Binding affinity prediction with Rosetta of HIV-1 protease inhbitor drugs

Rasmus Willsleff Andersen
KU-id: fvt270
Supervisor 1: Amelie Stein
Supervisor 2: Marion Lucia Silvestrini

Bioinformatics Project 1; Summer 2024

UNIVERSITY OF COPENHAGEN

Contents

1 Introduction

Human Immunodeficiency Virus (HIV) is a potent virus capable of inflicting severe damage to the human immune system. Especially prior to the 1980's, the viral infection were prevalent, but unknown and were refereed to as the "Gay Disease". Back then, the focus lied on the treatment and prevention of Acquired Immunodeficiency Syndrome (AIDS), which were established to be a result of the HIV retrovirus (Montagnier 2002).

Today, there exist a greater understanding of HIV and generally, two types of HIV exist: HIV-1 and HIV-2. Of these, HIV-1 is much more prevalent globally, whereas HIV-2 is predominantly present in West Africa. Because of the arguably higher transmission rate of HIV-1 indicated by the distribution; research, attention and funding were initially primarily afforded to HIV-1 (Montagnier 2002 & Quinn 1994). The focus of this paper will also lie within the domain of HIV-1 - specifically within the Protease Inhibitor (PI) drug class.

As HIV is a retrovirus and can be treated with Antiretroviral Therapy (ART), a number of different inhibitors exist as ART. These are divided into subclasses: nucleoside reverse-transcriptase inhibitor (NRTI), non-nucleoside reversetranscriptase inhibitor (NNRTI), PI, integrase strand transfer inhibitor (INSTI), and entry inhibitors. The scope of this paper is limited to PI. Within PI, different drugs exist: FPV, ATV, IDV, LPV, NFV, SQV, TPV and DRV of which DRV has been analyzed previously (Jiang 2024).

Previous work from the University of Copenhagen established the possibility of utilizing Rosetta to predict binding affinity of proteases to HIV-1 (Jiang 2024). In this paper, the establishment and assessment of Rosetta binding affinity prediction continues. Specifically, for the ATV, IDV, LPV and NFV drugs from the PI drug class. These are proteases which work by binding to HIV-1 which suppress replication - effectively stopping the spread of HIV-1 *in vivo* (Flexner 1998).

However, with evolutionary pressure from modern medicine, organisms and vira are prone to develop resistance, of which HIV is no exception. Resistance towards PI drugs come in the form of mutations which reduce protease binding affinity. Specifically, it has been assessed that mutations in position: 30, 32, 47, 48, 50, 54, 76, 82, 84 and 88 reduce binding affinity for the PI drug class (Stanford University n.d.).

A core concept of mutations is the concept of genetic barrier. This describes the genetic change required for an observable difference of the protein function, stability, binding etc. In this project, the mutational focus is on binding affinity. A low genetic barrier thus indicate that a low amount of genetic change is needed to cause an effect on binding affinity. This was particularly an issue with early versions of PI drugs. More recent development of the PI drugs showcased more differentiated and complex resistance patterns, but drug resistance may still occur, albeit requiring larger genetic change (Fletcher et al. 2018).

1.1 Software

Rosetta is a software tool for bioinformatics with multiple applications. Among these are de novo protein design, enzyme design, vaccine design, antibody engineering and much more. Crucially though, Rosetta is able to predict ligand docking, which is crucial for developing, discovering and understanding medicinal drugs. As previously described, Rosetta is used in this project to predict binding affinity of proteases to HIV. These predictions come in the form of $\Delta\Delta G$ predictions. A $\Delta\Delta G$ value represent the change in the Gibbs free energy from the wild-type to the mutated structure, where ΔG is the free energy. It is important to note, that when considering $\Delta\Delta G$ values, negative values are stabilising or increasing binding affinity mutations, whereas positive $\Delta\Delta G$ values are destabilising or decreasing binding affinity mutation (Rosetta commons n.d.).

Rosetta is able to handle multiple mutations, which is an important feature as the amount and type of mutations between organisms may vary greatly. This is especially the case for the HIV-1 protease enzyme, as it is under selection pressure coupled with the swift generation time of vira. This highlight both the importance of investigating and understanding PI-resistance positions as well as the need for any given prediction software to handle multiple mutations. However, as with any machine learning model, it becomes increasingly unreliable with increasing amounts of unknown and confounding variables (Ouyang-Zhang et al. 2024).

2 Methodology

At the core of investigating prediction of protease binding affinity to HIV-1 is the Rosetta software. This software is integrated at the University of Copenhagen. In fact, a pipeline has been set up for running $\Delta\Delta G$ calculations. This pipeline is heavily utilized and slightly modified to fit the scope of this paper.

