FINAL REPORT MEET-EU 2023 / 2024

BALLESTRA Lorenzo   GUILHON Henri   DELATTRE Mathys
DESPRETZ Quentin

January 31, 2024

Contents

1 Introduction

Following the COVID-19 pandemic, there has been an urgent need to develop effective therapeutic interventions to mitigate the disease's impact on public health. The unprecedented challenges posed by the virus, such as its rapid transmission rates and the continuous emergence of new variants, highlight the importance of developing various therapies.

Disposing of effective medications can contribute to the containment of outbreaks, reduce the burden on healthcare systems and improve the conditions of the patients. Moreover, having a wide repertoire of antiviral drugs can be useful to address future coronaviruses or emerging diseases.

The non-structural protein 13 (nsp13) is an ATP-dependent helicase that plays a crucial role in viral replication. It has been identified as a promising target for antiviral drugs due to its high sequence conservation and pivotal role. This project aims to identify potential therapeutic molecules targeting Sars-Cov2's nsp13 using docking and machine learning tools. We will be focusing on the ATP pocket of nsp13 as a therapeutic target.

The process of filtering promising ligands is very time consuming. Indeed, docking every single ligand is computationally expensive. The aim of our project was therefore to develop a deep learning neural network taking ligands as inputs and predicting a score reflecting the capability of the ligand to occupy the ATP pocket and therefore to be an inhibitor of Nsp13. This would allow it to bypass the docking process and to rapidly perform screening on several databases of small molecules looking for hits.

2 Method

All analyzes were conducted on pre-selected ligands given in the pilot_library.csv containing some keys to databases and smiles sequences of ligands.

2.1 Docking

Autodock Vina has been used to compute the affinity of several ligands to the nsp13 ATP pocket. Vina's affinity score informs us about the strength of the bonding between the tested ligand and the ATP pocket. Starting from the crystallographic structure of nsp13 in pdb format, we first converted this file to pdbqt format adding charges information using rdkit.

The presence of Selenium atoms was problematic as these atoms are not treated by Autodock Vina (only few atom charges are taken in account). Since selenium atoms were only present in a few selenocystein amino acids, we fixed this problem by substituting them with methionine, the most similar amino acid in terms of structure and properties. This change was realised by substituting selenium atoms with sulphur atoms.

The ATP-pocket was identified by searching in the literature for nsp13 amino acids that experimentally interact with ATP. We then used those amino acids to define a search space for AutodockVina. Pymol was used to take the center of the pocket and set the dimensions of a box around it such that the whole pocket would fit inside it. The restriction box of the ATP pocket is bound to 30.30.30 Å .

The DNA/RNA pocket box search was bounded to 30.40.40 Å, this box is wider as it takes multiple and composed ligands such as double strand DNA/RNA. Such docking seemed to be much harder than the ATP pocket, thus leading the next searches towards the ATP pocket that is more convenient to study. Even if main searches were not focused on DNA/RNA pocket, ligands were nevertheless docked in both pockets, thus some fast observations could be done later if needed.

The first score results taken are the best affinities of each docked ligand. Those affinities will subsequently be used to train the neural network and to assess the potential of the ligand.

2.2 Creating graphs

Once affinity scores have been computed on enough ligands, a neural network will be trained with the goal to predict the affinity of other ligands. Graph representation of molecules has been chosen as input data for our network as it conveys more information. Using RDKIT [1], we were able to obtain the graph representations of the ligands using their SMILES formula. In the graphs, the atoms are represented as nodes and the bonds as edges. The parameters chosen for a single ligand are :

- atomic number of the atoms
- degree of the nodes
- formal charge
- presence in an aromatic cycle

The data-frame is then converted into a list where every row (ligand) is converted to a graph containing all above parameters needed by the network. Two types of graphs were tried : with and without hydrogen atoms. These atoms are by default removed by Rdkit but were added to investigate if performances could be improved.

2.3 Training the model

The model was built using Graph Convolutional layers referenced in a paper [2], which are a type of layer especially adapted to our type of data (molecules and their features).
Many similar networks architectures can be found through the net, the template we used is the one described on a specific github repository [3][4] The goal of this model is to predict the affinity of the molecules in a batch, based on their graph representation.

