rank can treat multiple heterogeneous experimental data measured for different targets or platforms. This is major advantage of learning-to-rank because a traditional VS approach, such as classification and regression, cannot integrate multiple heterogeneous experimental data. Their method was based on the tensor product of the feature vectors of a compound and a target protein.

Drug–target interaction problems [8] have been studied as well as LBVS. This problem involves multiple compounds and proteins, where a multiplicity of interactions between compounds and proteins is predicted. We note that the drug–target interaction problem differs from LBVS because it focuses on the predictive accuracy of entire interactions involving multiple compounds and multiple proteins, whereas LBVS focuses on the predictive accuracy of interactions involving multiple compounds and a specific protein as drug target. A pairwise kernel method was proposed for the drug–target interaction problem [9]. It is a kernel-based machine learning method. A pairwise kernel is defined as the product of a compound kernel and a protein kernel, thus the pairwise kernel method has extensibility in terms of selecting the compound kernel and the protein kernel.

We note that the method involving the tensor product proposed by Zhang et al. is a special case of the pairwise kernel method. If both the compound kernel and the protein kernel are represented as a linear kernel, the pairwise kernel method is equivalent to the method that uses a tensor product.

In this paper, we propose a novel VS method called PKRank, which is a learning-to-rank-based VS method, using a pairwise kernel and RankSVM. PKRank has several advantages over the method that uses a tensor product: (a) PKRank can handle high-dimensional feature vectors. (b) Any kernel function can be used for the compound kernel and the protein kernel. (c) PKRank can handle similarity measurement for prediction.

The purpose of this study is to obtain a more accurate prediction model through PKRank than the method that uses the tensor product [7]. A comparison in terms of prediction accuracy between PKRank and the previous method with compound activity data recorded in BindingDB [10] showed that the former is superior.

2 Methods

We use the following notation throughout this paper: let $\mathbf{c} \equiv (c_1, ..., c_{d(\mathbf{c})})^\top$ be a feature vector of a compound and $\mathbf{p} \equiv (p_1, ..., p_{d(\mathbf{p})})^\top$ be a feature vector of a protein, where $d(\mathbf{c})$ and $d(\mathbf{p})$ are the number of dimensions of vector $\mathbf{c}$ and vector $\mathbf{p}$, respectively.

The ranking prediction model f of learning-to-rank is represented as $f(\mathbf{x}) = f(\mathbf{\Phi}(\mathbf{c}, \mathbf{p}))$, where $\mathbf{x} \equiv \mathbf{\Phi}(\mathbf{c}, \mathbf{p})$ is an input feature vector and $\mathbf{\Phi}$ is a feature map. In this section, we explain the method proposed by Zhang et al. that uses the tensor product as $\mathbf{\Phi}$ [7] as well as the proposed method PKRank, which is a learning-to-rank-based VS using a pairwise kernel. The former method is a special case of the latter, as described presently.

2.1 Previously proposed method (tensor product)

Zhang et al. [7] introduced the tensor product as feature map $\mathbf{\Phi}$ as follows:

$$\mathbf{\Phi}(\mathbf{c}, \mathbf{p}) = \mathbf{c} \otimes \mathbf{p}, \tag{1}$$

where $\otimes$ is the tensor product operator. If $\mathbf{c}$ is a $d(\mathbf{c})$-dimensional feature vector and $\mathbf{p}$ is a $d(\mathbf{p})$-dimensional feature vector, $\Phi(\mathbf{c}, \mathbf{p}) = \mathbf{c} \otimes \mathbf{p}$ is a $d(\mathbf{c}) \times d(\mathbf{p})$-dimensional feature vector. Zhang et al. used a general descriptor [11] (GD, 32 dimensions) as compound feature vector $\mathbf{c}$, and composition transition and the distribution feature [12] (CTD, 147 dimensions) as protein feature vector $\mathbf{p}$. Hence, they used a 4,704-dimensional feature vector as input to the ranking prediction model *f*. GD and CTD represent the physicochemical properties of a compound and a protein, respectively.