The pipeline consists of two crucial steps; a relax step and a $\Delta\Delta G$ calculation step. The goal of the relax step is to refine the atoms present within the structure file. Here, the conformations around the structure are searched for the optimal relaxed conformation (Rosetta commons n.d.). The next step, the $\Delta\Delta G$ calculation step, is the actual prediction step where the Rosetta algorithm performs $\Delta\Delta G$ calculation.

However, even before the relax step, the structure input must coincide with the analysis in question. In this instance, the requirement is to remove unnecessary information as well as information not able to be modelled, such as water molecules. Furthermore, the protein structure sequence must be aligned. Lastly, the ligand to be modelled and predicted upon needs to be included at the end of the structure file.

Another crucial input of the pipeline is the mutations to be predicted upon. These are given as a mutation *.txt* file. Further optional but relevant flags for the pipeline are also given. For instance, the *chainid*and *run_struc* parameters describes the protein chains to model. In the case of the ATV, IDV, LPV and NFV drugs, only chain-id A and B are present and are both modelled. An example of a relax step for the NFV drug script submission can be seen below.

```
#!/bin/sh
#SBATCH --job-name=hiv_NFV
#SBATCH --time=48:00:00
#SBATCH --mem 5000
#SBATCH --partition=sbinlab_ib

conda activate /groups/sbinlab/software/PRISM_tools/py3_ros_ddG_env

dir_py=/groups/sbinlab/fvt270/PRISM/software/rosetta_ddG_pipeline
dir_run=/groups/sbinlab/fvt270/summer_project/drugs/NFV

# Relax:
python3 $dir_py/run_pipeline.py \
    -s $dir_run/1ohr_aligned.pdb \
    -o $dir_run/output_NFV \
    -i create \
    -mm mut_file \
    -m $dir_run/NFV_mutfile.txt \
    --chainid AB \
    --run_struc AB \
    --overwrite_path True \
    --slurm_partition sbinlab_ib \
    --ligand True \
    --dump_pdb True
```

If one were to run $\Delta\Delta G$ calculations instead of the relax step, one should simply change the $-i$ flag to $-i$ *proceed*. Otherwise, as clearly seen in the bash command, the majority of inputs are pathing and $\Delta\Delta G$ calculation specifics. This means the other drugs, ATV, IDV and LPV, just need updated pathing, which is easily automated.

2.1 Optimizing analysis

As the vast majority of elements of filtering, job submission, $\Delta\Delta G$ output, plotting and analysis are identical across drugs, a class is utilized for optimizing these. Thus, a *data_manipulation* class is created. Within this, all of the steps from data filtering all the way to plotting is included.

Filtering of the data is done by removing invalid amino acids, indels, stop codons and non-assigned drug resistance values from the experimental mutational data. Furthermore, the wild-type mutational data should line up with the pdb structure data, which is checked with the *pdb_to_seq*. This returns a dictionary with the non-matching wild-type amino acids. These are changed in the pdb structure file. The mutations changed can be seen in the *pdb_mutations.py* file. Changing of the amino acids are performed with PyMol (Schrödinger, LLC 2015). Lastly, the Rosetta pipeline set up required rigid conformity to a predefined pdb file template. Failure to comply could cause an *IndexError*. The potential pitfalls surrounding these errors have been fixed manually in the pdb files.

Even though the data exists for more mutations, the amount of mutations considered for this paper is 5. This is due to the nature of predictions which gets increasingly unreliable with the amount of mutations and thus parameters (Ouyang-Zhang et al. 2024).

The next step, job submission, is simply changing pathing to fit the drug in question, as previously stated. The job submission itself is done through Slurm, which is a job submission manager. Slurm handles CPU and GPU resource allocation (Jette et al. 2002). Each individual relax and $\Delta\Delta G$ calculation script is submitted on the server-side by *sbatchrelax.sh*. This will submit the script and create a log file.

As can be seen on the NFV drug submission script, there are two variable crucial inputs: mutation file and pdb structure file input. The structure file of course gives the structure of the protein with coordinates. This structure should be aligned with the sequence for the given drug and fulfill the requirements previously set out. The other input parameter, the mutation file, should contain the amino acids to mutate. These are mutated on both chains.

```
I A50 L I B50 L A A71 V A B71 V
I A50 L I B50 L A A71 V A B71 V
L A10 I L B10 I I A13 V I B13 V R A41 K R B41 K
```

An example of such mutation file for the ATV drug can be seen above. Each mutation consist of vital information and is in order from left to right: The wild-type amino acid, the chain-ID, the position and the mutated amino acid. From this file snippet, it is also possible to see that each mutation occur on both chain A and B.

After the *relax.sh* script has run once for relaxation, the actual $\Delta\Delta G$ calculation can commence. Calculation of $\Delta\Delta G$ utilizes the same pipeline, and thus the only thing which should be altered is the flag from $-i$ *create* to $-i$ *proceed*. This will predict $\Delta\Delta G$ of the given input mutations and write them to a *.txt* file.

2.2 Data

The data for this project comes from the Stanford HIV database. This database contains experimental data for different drugs, hereunder the PI drug class, which is the focus of this project. The experimental data is the drug resistance for the given mutations (Stanford University n.d.). For this project, the following drugs are selected: ATV, IDV, LPV and NFV.

These mutations need to be in the correct format for the pipeline to predict on them, as previously explained. Furthermore, there has been conducted filtering and quality control of the experimental data and mutations. Noteworthy are the removal of indels, nonsensical mutations, mutations where the experimentalists were unsure of the end mutations, single mutations, non-applicable values (NA's) and removal of mutations count above 5.

As previously described, the data and the structures given must align with the sequences. This was done with PyMol. Lastly, the ligands were also added at the end of the structures.

2.3 Object oriented solution

To standardize the process of quality control, filtering, script generation, plotting and analysis, a python class has been created. This expedites the process, as the format of the drug resistance data as well as the pipeline is identical. The class is written in the *data_manipulation.py* file and contains everything needed to run prediction and analysis should a new drug to explore arise. The *data_manipulation.py* file can be found in the associated Github repository (Andersen 2024).

To get started using the class, one should simply supply the class initiation with the path to an unaltered Stanford HIV dataset. Then, the *data_manipulation* will know the column names and how to read the *.txt* datafile. The ingestion of the data is handled by Pandas, which is a handy tool for data manipulation and visualization (McKinney et al. 2010).

Options for filtering are given with the class. Primarily, these options are centered around this project. This means that the filtering options will most likely align with new drugs. The class functions which handles filtering and quality control are: $filter\_mutations$, $remove\_indels$, $filterna\_column$ and *count_mutations*. *count_mutations* is specifically made to investigate the amount of mutations present. Otherwise, the naming of the functions are self-explanatory when it comes to explaining their function. However, if one is unsure about a function or its input parameters, the class' functions contains detailed docstrings.

The next step of the project is to create a script for Slurm submission. This is handled by the *relax_structure_script*, which despite the name, also generates the bash code for $\Delta\Delta G$ calculation submission.

Once the Rosetta results are generated, the python class is also able to handle plotting and analysis. This comes in the form of plotly plots, which is a tool for plotting python data, hereunder pandas DataFrames (Plotly Technologies Inc. 2015). However, before plotting, the output data must be merged with the experimental data. This is handled by the *add_output* function.

The main plots utilized for this project are histograms, heatmaps and scatter plots. The histograms and heatmaps are investigated to get an overview of the mutational data and how it is spread along the amount of mutations. The histograms are made with the *plot_n_mutation_histogram* function, which is capable of plotting data only up to n mutations. Likewise, the heatmaps plotted are used to see the relationship of wild-type to mutated amino acids.

Scatter plots are used to gander the relationship between experimental and predicted data. These are made with the *plot_pred_exp* function. Crucially, this function has an *n_muts* parameter, which controls how many mutations should be plotted to avoid clutter. Along each mutational count, a trend line is included.

To check a crucial point about Rosetta, namely the fact that Rosetta has a harder time predicting proline compared to other amino acids, a function is made for checking this: *check_outlier*. The fact that Rosetta has a harder time predicting mutations to proline is likely due to the unique nature of proline, as it is the only amino acid where the side chain is connected to the backbone twice. This means that a ring containing carbon and nitrogen with a length of 5 is formed. The *check_outlier* function checks mutations to proline and summarises the findings within the data.

3 Results

Utilizing the object oriented python class created, it is possible to expedite the process of $\Delta\Delta G$ predictions from Rosetta. The resulting analysis, plots and statistics are generated. These are displayed in this section.

3.1 Mutational landscape

To investigate the mutational landscape, heatmaps and histograms are utilized. These provide insight into the mutational data. Histograms are used for looking into the amount of data present for each mutational count. The histograms for the drugs: ATV, IDV, LPV and NFV can be seen below in figure 1.

