Deep Learning Network for activity prediction of PROTACs

**Runshi Yang, Kevin Umaiyalan, Evianne Rovers**

**Abstract**

For certain diseases, like cancer and autoimmune disorders, accumulation of proteins in human cells are at the source for the development of the disease. Proteolysis Targeting Chimeras (PROTACs) are drugs that target proteins for degradation in the cells. Therefore, PROTAC is a therapeutic option to treat diseases with overproduction of proteins. The design of PROTACs is a trial-and-error process with limited known computational methods to aid in the design process. DeepPROTACs is the first machine learning based method to screen PROTACs for activity. The model has an accuracy of 78% on the PROTAC-database and 68% on a test case study of unseen data. The accuracy leaves room for improvement in prediction accuracy on both the PROTAC-database and test case study. Additionally, the neural network does not consider (graph) attention layers in the network which could improve the accuracy. Therefore, the project aimed to recreate the DeepPROTACs model and develop a new deep learning model that includes attention layers to improve the accuracy of prediction.

**Introduction and related works**

The production and degradation of proteins in cells - cellular homeostasis – is a critical process often deregulated in disease. It is controlled by post-translational modifications that include the addition and removal of markers attached to proteins that serve as flags for downstream processes. For example, ubiquitination - the addition of a ubiquitin tag to a protein - can flag a protein for subsequent degradation. Certain diseases, including cancer, autoimmune diseases, and neurodegenerative diseases are associated with increased protein levels in the cells1,2. Adding the ubiquitin tag to these proteins to target the proteins for degradation could have a therapeutic benefit. E3 ligases which are part of the degradation system can be brought in close proximity to target proteins by compounds that simultaneously engage both proteins. These compounds are called Proteolysis Targeting Chimeras (PROTACs) and consist of two small molecules - each binding to one protein - and a linker that connects these small molecules. Many PROTACs have been designed to prove this concept with 10 PROTACs reaching clinical trials (e.g. clinicaltrials.gov identifiers NCT03888612, NCT04772885, NCT04830137)1. However, the design of the PROTACs have been mostly a trial-and-error process with limited computational rational design. The limiting factor in PROTAC design is determining linker length and composition. A structural study showed that efficacy of PROTACs is determined by their ability to induce at least one stable ternary complex which is determined by the size, composition and flexibility of the linker3. It is not known whether the PROTAC only induces one conformation or can induce multiple conformations to induce degradation. Therefore, it is difficult to determine which predicted ternary complex is the correct one and prospectively which PROTAC can induce the correct conformation. Current methods to screen for new PROTACs mostly consist of screening for all possible ternary complexes and the PROTAC that can induce most complexes is predicted to be most active4–6.

Machine learning can detect patterns in data that cannot be seen by the human eye. Therefore, machine learning models can potentially detect the patterns in the 3D complexes of the ternary complexes (E2-PROTAC-Target) that could elucidate the mechanism of active PROTACs versus non-active PROTACs. DeepPROTAC is the first deep-learning neural network that uses deep learning to classify PROTACs for activity based on the information of the individual proteins and the PROTAC7. This model has an accuracy of 78%. The model does not take into consideration the information that can be extracted from these ternary complexes about the interactions between the components, such as hydrogen bonds, shape complementarity, number of residues in the interface, etc. which can determine how well the proteins and PROTAC interact with one another. Therefore, we propose to create a neural network which is trained on a dataset containing information about the ternary complex interactions for PROTACs listed in the PROTAC-database8,9, which will feed into the DeepPROTAC model to improve accuracy. We will test the ability to reproduce the results obtained in the DeepPROTAC paper and see whether including information about the interactions of the components can increase the accuracy.

