R-DTI: Drug Target Interaction Prediction based on Second-Order Relevance Exploration

Yang Hua[1], Zhenhua Feng[2], Xiaoning Song[1*], Hui Li[1], Tianyang Xu[1], Xiao-Jun Wu[1]

[1]School of Artifcial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu, P.R. China
[2]Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK
7211905018@stu.jiangnan.edu.cn; z.feng@surrey.ac.uk; x.song@jiangnan.edu.cn

Appendix A: The Basic Layers of SPDNet

Following the main paper, this section introduces the three basic SPDNet layers, BiMap, ReEig, and LogEig, in DTI prediction formulation. The BiMap layer is used to reduce the dimension of the SPD feature, effectively eliminating its redundant information. The BiMap layer is formulated as:

$$\mathbf{F}_r^k = f_b^k(\mathbf{F}_r^{k-1}; \mathbf{W}_k) = \mathbf{W}_k\, \mathbf{F}_r^{k-1}\, \mathbf{W}_k^\top, \tag{1}$$

where $\mathbf{F}_r^k$ is the input SPD matrix of the k-th layer, $\mathbf{W}_k \in \mathbb{R}^{C\times C_k}$ is the transformation matrix, and C is a generic dimension of the input SPD matrices. The SPD matrices include the channel-based Riemannian features $\mathbf{F}_{r^C} \in \mathbb{R}^{C_j\times C_j}$ and the length-based Riemannian features $\mathbf{F}_{r^L} \in \mathbb{R}^{L_j\times L_j}$. Since $\mathbf{F}_{r^L}$ is more sparse than $\mathbf{F}_{r^C}$, their corresponding transformation dimensions, C_{k^C} and C_{k^L}, are different, and we list their values in Table 1.

The ReEig and LogEig layers could introduce a nonlinearity into the context of the SPDNet and perform Riemannian computation on the resulting SPD matrices for output layers (Arsigny et al. 2007), to improve their discriminative performance. The ReEig layer is formulated as:

$$\mathbf{F}_r^k = f_r^k(\mathbf{F}_r^{k-1}) = \mathbf{U}\, max(\epsilon I, \Sigma)\, \mathbf{U}^\top, \tag{2}$$

$$max(\epsilon I, \Sigma)_{ii} = \begin{cases} \Sigma_{ii}, & \Sigma_{ii} > \epsilon \\ \epsilon, & \Sigma_{ii} < \epsilon \end{cases}, \tag{3}$$

where $\mathbf{U}$ and Σ are the eigenvector and eigenvalue matrices obtained by singular value decomposition on the input SPD matrix, ϵ is the small rectification threshold, $max(\epsilon I, \Sigma)$ is the diagonal matrix of the max operations.

The LogEig layer is formulated as:

$$\mathbf{F}_r^k = f_l^k(\mathbf{F}_r^{k-1}) = log(\mathbf{F}_r^{k-1}) = \mathbf{U}\, log(\Sigma)\, \mathbf{U}^\top, \tag{4}$$

where $\mathbf{F}_r^{k-1} = \mathbf{U}\Sigma\mathbf{U}^\top$, $log(\Sigma)$ is the diagonal matrix of the eigenvalue logarithms.

Table 1: Hyper-parameter settings of R-DTI.

Hyper-parameter	Value
Textual and structural feature channels: C_t, C_s,	192, 64
Protein and drug feature lengths: L_p, L_d	1200, 100
The joint feature length: L_j	1300
The joint feature channel: C_j	256
The transformation dimension: C_{k^C}, C_{k^L}	32, 24
The kernel size in DenseNet: K_p, K_d	15, 7
Number of linear layers in Classifier	3
The 1D-CNN kernel size in Auxiliary K	7
Number of convolution layers in Auxiliary	3
Number of linear layers in Auxiliary	3

Table 2: Statistics of the benchmarks

Datasets	Protein	Drug	Interaction	Positive	Negative
Human	852	1052	6212	3369	2843
C.elegans	2504	1434	7511	4000	3511
BindingDB	811	49567	60780	33777	27003
DrugBank	4254	6645	35022	17511	17511
Davis	379	68	25772	7320	18452
KIBA	225	2068	116350	22154	94196
sc-PDB	4782	6326	16034	-	-

Appendix B: Loss Functions, Hyperparameter Settings, and Measure Metrics.