The Model's architecture was built on three hidden Graph convolution layers : 16 → 32 → 8 Then results are pooled across the batches Finally, a linear layer will take these values and return the predicted affinity score. The model was evaluated using Mean squared error loss (MSE) with a learning rate of 0.001 The model was trained using Adam optimizer

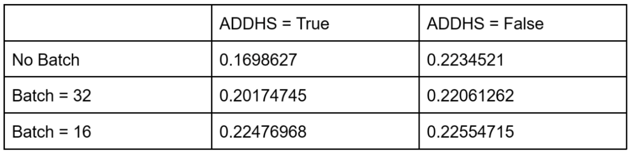
The training data were split into batches, we assumed this would help our model to avoid over-fitting and allow a faster convergence.

3 Results

3.1 Finding the optimal parameters for the model

A total of 1000 molecules were docked and split in two datasets. At the moment, only the first conformation of each ligand with the best affinity was considered for the training. Early results were obtained on half the dataset (534 molecules). The network will try to predict scores of a split test sample (30%) among these molecules.

The network learning step was performed with a learning rate of 0.001 considering our database was big enough for a supervised network. The variable parameters are BATCH_SIZE and ADDHS. They are respectively the size of the batch and if the input molecules have hydrogens. We trained our model on every combination of these parameters and calculated the mean of the last 500 converging epochs. The test portion was 30% of the data. Here are the results :



	ADDHS = True	ADDHS = False
No Batch	0.1698627	0.2234521
Batch = 32	0.20174745	0.22061262
Batch = 16	0.22476968	0.22554715

Table 1: Models performance depending of the choice of the parameters

As it can be seen, adding hydrogen atoms to the graph is improving the model's performance, but it lengthens the training. This is a little bit counter-intuitive since we thought making batches was an important part of the training. It appeared that even though they truly avoided over-fitting, the amount of time spared in the fastest convergence was silenced by the excessively long iteration time and they brought inaccuracy to the model.

Our training sets could be too small to make them efficient. Also, batches are really worth it while used on the GPU and we did not find any benefits in this implantation. In some cases, the no Batch model converges to irrelevant parameters leading to inaccurate predictions. This has some limits that to be taken into account.

All models were stored in the "Models" folder, the corresponding data in the "train-test" folder. For the rest of the project, the model will keep hydrogens and no batch.

3.2 Testing the model's affinity

Now that models were saved, we can assess their efficiency in the ligand pre-selection problem. The beginning of the notebook is dedicated to data exploration. (Readme.txt file explains how to properly reproduce results).

Model's performance were tested on training and testing data and gave the following results:

- Train Mean Squared Error : 0.1709
- Test Mean Squared Error : 0.1715

Since previous results, new docking molecules were obtained. To test how well our model is making predictions, we used these molecules (456) with known affinities. On this scatterplot **(fig2.A)**, each dot represents a ligand, its predicted affinity and the docking affinity. The relationship between the predictions and the docking results can be seen.

The barplot **(fig2.B)** represents the distribution of the affinity scores for both docking and predictions. We noticed that our predictions and the docking results have the same mean yet a different standard deviation value. The predicted affinities tend to be less spread out than the docking affinities.

