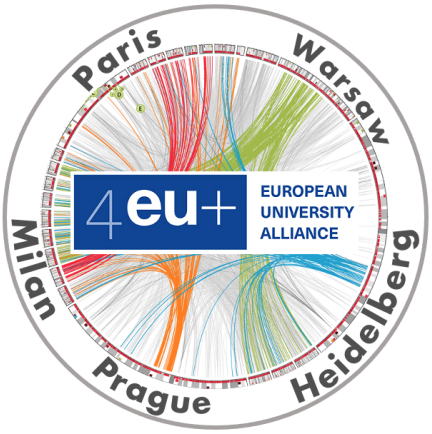


Sorbonne université

Meet-EU Report

Structural bioinformatics on the Sars-Cov-2 helicase protein (Nsp-13)

Written by :

Amina ALIOUA

Adel BIBI TRIKI

Késia DETHELOT-DELAG

Tiziri TERKMANI

Supervised by :

Vaitea OPUU

Elodie LAINE

Juliana SILVA BERNARDES

Contents

List of Figures

Abstract

The COVID-19 pandemic which is one of the most critical pandemics in human history has highlighted the need for effective therapies. Among the many potential therapeutic approaches, exploring specific viral components shows promise. One target of interest in the fight against COVID-19 is the non-structural protein 13 (nsp13). It is a vital component of the SARS-CoV-2 virus. Understanding and targeting nsp13 could lead to innovative therapies. This approach will contribute to ending the Covid-19 pandemic and preventing future pandemics. The project Meet-EU therefore fits in this context and could ultimately result in one of the participants finding a new inhibitor for the Nsp-13 helicase. Thus, the goal of our team is to have a final set of ligands that, according to us, could be viable inhibitors.

1 Material and methods

1.1 The dataset of compounds:

Our study relied on a database of 5016 molecules that we need to dock. For each of these molecules, we have the formula, the molecular weight and the most relevant feature for our analysis, the SMILES. The SMILES (Fig. 1) is a symbolic language for describing the structure of chemical molecules in the form of short ASCII strings and is widely used in chemistry and structural biology software.

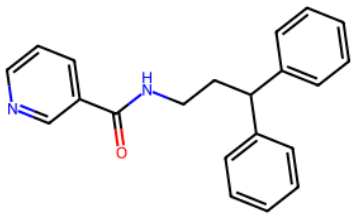
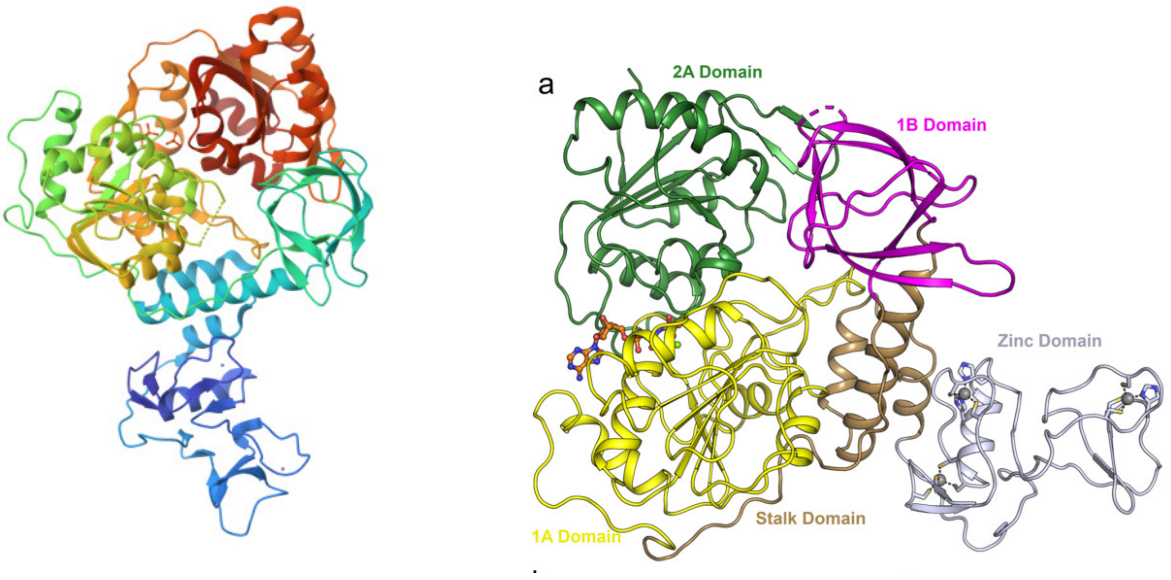