2.2 Proposed method (PKRank)

The pairwise kernel [9] was originally proposed in the context of the drug–target interaction problem [8]. Pairwise kernel $k : \mathbb{R}^{d(\mathbf{c}) \times d(\mathbf{p})} \times \mathbb{R}^{d(\mathbf{c}) \times d(\mathbf{p})} \to \mathbb{R}$ is defined between two pairs of proteins and compounds $(\mathbf{c}, \mathbf{p})$ and $(\mathbf{c}', \mathbf{p}')$ as follows:

$$k((\mathbf{c}, \mathbf{p}), (\mathbf{c}', \mathbf{p}')), \tag{2}$$

$$= \Phi(\mathbf{c}, \mathbf{p})^\top \Phi(\mathbf{c}', \mathbf{p}'), \tag{3}$$

$$= (\Phi_{\text{com}}(\mathbf{c}) \otimes \Phi_{\text{pro}}(\mathbf{p}))^\top (\Phi_{\text{com}}(\mathbf{c}') \otimes \Phi_{\text{pro}}(\mathbf{p}')), \tag{4}$$

$$= (\Phi_{\text{com}}(\mathbf{c})^\top \Phi_{\text{com}}(\mathbf{c}')) \times (\Phi_{\text{pro}}(\mathbf{p})^\top \Phi_{\text{pro}}(\mathbf{p}')), \tag{5}$$

$$= k_{\text{com}}(\mathbf{c}, \mathbf{c}') \times k_{\text{pro}}(\mathbf{p}, \mathbf{p}'), \tag{6}$$

where $k_{\text{com}} : \mathbb{R}^{d(\mathbf{c})} \times \mathbb{R}^{d(\mathbf{c})} \to \mathbb{R}$ is a compound kernel between two compounds, and $k_{\text{pro}} : \mathbb{R}^{d(\mathbf{p})} \times \mathbb{R}^{d(\mathbf{p})} \to \mathbb{R}$ is a protein kernel between two proteins.

RankSVM [13] is a learning-to-rank model based on a pairwise approach using SVM. Zhang et al. compared several learning-to-rank prediction models and concluded that RankSVM is the best. RankSVM can be extended to use the

kernel method as well as SVM; thus, the pairwise kernel can be used.

Our proposed PKRank is a learning-to-rank method that uses a pairwise kernel and RankSVM. There are two steps in the training of PKRank: (1) A Gram matrix of the pairwise kernel K is generated. (2) RankSVM is trained with the input of the Gram matrix of the pairwise kernel K and the order of activity of compounds against target proteins. PKRank requires k_{com} and k_{pro} to generate the Gram matrix of the pairwise kernel K. Figure 1 shows the training overview of PKRank and Fig. 2 shows the overview of the generation of the Gram matrix of the pairwise kernel K.

If both k_{com} and k_{pro} are represented as linear kernel $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^{\top}\Phi(\mathbf{x}') \equiv \mathbf{x}^{\top}\mathbf{x}'$, we obtain $\Phi_{\mathrm{com}}(\mathbf{c}) = \mathbf{c}$ and $\Phi_{\mathrm{pro}}(\mathbf{p}) = \mathbf{p}$ from (5) and (6). Then, we get $\Phi(\mathbf{c}, \mathbf{p}) = \mathbf{c} \otimes \mathbf{p}$ from (3) and (4). This is equivalent to the method that uses the tensor product; hence, this method is a special case of PKRank.

PKRank has several advantages: (a) The tensor product method cannot handle high-dimensional feature vectors because the number of dimensions of the tensor product is large [as previously described, $\mathbf{c} \otimes \mathbf{p}$ is a $d(\mathbf{c}) \times d(\mathbf{p})$ -dimensional feature vector]. However, PKRank can handle it because the pairwise kernel uses not a feature map $\Phi(\mathbf{c}, \mathbf{p})$, but the compound kernel k_{com} and the protein kernel k_{pro}. (b) In case of the tensor product method, the compound kernel k_{com} and the protein kernel k_{pro} are fixed to use a linear kernel, as described. However, PKRank can use any kernel function in addition to the linear kernel. (c) PKRank can handle similarity measurements for prediction. This is because the pairwise kernel method needs only the Gram matrix, whose elements represent the similarity between compounds or proteins; thus, a feature vector representation of compound Φ_{com} or protein Φ_{pro} is not always required.

To make full use of advantage (a) of PKRank, we introduce Extended-connectivity Fingerprints [14] (ECFP4, 2,048 dimensions) as a compound feature vector, which is a topological fingerprint representing the presence of substructures. ECFP4 cannot be dealt with by the method of tensor product because of its large dimensionality (if ECFP4

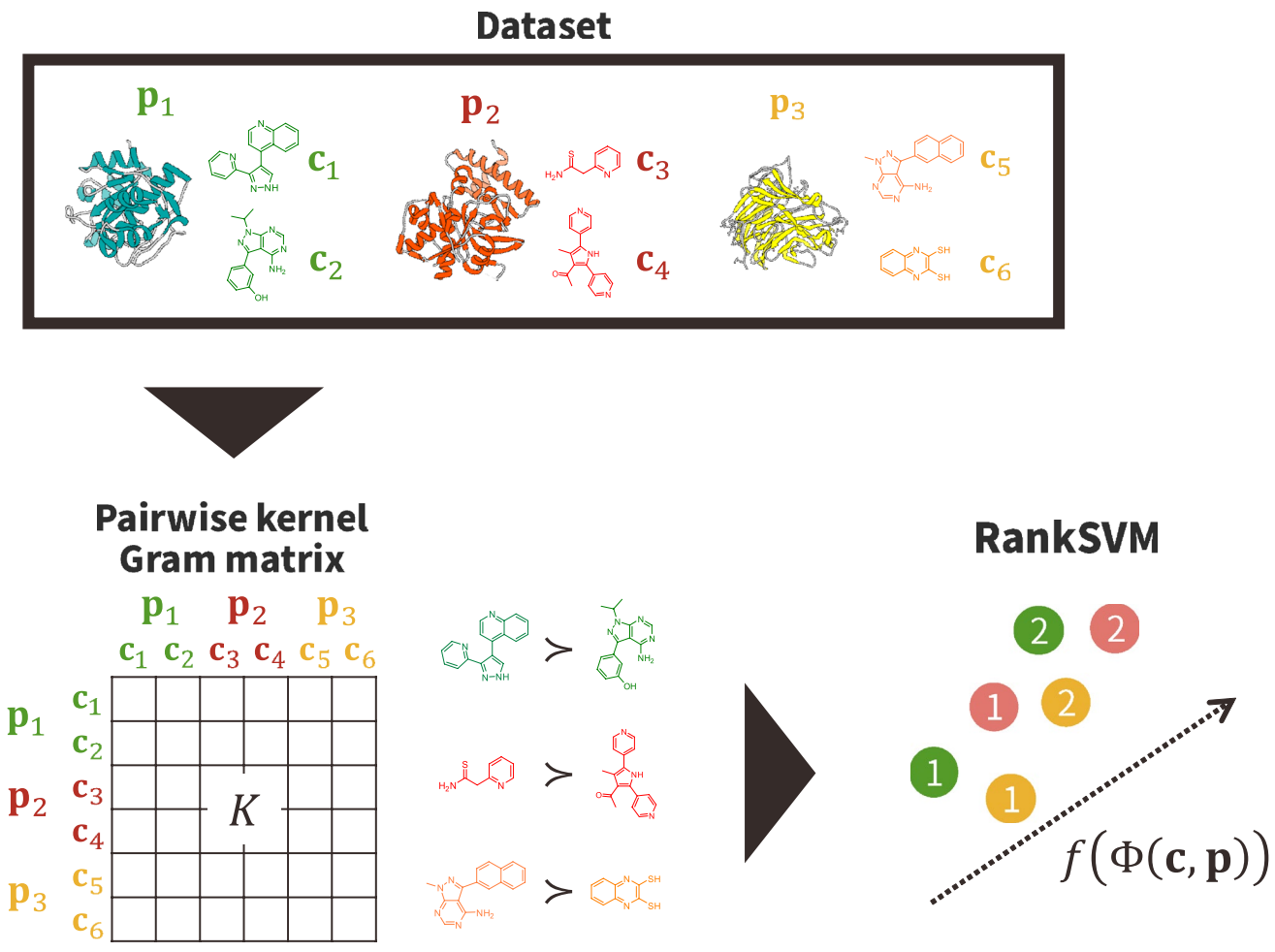

Fig. 1 The overview in the training of proposed method: PKRank

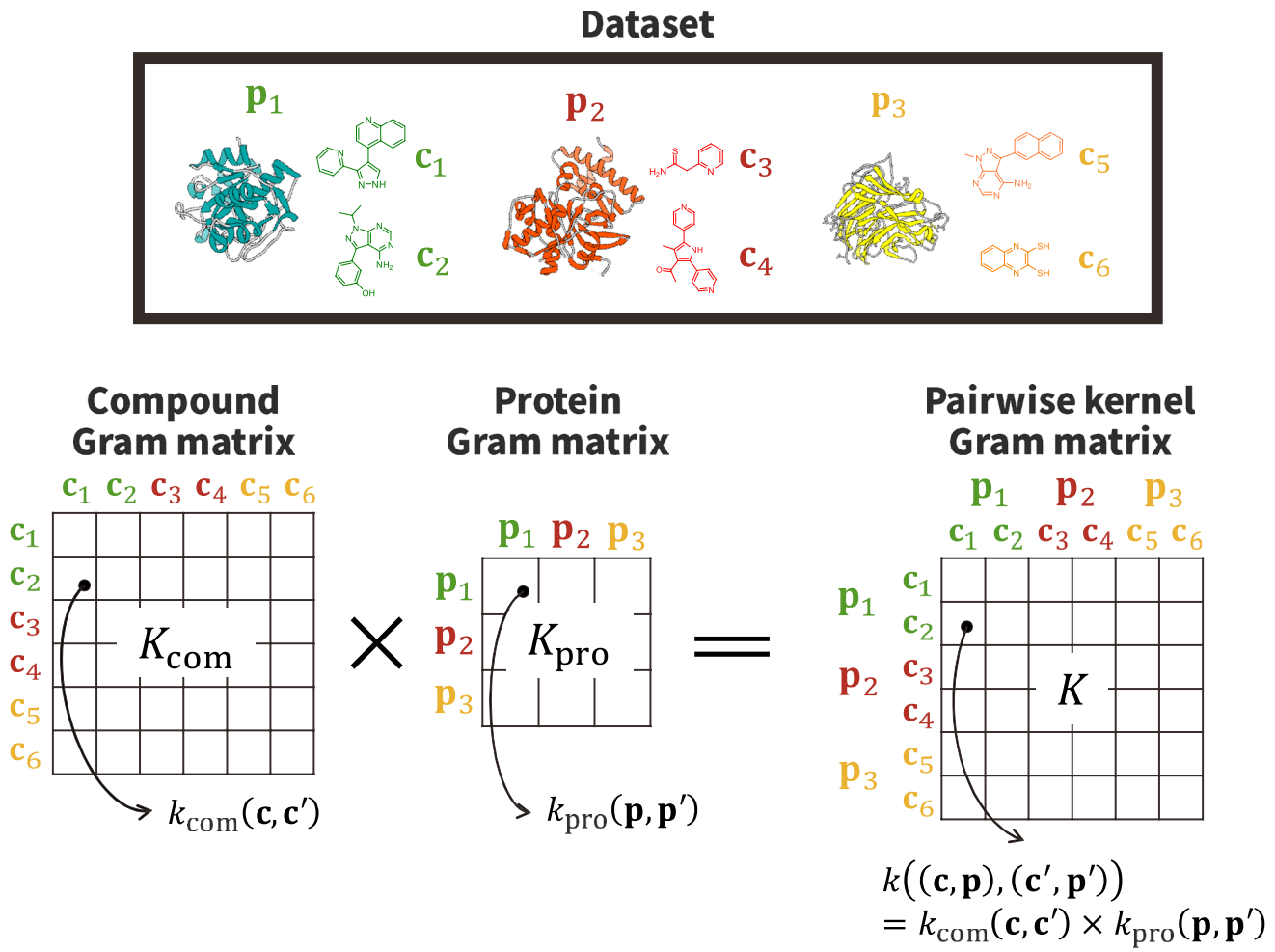

Fig. 2 Generating a Gram matrix of the pairwise kernel K

and CTD are used in this method, a 301,056-dimensional feature vector is the input to the ranking prediction model). With regard to advantage (b), we introduce a polynomial kernel, a radial basis function (RBF) kernel and the Tanimoto kernel [15], for the compound kernel k_{com}, and a polynomial kernel, an RBF kernel, for the protein kernel k_{pro}. To exploit advantage (c), we introduce the normalized Smith–Waterman score (nSW) [16] for protein kernel k_{pro}, which is a normalized local alignment score between sequences. The nSW is calculated only using two amino acid sequences, and shows the similarity between proteins. Thus, the nSW can also be used as protein kernel k_{pro}.

2.3 Kernels

We use a linear kernel, a polynomial kernel, an RBF kernel, and the Tanimoto kernel for the compound kernel k_{com}, and a linear kernel, an RBF kernel, and the normalized Smith–Waterman score (nSW) for protein kernel k_{pro}. Here, we explain these kernels.