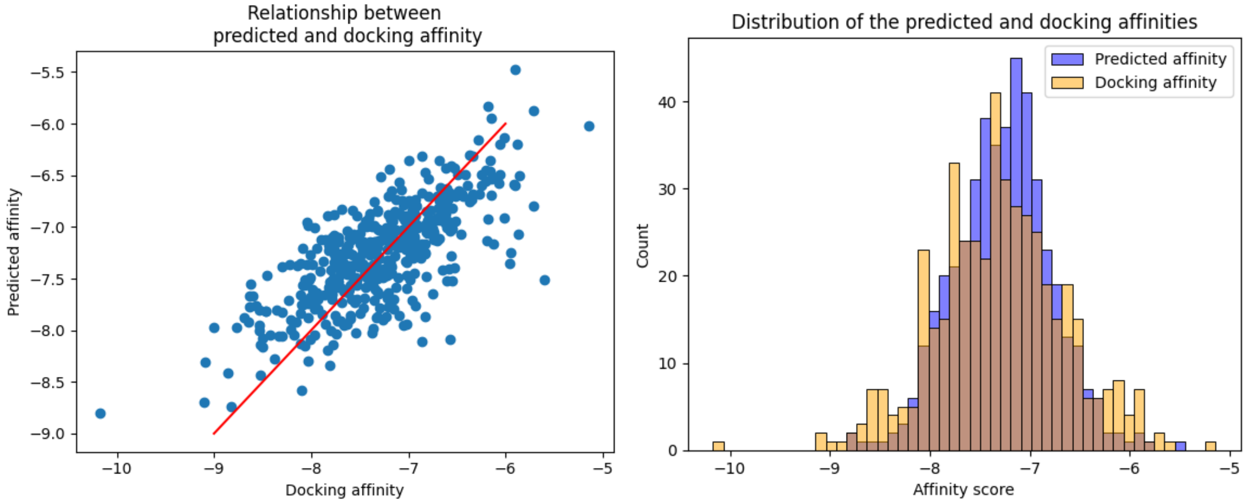


Figure 1: **fig2.A** : Scatterplot comparing the predicted affinity over the docked affinity for each molecule. **fig2.B** : Barplot representing the distribution of the predicted and docked affinities. The mean of prediction is -7.27 compared to -7.30. The standard deviation of the prediction is 0.48 compared to 0.65 for the docked affinities.

That doesn't actually mean that our model is no good at predicting good candidates, what we actually want is to select the top scoring ligands for further analysis, even if their predicted affinity score is not exactly accurate.

3.3 Compare the best predicted ligands and the known best ligands

In this part, we wanted to see if the molecules our model predicted with the best accuracies were also the ground truth best. More generally we wanted to see if, when sorting molecules from best to worst accuracy, we obtained the same order in the prediction and validation set. This would allow us to assess if the model can still be used in a pipeline.

We first tried to compare the top and bottom 20 molecules of the prediction and validation sets. A significantly high amount of common molecules was found, but it was not a relevant way of doing. We then tried some methods to better visualize this. The position of each ligand in the sorted sets were compared and the gap between predicted and true position was plotted **(fig3)**. This figure shows that most of the molecules did not have a huge gap, the good scored ligands are often predicted as so and vice-versa. This shows that despite having some inaccuracies in the affinity prediction, the relative affinity between each molecule is relatively conserved. Though, it should be noted that this is a tendency that is far from perfect.

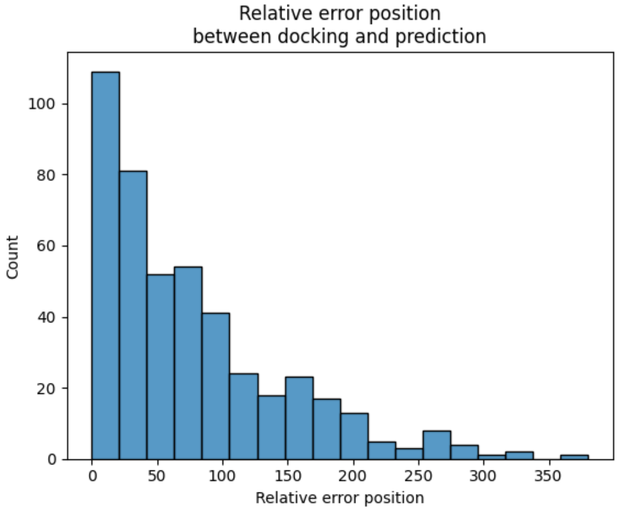


Figure 2: Barplot showing the position gap between the predicted and validation ligands.

