Title: Q.E.D's solution to IDG-DREAM Drug-Kinase Binding Prediction Challenge

Authors

Fangping Wan[1, 4] (Username: Stig, Email: wfp15@mails.tsinghua.edu.cn), Shuya Li[1, 4] (Username: Shuya Li, Email: lishuya17@mails.tsinghua.edu.cn), Yunan Luo [2](Username: Yunan Luo, Email: yunan@illinois.com), Hailin Hu[3, 4] (Username: Corgi, Email: huhl16@mails.tsinghua.edu.cn), Jian Peng[2] (Email: jianpeng@illinois.edu), Jianyang Zeng[1, 4] (Email: zengjy321@tsinghua.edu.cn)

1: Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China.
2: Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, USA.
3: School of Medicine, Tsinghua University, Beijing, China.
4: Silexon Co., LTD, Beijing, China.

1. Summary

This project contains the submission 2 (objectID 9686285) method from Q.E.D team, for both sub-challenges of IDG-DREAM Drug-Kinase Binding Prediction Challenge round2.

2 .Methods - submission 2 (objectID 9686285)

2.1 Data pre-processing

Generally, we used the compound-protein affinity data from Drug Target Commons (DTC) (Tang J, Ravikumar B, Alam Z, et al. DrugTargetCommons:a community effort to build a consensus knowledgebase for drug-target interactions. Cell Chemical Biology, 2018, 25(2):224-229.e2.) (downloaded from https://www.synapse.org/#!Synapse:syn17017461).

2.1.1 Used compounds:

Among the compounds curated in DTC, we only considered compounds that (1) have CHEMBL ID or (2) have PKD affinity. Compounds that have CHEMBL ID have structure information encoded by InChI or SMILES (stored in DTC). Compounds that have PKD affinity are encoded as InChIkey in DTC. We used PubChem Identifier Exchange Service (https://pubchem.ncbi.nlm.nih.gov/idexchange/idexchange.cgi) to map InChIkey to corresponding SMILES (stored as "DTC_pkd_inchikey_to_smiles" in the data folder in docker).

2.1.2 Used proteins:

Among the proteins curated in DTC, we only considered proteins that (1) have Uniprot ID and were labeled as kinase in Uniprot or (2) came from round_2_template.csv (i.e., test compounds, downloaded from https://www.synapse.org/#!Synapse:syn16809885).

2.1.3 Used affinities:

Among the binding affinity pairs curated in DTC, we only considered the pairs that satisfied the following conditions:

(1) compounds that came from 2.1.1; (2) proteins that came from 2.1.2; (3) the binding affinity measurement type was Ki or KI or Kd or KD or EC50 or PKD; (4) the binding affinity measurement relation was "=" (i.e., equal); (5) if the binding affinity measurement type was Ki or KI or Kd or KD or EC50, the standard unit should be "NM"; (6) In DTC dataset, each row represents a binding affinity, the "target id" column (i.e., the fifth column) should contain only one Uniprot ID, otherwise (e.g., multiple Uniprot IDs) we did not use this row.

We used the binding affinity pairs that satisfied the above conditions. Then, for the pairs that had the type Ki or KI or Kd or KD or EC50, we converted the binding affinity x using the transformation: $-\log_{10}(x / 10^9)$. For pairs that had the type PKD, we did not do such transformation. Some compound-protein pairs can have multiple binding affinities. In such cases, the median of the binding affinities was used. Note that, the median operation was applied after the $-\log_{10}(x / 10^9)$ transformation.

After the above pre-processing, the total number of compounds we used in training process is 13,608; the total number of proteins we used in training process is 527; and the total number of binding affinities we used in training process is 60,462.