In this section, we will introduce our training strategy and loss function. The training process of R-DTI consists of two steps. First, we train the pre-trained protein structural feature extractor. This process involves two supervions, contrastive learning and coordinate reconstruction. Specifically, we use the InfoNCE loss as a contrastive learning loss to bring the sequence hidden features closer to their structure hidden features. The InfoNCE loss is formulated as:

$$\mathcal{L}_q = -log \frac{\exp(q \cdot k_+)/\tau}{\sum_{i=0}^{K} \exp(q \cdot k_i)/\tau}, \tag{5}$$

where q is the sequence hidden feature. k_+ is the structural hidden feature of the same sample, and k_i is the ith structural hidden feature of other negative samples in a batch. τ is a temperature hyper-parameter per, and K is the number of the negative samples. Besides, we use SmoothL1 loss to reconstruct the amino acid coordinates based on the sequence hidden features. The SmoothL1 loss is formulated as:

$$\mathcal{L}_s = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{other} \end{cases}, \tag{6}$$

Table 3: The ablation study results for prediction formulation methods in terms of AUC-ROC and AUC-PR on the **BindingDB** and **DrugBank**.

Feature joint methods	**Relevance (space)**	**BindingDB**		**Davis**	
		AUC-ROC↑	***AUC-PR***↑	***AUC-ROC***↑	***AUC-PR***↑
Concatenation	**First-order**$_{\text{(Euclidean)}}$	0.965$_{\pm 0.001}$	0.968$_{\pm 0.002}$	0.897$_{\pm 0.001}$	0.831$_{\pm 0.001}$
Concatenation + LSTM	**First-order**$_{\text{(Euclidean)}}$	0.961$_{\pm 0.001}$	0.963$_{\pm 0.002}$	0.908$_{\pm 0.002}$	0.839$_{\pm 0.002}$
Concatenation + CNN	**First-order**$_{\text{(Euclidean)}}$	0.967$_{\pm 0.002}$	0.968$_{\pm 0.002}$	0.910$_{\pm 0.003}$	0.843$_{\pm 0.003}$
Inner product	**Second-order**$_{\text{(Euclidean)}}$	0.968$_{\pm 0.003}$	0.970$_{\pm 0.002}$	0.903$_{\pm 0.002}$	0.834$_{\pm 0.002}$
Outer product	**Second-order**$_{\text{(Euclidean)}}$	0.971$_{\pm 0.002}$	0.973$_{\pm 0.002}$	0.909$_{\pm 0.002}$	0.841$_{\pm 0.003}$
Transformer-based fusion	**Second-order**$_{\text{(Euclidean)}}$	0.973$_{\pm 0.003}$	0.975$_{\pm 0.003}$	0.915$_{\pm 0.002}$	0.847$_{\pm 0.003}$
Informer-based fusion	**Second-order**$_{\text{(Euclidean)}}$	0.974$_{\pm 0.002}$	0.974$_{\pm 0.001}$	0.911$_{\pm 0.002}$	0.842$_{\pm 0.001}$
Ours	**Second-order**$_{\text{(Riemannian)}}$	**0.983**$_{\pm 0.001}$	**0.982**$_{\pm 0.002}$	**0.937**$_{\pm 0.001}$	**0.868**$_{\pm 0.002}$

Table 4: A comparison of different methods in predicting the drug-target binding regions, evaluated in terms of accuracy.