- A linear kernel between two features $\mathbf{x}, \mathbf{x}'$ is

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'. \tag{7}$$

  As seen above, if both k_{com} and k_{pro} are represented as a linear kernel, PKRank is equivalent to the tensor product method.
- A polynomial kernel between two features $\mathbf{x}, \mathbf{x}'$ is

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + 1)^z. \tag{8}$$

  It has a hyper-parameter z, the manner of tuning which is explained in Sect. 3.4.
- An RBF kernel between two features $\mathbf{x}, \mathbf{x}'$ is

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2). \tag{9}$$

  It is widely used in kernel-based machine learning. It has a hyper-parameter γ, the manner of tuning which is explained in Sect. 3.4.
- The Tanimoto kernel between two binary vectors $\mathbf{x}, \mathbf{x}'$ is

$$k(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^\top \mathbf{x}'}{\mathbf{x}^\top \mathbf{x} + \mathbf{x}'^\top \mathbf{x}' - \mathbf{x}^\top \mathbf{x}'}. \tag{10}$$

The Tanimoto coefficient is used to measure similarity between compounds using a binary feature. In the drug–target interaction problem, the Tanimoto coefficient is used as a compound kernel. We use the Tanimoto kernel only with ECFP4 (binary feature).

- The normalized Smith–Waterman score (nSW) between two proteins seq, seq' is:

$$k(seq, seq') = \frac{\mathrm{SW}(seq, seq')}{\sqrt{\mathrm{SW}(seq, seq)}\sqrt{\mathrm{SW}(seq', seq')}}, \tag{11}$$

where $\mathrm{SW}(\cdot,\cdot)$ is the Smith–Waterman score, which is a local alignment score between amino acid sequences.

3 Experiments

3.1 Data

To compare the predictive accuracy of the method that uses the tensor product and PKRank, compound activity data measured in IC_{50} against proteins of phosphodiesterase (PDE), cathepsin (CTS), and the adenosine receptor (ADOR) family, recorded in BindingDB [10], were used as benchmark. We note that the compound activity data against the PDE and the CTS families had been used in a previous study [7]. To remove highly similar compounds from the dataset, clustering based on Butina's method [17] with ECFP4 and the Tanimoto coefficient was performed. We set the threshold of similarity cutoff of Butina's algorithm to 0.8. The non-redundant datasets compiled are shown in Table 1.

3.2 Evaluation criteria

In information retrieval, normalized discounted cumulative gain (NDCG) [19] is widely used for evaluation. There are two types of NDCG: NDCG1 and NDCG2. NDCG1 is defined as follows:

$$\mathrm{DCG1@}m = rel_1 + \sum_{i=2}^{m} \frac{rel_i}{\log_2 i}, \tag{12}$$

$$\mathrm{NDCG1@}m = \frac{\mathrm{DCG1@}m}{\mathrm{IdealDCG1@}m}, \tag{13}$$

where m is the number of items considered for evaluation, rel_i is the relevance of the item at position i in predicted ranking, and IdealDCG1@m is the normalization term defined as DCG1@m if all items contained in the dataset are sorted according to their true relevance. On the other hand, NDCG2 is defined as follows:

$$\mathrm{DCG2@}m = \sum_{i=1}^{m} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{14}$$

$$\mathrm{NDCG2@}m = \frac{\mathrm{DCG2@}m}{\mathrm{IdealDCG2@}m}. \tag{15}$$

The study where the method using the tensor product was proposed [7] used NDCG2@10 for evaluation, but this is unstable because it changes drastically due to the rel_i-th power of 2, even if the predicted ranking changes only slightly. Thus, we used NDCG1@100 and NDCG1@10 in addition to NDCG2@10 for evaluation.

We note that $\mathrm{pIC_{50}} \equiv -\log_{10}(\mathrm{IC_{50}})$ is used for the relevance of a compound. The higher the $\mathrm{pIC_{50}}$ of a compound, the more strongly it binds to a target protein; hence, $\mathrm{pIC_{50}}$ shows the relevance of a compound.