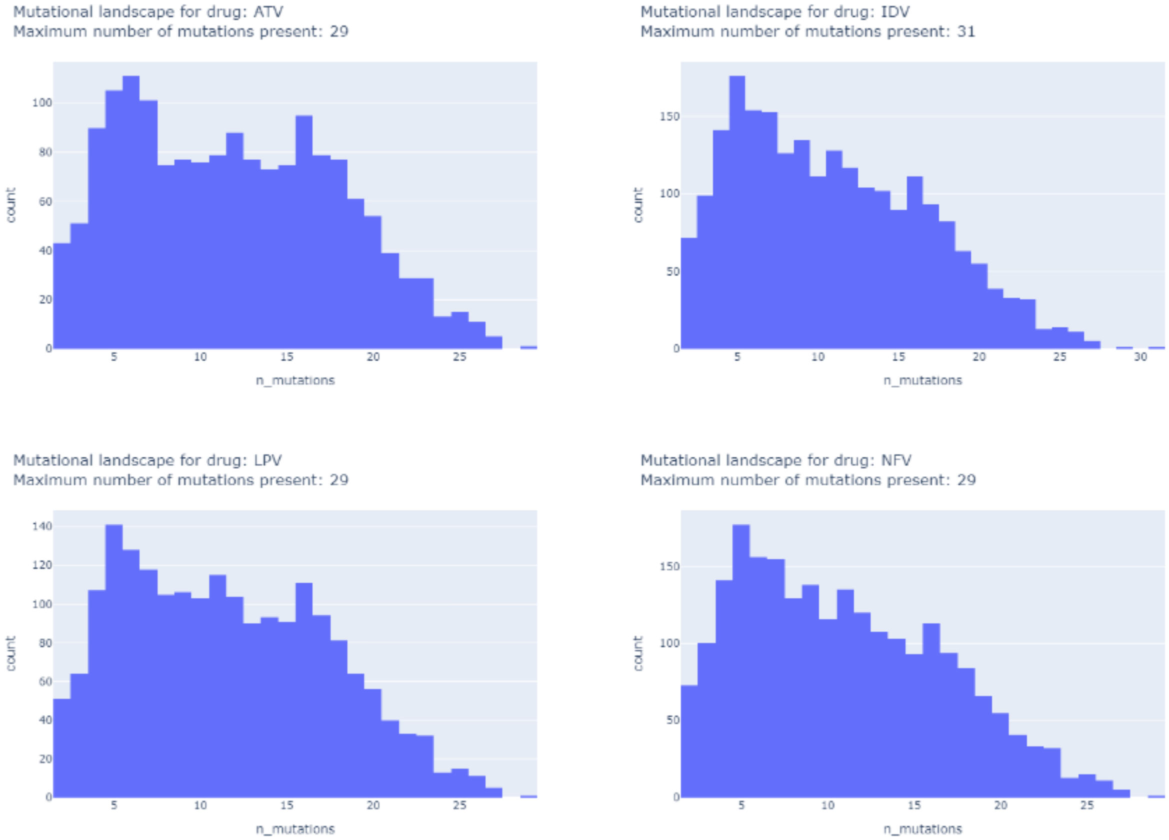
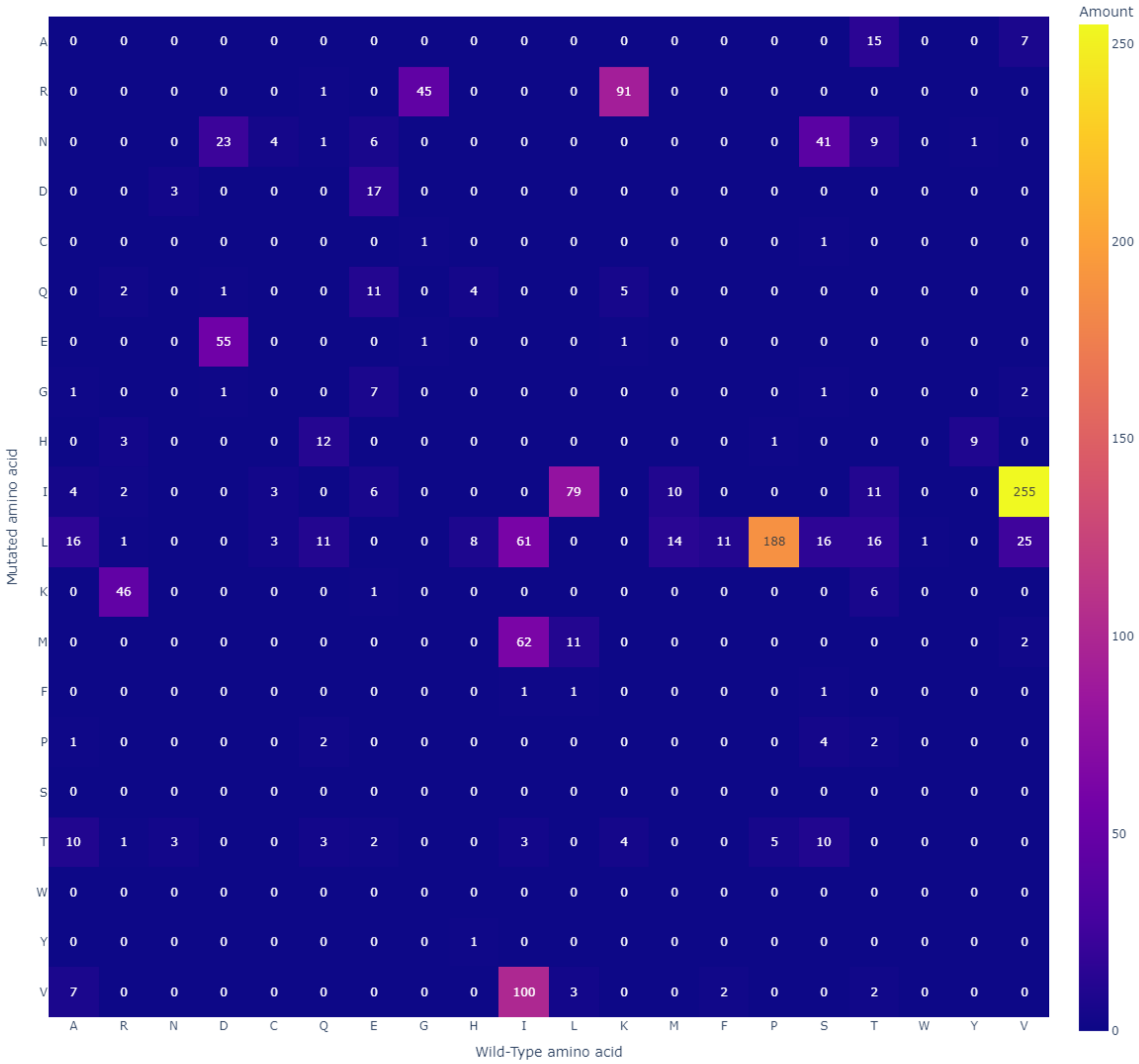