Figure 1: Topological formula of the SMILES CC1=CC(=O)c2ccccc2C1=O

1.2 Target protein:

The aim of this study is to target the SARS-CoV-2 Non structural protein 13 (Nsp13). This 67 kDa protein is highly conserved and essential for viral replication. Moreover, according to

previous works, Nsp13 has two pockets which could interact with drugs. Our target structure is the 6ZSL, crystal structure of the SARS-CoV-2 helicase at 1.94 Angstrom resolution.



(a) 2a (b) 2b

Figure 2: Structure of the 6ZSL and its several domains

1.3 Fingerprints:

To classify molecules, we need to assign them a weight based on common criteria. This can be achieved using fingerprints, which are pre-implemented in the RDKit python package. Fingerprints create a binary vector for each molecule from a SMILES .There are several types of fingerprints such as MACCS, RDkit, and Morgan.

The Morgan fingerprints method generates a bit vector whose size varies. It allows the adjustment of the radius parameter. In addition, it distinguishes stereoisomers, which can make it computationally intensive.

MACCS fingerprints consist of a fixed length of 166 bits. Each bit in the fingerprint corresponds to a specific chemical feature or substructure. This makes it easier to understand the molecular characteristics that contribute to similarity. A value of 0 indicates the absence of a character, while 1 shows its presence. However, MACCS fingerprints do not differentiate between stereoisomers, considering them identical.

Using fingerprints to characterize molecules enables comparison through measures of similarity distances. We employ Tanimoto distance which consists of comparing the fingerprints

and calculating the number of shared bits. Molecules with closer Tanimoto scores are then regrouped using Kmeans. Kmeans is an unsupervised machine learning algorithm. It divides all observations into k clusters. It is a very simple and fast algorithm but very sensitive to outliers. Other clustering algorithms, such as DBScan, Hierarchical clustering, and spectral clustering, are available. DBScan is known for its robustness to outliers and ability to detect clusters of varying shapes. Hierarchical clustering provides a tree-like structure of clusters. Spectral clustering uses graph theory, and Gaussian Mixture Models allow for more flexible cluster shapes by modeling data as a mixture of Gaussian distribution.

1.4 Docking engine:

For both the docking and scoring, we decided to rely on DiffDock, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, it maps this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develops an efficient diffusion process on this space. Empirically, DiffDock obtains a 38% top-1 success rate (RMSD¡2A) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DiffDock maintains significantly higher precision (21.7%). Finally, DiffDock has fast inference times and provides confidence estimates with high selective accuracy. It is also relevant to specify that DiffDock scores are mainly negative, and that the closer they are to zero the better. For instance, the DiffDock score for an already known inhibitor of the Nsp-13 is -0.2, which is much superior to the other scores that we had during the preliminary tests (between -1.2 and -0.6).

1.5 Regression model:

We kept two regression models: A random forest model and a neural network model. The architecture of the neural network is simple with one input layer, one hidden layer with 64 neurons, another hidden layer with 32 neurons, and an output layer. The model is trained for 10 epochs using Adam optimizer and mean squared error (MSE) loss.

2 Approach:

The key to our strategy was DiffDock because it did both the docking and the scoring, but it also comes with a price. DiffDock is very slow (5 min per molecule in the best case) and using it to dock the whole database would take more than two weeks. So, our first idea was to reduce the size of the dataset until having a few molecules that we dock. To do so, we decided

to clusterize our database: we start by computing the fingerprints of all the molecules. This allows us to have numeric representations and thus have a distance matrix that will be used for the clustering. We then extract representatives of each cluster. These representatives are docked and help decide which clusters we can get rid of. This also means that we may need to repeat all this process several times.