2.2 Prediction method

Under the problem setting that requires fine-grained discrimination between similar compounds or targets, we found the explicit introduction of similarity metrics as model input generally outperformed other more complex model architectures that attempt to organize representative features from scratch. In particular, we defined a comprehensive set of compound structure similarity and protein sequence similarity metrics as the input of our model. Then, we leveraged CGKronRLS method (Pahikkala T. Fast gradient computation for learning with tensor product kernels and sparse training labels[C]//Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg, 2014: 123-132.) (implemented in Pahikkala T, Airola A. RLScore: regularized least-squares learners[J]. The Journal of Machine Learning Research, 2016, 17(1): 7803-7807. https://github.com/aatapa/RLScore) as the regression model to predict the binding affinity.

2.2.1 Features of compounds

We computed the following compound similarity matrices as compound features (computed by RDKit: https://github.com/rdkit/rdkit):

1: Tanimoto similarity of morgan fingerprint with arguments radius=2, nBits=1024, useChirality=True.

2: Tanimoto similarity of morgan fingerprint with arguments radius=2, nBits=1024, useChirality=False.

3: Tanimoto similarity of morgan fingerprint with arguments radius=3, nBits=1024, useChirality=True.

4: Tanimoto similarity of morgan fingerprint with arguments radius=3, nBits=1024, useChirality=False.

5: Dice similarity of morgan fingerprint with arguments radius=2, nBits=1024, useChirality=True.

6: Dice similarity of morgan fingerprint with arguments radius=2, nBits=1024, useChirality=False.

7: Dice similarity of morgan fingerprint with arguments radius=3, nBits=1024, useChirality=True.

8: Dice similarity of morgan fingerprint with arguments radius=3, nBits=1024, useChirality=False.

2.2.2 Features of proteins

We computed the protein similarity matrix as protein features (computed by https://github.com/mengyao/Complete-Striped-Smith-Waterman-Library). Specifically, protein similarity is defined as the normalized Striped-Smith-Waterman similarity. Let sw(s1, s2) be the alignment score of Striped-Smith-Waterman algorithm between protein sequences s1 and s2. The protein similarity between s1 and s2 can be defined as sw(s1, s2) / sqrt(sw(s1, s1) * sw(s2, s2)).

2.2.3 Regression model

We used CGKronRLS as the regression model. It took the compound and protein similarity matrices as input and output the binding affinity.

2.2.4 Model ensemble

Instead of using single model, we used the ensemble of multiple CGKronRLS (with different iterations, regularization parameters and input features) models. We ensembled 440 CGKronRLS models with the following setting: protein feature ∈ {the protein similarity matrix from 2.2.2} x compound feature ∈ {eight compound similarity matrices from 2.2.1} x regularization parameter of CGKronRLS ∈ {0.1, 0.5, 1.0, 1.5, 2.0} x iteration of CGKronRLS ∈ {400, 410, 420, 430, 440 ,450, 460, 470, 480, 490, 500}, where x means cartesian product.

After training the 440 CGKronRLS models, we averaged the predictions among them to produce the final prediction.

3. Testing environment

Submission 2 (objectID 9686285) model was trained and tested on a server with the following configuration: System version: Ubuntu 16.04.2 LTS; Cores: Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz, 32 in total; Memory: 264024700 kB.

4. Running the final model

To run our docker image and get only the final output file for our submission 2 (objectID 9686285), run the following command:

```
$ docker run -it --rm -v ${PWD}/input:/input -v ${PWD}/output:/output
docker.synapse.org/syn18519352/qed-sub2:9686285
```

To run our docker image and get all the output files of intermediate processes for our submission 2 (objectID 9686285), run the following command:

```
$ docker run -it --rm -v ${PWD}/input:/input -v ${PWD}/output:/output -v
${PWD}/data:/data -v ${PWD}/SW_based_prediction:/SW_based_prediction
docker.synapse.org/syn18519352/qed-sub2:9686285
```

5. Author contribution

F.W., S.L., Y.L., H.H., J.P., and J.Z. conceived the method. F.W. and S.L. conducted the data pre-processing. F.W., S.L., Y.L. and H.H. calculated the features. F.W. and S.L. wrote and ran the regression model. S.L. prepared the docker with the help of F.W. F.W. wrote the writeup with support from all authors.