Figure 1: The mutational landscape of ATV (top left), IDV (top right), LPV (bottom left) and NFV (bottom right). Along with each subplot is the amount of mutations present. In this histogram, the amount of mutations for the patient is on the x-axis and the amount of patient is on the y-axis.

From figure 1 it is quite noticeable that the amount of mutations present for each patient varies greatly; all the way from 1 to 31 mutations present. It is also noteworthy that data exists along this mutational landscape. This means that even though a lot of mutations are present, the protein is still functional. It can be observed, that consistently, the most data exists for around 5 mutations. Coincidentally, 5 is also the cutoff for the amount of mutations to consider for Rosetta predictions.

Heatmaps are useful in the context of investigating the mutational landscape by providing insight into the wild-type amino acids being mutated as well as mapping the mutations. The heatmaps generated by this project are quite similar. Thus, only a single heatmap is shown for this results section. However, if one is interested in investigating more heatmaps, look no further than the github repository associated with this project (Andersen 2024).



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	7
R	0	0	0	0	0	1	0	45	0	0	0	91	0	0	0	0	0	0	0	0
N	0	0	0	23	4	1	6	0	0	0	0	0	0	0	0	41	9	0	1	0
D	0	0	3	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
Q	0	2	0	1	0	0	11	0	4	0	0	5	0	0	0	0	0	0	0	0
E	0	0	0	55	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
G	1	0	0	1	0	0	7	0	0	0	0	0	0	0	0	1	0	0	0	2
H	0	3	0	0	0	12	0	0	0	0	0	0	0	0	1	0	0	0	9	0
I	4	2	0	0	3	0	6	0	0	0	79	0	10	0	0	0	11	0	0	255
L	16	1	0	0	3	11	0	0	8	61	0	0	14	11	188	16	16	1	0	25
K	0	46	0	0	0	0	1	0	0	0	0	0	0	0	0	0	6	0	0	0
M	0	0	0	0	0	0	0	0	0	62	11	0	0	0	0	0	0	0	0	2
F	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0
P	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	4	2	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	10	1	3	0	0	3	2	0	0	3	0	4	0	0	5	10	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V	7	0	0	0	0	0	0	0	0	100	3	0	0	2	0	0	2	0	0	0

Figure 2: Heatmap of the mutations with data available for the LPV drug. The mutation amount are in absolute numbers and include only a mutation count up to and including 5. On the x-axis are the wild-type, whereas the y-axis displays the mutation.

The mutational landscape heatmap for the LPV drug up to 6 mutations shown in figure 2 shows a clear pattern with few amino acids disproportionately occupying much of the mutational landscape. These amino acids include: isoleucine, leucine, proline and valine. As previously mentioned, this pattern is repeated in heatmaps for the other drugs included in this project: ATV, IDV and NFV. As well as gandering heatmaps for the other drugs, it is also possible to investigate heatmaps with differing mutational amounts as well as look into relative count as opposed to the absolute count showcased in figure 2.

3.2 Rosetta predictions

The results of the Rosetta pipeline is an output *.txt* file with predicted $\Delta\Delta G$ values. This can be compared with the experimental values present for each drug. This has been done by plotting the log of the experimental drug resistances versus Rosetta predicted $\Delta\Delta G$. Furthermore, Spearman rank correlation and Pearson correlation coefficients are also calculated to investigate the relationship between experimental and predicted data.

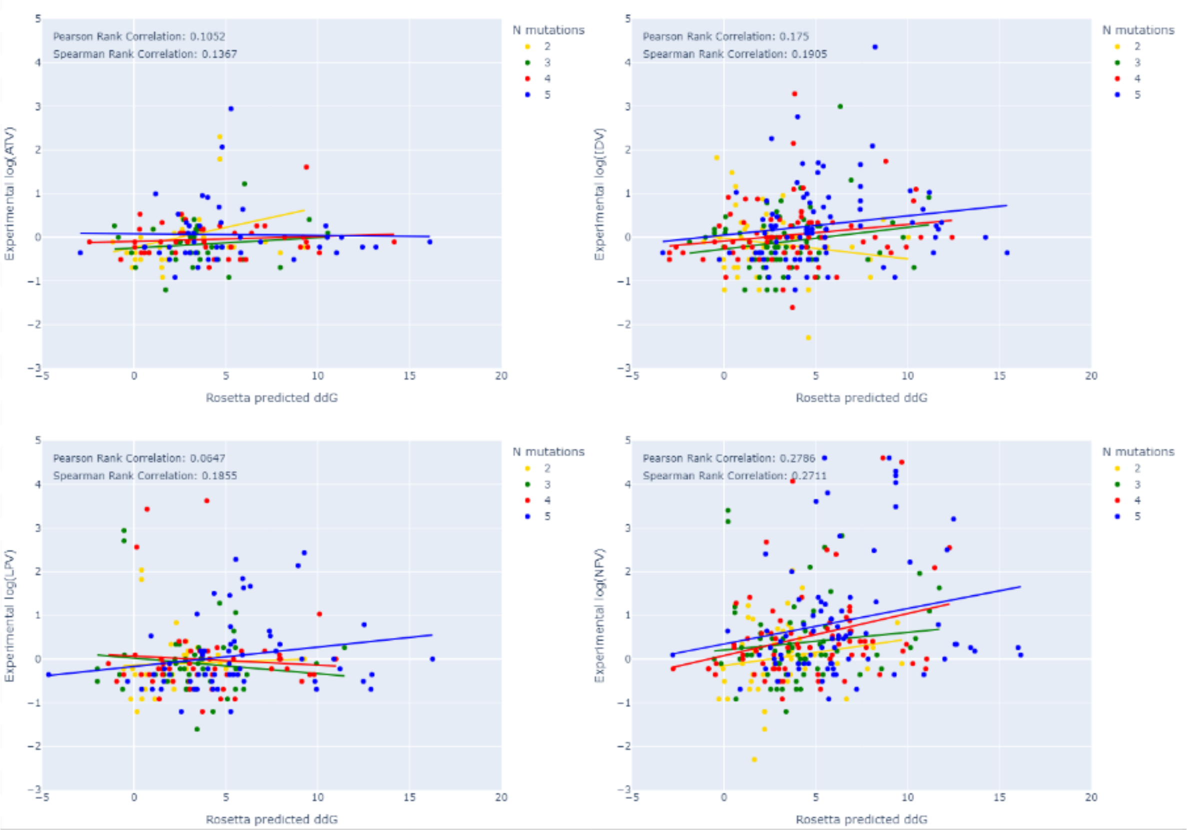


Figure 3: Logarithm of the experimental drug resistances (y-axis) plotted against the predicted $\Delta\Delta G$ scores from Rosetta (x-axis). The scatter plots are for the ATV (top left), IDV (top right), LPV (bottom left) and NFV (bottom right) PI drugs. Furthermore, the points are colored by the amount of mutations for the given patient.

From figure 3 there seems to be no clear correlation between the experimental drug resistances and the Rosetta predicted $\Delta\Delta G$ scores. However, it is important to note that as the relationship between the two variables is unknown, linear regression might not be the best metric for assessing the correlation. Considering the pearson correlation coefficients aggregated for 2, 3, 4 and 5 mutation counts for the ATV, IDV, LPV and NFV drugs, which are 0.11, 0.18, 0.06 and 0.28 respectively, they do not indicate that Rosetta can reliably predict drug resistances for these drugs with the PI drug class. The same pattern is true for the Spearman rank correlations, which are 0.14, 0.19, 0.19 and 0.27 respectively. However, it seems that the prediction capability of Rosetta is not equally distributed among the drugs, where the NFV drug scores higher in both metrics. If one wishes to further investigate specific drugs or look into metrics for specific mutation counts for specific drugs, one can look into the associated github repository (Andersen 2024).

As the Rosetta software struggles with mutations to proline, looking into the amount of mutations to proline in the outliers becomes paramount. An overview of these outliers can be seen in table 1. Here, data for the ATV, IDV, LPV and NFV PI drugs are present with the amount of outliers, amount of proline outliers and the ratio of the two.

