**REINDEER: A Protein-Ligand Feature Generator Software for Machine Learning Algorithms**

Milad Rayka1\*

1. Unaffiliated

Corresponding author:

Milad Rayka, milad.rayka@yahoo.com

**Abstract**

**Keywords:** Cheminformatics, Protein-Ligand Complex, Software, Machine Learning, Feature Engineering

1- Introduction

In the past decade, we have witnessed a proliferation of data-driven approaches for designing protein-ligand scoring functions, which scientists apply to predict the binding affinity of a protein-ligand complex. These new scoring functions employ traditional machine learning (ML) and deep learning algorithms (DL) for deriving a non-linear relationship between binding affinity quantity and a representation of the protein-ligand complex [1-9].

A designed scoring function can be evaluated based on its ability to perform four tasks: scoring, ranking, docking, and screening. The first two tasks involve predicting binding affinity values, while the other two tasks evaluate the scoring function's ability to distinguish between native or near-native poses and decoys, as well as true binders from non-binders [10, 11]. The PDBbind dataset [12], DUD-E [13], and DEKOIS2.0 [14] are among the popular datasets for training and testing, in spite of the fact that there are some hidden biases in these datasets [15, 16].

In the ML case, it is needed to represent a protein-ligand complex in terms of a feature vector by applying feature engineering techniques, while in the DL case, these representations are mostly learned end-to-end during the training phase.

RF-Score [17], ECIF∷LD-GBT [18], and multi-shelled ECIF [19] are ML-based scoring functions that exploit the occurrence of interatomic contacts and ligand descriptors in ECIF∷LD-GBT case to generate a feature vector for ML algorithms like Random Forest (RF) or Gradient Boosting Decision Trees (GBT). ET-Score [20] and its’ enhanced versions, i.e., GB-Score [21] and ENS-Score [22], employed distance-weighted interatomic contacts and Extremely Randomized Trees (ERT) for constructing an ML-based scoring function. ∆vinaRF20 [23], ∆vinaXGB [24] and ∆LinF9XGB [25] employ some Autodock Vina empirical terms alongside other descriptors, e.g., RDKit descriptors for ligand, for feature generation procedures, and RF and eXtreme Gradient Boosting (XGB) algorithms as learners. Proteo-chemometrics IFPs [26] and PLEC-FP [27] apply different generalized versions of extended-connectivity fingerprints (ECFPs) for protein-ligand representation. SMPLIP-Score [28] utilizes a featurization method to vectorize and embed the interaction fingerprint pattern between the ligand‑binding site environment and fragments of ligands. TB-IECS [29] employs some theory-based energy terms, e.g., van der Waals, for making a representation for an XGB algorithm.

Tomorrow add Topology-based methods

Next day add DL methods

Emphasis of ML-based

Lack of package to feature engineering

Comparing to existence one, use table, ML-PLIC paper

Reason for selection of these methods, exclusion of enery term based like DeltaVinaRF20, new, geometry-based

Last paragraph of introduction

2- Theory

Occurrence of Interatomic Contact

Occurrence of interatomic contact (OIC) was introduced by Ballester et al. for developing the RF-Score scoring function [17]. In this technique, the authors represented a protein-ligand complex by counting the number of occurrences of a specific pair of protein and ligand atoms below a distance threshold. Ten elemental atom types (H, C, O, N, F, P, S, Cl, Br, and I) were allocated for protein and ligand , although in its original implementation, the Hydrogen element was omitted. The following formula calculates the occurrence (1):

|  |  |
| --- | --- |
|  | (1) |

Where xi,j is the number of contacts between i and j atom types. k and l are protein and ligand atoms belonging to the i and j atom types. dkl is the Euclidean distance between k and l atoms, and Θ is the Heaviside step function that counts contacts below dcutoff=12 Å. The feature vector contains 50 integer-valued features.