Dataset	**Methods**	S=5	S=10	S=15	sc-PDB
Davis	DeepCDA	0.166	0.257	0.339	0.191
	TransformerCPI	0.085	0.213	0.289	0.237
	CPInformer	0.101	0.158	0.318	0.213
	MFR-DTA	0.174	0.459	0.968	0.621
	PSC-CPI	0.253	0.541	0.971	0.643
	R-DTI	**0.313**	**0.598**	**0.979**	**0.703**
KIBA	DeepCDA	0.325	0.472	0.501	0.151
	TransformerCPI	0.287	0.374	0.419	0.196
	CPInformer	0.301	0.488	0.529	0.218
	MFR-DTA	0.651	0.902	0.942	0.513
	PSC-CPI	0.693	0.910	0.951	0.556
	R-DTI	**0.757**	**0.932**	**0.963**	**0.621**

where x is the difference in values between the predicted and target coordinates.

The second step is to train the R-DTI model, which produces two outputs, the results of the classifier and the auxiliary. We calculate the two losses by the binary cross-entropy loss function, which is formulated as:

$$\mathcal{L}_b = \frac{1}{N}\sum_i - \left[y_i \cdot \log\left(p_i\right) + \left(1-y_i\right) \cdot \log\left(1-p_i\right)\right], \tag{7}$$

where y_i is the label of sample i, p_i denotes the probability that sample i is predicted to be a positive class.

Our method was implemented in Python 3.8 with PyTorch 1.8.0. The experiments were carried out on a machine with the Ubuntu 20.04 operating system, Intel Core i7-11700K CPU and one NVIDIA GeForce RTX 3090 card. We used the AdamW optimizer (Loshchilov and Hutter 2018) for network training. In the training process, we set the learning rate as 5×10^{-4}, the weight decay as 7×10^{-2}, the batch size, as 16, and the dropout ratio as 0.1. More hyperparameter settings are listed in Table 1.

We use the Area Under Receiver Operating Characteristic Curve (AUC-ROC) and Area Under Precision-Recall Curve (AUC-PR) as our main metrics to evaluate DTI tasks, consistent with the previous work (Hua et al. 2023b). Besides, we also evaluate the proposed model in DTA tasks by using the Concordance Index (CI) ((Gönen and Heller 2005)) and Mean Squared Error (MSE), consistent with the previous work (Hua et al. 2023a). Specifically, CI is calculated as $CI = \frac{1}{Z}\sum_{\delta_i > \delta_j} h\left(b_i - b_j\right)$, where δ_i and δ_j are the label values of the affinity of two samples. δ_i is larger than δ_j. b_i is the prediction value of the ith sample, b_j is the prediction value of the jth sample, Z is the normalised constant, and $h(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0.5, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases}$ is the step function.

Appendix C: Statistics of the benchmarks

We present the statistics of the six benchmarking DTI datasets and a 3D dataset, sc-PDB, in Table 2. In particular, sc-PDB (Gaber, Rashad, and Fathy 2019) is used to evaluate the performance of DTI methods in predicting drug-target binding regions, and the relevant experiments are reported in ***Appendix E***.

Appendix D: Ablation Study Results for Prediction Formulation Methods

To validate our proposed Riemannian classifier, we compare the performance of different prediction formulation methods that capture different relevance of DTI features. We validate the robustness of these methods on both the BindingDB and Davis datasets under two imbalance conditions, the imbalance in the number of proteins and drugs, and the imbalance in the number of positive and negative sample pairs. Note that all experimental results are based on the same framework, which uses MLP to predict DTI based on the extracted relevance of DTI features. As shown in Table 3, concatenation is a simple method to combine protein and drug, and its performance can be further improved together with Convolutional Neural Networks (CNN). However, concatenation-based first-order relevance extraction methods are worse than second-order relevance extraction methods when the number of drugs is much higher than the number of proteins in the BindingDB dataset. This observation confirms our previous argument that the first-order relevance of the DTI feature is used to bias the protein, and its performance will rapidly decrease when the number of proteins and drugs is unbalanced. In contrast, when the positive and negative samples are unbalanced, the existing second-order methods do not have much advantage, and their performance is limited to the unstable semantic information of the DTI features. Our proposed method addresses the above two issues effectively by compensating the bias of the DTI feature and making its semantic information more stable. Therefore, our method against the second best feature joint method, outperforming 0.9% and 2.2% in terms of AUC-ROC on the BindingDB and Davis datasets, respectively.