Drug	**Outliers**	**Proline outliers**	**Proline ratio**
ATV	11	8	0.73
IDV	18	14	0.78
LPV	10	6	0.6
NFV	22	16	0.73
Total	61	44	0.72

Table 1: Overview of proline appearance in Rosetta outliers for each drug. The *Outliers* column describes predicted $\Delta\Delta G > 10.0$.

The table in table 1 shows a pattern of proline being quite prevalent in Rosetta predicted outliers. Here, the difficulty of predicting proline for Rosetta is revealed. Interestingly, the same pattern is also present for the experimental outliers, although not as striking. These can be seen in table 2.

Drug	Outliers	Proline outliers	Proline ratio
ATV	1	0	0.00
IDV	4	2	0.50
LPV	5	0	0.00
NFV	21	13	0.62
Total	31	15	0.48

Table 2: Overview of proline appearance in experimental outliers for each drug. The *Outliers* column describes experimental log(drug resistance) > 2.5.

From both the experimental proline outlier ratio in table 2 and the predicted proline outlier ratio in table 1, the amount of proline present in outlier seems to be rather high. However, predicted outliers seem to have especially high prevalence of proline. Especially interesting is the amount of outliers found for the NFV drug, which contains a large portion of the total outliers. This can be seen in both table 1 & 2, but is very noticeable in table 2 and is visualized on figure 3.

4 Discussion

The findings from this project demonstrate a lacking capability of Rosetta to accurately predict ligand docking for the ATV, IDV, LPV and NFV drugs of the PI drug class. The Spearman rank correlations are rather low, but the Spearman rank correlation generally increase when considering decreasing mutation amount. This can be seen on the experimental versus predicted scatter plots on the associated Github page (Andersen 2024). This pattern of increasing accuracy was also found by the previous study on this topic (Jiang 2024).

It is important to consider the scale and nature of $\Delta\Delta G$ values. The topic for this project is binding affinity. This means that positive $\Delta\Delta G$ values have decreased predicted binding affinity and negative $\Delta\Delta G$ values have increased predicted binding affinity. In this case, this means that large $\Delta\Delta G$ values have increased drug resistance. This is explained by the goal of the PI drug class is, as the name suggest, protease binding thereby inhibiting HIV-1 proliferation.

When comparing to the previous work surrounding this topic, the mutational landscape looks quite similar. This can be assessed by consulting the heatmap in figure 2 as well as the other heatmaps on the associated Github page (Andersen 2024). Comparing these heatmaps to the previous paper on this topic, the main amino acid being mutated to and from are also leucine, isoleucine and valine (Jiang 2024).

4.1 Proline prediction predicament

The previous study did not include many mutations to and from proline. Thus, there can be no comparison between proline outlier analysis, as no proline outlier analysis were performed (Jiang 2024). However, the high content of proline within the outliers is still damning for proline mutation prediction. The relatively high presence of proline in predicted $\Delta\Delta G$ outliers showcased in table 1 supports the difficulty of proline predicted found in the literature, as it is a known phenomenon. Entire studies have been dedicated to understanding and predicting proline mutations. A mutation to or from proline is is interesting, as it requires the formation or breaking of a covalent bond, which can have unforeseen stability consequences. Furthermore, proline is less flexible, and during protein folding, this results in entropy loss. However damning this is, mutating to a proline in certain key positions can increase stability (Duan, Lupyan, and Wang 2020).

In table 2 and 1 there is a disparity between the proline ratio of the outliers. These are likely explained by the difficulty of predicting proline mutations. From these tables, the amount of total outliers find varies. In this instance the outliers were selected visually by consulting the scatter plots in figure 3. However, other outlier detection methods could also have been utilized. For instance, one could have chosen to define an outlier as being 1, 2 or 3 standard deviations from the mean, although 2 standard deviations from the mean is commonly applied: $\mu \pm 2\sigma$.

4.2 Future of HIV-1 drug resistance prediction

As the HIV-1 protease is subject to somewhat artificial selection due to administrating of PI drugs, the need to continually assess drug resistance becomes paramount. Thus, it would be neat to have a tool or software that can accurately predict effects on drug resistance of mutations. Unfortunately, it does not seem as if Rosetta can reliably predict these. This highlight the need to continue developing Rosetta and other similar tools.

Luckily, application and benchmarking of new tools or updated Rosetta can be made easy with the introduced Python class. Here, the adapter design pattern can be utilized to integrate any new tool´s output into the existing Python class (Shvets, Skobeleva, and Zhart 2014). However, this depend on the Stanford HIV database to have a similar structure. Once more, one could utilize an adapter to integrate new data into the Python class to test drug resistance prediction from other datasets. This highlight the utility of the created Python class, and if it properly used and maintained, it could expedite the process of testing tools for HIV-1 PI drug resistance (Andersen 2024).