Table 1 Three datasets used as benchmark in this study (PDE, CTS, ADOR)

PDE family (15 subfamilies)			
PDE5 [P54750] (835)	PDE1A [P54750] (12)	PDE1B [Q01064] (132)	PDE1C [Q14123] (141)
PDE2A [O00408] (324)	PDE3A [Q14432] (177)	PDE3B [Q13370] (22)	PDE4A [P27815] (356)
PDE4B [Q07343] (514)	PDE4C [Q08493] (83)	PDE6A [P35913] (32)	PDE6C [P51160] (13)
PDE9A [O76083] (72)	PDE10 [Q9Y233] (1307)	PDE11A [Q9HCR9] (76)	
CTS family (10 subfamilies)			
CTSK [P43235] (735)	CTSB [P07858] (440)	CTSD [P07339] (686)	CTSE [P14091] (20)
CTSF [Q9UBX1] (20)	CTSG [P08311] (186)	CTSH [P09668] (15)	CTSL [P07711] (566)
CTSS [P25774] (771)	CTSZ [Q9UBR2] (6)		
ADOR family (4 subfamilies)			
ADORA3 [P0DMS8] (201)	ADORA1 [P30542] (390)	ADORA2A [P29274] (141)	ADORA2B [P29275] (199)

Test sets (PDE5, CTSK, ADORA3) are shown in bold. IDs in brackets show the UniProt [18] accession numbers and the numbers in parentheses show the number of records of compound activity against each protein

3.3 Training method

We used RankSVM with kernel method implemented by Kuo et al. [20].

We tested ten methods by changing the feature vectors and kernels. For the compound feature vector, GD and ECFP4 were calculated by RDKit (version 2016.09.1) [21]. For the protein feature vector, CTD was calculated by PROFEAT (version 2016) [12], and the normalized Smith–Waterman score (nSW) was calculated by EMBOSS (version 6.6.0) [22].

3.4 Parameter settings

We tuned the hyper-parameters as follows: (1) We randomly split the test data into two parts (a validation part for hyper-parameter tuning and a test part for evaluation). (2) We used NDCG1@100, NDCG1@10, and NDCG2@10 to assess the test part using a hyper-parameter combination that maximized the evaluation score for the validation part. (3) We repeated (1) and (2) five times and reported the mean of the five evaluation scores.

RankSVM has a cost parameter C as SVM does, and we chose a values for it from the set $\{10^{-9}, 10^{-8}, ..., 10^{0}\}$. The polynomial kernel has a degree parameter z, which was chosen from $\{2, 3\}$. The RBF kernel has a bandwidth parameter γ, which was chosen from $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$.

4 Results

The evaluation scores (NDCG1@100, NDCG1@10 and NDCG2@10) are shown in Tables 2, 3 and 4. The best score among the 10 methods is shown in bold. The italicized

Table 2 Experimental results of PDE

Line no.	Compound feature	Compound kernel	Protein feature	Protein kernel	NDCG1@100	NDCG1@10	NDCG2@10
1	*GD*	*Linear*	*CTD*	*Linear*	*0.821*	*0.729*	*0.258*
2	GD	Polynomial	CTD	Polynomial	0.811	0.762*	0.256
3	GD	RBF	CTD	RBF	0.834*	0.830*	0.336*
4	GD	RBF	Sequence	nSW	**0.855***	**0.847***	0.371*
5	ECFP4	Linear	CTD	Linear	0.776	0.715	0.275
6	ECFP4	Polynomial	CTD	Polynomial	0.817	0.781*	0.326*
7	ECFP4	Tanimoto	CTD	RBF	0.827	0.740	0.313*
8	ECFP4	Tanimoto	Sequence	nSW	0.827	0.745	0.329*
9	ECFP4	RBF	CTD	RBF	0.838*	0.811*	0.390*
10	ECFP4	RBF	Sequence	nSW	0.849*	0.835*	**0.399***

The first italicized line shows the results of the method using tensor product [7]. The bold type shows the best score for each evaluation criterion. Scores with asterisk indicate significance at $P < 0.05$, calculated by the Wilcoxon signed-rank test between each of proposed methods (regular lines) and the method that uses tensor product (italicized line)

Table 3 Experimental results of CTS

Line no.	Compound feature	Compound kernel	Protein feature	Protein kernel	NDCG1@100	NDCG1@10	NDCG2@10
1	*GD*	*Linear*	*CTD*	*Linear*	*0.789*	*0.669*	*0.117*
2	GD	Polynomial	CTD	Polynomial	0.820*	0.730*	0.149*
3	GD	RBF	CTD	RBF	0.855*	0.839*	**0.599***
4	GD	RBF	sequence	nSW	0.843*	0.807*	0.405*
5	ECFP4	Linear	CTD	Linear	0.803*	0.644	0.122
6	ECFP4	Polynomial	CTD	Polynomial	**0.867***	**0.847***	0.525*
7	ECFP4	Tanimoto	CTD	RBF	0.819*	0.722*	0.234*
8	ECFP4	Tanimoto	sequence	nSW	0.820*	0.730*	0.273*
9	ECFP4	RBF	CTD	RBF	0.865*	0.821*	0.524*
10	ECFP4	RBF	sequence	nSW	0.846*	0.745*	0.262*