3.4 Predict the most promising ligands of the whole dataset

We computed the predicted scores on the reference pilot library containing all of the 5016 molecules. This prediction took only a few seconds while it would have taken days with vina. The 16 best scoring molecules found this way are : ['EOS101357' 'EOS102307' 'EOS101186' 'EOS100233' 'EOS101072' 'EOS101472' 'EOS93' 'EOS1195' 'EOS100814' 'EOS101803' 'EOS264' 'EOS100811' 'EOS1824' 'EOS100687' 'EOS101096' 'EOS98622']

Autodock Vina was then used to check if the predicted molecules were indeed good candidates. Vina's predictions were : [-9.631, -7.507, -6.559, -8.048, -8.513, -8.083, -9.247, -10.18, -7.56, -8.026, -8.78, -7.993, -7.304, -7.708, -8.189, -8.45]. This set presents a high amount of great affinity molecules and a few ones with extreme affinity (e.g. under -9). The mean affinity is -8.26. It should be noted that even though we have extremely promising candidates for further experiments, this set is not exhaustive and some better ligands are probably still missing. Also, some molecules with poor affinity are also present in this set. Using our model to sort a dataset is still very interesting to remove the less promising molecules and save a huge amount of time. It should be used only to extract a sample of good candidates, a big sample will allow not to miss any promising molecule but will allow the presence of more poor candidates too.

4 Limits and perspectives

4.1 Problem introduction

One significant constraint in our methodology lies in relying solely on Affinity as an absolute measure of ligand's probability to be a good candidate. While affinity serves as a valuable indicator, it cannot be used alone. To illustrate this, the **fig4** shows the docking of two ligands with identical affinity values. Despite sharing the same affinity, their spatial orientations differ significantly. This observation underscores the inadequacy of relying solely on affinity as an absolute measure.

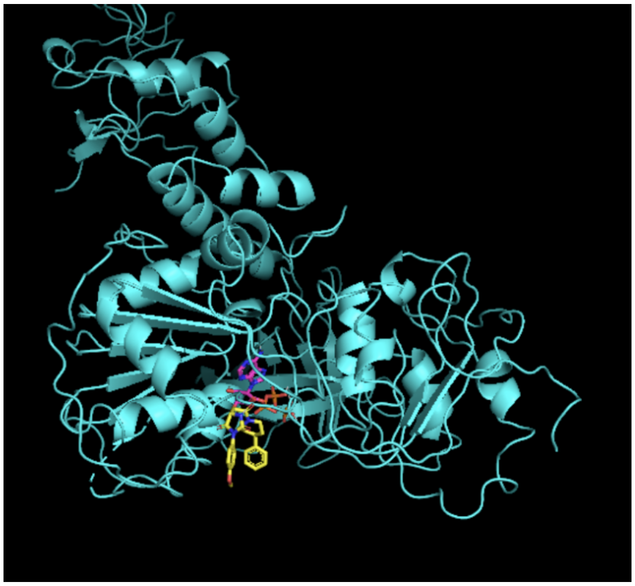


Figure 3: Visualization of the binding of two different ligands with similar affinities. We can see that despite having similar affinities, the orange ligand goes deeper in the pocket which makes it a more promising candidate of potential inhibitor.

4.2 Possible approach

The approach we chose was to introduce a value representing the pocket occupation of a ligand. When performing docking, Vina will try to dock several conformations for each ligand, returning an affinity score and the coordinates of the atoms for every conformation.

The default parameter set on 9 conformations was used half the time and afterward set to 20 to explore more possible conformations. This time, we did not choose the conformation with the best affinity. The selected conformation was the one which maximizes the affinity score and the pocket occupation score. We then made a dataset containing for each molecule, the affinity score of the most probable bind to the ATP pocket.

A model was trained on this dataset and we tested most of the pipeline pictured in the "Results" part. It appeared that the prediction of our model was slightly worse than the regular model we made, with a mean squared error around 0.28. We then sorted our predicted and validation results and performed the same analysis as before. The **fig5** shows similar results as before, the ligands rank is relatively conserved between the prediction and the validation set, which allows to select a batch of promising elements.