Unfortunately, the previous approach was only effective if we had many clusters, and in the worst case scenario the clustering fails and we find that the data can not be separated. So we had to find another approach to cover our basis. Our teachers suggested that instead of reducing the size of the database, we should try to predict the scores of the molecules that we do not dock. In other words, we randomly pick molecules from the database that we score with DiffDock. Then these scores will be used to train a regression model that should predict the scores for the rest of the molecules in the database.

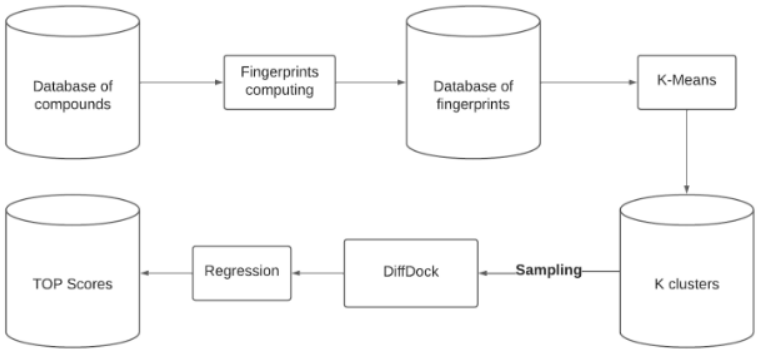


Figure 3: Diagram of the full approach

3 Results:

3.1 Fingerprints:

We initially used the Morgan fingerprint (or circular fingerprint) to be our main mathematical representation of our molecules, since this type of vector is known for being good at representing biological molecules and especially drug interactions. However, we later found out that this type of fingerprints couldn't help us divide the dataset. So we decided to change the fingerprint method because another fingerprint could be more suitable for our problem. Ultimately, we chose MACCS keys (MDL keys) fingerprints which were better than the Morgan fingerprints.

3.2 Clustering:

Then, with an appropriate fingerprint (here MACCS) we decided to do a clustering. We did multiple methods such as : KMeans, DBSCAN, Gaussian mixture and Hierarchical clustering. After these clustering we choose to only keep the KMeans method due to the better visualization we get after a principal component analysis (Fig. 3). When we were doing DBSCAN and Hierarchical clustering the results were not sufficient and by looking visually at it we could say that it wasn't the best way to do the clustering. The Gaussian mixture model was also a little bit better but not as much as KMeans. So in the further analysis we stuck to Kmeans clustering.