The first italicized line shows the results of the method using tensor product [7]. The bold type shows the best score for each evaluation criterion. Scores with asterisk indicate significance at $P < 0.05$, calculated by the Wilcoxon signed-rank test between each of proposed methods (regular lines) and the method that uses tensor product (italicized line)

Table 4 Experimental results of ADOR

Line no.	Compound feature	Compound kernel	Protein feature	Protein kernel	NDCG1@100	NDCG1@10	NDCG2@10
1	*GD*	*Linear*	*CTD*	*Linear*	*0.942*	*0.799*	*0.321*
2	GD	Polynomial	CTD	Polynomial	0.943	0.805	0.420*
3	GD	RBF	CTD	RBF	**0.948***	**0.812**	**0.439***
4	GD	RBF	sequence	nSW	0.941	0.802	0.370
5	ECFP4	Linear	CTD	Linear	0.937	0.790	0.339
6	ECFP4	Polynomial	CTD	Polynomial	0.935	0.787	0.301
7	ECFP4	Tanimoto	CTD	RBF	0.915	0.695	0.280
8	ECFP4	Tanimoto	sequence	nSW	0.901	0.701	0.221
9	ECFP4	RBF	CTD	RBF	0.922	0.744	0.361
10	ECFP4	RBF	sequence	nSW	0.916	0.728	0.318

The first italicized line shows the results of the method using tensor product [7]. The bold type shows the best score for each evaluation criterion. Scores with asterisk indicate significance at $P < 0.05$, calculated by the Wilcoxon signed-rank test between each of proposed methods (regular lines) and the method that uses tensor product (italicized line)

first lines in Tables 2, 3 and 4 are equivalent to the method using tensor product [7], as shown in Sect. 2.2. The other lines (from the second to the last line) correspond to the proposed method PKRank. To check the significance of the evaluation score of PKRank, the Wilcoxon signed-rank test was performed between each of the proposed methods (regular lines) and the method using tensor product (the italicized first line). The scores with an asterisk (*) show that the score of the combination of feature vectors and kernels was significant at $P < 0.05$.

For the three datasets, PKRank outperformed the method using tensor product in all three evaluations. The best combination of feature vectors and kernels was different for each dataset and evaluation criterion, but the combination of GD compound feature, RBF compound kernel, CTD protein feature, and RBF protein kernel (Line 3 in Tables 2, 3 and 4) comprehensively worked well.

5 Discussion and conclusion

The experimental results showed that the proposed PKRank is superior to the method using tensor product in NDCGs evaluation.

It is not advisable to use a linear kernel for the compound kernel or the protein kernel, but other kernels worked well. This is because non-linear kernels can represent more complicated ranking models. Since PKRank has extensibility in terms of kernel selection, these non-linear kernels can be used.

Since the evaluation scores of the number of cases were significant, it can be said that the settings of Line 3 in Tables 2, 3 and 4 worked well. There was only one case where the evaluation score was not significant for the ADOR dataset on NDCG1@10 evaluation, but was still better than that for methods using tensor product. One way to determine the best combination of features and kernels is by using cross-validation by changing features and kernels.

The nSW for the protein kernel worked well in the PDE dataset, but this tendency was not replicated in the results for the CTS and ADOR datasets. The training set of the PDE dataset had many proteins related to PDE5 (test data). This might have caused such a result, and further study is needed to determine when the nSW works well.

The purpose of this study was to obtain a more accurate prediction model by PKRank than the previous method that uses tensor product. This study showed that PKRank outperforms the tensor product-based method due to its several advantages. We believe that methods that can cope with multiple heterogeneous experimental data, like PKRank can, are important for drug discovery research. Moreover, classifying objects that are sampled jointly from two or more domains has many applications, such as in bioinformatics [23, 24], social network analysis [25], and world wide web [26]. The tensor product feature space is useful for modeling interactions between feature sets in different domains where PKRank can be applied to yield better performance.