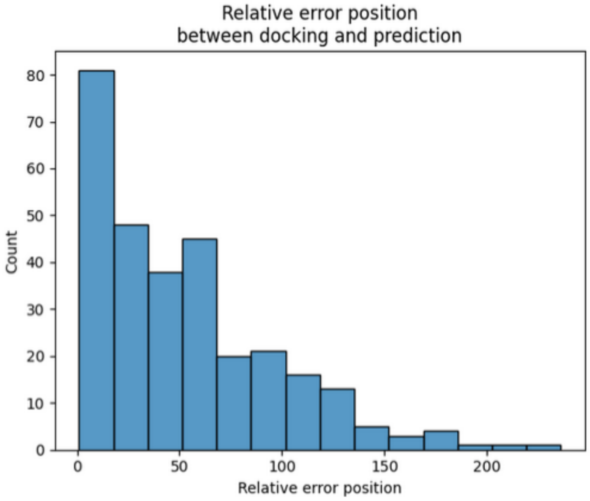


Figure 4: Barplot showing the position gap between the predicted and validation ligands obtained from data that takes into account the pocket occupation.

Further analysis should have been done to assess if the selected ligands are more relevant with this method of introducing pocket occupation or not.

5 Conclusion

The results are very encouraging, we obtained a model capable of sorting the molecules and extracting the potential best hits within seconds. It was found out that the best use for it was to select a range of ligands since its accuracy did not allow to select only the very best molecules but the best pack of molecules. Hence, we still need to dock every single one of the selected molecules afterward to find the best candidates as enzyme ligands.

We tried different input parameters to attest what is better for an accurate prediction. We saw that smaller batches lead to better results overall as well as a much smaller training time. But this comes with the limitations of a model more sensitive to local minimums. Also adding the hydrogens gives best performance at the expense of the training time.

At the end of the pipeline, we predicted the 16 best candidates of the pilot library that was given to us. These candidates were afterwards compared to the real docking affinity. The results confirmed what we observed beforehand: our prediction contains mostly high scoring molecules which are promising, but also some relatively bad candidates.

We highlighted the drawbacks of relying solely on affinity in a ligand prediction pipeline and proposed an alternative approach. While our model's predictions were slightly affected, the results remained informative. Even though we did not entirely explore this method, we introduced the ligand occupation score as a potentially valuable tool to implement in a pipeline of predicting promising inhibitors to an enzyme.

6 Discussion

About the network : Even though we tried a lot of different parameters and inputs, we did not explore every aspect of the network. For example, the selected features of the atoms were selected a bit arbitrarily. Adding other relevant information could bring more information on the molecule and lead to better prediction. There is also one feature that we kept mostly containing 0 all over the data, which may lead to some inaccuracies in the prediction.

Concerning the size of the batch, it would be interesting to re-train the model on a larger amount of data, that would maybe lead to different conclusions on the best approach. Regarding affinity, our findings indicate that it may not be a flawless metric for assessing whether a molecule can serve as an effective ligand. Introducing alternative decision-making criteria, such as pocket occupation, could offer a viable solution to address this limitation. Our exploration has only scratched the surface of this issue, and for the development of a robust pipeline, further in-depth investigation is warranted.

The majority of our findings indicated that employing a neural network was effective in predicting molecules with the highest affinity scores. However, we demonstrated that having the best affinity score did not necessarily make a molecule the optimal candidate. To address this, a crucial next step involves verifying whether the implementation of the pocket occupation score genuinely corrects this issue.

7 Bibliography

[1] : RDKIT tools used : Chem

[2] : Thomas N. Kipf, Max Welling, 2016, Semi-Supervised Classification with Graph Convolutional Networks, https://arxiv.org/abs/1609.02907

[3] : vaiteaopuu/gnn_mol_example (github.com)

[4] : vaiteaopuu/molzip_adapted: molzip adaptation for teaching purposes (github.com)

8 Remerciements

We would like to thank :

- Vaitea OPUU for his help on building the neural network and on the orientation of our project.
- Elodie LAINE for introducing us to other criteria to assess the quality of the ligands.
- Juliana SILVA BERNARDES for supervising the project.