Another interesting avenue regarding the topic of PI drug resistance is to investigate where along the protein the mutations occur. For instance, it might be interesting and revealing if they occurred within or near the binding site, as PI drugs rely on binding. Mutations in other protein domain could also have knock-off effects on stability, which could be interesting to look into. A relevant entry point into this topic might be to look into positions: 0, 32, 47, 48, 50, 54, 76, 82, 84 and 88, as these were stated to be reducing protease binding affinity (Stanford University n.d.).

One could also investigate improvement of prediction accuracy, whereby inclusion of more mutations could be highly relevant. The inclusion of more mutations is really common within patients with HIV-1. This can be seen on figure 1 where up to 31 mutations can occur! Improving prediction accuracy and understanding of these mutations could expedite drug development and, more cynically, save on experimental costs.

A valid approach to increasingly reliable predictions could be to look into other tools capable of predicting binding. As deep learning and machine learning are fast paced fields, new tools emerge constantly (Zhang et al. 2021). Looking into available tools on public benchmarking platforms could reveal promising tools for binding predictions. A platform such as ProteinGym tries to do just that. ProteinGym provides standardizes deep mutational scanning data (DMS assays) containing both substitution and indel data. From this data, publishers of tools can upload their prediction results (Notin et al. 2023).

5 Conclusion

Rosetta was not able to convincingly predict binding affinity for patient present in the Stanford HIV database for the ATV, IDV, LPV and NFV drugs. The results found in this project more or less aligns with the findings of the previous study done on the topic (Jiang 2024). This means that the mutational landscape and prediction capability are quite similar. Additional proline content analysis were also performed, which highlight the prediction predicament that proline is - which include Rosetta predictions. However, major progress within the field of bioinformatics and machine learning may be able to better the results in the future.

References

Andersen, Rasmus Willsleff (2024). *Summer project.* URL: https://github.com/fvt270/summer_project.

Duan, Jianxin, Dmitry Lupyan, and Lingle Wang (2020). "Improving the accuracy of protein thermostability predictions for single point mutations". In: *Biophysical Journal* 119.1, pp. 115–127.

Fletcher, Courtney V, J Bartlett, PE Sax, and J Mitty (2018). "Overview of antiretroviral agents used to treat HIV". In: *UpToDate https://www. uptodate. com/contents/overview-of-antiretroviral-agents-used-to-treat-hiv.*

Flexner, Charles (1998). "HIV-protease inhibitors". In: *New England Journal of Medicine* 338.18, pp. 1281–1293.

Jette, M, C Dunlap, J Garlick, and M Grondona (2002). "SLURM: Simple Linux Utility for Resource Management". In: URL: https://www.osti.gov/biblio/15002962.

Jiang, Yinghan (2024). "Exploring the Impact of Multiple Mutations on Protein-Ligand Interactions on HIV Drug Resistance databaset". In: *University of Copenhagen - Bioinformatics Project 2.*

McKinney, Wes et al. (2010). "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference.* Vol. 445. Austin, TX, pp. 51–56.

Montagnier, Luc (2002). "A history of HIV discovery". In: *Science* 298.5599, pp. 1727–1728.

Notin, Pascal, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks (2023). "ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design". In: *Advances in Neural Information Processing Systems.* Ed. by A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., pp. 64331–64379. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/cac723e5ff29f65e3fcbb0739ae91bee-Paper-Datasets_and_Benchmarks.pdf.

Ouyang-Zhang, Jeffrey, Daniel Diaz, Adam Klivans, and Philipp Krähenbühl (2024). "Predicting a Protein's Stability under a Million Mutations". In: *Advances in Neural Information Processing Systems* 36.

Plotly Technologies Inc. (2015). *Collaborative data science.* URL: https://plot.ly.

Quinn, Thomas C (1994). "Population migration and the spread of types 1 and 2 human immunodeficiency viruses." In: *Proceedings of the National Academy of Sciences* 91.7, pp. 2407–2414.

Rosetta commons (n.d.). URL: https://rosettacommons.org.

Schrödinger, LLC (2015). "The PyMOL Molecular Graphics System, Version 1.8".

Shvets, Oleksandr, Olga Skobeleva, and Dmitry Zhart (2014). *Structural design patterns.* URL: https://refactoring.guru/.

Stanford University (n.d.). *Stanford University HIV DataBase.* URL: https://hivdb.stanford.edu/pages/genopheno.dataset.html (visited on 08/01/2024).

Zhang, Yanzhe, Jianqi Zhang, Zhuoqi Zheng, Bo Zhang, Jianmin Wang, Xiaotian Liao, Meng Liu, Albert Vilella, Alexander Tong, Christoph Feinauer, Jinyuan Sun, Minji Lee, and Anar Rzayev (2021). *List of papers about Proteins Design using Deep Learning.* https://github.com/Peldom/papers_for_protein_design_using_DL.