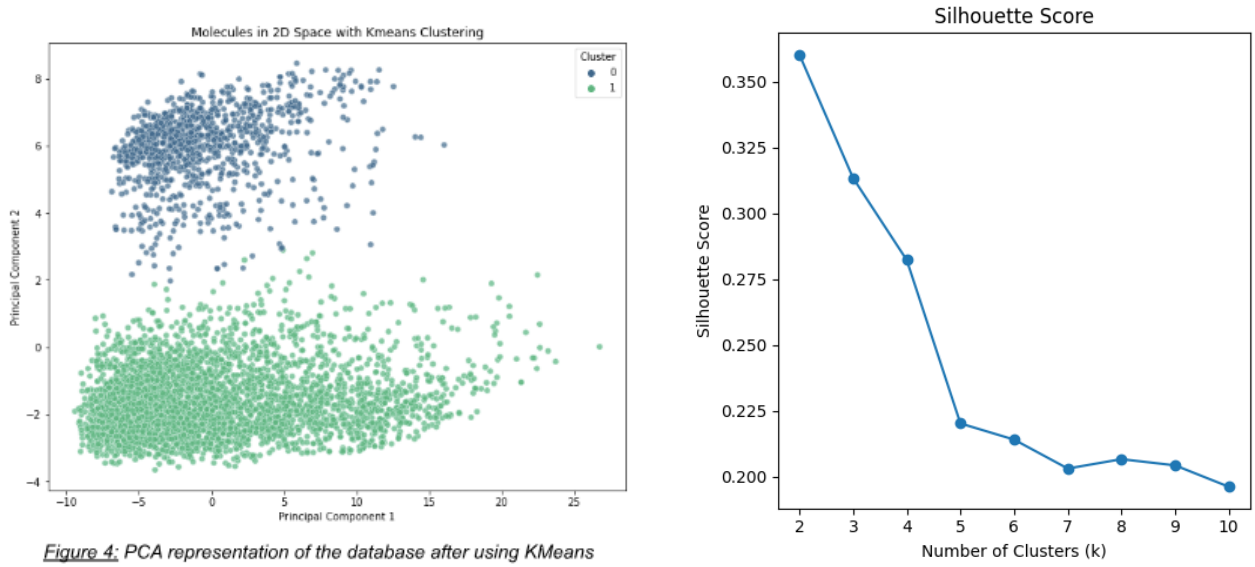


Figure 4: PCA representation of the database after using KMeans

(a) 4a (b) 4b

Figure 4: PCA after clustering and silhouette score

3.3 Regression:

In order to execute a regression, Diffdock was used to get the scores of a sample of molecules in both clusters. It was used against the whole 6ZSL protein. The sizes of the samples are different, from the first cluster we took 66 molecules and for the second cluster we took 89 molecules. Docking scores were quite similar between the 2 clusters.

3.4 Prediction:

We used the neural network for the prediction of the scores for both clusters and since we have no concrete way to check the reliability of our model, the only reference we have are DiffDock

scores but even those give different results for the same molecule,we took the list of ligands that we already calculated the scores for and checked the MSE, MAE. We obtained the following results:

Cluster	MAE	MSE
Cluster 1	0.3163	0.1264
Cluster 2	0.3322	0.1459

Figure 5: MSE and MEA results for the Clusters

Which are quiet good results considering the fact that our training set is not as big as it should be.

Then, we predict the scores for the rest of the database based on the scores found by Diffdock. The initial thought was to eliminate the cluster that appears to have lower values, but since the scores were similar we keep both of the clusters and just assign a docking score to the rest. To evaluate our prediction, we use DiffDock on randomly chosen molecules, and compare predicted values with DiffDock scores.

4 Discussion:

4.1 Clustering:

The K-means clustering was disappointing since we only had 2 clusters. The plot of the silhouette scores, a measure of how similar a data point is within-cluster (cohesion) compared to other clusters (separation), shows that the optimal K is K=2 (figure 3). The PCA also shows a good separation of the dataset into two clusters. Using a higher K would be counterproductive.

4.2 Prediction of the scores:

We notice that the neural network predicts different values for one molecule. Consequently, we decided to keep the average of 20 estimations. Furthermore, we filter our predictions by applying a threshold, and we only keep the molecules with a predicted score higher than -0.5. This threshold enables the retention of molecules that are most likely to be suitable candidates.

Indeed, the closer the score is to 0, the more accurate it is.

Among those selected molecules, we picked 15 molecules from the first cluster and 10 from the second cluster. We then run DiffDock on those molecules in order to assess whether the results are coherent or not. Sadly, the predictions and the DiffDock scores are very different and are almost random, only very few molecules have satisfying results. The problem is obviously how small the training set is compared to the whole database, and unfortunately this a problem that we could not solve unless we ran more docking, running DiffDock on more molecules would help widen our training set for a better performance of the model, but it does take a lot of time and we do have a large dataset. Another reason could be the model that is too simple and not suited enough for our problem, as it was a last minute idea that was not explored deeply enough.

4.3 Analysis of the top hits:

Our best molecules predicted by our neural network are long molecules. Most of them contain sulfur atoms (Fig. 5). We know that disulfide bonds between polypeptides are essential for protein assembly and structure. Moreover, literature shows that drugs with sulfur have a large range of biological activities. So our top hits are promiscuous molecules to inhibit Nsp13.

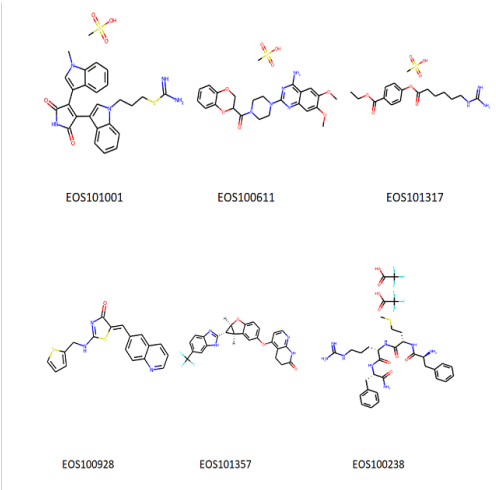


Figure 6: Top 3 best hits for the each cluster

Our pipeline on the Charles's team work : We try to use our pipeline on the work Charles's university gave us, but when we look for the differences between the predicted value by our neural network and their scores we find a significant difference.

5 Conclusion and Futur work:

Throughout the project, we knew that DiffDock was very slow, not completely reliable, and yet we decided to use it anyway because we wanted to try a deep learning method for docking and scoring. We would suggest using DiffDock for small datasets and certainly not datasets as big as ours unless you have the material to cope with the computational load. We faced various issues during the project: the main difficulty was computational whether for using DiffDock or computing the similarity matrix for the clustering. Another issue was obviously having deceiving or unexpected results that made us switch strategies more times than we would've wanted: for example we expected to have more than 2 clusters and hoped to gradually reduce the size of the database, but unfortunately having 2 clusters with very close scores was a bad surprise that pussed to use a regression model to predict the scores for all the dataset.

However, the project was still an interesting introduction to the drug design field. As we just said, we learned how to handle a research project with so many possible tools and approaches. This also means that we had many more tracks that we would've liked to explore if we had more time. Firstly, using different methods for the clustering, the fingerprints and the regression model since a lot of the methods had the same effectiveness and the lack of time prevented us from finding the most suited ones. Secondly, using the other features that were in the database (hydrophobicity,...) for the clustering since these features could be more relevant regarding our approach. Finally, optimizing the docking by only using the pocket pdb instead of the whole 6ZSL, this aims at reducing computation time and may give better docking scores.

References

[1] Newman, J. A., Douangamath, A., Yadzani, S., Yosaatmadja, Y., Aimon, A., Brandão-Neto, J., Dunnett, L., Gorrie-Stone, T., Skyner, R., Fearon, D., Schapira, M., Von Delft, F., & Gileadi, O. (2021). Structure, mechanism and crystallographic fragment screening of the SARS-COV-2 NSP13 helicase. Nature Communications, 12(1). https://doi.org/10.1038/s41467-021-25166-6

[2] Source, D.L.(s.d.). NSP13 Helicase Crystal Structure and XCHEM Fragment Screen Diamond Light Source. https://www.diamond.ac.uk/covid-19/for-scientists/NSP13-Helicase-Structure-and-XChem.html

[3] Corso, G., Stark, H., Jing, B., Barzilay, R., & Jaakkola, T. S. (2023). DiffDock : diffusion steps, twists, and turns for molecular docking. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2210.01776

[4] Bajusz, D., Rácz, A., & Héberger, K. (2015). Why is Tanimoto Index an appropriate choice for fingerprint-based similarity calculations ? Journal of Cheminformatics, 7(1). https://doi.org/10.1186/s13321-015-0069-3

[5] Hamid Safizadeh, Scott W. Simpkins, Justin Nelson, Sheena C. Li, Jeff S. Piotrowski, Mami Yoshimura, Yoko Yashiroda, Hiroyuki Hirano, Hiroyuki Osada, Minoru Yoshida, Charles Boone, and Chad L. Myers.(2021) Improving Measures of Chemical Structural Similarity Using Machine Learning on Chemical–Genetic Interactions. Journal of Chemical Information and Modeling 61 (9),4156-4172 doi: 10.1021/acs.jcim.0c00993

[6] GitHub - gcorso/DiffDock: Implementation of DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. https://github.com/gcorso/DiffDock?tab=readme-ov-file

[7] K Means Clustering Method to get most optimal K value (analyticsvidhya.com). https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/#:~:text=The%20silhouette%20coefficient%20or%20silhouette,scikit%2Dlearn%2Fsklearn%20library.

[8] Structure, mechanism and crystallographic fragment screening of the SARS-CoV-2 NSP13 helicase - PubMed (nih.gov). https://pubmed.ncbi.nlm.nih.gov/34381037/#&gid=article-figures&pid=fig-1-uid-0