Distance-Weighted Interatomic Contact

Distance-weighted interatomic contact (DWIC) was employed by ET-Score [20] for vectorial representation of a protein-ligand complex and further used by GB-Score [21] and ENS-Score [22]. Like OIC, ten elemental atom types are chosen for both ligand and protein. However, protein atom types are further augmented by considering the nature of amino acid side chains. To reflect the different characteristics of amino acids, they are classified into four groups (Charged (c), Polar (p), Amphipathic (a), and Hydrophobic (h)):

Charged={Arg, Lys, Asp, Glu}

Polar={Gln, Asn, His, Ser, Thr, Cys}

Amphipathic={Trp, Tyr, Met}

Hydrophobic={Ile, Leu, Phe, Val, Pro, Gly, Ala}

Therefore, each elemental protein atom type belongs to four groups. As an example, CP denotes a carbon atom of polar residues. At its’ core, DWIC is similar to OIC in the feature generation by considering atom types pair of protein and ligand, but, in DWIC, the Heaviside step function is replaced by a function, which differentiated close and distant interatomic contacts by applying an inverse-square factor. The following equation (2) describes this function:

|  |  |
| --- | --- |
|  | (2) |

The definition of symbols is similar to equation (1), and like OIC, dcutoff=12Å is applied. The DIWC feature vector includes 200 float-valued features.

Extended Connectivity Interaction Features

Extended Connectivity Interaction Features, or concisely ECIF, share the same essence, like DIWC, with OIC [18]. However, ECIF applies different atom type representations and dcutoff value (6Å) and only retains the counting atom type pairs scheme of OIC. The authors defined atom types in ECIF by considering the atom environment. Atom symbol, explicit valence, number of attached heavy atoms, number of attached hydrogens, aromaticity, and ring membership were employed for this definition, which results in 22 and 70 atom types for protein and ligand, respectively. The final feature vector comprises 1540 integer-valued features.

Proteo-Chemometrics Interaction Fingerprints

Proteo-chemometrics interaction fingerprints (PrtCmm IFPs) method combines ECFPs with proteo-chemometrics approaches. In this procedure, protein and ligand are separately characterized for the construction of a predictive model. Despite the PLEC FPs technique, which generates fingerprints by considering and hashing atom pairs in protein-ligand complex, the current method generates fingerprints for ligand and protein separately and concats them together. Also, the PrtCmm IFPs method applies a novel noniterative (NI) ECFP algorithm to remove the information collision possibility and nested-hash operations. This new algorithm hashes the average properties of atoms in each neighborhood for making identifiers. It is worth mentioning that atomic mass, total number of connections, number of heavy-atom neighbors, number of attached hydrogen atoms, and formal charge are considered properties of an atom {Wang, 2021 #28}. The current implementation of PrtCmm IFPs in REINDEER uses R1=1 and R2=1 (the same radius for ligand and protein), 29-bit string, and NI ECFP for making fingerprints.

decode protein–ligand binding structures as binary strings where each bit corresponds to a specific interaction type

Proteo-chemometrics approaches (Lapinsh et al., 2001), which separately characterize the protein (primarily the binding site) and ligand in the construction of predictive models, are an alternative approach. Based on the proteo-chemometrics concept, we have developed ECFPs-based, proteo-chemometrics IFPs for protein–ligand complexes.

PLEC FPs were developed based on ECFPs, and proceed

from interacting atom pairs in protein–ligand binding complexes.

They start from searching for interacting atom pairs (e.g. distance < 4:5A) in the protein–ligand interfaces and iteratively hash the neighborhoods of the two atoms in each pair to integers, and then fold the virtual string. PLEC FPs account for multiple depth combinations

On the basis of the proteo-chemometrics concept and the ECFP routines, we developed a proteo-chemometrics IFP (PrtCmm IFP)

The conventional ECFP algorithm (Algorithm 1) starts from hashing the information of all heavy atoms to identifiers, and then iteratively updates the identifier list by hashing the information of atom neighborhoods toward a given radius. This involves large-scale nestedhash operations and increases the possibility of information collision. To mitigate this and simplify the procedure, we develop a noniterative (NI) ECFP algorithm that removes the nested-hash operations, but directly hashes the average properties of atoms in each neighborhood to identifiers.

atomic mass, total number of connections, number of heavy-atom neighbors, number of attached hydrogen atoms and formal charge

PrtCmm

IFPNI: ½R1; R2 ¼ ½1; 1, l=9

3- Method

4- Case Study

5- Conclusion

6- References

1. Ain QU, Aleksandrova A, Roessler FD, Ballester PJ. Wiley Interdisciplinary Reviews: Computational Molecular Science 5, 6 (2015): 405-424.