**Methods**

*Dataset creation*

PROTACs are extracted from the PROTAC-db, a database listing all the PROTACs currently developed and their activity profile. The number of PROTACs available in this database is ???. The PROTACs are matched up to corresponding 3D crystal structures from the Protein Data Bank [ref]. After curation of the database for the presence of 3D crystal structures, the database for training consists of 949 PROTACS. Using ICM [Molsoft, San Diego], the 3D crystal structures of the proteins were prepared to remove non-relevant atoms and add missing atoms. The graph representation of the proteins and PROTAC were generated using PyMol [ref] and OpenBabel [ref] respectively. Each PROTAC is labelled 0 for ‘non-active’ if degradation efficacy is <80% or >100nM, otherwise labelled 1 for ‘active’ if degradation efficacy is >80% or <100nM. The database did not contain information about degradation efficacy for every PROTAC, therefore the database was manually labelled using the descriptions for the PROTACs in the corresponding papers. The number of data points is limited by the availability of 3D crystal structures for the proteins and data augmentation is not possible due to the nature of the data. Adding atoms or removing atoms could potentially lead to active or inactive PROTACs resulting in adding noise to the model rather than increasing the database. The database ratio of active/total is 676/949 making the data heavily skewed towards active PROTACs.

*Reproduction of DeepPROTAC model*

First, we want to reproduce the results obtained in the DeepPROTAC model including the hyperparameter search. The github repository was downloaded and combinations of hyperparameters described in the publication were tested to validate the created database and to validate the ability to reproduce the results published.

*Creation of Deep Learning network*

Since DeepPROTAC separates all the components, we tested the hypothesis of minimising the separation of the components. We made a deep neural network beginning with 3 neural networks processing either the E3 protein, PROTAC or target protein. Rather than representing the PROTAC in the SMILES representation, we use the graph representation as this will provide information about atom bonds and types. Each neural network consists of an embedding layer, graph convolutional layer and graph attention layer. Afterwards, the output of the neural networks are concatenated and run through an attention layer followed by two fully connected layers. See visualisation of model architecture in Figure XX).

**Preliminaries**

*Graph convolutional layers*

XXX

*Graph attention layers*



Something like this?

<https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GATConv.html#torch_geometric.nn.conv.GATConv>

**Results**

*Reproduction of DeepPROTACs model*

Xxxx

*Creation of new deep learning model*

XXX

Final\_model\_b64\_l0.1\_w0.0001: INFO:root:Train epoch 91, loss: 316.3803, val\_loss: 54.0050, val\_acc: 0.6421, val\_auroc: 0.6310

Final\_model\_b64\_l0.01\_w0.0001: INFO:root:Train epoch 63, loss: 0.4888, val\_loss: 0.5009, val\_acc: 0.7474, val\_auroc: 0.7789

Final\_model\_b64\_l0.001\_w0.0001: INFO:root:Train epoch 88, loss: 0.3570, val\_loss: 0.3965, val\_acc: 0.8053, val\_auroc: 0.8887

Final\_model\_b64\_l0.0001\_w0.0001: INFO:root:Train epoch 285, loss: 0.4013, val\_loss: 0.4384, val\_acc: 0.7842, val\_auroc: 0.8487

final\_model\_b128\_l0.1\_w0.0001: INFO:root:Train epoch 16, loss: 220270.2279, val\_loss: 7219.7847, val\_acc: 0.3105, val\_auroc: 0.5000

Final\_model\_b128\_l0.01\_w0.0001: INFO:root:Train epoch 26, loss: 0.5534, val\_loss: 0.5546, val\_acc: 0.7053, val\_auroc: 0.6916

Final\_model\_b128\_l0.001\_w0.0001: INFO:root:Train epoch 104, loss: 0.3716, val\_loss: 0.4496, val\_acc: 0.7789, val\_auroc: 0.8378

Final\_model\_b128\_l0.0001\_w0.0001: INFO:root:Train epoch 62, loss: 0.5285, val\_loss: 0.5114, val\_acc: 0.6789, val\_auroc: 0.7034

Final\_model\_b256\_l0.1\_w0.0001: INFO:root:Train epoch 9, loss: 4049.4972, val\_loss: 1269.1311, val\_acc: 0.6895, val\_auroc: 0.5000

Final\_model\_b256\_l0.01\_w0.0001: INFO:root:Train epoch 86, loss: 0.4494, val\_loss: 0.4519, val\_acc: 0.7947, val\_auroc: 0.8386

Final\_model\_b256\_l0.001\_w0.0001: INFO:root:Train epoch 208, loss: 0.4503, val\_loss: 0.4589, val\_acc: 0.7895, val\_auroc: 0.8371

final\_model\_b256\_l0.0001\_w0.0001: