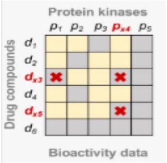
IDG-DREAM Drug-Kinase Binding Prediction Challenge

Mayukh Majumdar
maymaj@stanford.edu
Advisor : David A Knowles (dak33@Stanford.edu)

Introduction

The bioactivity of all compounds would be great to know for the therapeutic potential of drugs as well as to predict their usage or manage bad effects of their use.

Protein kinase are kinase enzymes that modify proteins by chemically adding phosphate groups to them. Human genome has 518 protein kinase genes. 30% of all human proteins maybe modified by kinase activity.



As part of this Challenge, we are trying to predict which of the drugs will pair with which protein kinase. This deep learning based prediction model would provide a systematic way to reduce drug discovery time.

DataSet

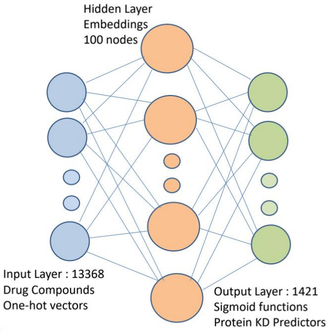
Data provided by the Challenge makes use of open-data web-platform, DrugTargetCommons, available at : https://drugtargetcommons.fimm.fi. The original data was a 1.6GB .csv file, having 1746997 compounds and 13023 protein targets. The file was parsed to extract only the relevant lines with KD affinity values available.

Final Dataset only had KD pair values for 13368 unique drug compounds paired with 1421 unique protein kinases. Affinity vectors for training of Drug->Protein and Protein->Drug mappings were created from this dataset. Protein->Drug mappings were more uniform while Drug->Protein were spiky. The range of KD values was quite large : from 0.008 to 70000.

Typical Data Format
Compound_id standard_inchi_key standard_type standard_relation standard_value. assay_format assay_subtype target_id

```
CHEMBL3545284 NA KD = 19155.14 cell_free binding_reversible Q9Y4K4
CHEMBL3545284 NA KD = 1565.72 cell_free binding_reversible Q9Y478
CHEMBL3545284 NA KD = 746.77 cell_free binding_reversible Q9Y2U5
CHEMBL3545284 NA KD = 13558.67 cell_free binding_reversible Q9Y2K2
```

Model – Embeddings



1. Affinities between drug compounds and protein kinase analogous to movie-user pairs; recommender system techniques are a natural fit.
2. Embeddings were an obvious solution to the problem statement.
3. Input layer provides one-hot vectors for all drug compounds in database.
4. Output layer is trained to the known KD affinity for each protein kinase.
5. A single hidden layer of 100 nodes provides the embeddings to be learnt.
6. Training was provided by affinity vectors for proteins for each drug input.
7. L2 loss function used : MSE = 1/n(Sum over all i[$(y_i – y_i^p)^2$])
8. After training, given an input drug compound as a one-hot vector input output would provide KD prediction for all proteins given.
9. Known drug-protein KD affinities would match the ground truth.
10. Two models derived from dataset :
    A. Drug Compounds as Input and Protein Kinases as Output.
    B. Protein Kinases as input and Drug Compounds as Outputs.

Discussion

1. Given that the number of drug compounds were 10x of protein kinases, the mapping of protein->drug compounds was denser.
2. The sigmoid for output layers would help predict the ground truth for KD values directly.
3. Normalization so that all affinity add up to 1 would become softmax outputs. Then we would need another network to use the embedding as inputs to predict the affinities.
4. Better normalization could be :
   - normalize affinity of each drug compound so that they sum to 1
   - normalize all KD values to a standard normal distribution.
5. Multiple hidden layers could possibly provide better results.
6. L2 loss was chosen as it is a continuous loss function for regression. Usually affected by outliers.
7. Need to check results with Huber Loss or Log-Cosh Loss functions.

Results

Model	Train RMSE (Samples)	Test RMSE (Samples)
Drug to Protein	0.6823 (12031)	1.835 (430)
Protein to Drug	0.8659 (12031)	1.746 (430)

Future Work

- Changing output to softmax; adding another network to take embeddings as input.
- Other algorithms for Recommendation Systems like Collaborative Filtering (CF).
- Extract more correlations available from the dataset like IC50 – which is unused now.

References

1. IDG-DREAM Drug-Kinase Binding Prediction Challenge - Wiki Page.
2.https://www.tensorflow.org/tutorials/representation/word2vec
3.https://developers.google.com/machine-learning/crashcourse/embeddings/video-lecture
4. Greg Linden, et al "Amazon.com Recommendations", IEEE Internet Computing.
5. Netflix Recommender System. https://www.wired.co.uk/