Appendix E: Comparison for Predicting Drug-target Binding Regions

In this section, we introduce binding region information for training the R-DTI model to improve its functionality. Specifically, we obtain the protein-drug relevance, $\mathbf{F}_{relevance} \in \mathbb{R}^{L_p \times L_d}$, from the length-based Riemannian feature, $\mathbf{F}_{rL} \in \mathbb{R}^{(L_p+L_d)\times(L_p+L_d)}$. Then the full connected layer is used to predict binding regions, denoted as $\mathbf{P}_{region} \in \mathbb{R}^{L_p}$. Following the previous methods (Hua et al. 2023a), we adopt the Rectified Wing (RWing) loss function (Feng et al. 2020) to train the model for binding region prediction. The loss function is defined as:

$$RWing(x) = \begin{cases} 0 & \text{if } |x| < r \\ w \ln(1 + (|x| - r)/\epsilon) & \text{if } r \leq |x| < w \\ |x| - C & \text{otherwise} \end{cases} \tag{8}$$

where the non-negative parameter r sets the range of rectified region to $(-r, r)$ for very small values. For small-medium range values with the absolute value in $[r, w)$, RWing uses a modified logarithm function, where ϵ limits the curvature of the nonlinear region and $C = w - w\ln(1 + (|x| - r)/\epsilon)$ is a constant that smoothly links the linear and nonlinear parts.

During the testing, we take the probability of the actual binding site falling into the prediction regions as the metric to measure the accuracy of these approaches. The length of the prediction region is S amino acid elements, and the midpoint is where with the highest value in the drug-target response vector. We measure the prediction regions of different methods according to three scales (S = 5, 10, and 15). These approaches are evaluated on three datasets, including two DTA benchmarks, Davis and KIBA, and a 3D dataset, sc-PDB (Gaber, Rashad, and Fathy 2019). We use the sc-PDB dataset, which contains drug-target binding sites, to evaluate these models in predicting unseen drug-target binding regions (testing only, no training) with the scale S = 15. We compare R-DTI with the existing approaches in predicting drug-target binding regions and report the experimental results in the Table 4. Notably, the experimental results of the first four methods, DeepCDA, TransformerCPI, CPInformer, and MFR-DTI, in the table are excerpted from previous work (Hua et al. 2023a), and we supplement the results of PSC-CPI and R-DTI. Obviously, our method achieves the best performance in all size conditions on three datasets, validating the accuracy and robustness of using second-order Riemannian relevance to predict binding sites.

References

Arsigny, V.; Fillard, P.; Pennec, X.; and Ayache, N. 2007. Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1): 328–347.

Feng, Z.-H.; Kittler, J.; Awais, M.; and Wu, X.-J. 2020. Rectified wing loss for efficient and robust facial landmark localisation with convolutional neural networks. *International Journal of Computer Vision*, 128(8): 2126–2145.

Gaber, Y.; Rashad, B.; and Fathy, E. 2019. Biological 3D Structural Databases. In *Essentials of Bioinformatics, Volume I*, 47–73. Springer.

Gönen, M.; and Heller, G. 2005. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4): 965–970.

Hua, Y.; Song, X.; Feng, Z.; and Wu, X. 2023a. MFR-DTA: a multi-functional and robust model for predicting drug–target binding affinity and region. *Bioinformatics*, 39(2): btad056.

Hua, Y.; Song, X.; Feng, Z.; Wu, X.-J.; Kittler, J.; and Yu, D.-J. 2023b. CPInformer for Efficient and Robust Compound-Protein Interaction Prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(1): 285–296.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 1–18.