Acknowledgements This work used computational resources of the TSUBAME 2.5 supercomputer provided by the Global Scientific Information and Computing Center (GSIC), Tokyo Institute of Technology, through the support of the Education Academy of Computational Life Sciences (ACLS), Tokyo Institute of Technology. This work was partially supported by a Grant-in-Aid for Scientific Research (A) (Grant number 24240044), a Grant-in-Aid for Young Scientists (B) (Grant number 15K16081) and a Grant-in-Aid for Scientific Research (B) (Grant number 17H01814) from the Japan Society for the Promotion of Science (JSPS), Core Research for Evolutional Science and Technology (CREST) "Extreme Big Data" from the Japan Science and Technology Agency (JST), and AIST-Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory (RWBC-OIL), National Institute of Advanced Industrial Science and Technology (AIST).

References

1. Mullard A (2014) New drugs cost US$2.6 billion to develop. Nat Rev Drug Discov 13:877
2. Lavecchia A, Di Giovanni C (2013) Virtual screening strategies in drug discovery: a critical review. Curr Med Chem 20(23):2839–2860
3. Lavecchia A (2015) Machine-learning approaches in drug discovery: methods and applications. Drug Discov Today 20(3):318–331
4. Liu T (2009) Learning to rank for information retrieval. Springer, Berlin Heidelberg
5. Agarwal S, Dugar D, Sengupta S (2010) Ranking chemical structures for drug discovery: a new machine learning approach. J Chem Inf Model 50(5):716–731
6. Rathke F, Hansen K, Brefeld U, Muller KR (2011) Structrank: a new approach for ligand-based virtual screening. J Chem Inf Model 51(1):83–92
7. Zhang W, Ji L, Chen Y, Tang K, Wang H, Zhu R, Jia W, Cao Z, Liu Q (2015) When drug discovery meets web search: learning to rank for ligand-based virtual screening. J Cheminform 7:5
8. Ding H, Takigawa I, Mamitsuka H, Zhu S (2014) Similarity-based machine learning methods for predicting drug-target interactions: a brief review. Brief Bioinform 15(5):734–747
9. Jacob L, Vert JP (2008) Protein-ligand interaction prediction: an improved chemogenomics approach. Bioinformatics 24(19):2149–2156
10. Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J (2016) BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic Acids Res 44(D1):D1045–D1053
11. Labute P (2000) A widely applicable set of descriptors. J Mol Graph Model 18(4–5):464–477
12. Rao HB, Zhu F, Yang GB, Li ZR, Chen YZ (2011) Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucleic Acids Res 39(Web Server issue):385–390
13. Herbrich R, Graepel T, Obermayer K (2000) Large margin rank boundaries for ordinal regression. In: Smola AJ, Bartlett P, Scholkopf B, Schuurmans D (eds) Advances in large margin classifiers. MIT Press, Cambridge, pp 115–132
14. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50(5):742–754
15. Willett P, Barnard JM, Downs GM (1998) Chemical similarity searching. J Chem Inf Comput Sci 38(6):983–996
16. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147(1):195–197
17. Butina D (1999) Unsupervised database clustering based on daylight's fingerprint and Tanimoto similarity: a fast and automated way to cluster small and large datasets. J Chem Inf Comput Sci 39(4):747–750
18. The UniProt Consortium (2017) UniProt: the universal protein knowledgebase. Nucleic Acids Res 45(D1):D158–D169
19. Jarvelin K, Kekalainen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inform Syst 20(4):422–446
20. Kuo T-M, Lee C-P, Lin C-J (2014) Large-scale kernel RankSVM. In: Proceedings of the 2014 SIAM international conference on data mining (SDM14), pp 812–820
21. RDKit: Open-source cheminformatics; http://www.rdkit.org. Accessed 13 Nov 2017
22. Rice P, Longden I, Bleasby A (2000) EMBOSS: the European molecular biology open software suite. Trends Genet 16(6):276–277
23. Ben-Hur A, Noble WS (2005) Kernel methods for predicting protein-protein interactions. Bioinformatics 21(Suppl 1):i38–46
24. Faulon JL, Misra M, Martin S, Sale K, Sapra R (2008) Genome scale enzyme-metabolite and drug-target interaction predictions using the signature molecular descriptor. Bioinformatics 24(2):225–233
25. Oyama S, Manning DC (2004) Using feature conjunctions across examples for learning pairwise classifiers. In: Proceedings of 15th European conference on machine learning (ECML2004), pp 322–333
26. Raymond R, Kashima H (2010) Fast and Scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In: Proceedings of the 2010 European conference on machine learning and knowledge discovery in databases (ECMLPKDD2010), pp 131–147