2. Colwell LJ. Current opinion in structural biology 49, (2018): 123-128.

3. Shen C, Ding J, Wang Z, Cao D, Ding X, Hou T. Wiley Interdisciplinary Reviews: Computational Molecular Science 10, 1 (2020): e1429.

4. Li H, Sze KH, Lu G, Ballester PJ. Wiley Interdisciplinary Reviews: Computational Molecular Science 11, 1 (2021): e1478.

5. Pellicani F, Dal Ben D, Perali A, Pilati S. Molecules 28, 4 (2023): 1661.

6. Yang C, Chen EA, Zhang Y. Molecules 27, 14 (2022): 4568.

7. Schapin N, Majewski M, Varela-Rial A, Arroniz C, De Fabritiis G. Artificial Intelligence Chemistry (2023): 100020.

8. Hagg A, Kirschner KN. Journal of Chemical Information and Modeling 63, 15 (2023): 4505-4532.

9. Zhang Y, Li S, Meng K, Sun S. Journal of Chemical Information and Modeling (2024).

10. Li Y, Liu Z, Li J, Han L, Liu J, Zhao Z, Wang R. Journal of chemical information and modeling 54, 6 (2014): 1700-1716.

11. Li Y, Han L, Liu Z, Wang R. Journal of chemical information and modeling 54, 6 (2014): 1717-1736.

12. Su M, Yang Q, Du Y, Feng G, Liu Z, Li Y, Wang R. Journal of chemical information and modeling 59, 2 (2018): 895-913.

13. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Journal of medicinal chemistry 55, 14 (2012): 6582-6594.

14. Bauer MR, Ibrahim TM, Vogel SM, Boeckler FM. Journal of chemical information and modeling 53, 6 (2013): 1447-1462.

15. Shen C, Hu Y, Wang Z, Zhang X, Pang J, Wang G, Zhong H, Xu L, Cao D, Hou T. Briefings in Bioinformatics 22, 3 (2021): bbaa070.

16. Li J, Guan X, Zhang O, Sun K, Wang Y, Bagni D, Head-Gordon T. ArXiv (2023).

17. Ballester PJ, Mitchell JB. Bioinformatics 26, 9 (2010): 1169-1175.

18. Sánchez-Cruz N, Medina-Franco JL, Mestres J, Barril X. Bioinformatics 37, 10 (2021): 1376-1382.

19. Shiota K, Akutsu T. Bioinformatics Advances 3, 1 (2023): vbad155.

20. Rayka M, Karimi‐Jafari MH, Firouzi R. Molecular Informatics 40, 8 (2021): 2060084.

21. Rayka M, Firouzi R. Molecular Informatics 42, 3 (2023): 2200135.

22. Rayka M, Mirzaei M, Mohammad Latifi A. Molecular Informatics (2024): e202300292.

23. Wang C, Zhang Y. Journal of computational chemistry 38, 3 (2017): 169-177.

24. Lu J, Hou X, Wang C, Zhang Y. Journal of chemical information and modeling 59, 11 (2019): 4540-4549.

25. Yang C, Zhang Y. Journal of chemical information and modeling 62, 11 (2022): 2696-2712.

26. Wang DD, Xie H, Yan H. Bioinformatics 37, 17 (2021): 2570-2579.

27. Wójcikowski M, Kukiełka M, Stepniewska-Dziubinska MM, Siedlecki P. Bioinformatics 35, 8 (2019): 1334-1341.

28. Kumar S, Kim M-h. Journal of cheminformatics 13, (2021): 1-17.

29. Zhang X, Shen C, Jiang D, Zhang J, Ye Q, Xu L, Hou T, Pan P, Kang Y. Journal of Cheminformatics 15, 1 (2023): 63.